République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Ibn Khaldoun - Tiaret

Faculté des Sciences de la Technologie et Sciences de la Matière
Département d'Informatique

Mémoire présenté pour l'obtention du diplôme de

Magister en Informatique

Par BENABID Houari

Titre

Intégration sémantique de schémas de données guidée par les ontologies

Membres du Jury

Président :
Dahmani Youcef : Maître de Conférences, Université Ibn Khaldoun, Tiaret

Examinateurs :
Benabdellah Mohammed: Professeur, Université d'AbouBekr Belkaid, Tlemcen
Bessaid Abdelhafid : Professeur, Université d'AbouBekr Belkaid, Tlemcen

Directeur de Mémoire :
Chikh Mohammed Amine : Maître de Conférences, Université d'AbouBekr Belkaid, Tlemcen

Invité :
Chadli Abdelhafid: Chargé de Cours, Université Ibn Khaldoun, Tiaret

ANNEE 2010/ 2011

Remerciements

Je souhaiterais tout d'abord remercier mon directeur de mémoire, Chikh Mohammed Amine, Maître de conférences à l'université d'AbouBekr Belkaid-Tlemcen pour l'encadrement de mon travail, ses conseils avisés et ses remarques pertinentes afin d'améliorer la qualité de ce travail et de bien cerner toute la problématique.

Merci également aux personnes qui m'ont fait l'honneur d'accepter de participer à mon jury Dahmani Youcef, Maître de conférences à l'université IbnKhaldoun-Tiaret, Benabdellah Mohammed, Professeur à l'université d'AbouBekr Belkaid-Tlemcen, Bessaid Abdelhafid, Professeur à l'université d'AbouBekr Belkaid-Tlemcen et Chadli Abdelhafid , chargé de cours à l'université Ibn Khaldoun - Tiaret.

Je remercie tous les membres du personnel du CRSC pour leur amitié et leur aide pendant cette période de travail. Merci à Meslem Youcef et Smail M'hamed.

Je tiens à remercier mes collègues et amis de la promotion pour les bons moments que l'on a vécus ensemble à Tlemcen, Je remercie tout particulièrement Mr,Chikh Azzedine, pour son soutien pendant notre étude à Tlemcen.

Je remercie enfin ma famille, mes proches et amis pour leur compréhension et leurs encouragements.

Résumé

L'explosion du nombre de sources d'informations accessibles via le Web multiplie les besoins de techniques permettant l'intégration de ces sources. En définissant les concepts associés à des domaines particuliers, les ontologies sont un élément essentiel des systèmes d'intégration, car elles permettent à la fois de décrire le contenu des sources à intégrer et d'expliciter le vocabulaire utilisé dans les requêtes des utilisateurs. La tâche d'alignement d'ontologies (recherche de mappings, appariements ou mises en correspondance) est particulièrement importante dans les systèmes d'intégration puisqu'elle autorise la prise en compte conjointe de ressources décrites par des ontologies. Dans ce cadre nous avons proposé une architecture d'un système d'intégration de schémas basé sur les ontologies ISGO (Intégration des Schémas de données Guidée par les Ontologies). Les enjeux de l'intégration de schéma sont la prise en compte de l'hétérogénéité des sources, la définition des requêtes de médiation, la définition des correspondances sémantiques entre ces sources et du schéma de médiation. Pour cela nous avons doter le système d'intégration au (niveau du médiateur) par un module d'alignement d'ontologies pour découvrir ces correspondances afin de les utiliser dans la phase de réécriture de requête. L'algorithme permet non seulement le traitement efficace de ces correspondances par le calcul de mesure de similarité mais aussi l'automatisation de ce processus.

Mots clés: Intégration de Schémas, Médiation, Correspondances sémantiques, Ontologie, Alignement.

Abstract

The explosion in the number of sources information accessible via the Web multiplies the needs of technical allowing the integration of these sources. By defining the concepts associated with particular areas, ontology are an essential element of systems integration because they allow both to describe the content sources to integrate and explain the vocabulary used in user queries. The task of ontology alignment is particularly important in integration of systems as it enables the joint consideration of resources described by ontology. In this framework we have proposed architecture of a system integration based on ontology **ISGO** (Integration of data schemas Guided by the Ontology).The challenges of integration schema are taken into account the heterogeneity of sources, the definition of mediation queries, the definition of **semantic correspondences** between the sources and pattern of mediation. For that we provide the system integration (the mediator level) by a module ontology alignment to discover these connections for use in the query rewriting phase. The algorithm not only allows the efficient handling of these matches by the calculated similarity measure but also automation of this process.

Keywords: Schemas integration, Mediation, Semantic mapping, Ontology, Alignment.

Introduction Générale

1. Introduction

L'utilisation avancée de l'Internet et l'Intranet a conduit à la multiplicité et à l'expansion des systèmes d'information sur le web. La quantité de données gérées au sein de ces systèmes a rendu indispensable l'intégration de données provenant de différentes sources. Cette tache est devenue cruciale pour un nombre important d'application telle que le traitement de données financières, la bioinformatique, l'ingénierie de document, etc. Cette intégration permet à ces applications d'exploiter cette masse d'information pour répondre aux besoins des utilisateurs à travers des interfaces d'accès aux données de sources. Ces sources de données sont le plus souvent **réparties, autonomes** et **hétérogènes**.

Les sources d'informations sont **réparties** : de plus en plus d'informations sont créées partout dans le monde et publiées sur le Web, de nombreuses entreprises ont des ramifications dans plusieurs pays et les états décentralisent leurs administrations. Elles sont **autonomes** car les sources de données sont conçues par différentes personnes, à différents moments et pour répondre à différents besoins applicatifs. En fin, les sources d'informations sont **hétérogènes** : des logiciels différents sont utilisés pour créer et gérer les données par exemple (Oracle, O2, SQLServer), les données sont publiées dans des formats divers (HTML, PDF, images) et des modèles de données différents sont utilisés pour les représenter (modèles relationnel, objet, semi-structuré).

Un système d'intégration de données a pour rôle d'offrir à l'utilisateur une vue uniforme et une interrogation transparente des informations sans que l'utilisateur n'ait le souci de la provenance des informations ni de leur format d'origine.

2. Systèmes d'intégration de données

La combinaison des sources de données hétérogènes et de les interroger via une seule interface de requête peut être effectuée de différentes façons et à différents niveaux de l'architecture du système. 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 :

- **l'intégration virtuelle de données**, où la vue unifiée est virtuelle et les données restent stockées dans les sources d'origine. L'architecture type pour l'intégration virtuelle de données est l'architecture **médiateur**.

- **l'intégration matérialisée de données**, où la vue unifiée des données est matérialisée et les données sont rapatriées des sources d'origine et stockées dans un **entrepôt** de données.

Dans les systèmes de médiation, les besoins des utilisateurs sont représentés par un schéma de médiation pouvant être créé à partir des schémas des sources de données, ou indépendamment des sources par des experts du domaine. Il existe essentiellement deux types de liens entre le schéma de médiation et les schémas des sources de données : les correspondances sémantiques et les requêtes de médiation. Une correspondance sémantique met en relation deux concepts équivalents. Les requêtes de médiation décrivent le mode de calcul des instances du schéma de médiation à partir des instances des schémas sources ou inversement. Ces requêtes peuvent être élaborées suivant deux approches : dans la première, dite Global-As-View (GAV), chaque objet du schéma de médiation est défini par une requête exprimée sur les données sources. Dans la deuxième approche, dite Local-As-View (LAV), chaque objet dans un schéma source est défini par une requête exprimée sur le schéma de médiation. Dans les deux cas, les requêtes des utilisateurs sont exprimées sur le schéma de médiation et des algorithmes de réécriture permettent de transformer ces requêtes sur les données sources. L'architecture générale d'un médiateur est représentée par trois couches: médiateur, adaptateurs et sources

L'approche entrepôt de données consiste à voir l'intégration comme la construction d'une base de données réelle appelée entrepôt, stockant des données intégrées provenant de différentes sources. L'intégration selon une approche entrepôt de données est fondée sur un schéma global de l'entrepôt fournissant une vue intégrée des sources. Une fois que le schéma de l'entrepôt est conçu, les données sont extraites à partir des sources, transformées au format de représentation des données de l'entrepôt par des extracteurs, elles sont éventuellement filtrées pour ne garder que les données pertinentes, et enfin stockées dans l'entrepôt. Les magasins de données correspondent à un ensemble de vues sur l'entrepôt qui peuvent être matérialisées ou abstraites. L'interrogation s'appuie sur des techniques classiques d'interrogation du domaine des bases de données. L'utilisateur interagit avec l'entrepôt pour une interrogation directe des données de l'entrepôt ou à travers les magasins de données soit pour effectuer de la fouille de données soit pour générer des rapports statistiques.

3. Intégration sémantique de données

L'intégration sémantique présente un défi majeur dans le processus d'élaboration des systèmes d'intégration quelle que soit l'architecture du système utilisée. Elle est due au problème de l'hétérogénéité des sources de

données qui est généralement classée en deux types:(i) hétérogénéité des schémas et (ii) hétérogénéité des données.

L'hétérogénéité des schémas ou de structures, apparaît, d'une part, lorsque des modèles de données différents sont utilisés pour décrire les données mais également lorsque des schémas décrits dans un même modèle sont différents. En effet, les choix considérés pour concevoir le schéma de la source de données, tels que les noms des relations, des attributs, des tags, des types de données et le degré de décomposition des attributs, diffèrent d'une source à une autre. La présence de variations dans les structures est inévitable puisque les humains pensent différemment et que les sources de données sont conçues pour des besoins applicatifs distincts.

L'hétérogénéité de données apparaît lorsque différents vocabulaires et référentiels sont utilisés pour représenter les données, lorsque les informations sont incomplètes c'est-à-dire lorsque certains attributs ne sont pas renseignés, et lorsque les données contiennent des erreurs. Ce type d'hétérogénéité est également inéluctable lorsqu'on souhaite intégrer des données provenant de différentes sources. De plus, certaines sources de données sont alimentées par des informations extraites automatiquement à partir du Web ce qui peut engendrer de multiples erreurs dans les données. Ces erreurs peuvent être dues par exemple à une mauvaise segmentation des informations ou une association incorrecte des valeurs aux attributs.

Pour traiter l'hétérogénéité des schémas, la majorité des travaux menés se contentent de l'exploitation des informations présentes dans les schémas. Des techniques de comparaison syntaxique, telle que les mesures de similarité ont été utilisées. Des heuristiques ont également été utilisées, par exemple, le fait que des noms des attributs comportent des informations sur leur sémantique. Pourtant ces derniers peuvent contenir des abréviations, des concaténations de plusieurs noms d'attributs et des homonymes. Par exemple, deux entreprises qui souhaitent fusionner leur base de données ont dans leur schéma respectif des attributs homonymes qui possèdent des sens différents. Les mêmes noms d'attributs peuvent ne pas représenter la même information et n'ont donc pas la même sémantique. Le problème de l'hétérogénéité des schémas ne peut donc pas être résolu simplement avec ce type de méthodes. C'est pour cela que rendre explicite la sémantique précise des données intégrées est essentiel pour avoir une intégration sémantiquement correcte des données. Notre approche est fondée sur l'explicitation et l'exploitation de la sémantique des schémas de ces données à intégrer. Nous parlerons alors d'intégration sémantique de données.

Une première idée pour rendre explicite la sémantique des éléments du schéma des sources de données est de la spécifier de manière exhaustive pour tous les éléments des schémas considérés.

Les ontologies qui sont définies comme " **Une spécification explicite et formelle d'une conceptualisation commune** ", peuvent contribuer à la résolution du problème de l'hétérogénéité sémantique. En effet, elles offrent une description formelle des concepts et de leurs relations existant dans certains domaines. Un utilisateur ou concepteur peut grouper et déclarer explicitement la sémantique des concepts et des données fournis par les sources. Ainsi, en exploitant cette sémantique explicite et partagée, l'impact nuisible de l'hétérogénéité sémantique peut être réduit. Il s'agit dans ce cas d'une approche d'intégration de données fondée sur les ontologies. Une fois que les données des sources sont rendues conformes à l'ontologie, les requêtes peuvent être exprimées en termes de l'ontologie et les données peuvent être interrogées via une seule interface de requêtes. Dans le cas d'une intégration matérialisée des données, les requêtes exprimées en termes de l'ontologie sont évaluées directement sur l'entrepôt ; dans le cas d'une intégration virtuelle des données ces requêtes sont réécrites en fonction des vues sur les sources et les extensions de ces vues sont ensuite intégrées et combinées pour pouvoir répondre à l'utilisateur.

4. Problématique abordée

Pour concevoir des systèmes d'intégration sémantique de données, nous nous sommes posé le problème principal qui est indépendant de l'architecture choisie pour le système d'intégration (entrepôt ou médiateur).

Le problème se pose lorsqu'on souhaite intégrer les sources de données avec un schéma : ce qui consiste à déterminer quel élément du schéma global est représenté par quel élément du schéma des sources. Ces travaux s'inscrivent dans la problématique plus générale de la mise en correspondance de schéma.

La mise en correspondance de schémas consiste à prendre deux schémas en entrée et à construire en sortie un ensemble de correspondances entre des éléments des deux schémas afin de pouvoir établir des liens entre les deux schémas. Ces Liens vont servir à la réécriture de requête et à la découverte les sources les plus appropriées. Afin de réduire cette hétérogénéité, de plus en plus d'approches d'intégration associent aux données des thésaurus, comme WordNet, visent à définir le sens des mots et les relations entre ces mots. Ces relations étant approximatives et fortement contextuelles elles permettent seulement une automatisation partielle du processus d'intégration, sous la supervision d'un expert humain. Ceci demande une forte implication de l'expert et le résultat est très incertain car les décisions sont subjectives.

Lorsque, méthode plus récente, des ontologies formelles sont utilisées, on suppose que chaque source contient sa propre ontologie issue d'une ontologie de domaine qui couvre la totalité des concepts utilisés. Le problème d'intégration sémantique devient alors un problème d'intégration d'ontologies. Dans ce cas les concepts sont définis avec plus de précision et l'intégration peut être plus argumenté et plus facilement outillable, en particulier, si les différentes ontologies sont basées sur le même modèle. Une correspondance entre ontologies, associées aux sources de données, peut alors être exploitée par le système d'intégration, notamment dans l'étape de réécriture de requête et identification des sources pertinente.

Les données extraites des sources doivent être décrites en utilisant le vocabulaire de cette ontologie globale afin que cette dernière puisse servir d'interface d'interrogation des sources de données intégrées. Pour atteindre cet objectif les deux problèmes suivants doivent être traités :

- Trouver des mises en correspondance entre les concepts des schémas sources et les concepts du schéma global.
- Ce processus doit être le plus automatique possible et donc minimiser l'intervention humaine.

5. Contribution

Pour traiter les deux problèmes dégagés dans la section précédente nous avons proposé une approche dans laquelle l'ontologie a un rôle central. Cette approche consiste à exploiter en plus des informations syntaxiques présentes dans les données, les connaissances du domaine déclarées explicitement dans l'ontologie. Dans notre approche, l'exploitation du contenu de l'ontologie a deux rôles :

1. homogénéisation des données hétérogènes en les décrivant relativement au contenu d'une même ontologie globale ,dite ontologie de domaine, en les enrichissant par les concepts, les termes et les relations du domaine représentés dans l'ontologie. Nous parlons donc de l'enrichissement sémantique des données.
2. elle fournit aux utilisateurs un vocabulaire de base approprié pour formuler leurs requêtes et interroger les sources auxquelles le médiateur a donné accès.

Nous avons proposé une solution pour l'intégration de systèmes d'informations hétérogènes, fondée sur la médiation et les ontologies pour résoudre les conflits sémantiques. Nous visons l'hétérogénéité sémantique aussi bien au niveau des schémas (structures) qu'au niveau du contenu (données), et ceci, tout en préservant l'autonomie des sources de données. Dans cette architecture et en première étape en associant à chaque source, plus particulièrement, à son

schéma local une ontologie qui en définit le sens, mais toutes les ontologies utilisent un vocabulaire partagé d'une ontologie globale dite « ontologie partagée ». Après cette association, on procède à doter l'architecture proposée par un module d'alignement pour aligner l'ontologie globale (schéma global) avec les ontologies associées aux différents sources utilisées en générant les correspondances sémantiques nécessaires entre les concepts du schéma global et ceux des schéma locaux.

Dans le cadre de notre travail, nous utilisons une approche d'intégration hybride de description de correspondances sémantiques entre ontologies. Nous proposons un modèle qui utilise un médiateur intelligent (comme outil de requête) permettant l'intégration (semi-automatique) des sources de données hétérogènes.

L'idée principale de notre approche consiste à déterminer un algorithme permettant un traitement efficace de ces mises en correspondances. Nous utilisons un enchaînement de techniques, linguistique et structurelle afin d'obtenir un ensemble de règles d'association entre les éléments. Le résultat de ces correspondances doit servir à la réécriture de requête et à l'indentification des sources pertinentes.

6. Organisation du mémoire

Après cette introduction montrant la problématique de ce travail ainsi que notre contribution, le reste de ce mémoire est structuré comme suit :

Le Chapitre I : Etat de l'art sur l'intégration de données. Défini la notion d'intégration de données, et les problèmes sémantiques rencontrés et présente un état de l'art sur les travaux proposant des solutions à ces problèmes sémantiques.

Le chapitre II : L'intégration de données et les Ontologies .Présente des notions sur les ontologies (langages, formalismes..), le rôle des ontologies dans le processus d'intégration de données et les différentes méthodes de mesure de similarité pour aligner l'ontologie globale avec les ontologies locales.

Le chapitre III : Proposition d'une Approche d'intégration sémantique. Consacré à l'approche proposée qui consiste à proposer un module d'alignement pour une intégration virtuelle (médiateur) basée sur les ontologies (ontologie globale, ontologies locales) pour le traitement efficace des correspondances sémantiques. Ce dernier permet d'une part d'automatiser le processus d'intégration et la réécriture des sous requêtes d'autre part.

Le chapitre IV: Développement et expérimentations. Dans ce chapitre, nous présentons le prototype ISGO qui met en oeuvre la méthode d'enrichissement

sémantique. Nous présentons également les résultats d'expérimentation dans le domaine voyage touristique.

Le chapitre V: Conclusion Générale. C'est une conclusion de ce mémoire qui synthétise les principales contributions de ce travail et donne quelques perspectives.

Chapitre I:Etat de l'art sur l'intégration de données

I.1 Introduction

Avec l'émergence de l'Internet et de l'Intranet, il est possible d'accéder aujourd'hui à de multiples sources d'informations réparties, hétérogènes et autonomes. Dans un tel contexte, il est souvent nécessaire pour une application d'accéder simultanément à plusieurs sources, du fait qu'elles contiennent des informations pertinentes et complémentaires. Pour ce faire, la solution des systèmes d'intégration a été proposée. Elle consiste à fournir une interface uniforme et transparente aux données pertinentes via un schéma global pour répondre aux besoins de utilisateurs.

De ce fait, l'interopérabilité entre ces systèmes d'information est complexe puisque les applications doivent être adaptées pour pouvoir déterminer, pour chaque requête, les sources de données pertinentes, la syntaxe requise pour l'interrogation, la terminologie (concept) propre à la source, et pour pouvoir combiner les fragments de résultats issus de chaque source en vue de construire le résultat final. Ce processus d'adaptation peut être plus ou moins complexe : exploitation des résultats d'une source pour interroger une autre, élimination des redondances, etc.

L'intégration virtuelle des sources de données hétérogènes, autonomes et réparties est une solution pour l'interopérabilité entre différents systèmes d'information, puisqu'elle simplifie l'accès aux données. Cette approche consiste à fonder l'intégration d'informations sur l'exploitation de vues abstraites d´écrivant le contenu des différentes sources d'information. Le médiateur a pour rôle de masquer l'hétérogénéité et la répartition des sources de données. Quant à l'adaptateur, il a pour fonction d'adapter les requêtes aux formats des sources de données.

I.2 Contexte d'intégration de données

L'objectif de l'intégration de données restant de pouvoir répondre aux requêtes de l'utilisateur sans imposer à celui-ci de pré requis, il est nécessaire de définir le problème que nous visons à résoudre

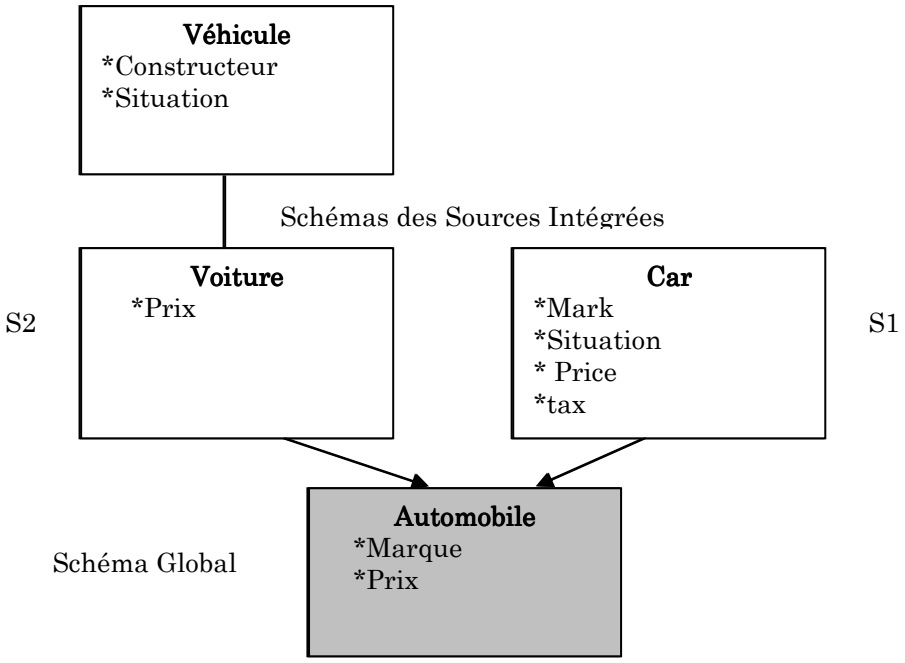


FIG I.1 - Exemple de catalogues électroniques hétérogènes

Un système d'intégration fournit un point d'accès unique à un ensemble de sources de données. Ces sources de données ont été conçues indépendamment par des concepteurs différents. Cela entraîne l'hétérogénéité de données, c'est-à-dire que les données relatives à un même sujet sont représentées différemment sur des systèmes d'information distincts. Cette hétérogénéité provient des choix différents qui sont faits pour représenter des faits du monde réel dans un format informatique. En effet, les données des sources sont structurellement indépendantes mais sont toujours supposées relever de domaines similaires.

Nous allons étudier l'exemple de la figure I.1 pour bien illustrer le problème de l'hétérogénéité de sources. Supposons qu'une société vend des voitures sur Internet. Les voitures mises en vente sont fournies par des fournisseurs différents, ou chacun organise ses produits dans son catalogue selon ses propres critères. Cette société doit donc traduire les catalogues de différents fournisseurs à son format, appelé schéma global. Etant donné deux catalogues S1 et S2 de deux fournisseurs et le catalogue de l'entreprise Schéma Global (SG).

Comme indique la figure I.1 les trois catalogues décrivent une voiture différemment. Cette différence concerne le nombre de concepts utilisés pour définir chaque source, ainsi que l'aspect sémantique de chaque concept. Le tableau I.1 récapitule les différentes propriétés de SG, S1 et S2.

	SG	S1	S2
Classes	**Automobile :** Voiture actionnée par un moteur	**Car :** autobus	**Véhicule :** Moyen de transport terrestre
			Voiture : voiture
Propriétés		**Situation** : (domaine :boolean) neuf ou occasion	**Situation** :(domaine :boolean) disponible ou non
	Marque (domaine:string) : marque de fabrique	**Mark : (domaine : string):** marque de fabrique	**Constructeur :(domaine : string) :** marque de fabrique
	Prix: (domaine number) : le prix d'une voiture avec un garantie de 30000 Km	**Price : (domaine :number) :** le prix hors taxe d'une voiture	**Prix : (domaine :number) :** le prix total (qui inclut la TVA) d'une voiture
		Tax : (domaine :number):la TVA d'une voiture	

Tab I.1–Sémantique de données de trois catalogues dans la FIG II.1

Tout processus d'intégration de données hétérogènes doit d'abord:

- Identifier les conflits entre les concepts dans des sources différentes qui ont des liens sémantiques.
- Résoudre les différences entre les concepts sémantiquement liés.

Dans un système multisources, une même donnée du monde réel peut être codée et/ou interprétée de plusieurs façons au cours de ses stockages dans les différentes sources de données. Dans ce contexte, Goh[1] a identifié quatre principaux types de conflit : conflits de nom, conflits d'échelle, conflits de représentation et conflits de contexte. Nous les commentons dans ce qui suit.

I.2.1 Conflit de noms : Les conflits de nom sont liés à des différences dans la désignation de concepts au de propriétés. Le cas le plus fréquent est la présence de (synonymes), on utilise des noms différents pour le même concept ou propriété, et de (homonymes) lorsque le même nom est utilisé pour des entités différentes. Par exemple Le même concept Automobile est nommé par Car dans la source S1, et par Voiture dans la source S2. La propriété Situation se trouve dans les deux sources, mais avec deux significations différentes (voir Tab I.1).

I.2.2 Conflits d'échelle : les conflits d'échelle ont lieu quand des systèmes de référence différents sont utilisés pour mesurer une grandeur. Par exemple l'unité de mesure du prix dans la source S1est le Dinar tandis que celle dans la source S2 est l'euro.

I.2.3 Conflits de représentation : les conflits de représentation surviennent quand les schémas de deux sources décrivent différemment un même concept. Par exemple le fournisseur de S1 utilise une seule classe Car et 4 propriétés : Mark, Situation, price et tax pour décrire une voiture. Tandis que le fournisseur de S2 utilise deux classes : véhicule et voiture et 3 propriétés constructeur, situation et prix. Un autre exemple de confit de représentation entre les deux fournisseurs à mettre en évidence, c'est le cas où le fournisseur de S1 utilise deux propriétés : price et tax pour calculer le prix d'une voiture, tandis que de S2 n'en utilise qu'une seule, à savoir prix.

I.2.4 Conflits de contexte : le contexte est une notion très importante dans les systèmes d'information répartis. Les conflits d'indéterminisme surgissent quand les concepts semblent avoir le même sens alors qu'ils sont différents. Ceci peut être dû à des contextes temporels différents.

I.3 Les approches d'intégration

La première étape de l'intégration de données consiste à élimer d'abord les conflits sémantiques et ensuite les représenter par un schéma global uniforme. Ce processus d'intégration dans ce domaine présente néanmoins trois difficultés:

- L'identification de **la correspondance sémantique** entre les différentes sources qui met en relation deux concepts équivalents ;
- L'élaboration d'une vue globale intégrée des données représentées à travers des conceptualisations différentes ;
- L'évolution des différentes sources de données et leurs conséquences sur le schéma global.

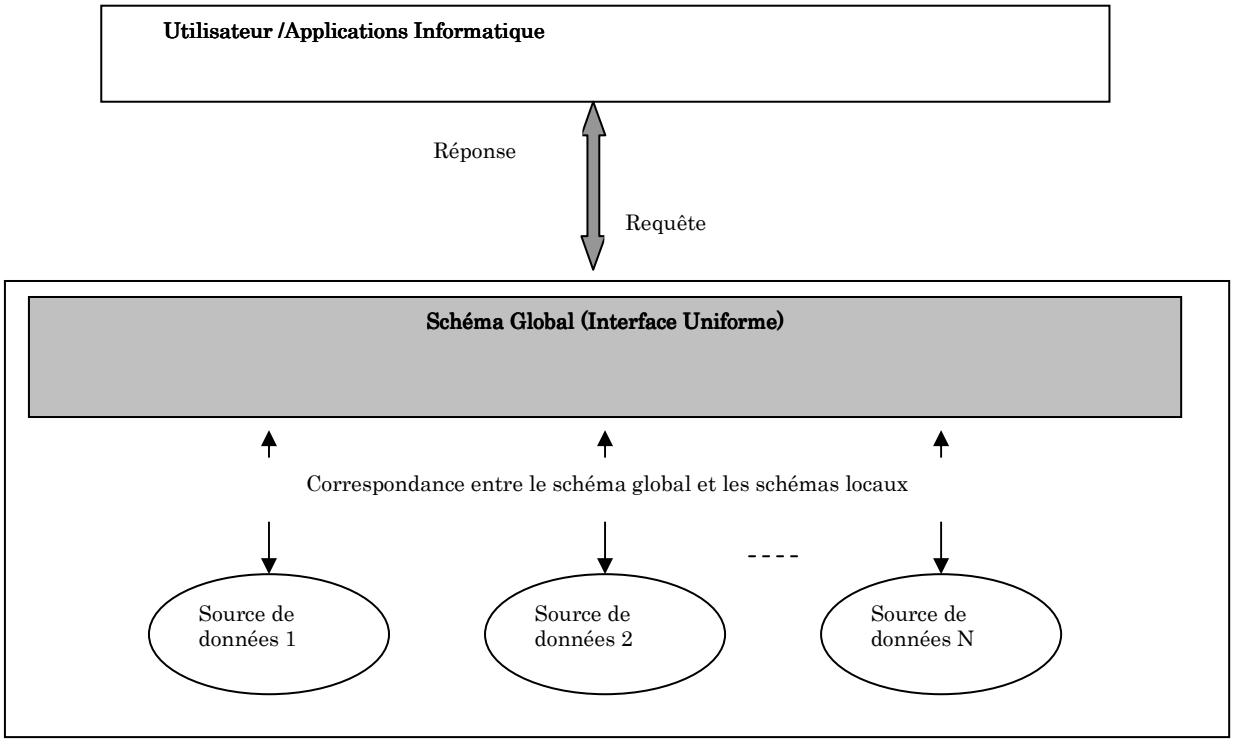


FIG I.2 - Système d'intégration de données

Généralement, un système d'intégration comprend d'une part, les sources d'informations participant dans le processus d'intégration et d'autre part,une interface (schéma global) permettant aux éléments de la partie externe d'accéder d'une manière transparente aux sources de données. Cette interface peut être une couche sans données propres (approche médiateur) ou une couche contenant, sous une forme qui lui est propre, une duplication des données pertinentes des sources (approche entrepôt). Les éléments externes au système ne voient pas les sources internes. Celles-ci sont encapsulées, cachées par l'interface. Les éléments externes doivent pouvoir agir sur le système d'intégration d'une manière transparente via un schéma global.

Cette vision générique d'un tel système se décline dans des types de systèmes souvent classifiés par rapport à des paramètres tels que le degré d'intégration des données, la gestion des données, l'expression et le traitement de requêtes. Gomez propose dans [15] une classification de ces systèmes selon les dimensions de distribution, hétérogénéité et autonomie. D'autres classifications ont été proposées dans [13]. Afin de nous permettre de mieux positionner notre travail par rapport à l'existant nous proposons une classification suivant trois critères, à savoir :

- le type d'intégration de données : virtuelle ou matérialisée,
- la correspondance entre schéma global et les schémas locaux: l'approche GAV ou LAV,
- le processus d'intégration de données : le degré d'automatisation.

I.4 Architecture virtuelle

Il faut noter que l'architecture virtuelle est une intégration virtuelle. Cette dernière n'est pas matérialisée, comme pour les entrepôts de données. Les données restent dans leur source d'origine.

I.4.1 Systèmes Multibases

Les systèmes multibases sont des systèmes dits faiblement couplés [34]. On les caractérise de cette manière car ils n'offrent pas une vision unifiée des données. Il n'existe pas de schéma global permettant un accès transparent aux différentes sources de données. La coopération est seulement assurée par l'intermédiaire d'un langage commun : le langage multibase (de type SQL notamment [35]).

L'utilisateur peut poser une requête aux différents systèmes à l'aide de ce langage commun, sans se soucier de l'hétérogénéité des systèmes sources. Ceci ne signifie pas qu'une requête nécessitant l'accès à diverses sources est exécutée en une seule fois .L'utilisateur doit envoyer autant de requêtes qu'il y a de sources impliquées. C'est donc à l'utilisateur de relier les différentes réponses aux requêtes formulées.

Ces systèmes sont donc faiblement intégrés et gardent une grande autonomie. Les sources peuvent évoluer de manière indépendante, sans conséquence sur leur accès. Cependant, la cohérence entre ces sources n'est pas assurée. L'hétérogénéité n'est pas traitée en amont. L'intégration est dynamique : les correspondances entre les données ne sont pas prédéfinies. Certains systèmes offrent la possibilité de rendre les correspondances persistantes entre les sources par la création de vues multibases. Les requêtes multibases sont exprimées sur ces vues multi-bases. C'est le cas du système MSQL [35].

I.4.2 Systèmes Fédérés

A l'inverse des systèmes multibases, les systèmes fédérés sont dits fortement couplés [34]. Ils se caractérisent par l'existence d'un schéma unifié appelé schéma fédéré qui constitue l'interface d'accès au système intégré. L'intégration se situe au niveau des schémas.

La conception de ce schéma fédéré suppose d'unifier les schémas source et de traiter leur hétérogénéité. Il est nécessaire d'identifier les correspondances et de résoudre les conflits entre les éléments des différents schémas. Ces

correspondances peuvent être exprimées à l'aide de différents langages, au moyen de règles ou à travers une ontologie. Il y a donc cette fois une vision unifiée des sources (ou plus justement d'une partie). L'intégration offre un accès commun aux sources et une représentation commune.

L'intégration des systèmes fédérés est statique : les liens de correspondances entre les schémas sont prédéfinis. Ce n'est plus à l'utilisateur d'établir ces liens. Ce sont les administrateurs des bases sources qui définissent les sous-ensembles des données qu'ils souhaitent intégrer. L'accès aux données peut se faire de deux manières différentes : via les schémas locaux (schémas des BD source) ou via le schéma fédéré. Les bases sources gardent leur autonomie et restent sous contrôle de leur administrateur. C'est une des particularités des systèmes fédérés. En général, seule une partie des données des différentes sources est mise en commun. Pour cette raison, ces systèmes sont parfois considérés comme faiblement couplés [36]. Le niveau d'intégration est en effet moins élevé que celui imposé par une base de données répartie ou distribuée. Cette dernière suppose une intégration totale des données sources [37]. Les sources de données initiales perdent donc complètement leur autonomie. La fédération est un compromis entre une intégration nulle et totale.
Les auteurs dans [38] ont défini une architecture de référence pour les systèmes fédérés (voir FIG I.3). Elle est composée de 5 couches différentes :

- **Les schémas locaux** : il s'agit des schémas conceptuels initiaux des différentes BD source. Il en existe autant qu'il y a de systèmes sources à intégrer. Ces schémas locaux peuvent être exprimés dans des modèles différents.
- **Les schémas pivots :** il s'agit des schémas locaux traduits dans le modèle commun (ou modèle canonique), c'est-à-dire le modèle utilisé pour la fédération.
- **Les schémas d'export :** ils correspondent à un extrait des schémas pivots. Seuls les éléments que les administrateurs des bases sources souhaitent fédérer sont exportés.
- **Les schémas fédérés :** il s'agit des schémas d'export intégrés. Il peut en exister plusieurs. Ils offrent une vue unifiée des schémas exportés selon le modèle canonique.
- **Les schémas externes :** il s'agit de vues définies pour des groupes d'utilisateurs particuliers du système fédéré (reposant sur le modèle canonique). Ils n'offrent l'accès qu'à un sous-ensemble de données.

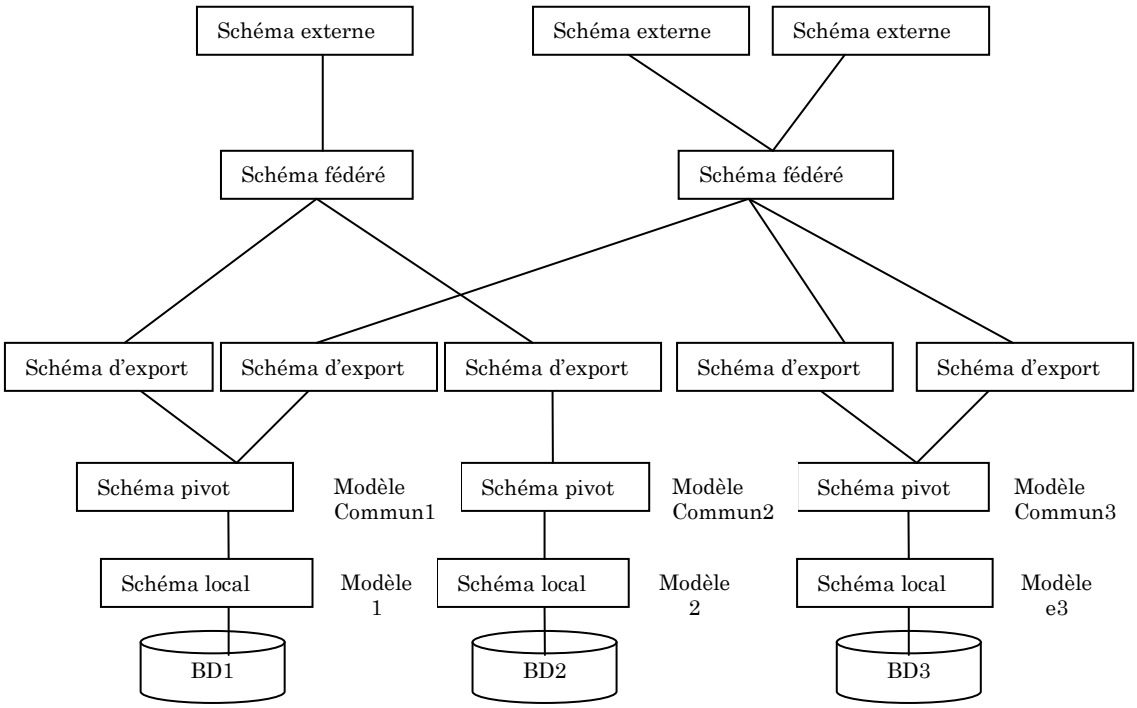


FIG I.3 - Architecture des systèmes fédérés

La conception de systèmes fédérés se fait de manière ascendante [34]. On définit le schéma fédéré à partir des schémas source après une analyse des correspondances entre les schémas source (plan horizontal), et après avoir traité les différents conflits entre les éléments des différents schémas (FIG I.3). L'identification des relations entre les schémas sources permet donc d'aboutir au schéma fédéré. La conception est différente de celle qui peut être adoptée pour les bases de données réparties. Pour celles-ci, les schémas locaux sont définis à partir du schéma global (approche descendante). Les correspondances entre le schéma global et les schémas locaux sont analysées dans un plan vertical (décomposition) et le principal problème est de traiter la répartition des données (fragmentation, duplication,…).

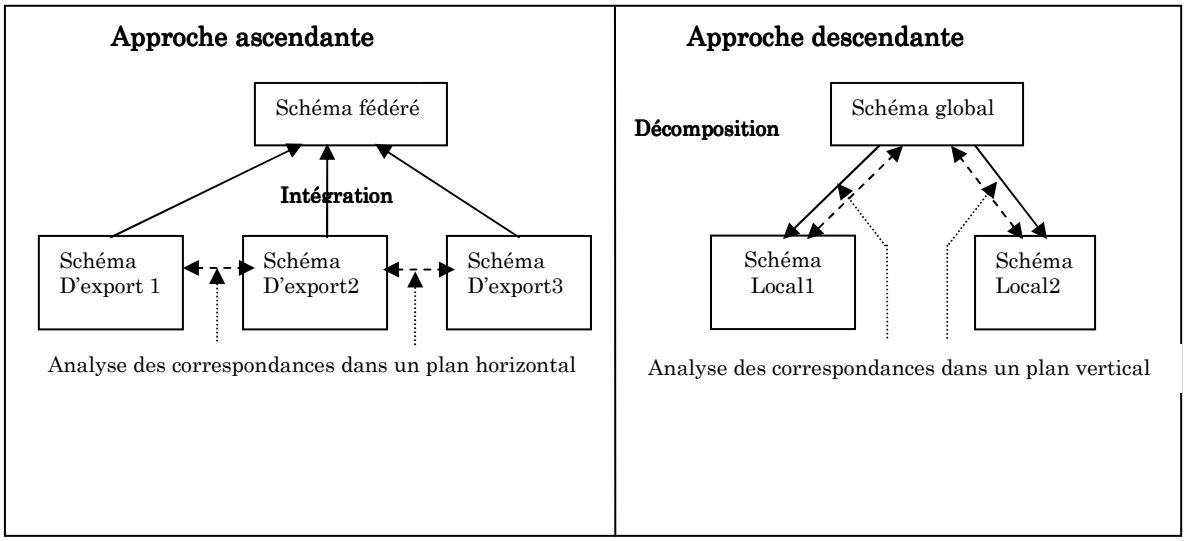


FIG I.4 - Stratégies de développement des systèmes fédérés et répartis

Il faut noter que la fédération est une intégration virtuelle. L'intégration n'est pas matérialisée, comme pour les entrepôts de données. Les données restent dans leur source d'origine. Grâce aux correspondances définies entre les schémas sources et à l'existence de mécanismes permettant de traduire les requêtes posées sur le schéma fédéré dans les termes des schémas source, il est possible d'accéder aux différentes bases [34]. Ce type de couplage a des conséquences importantes sur l'évolution du système. Un changement de configuration ou de schéma dans les bases sources doit se répercuter dans le schéma fédéré. Suivant le degré d'autonomie et d'hétérogénéité, des problèmes d'incohérences peuvent apparaître entre le système fédéré et les systèmes locaux. C'est la raison pour laquelle on considère généralement que cette intégration est bien adaptée lorsqu'il n'existe qu'un petit nombre de sources à intégrer.

En termes de fonctionnalité, les systèmes fédérés doivent en principe permettre d'accéder aux données en lecture et écriture à partir du schéma fédéré [34]. Des mises à jour peuvent être propagées dans les sources à partir de la base unifiée virtuelle. Cette capacité de mise à jour est une des caractéristiques qui distingue les systèmes fédérés des systèmes fondés sur la médiation. Ces derniers sont principalement conçus pour l'interrogation. Nous les présentons dans la partie suivante.

I.4.3 Système par médiateur

L'approche Médiateur consiste à définir une interface entre l'agent (humain ou logiciel) qui pose une requête et l'ensemble des sources accessibles via le Web potentiellement pertinentes pour répondre. L'objectif est de donner l'impression d'interroger un système centralisé et homogène alors que les

sources interrogées sont réparties, autonomes et hétérogènes. Il repose sur deux composants essentiels : le médiateur et l'adaptateur.

Le médiateur : chargé de la localisation des sources de données et des données pertinentes par rapport à une requête, il résout, de manière transparente, les conflits de données. Un ensemble de connaissances sur les sources permet au médiateur de générer un plan d'exécution pour traiter les requêtes d'utilisateurs.

L'adaptateur : comme dans l'approche entrepôt, c'est un outil permettant à un (ou plusieurs) médiateur(s) d'accéder au contenu des sources d'informations dans un langage uniforme. Il fait le lien entre la représentation locale des informations et leur représentation dans le modèle de médiation.

Dans cette approche, l'intégration d'information est fondée sur l'exploitation de vues abstraites décrivant de façon homogène et uniforme le contenu des sources d'information dans les termes du médiateur. Les sources d'information pertinentes, pour répondre à une requête, sont calculées par réécriture de la requête en termes de ces vues. Le problème consiste à trouver une requête qui, selon le choix de conception du médiateur, est équivalente ou implique logiquement, la requête de l'utilisateur mais n'utilise que des vues. Les réponses à la requête posée sont ensuite obtenues en évaluant les réécritures de cette requête sur les extensions des vues.

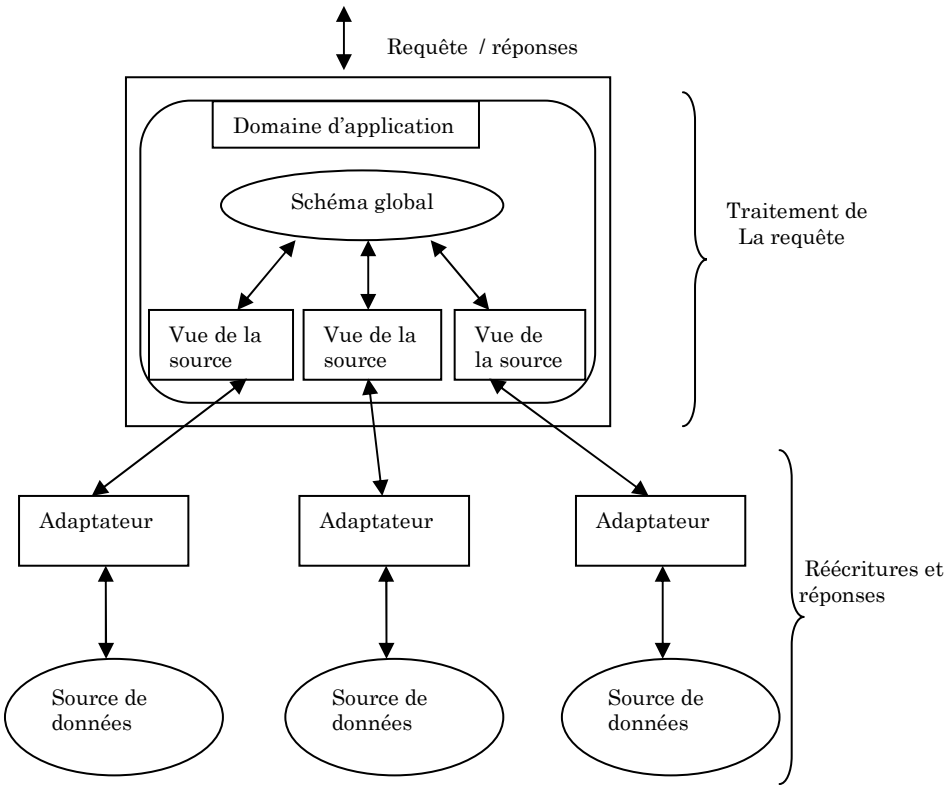


FIG I.5 - Architecture générale de l'approche Médiateur

les principaux problèmes pour lesquels les systèmes de médiation sont construits sont:

- le schéma global pour représenter les vues sur les sources à intégrer et pour exprimer les requêtes provenant des utilisateurs humains ou d'entités informatiques ;
- les mappings du schéma global avec les sources ;
- les fonctions de réécriture de requêtes et les fonctions de composition des résultats en termes de vues décrivant les sources de donnés pertinentes.

I.4.4 Quelques projets de l'approche « Médiateur »

Les différents systèmes d'intégration d'informations à base de médiateur se distinguent par:d'une part, la façon dont est établie la correspondance entre le schéma global et les schémas des sources de données à intégrer, d'autre part les langages utilisés pour modéliser le schéma global, les schémas des sources de données à intégrer et les requêtes des utilisateurs.

OBSERVER: (Ontology Based System Enhanced with Relationships for Vocabulary hEterogeneity Resolution) [27]. Le but de ce projet est de fournir un système permettant l'accès transparent et uniforme à des données indépendamment de leurs localisations, leurs formats de stockage et de leurs conceptualisations. OBSERVER associe à chaque source de données une ontologie qui conceptualise son contenu. Ce projet sera détaillé dans la section I.9.2.1.

TSIMMIS[1] : (The Stanford-IBM Manager of Multiple Information Sources) [10] est un projet visant à fournir des outils pour un accès intégré à des entrepôts d'information. Chaque source d'information est équipée d'un traducteur qui encapsule la source, convertissant les objets sous-jacents dans un modèle de données commun. Dans TSIMMIS, un modèle de données orienté-objet simple, appelé "Object Exchange Model" (OEM) est utilisé. Au dessus des traducteurs, TSIMMIS comporte un autre type de composants appelés médiateurs. Chaque médiateur obtient les informations d'un ou de plusieurs traducteurs ou d'autres médiateurs, affinent cette information par intégration et résolution de conflits entre les différents morceaux d'information issus des différentes sources, et fournit l'information résultant à l'utilisateur ou à d'autres médiateurs. Au niveau conceptuel, les médiateurs peuvent être considérés comme des vues sur les données d'une ou plusieurs sources qui sont soigneusement intégrées et traitées. Le médiateur est défini en terme de langage logique, appelé MSL, qui est essentiellement Datalog, étendu pour

[1] http://www-db.stanford.edu/tsimmis/

supporter les objets OEM. Les médiateurs fournissent des vues virtuelles puisqu'ils ne stockent pas les données localement.

I.5 Architecture matérialisée (Entrepôt)

La constitution d'entrepôts de données est une réponse au problème de l'intégration d'une grande quantité de données variées, relatives à un certain domaine d'application, et stockées physiquement dans différentes sources de données. L'entrepôt de données regroupe, sous une forme exploitable par des traitements utiles pour l'aide à la décision, les informations extraites de ces sources et qui sont potentiellement pertinentes pour telle ou telle catégorie de décideurs du domaine. Depuis son existence, les entrepôts de données se sont imposés comme une solution rentable pour faire face aux besoins des entreprises en termes de capitalisation de connaissances et d'aide à la décision.

I.5.1 Principe d'un entrepôt de données

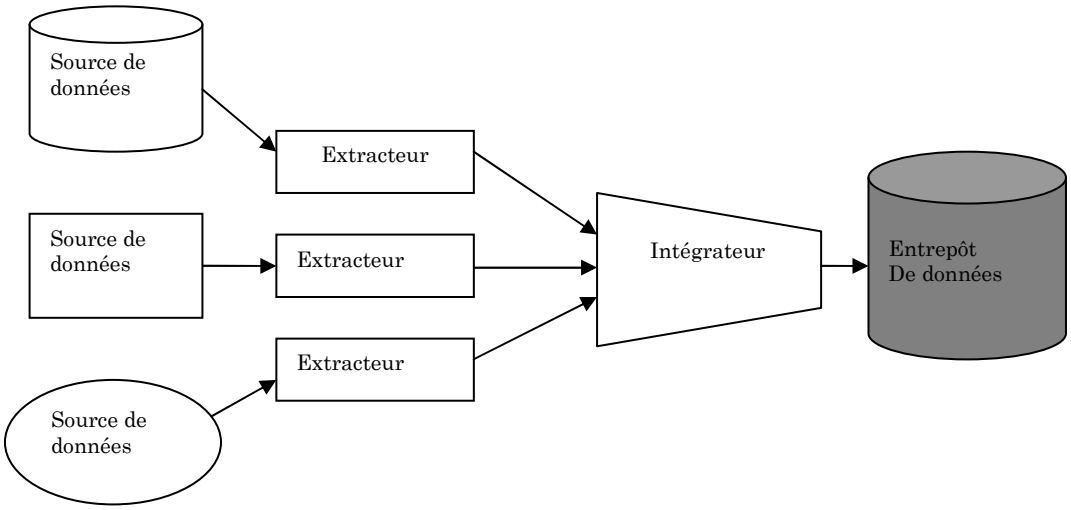
Dans son ouvrage de référence [2], W.H. Inmon définit l'entrepôt de données comme "une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse". Cette définition englobe différents termes que nous explicitons.

Intégrées:les données de l'entrepôt proviennent de différentes sources éventuellement hétérogènes. L'intégration consiste à résoudre les problèmes d'hétérogénéité des systèmes de stockage, des modèles de données, de sémantique de données…

Orienté sujet : les données de l'entrepôt peuvent être réorganisées autour de thèmes tels que le patient, les diagnostics, les médicaments…

Historisés : la prise en compte de l'évolution des données est essentielle pour la prise de décision qui par exemple utilise des techniques de prédiction en s'appuyant sur les évolutions passées pour prévoir les évolutions futures.

Non-volatiles : Les données de l'entrepôt sont essentiellement utilisées en mode de consultation. Les utilisateurs ne peuvent pas les modifier.



FIGI.6-Architecture générale de l'approche par entrepôt de données

Le processus de construction d'un système d'intégration matérialisé se compose en quatre étapes principales [23]:

- l'extraction des données des sources de données opérationnelles,
- la transformation des données aux niveaux structurel et sémantique,
- l'intégration des données,
- le stockage des données intégrées dans le système cible.

L'étape d'extraction et une partie de l'étape de transformation peuvent être groupées dans le même composant logiciel, tel qu'un «Extracteur» ou «Adaptateur ». Chaque adaptateur fournit une interface d'accès et de requêtes sur la source afin d'aboutir à des données représentées dans un même format. L'étape d'intégration consiste à éliminer les conflits et intégrer les données, puis l'étape de stockage qui charge les données dans l'entrepôt.

I.5.2 Quelques projets de l'approche « Entrepôt »

Quelques projets spécifiques d'entrepôts de données servent actuellement de références. Nous présentons ici quelques projets à titre d'exemple.

DWQ[2] **:** (Foundations of Data Warehouse Quality)[3,4] est un projet de la communauté Européenne visant à développer des fondements sémantiques qui permettront d'aider les concepteurs d'entrepôts de données dans le choix des modèles, des structures de données avancées et des techniques d'implantation efficaces en s'appuyant sur des facteurs de qualité de service. Ceci permet d'améliorer la conception, l'exploitation et l'évolution des applications d'entrepôts.

[2] http://www.dbnet.ece.ntua.gr/~dwq/

DWQ s'appuie sur des modèles formels pour la qualité. Les résultats comportent des méta modèles de données formels destinés à la description de l'architecture statique d'un entrepôt de données. Les outils associés comportent des facilités de modélisation incluant des caractéristiques spécifiques aux entrepôts comme la résolution de sources multiples, la gestion de données multidimensionnelles (éventuellement agrégées) et des techniques pour l'optimisation de requêtes et la propagation incrémentale des mises à jour.

SIRIUS[3] : Le projet SIRIUS (Supporting the Incremental Refreshment of Information warehoUseS) [5] développé à l'Université de Zurich, est un système d'entrepôt de données qui a pour objectif d'étudier des techniques permettant le rafraîchissement incrémental de l'entrepôt en réduisant les temps de mise à jour. Le schéma de l'entrepôt est défini sous la forme d'un schéma global UML.

WHIPS[4] :(WareHouse Information Prototype at Stanford) [6,7,8] est un système de gestion d'entrepôts de données utilisé comme banc d'essai. Le but de ce projet est de développer des algorithmes pour collecter, intégrer et maintenir des informations provenant de sources hétérogènes, distribuées et autonomes.

L'architecture du prototype WHIPS consiste en un ensemble de modules indépendants implantés comme des objets CORBA. Le composant central du système est l'intégrateur, auquel tous les autres modules sont reliés. Différents modèles de données peuvent être utilisés à la fois pour chacune des sources et pour les données de l'entrepôt. Le modèle relationnel est utilisé comme modèle unificateur : pour chaque source de l'entrepôt, les données correspondantes sont converties dans le modèle relationnel par un traducteur spécifique.

I.6 Discussion sur les deux architectures

Nous venons d'exposer brièvement différents systèmes intégrés, c'est-à-dire des architectures qu'il est possible de mettre en placer pour faire coopérer de manière relativement transparente (selon le niveau d'intégration) des sources de données initialement indépendantes. Nous avons ainsi vu les systèmes multibases, les systèmes fédérés et les systèmes de médiation. Nous avons également présenté les entrepôts de données car il existe une phase d'intégration de données pour les constituer.

[3] http://www.ifi.uzh.ch/arvo/dbtg/Projects/SIRIUS/sirius.html
[4] http://www-db.stanford.edu/warehousing/

L'avantage principal de l'approche matérialisé est que l'interrogation d'un entrepôt de données se fait directement sur les données de l'entrepôt et non sur les sources originales. On peut donc utiliser les techniques d'interrogation et d'optimisation des bases de données traditionnelles. On peut citer d'autres avantages comme la performance, personnalisation des données, versions et archivage .Par contre cette approche exige un coût de stockage supplémentaire et surtout un coût de maintenance causé par les opérations de mises à jour au niveau de sources de données (toute modification dans les sources locales doit être répercutée sur l'entrepôt de données).

L'approche virtuelle (système fédéré et médiateur) présente l'intérêt de pouvoir construire un système d'interrogation de sources de données sans toucher les données qui restent stockées dans leur source d'origine. Par contre le médiateur ne peut pas évaluer directement les requêtes qui lui sont posées car il ne contient pas de données, ces dernières étant stockées de façon distribuée dans des sources indépendantes.

I.7 Correspondances entre Schéma global et Schémas locaux

La correspondance entre le schéma global et les schémas locaux peut être élaboré suivant deux approches : Global-As-View (GAV) et Local as View (LAV).

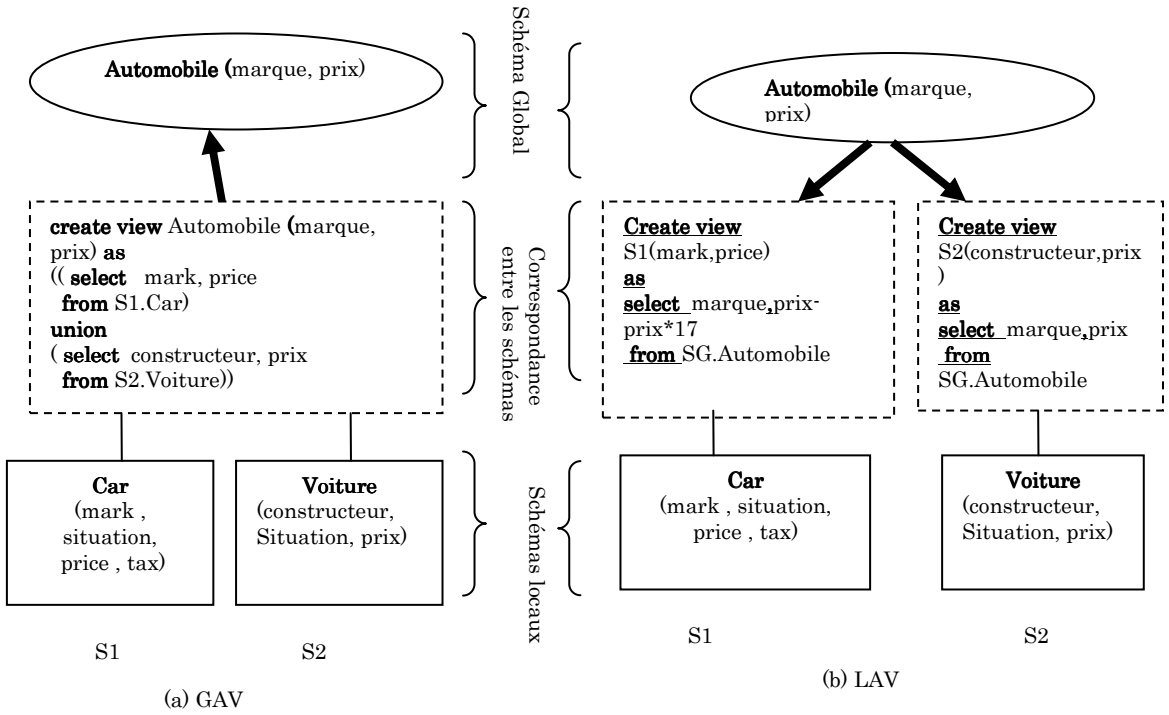


FIG I.7 -Exemple de mise en correspondance entre schéma global et schémas locaux

I.7.1 L'approche Global As View (GAV)

L'approche GAV consiste à définir le schéma global en fonction des schémas des sources de données à intégrer puis à le connecter aux différentes sources. En d'autres termes, pour chaque relation utilisée dans le schéma intégré, on définit une vue composée des termes des relations des sources.

L'avantage principal de cette approche réside dans le fait que la réécriture des requêtes est simple. En effet, on obtient facilement une requête en terme du schéma des sources de données intégrées, en remplaçant les prédicats du schéma global par leurs définitions. Parmi les systèmes utilisant GAV, on peut citer TSIMMIS [12] et MOMIS [13].

L'inconvénient essentiel de cette approche réside dans le fait qu'il est difficile d'ajouter des nouvelles relations dans l'expression du schéma intégré. Ceci revient pratiquement à la réécriture de toutes les relations du schéma intégré. La complexité de cette tache s'accroît si le nombre de source de données est important.

I.7.2 L'approche Local As View (LAV)

Contrairement à l'approche GAV, l'approche LAV consiste à définir les schémas des sources de données à intégrer comme des vues du schéma global. Les principaux systèmes développés autour de cette approche sont : Infomaster [14], PICSEL [20], Information Manifold [16].

L'ajout d'une nouvelle source est facile car il consiste à écrire l'ensemble des relations de la source en fonction des relations du schéma intégré global. Par contre le processus de réécritures de requêtes en fonction des vues est un peu complexe par apport à la méthode GAV.

I.8 Processus d'intégration

La nature de processus d'intégration de données spécifie la manière dont les données sont intégrées. Les données peuvent être intégrer manuellement, semi-automatique ou automatique. Ce critère devient essentiel lorsque l'on veut intégrer un nombre important de sources de données indépendantes.

I.8.1 Intégration manuelle de données

Le système de multi-base de données [17] permet de créer une véritable interopérabilité entre les bases de données sans besoin de générer explicitement un schéma global intégré. Le processus d'intégration est conçu au niveau du langage d'interrogation. De cette manière, l'utilisateur spécifie sa requête en précisant les sources de données interrogées. L'implémentation du système (langage d'interrogation) est aisée et toutes les hétérogénéités sont

traitées par les utilisateurs. Les interfaces d'interrogation sont textuelles. Ce processus d'intégration est donc complètement manuel. Par contre, ils assurent l'autonomie et l'intégrité de chaque base de données tout en permettant un accès partagé aux données.

La fédération des bases de données [18] vise à fournir un schéma conceptuel global représenté par l'union de toutes les bases de données. Le schéma global donne une vision homogène de toutes les sources et facilite l'interrogation de leurs données. Tout d'abord un schéma externe est défini pour chaque base de données qui permet de définir ce que l'on vise à exploiter. Ces schémas sont ensuite convertis à une représentation commune, un formalisme relationnel par exemple, puis les schémas canoniques sont intégrés dans un schéma global. Les requêtes d'utilisateurs sont spécifiées sur le schéma global et sont ensuite décomposées en sous-requêtes. Ces sous-requêtes sont envoyées aux sources de données. Le processus de la construction de schéma global est fait manuellement.

Une troisième technique utilisée dans les approches d'intégration manuelle de données est l'utilisation des standards de représentation et d'accès pour faciliter l'interopérabilité de systèmes hétérogènes. XML et EXPRESS, par exemple, qui sont les standards pour la représentation et l'échange de données, sont utilisés pour résoudre les problèmes liés à l'hétérogénéité syntaxique (uniformisation de la structure des données) et produire une vue logique des données. L'intégration sémantique doit entièrement être faite par l'utilisateur.

Dans les environnements qui nécessitent d'intégrer un nombre important de sources de données réparties qui évoluent fréquemment, l'intégration de données manuelle devient très coûteuse et souvent même impossible. Les traitements plus automatisés ont donc été conçus pour faciliter la résolution des conflits sémantiques. Ceci correspond aux deux classes étudiées ci-dessous.

I.8.2 Intégration semi-automatique de données

La deuxième génération de systèmes d'intégration utilise des ontologies linguistiques (OL) (qui sont également appelées des thésaurus) pour identifier automatiquement ou semi-automatiquement quelques relations sémantiques entre les termes utilisés dans les sources de données.

On peut alors automatiquement comparer les noms de relations ou d'attributs et essayer d'identifier les éléments similaires en utilisant des OL. Les OL restent cependant orientés "terme" et non "concept". Ce type d'ontologies est également très lourd à développer et à mettre à jour. Ainsi le traitement par OL est nécessairement supervisé par un expert et ne peut être que partiellement automatique. Certains travaux d'intégration à base OL utilisent des mesures d'affinité et de similarité afin de calculer la vraisemblance de

relations entre concepts. Ces mesures sont associées aux seuils définis par l'administrateur de la base de données utilisée pour décider si la relation est retenue.

Deux thésaurus assez connus sont utilisés dans ce genre d'approche : WORDNET et MeSH. MeSH (Medical Subject Heading)5 est un thésaurus médical. C'est le thésaurus d'indexation de la base bibliographique MEDLINE6. WORDNET [19] est une base de données lexicales. Les termes y sont organisés sous formes d'ensembles de synonymes et de synsets. Chaque synset est un concept lexicalisé. Ces concepts sont reliés par des relations linguistiques. WORDNET est un énorme dictionnaire hypermédia de l'anglais américain (plus de 100000 synsets). Sa richesse et sa facilité d'accès le positionnent comme un intéressant outil pour la recherche d'information ou d'autres tâches comme le traitement du langage naturel mais ce n'est pas une ontologie car les relations ne sont en aucun cas formelles.

Le projet MOMIS [13] est un exemple de projet d'intégration utilisant des OL qui vise à intégrer semi-automatiquement des données de sources structurées et semi structurées. Pour cela, chaque source de données est associée à un adaptateur qui consiste à traduire le schéma de cette source dans un modèle orienté objet commun (OQLi3) [20]. Un thésaurus global représentant les relations terminologiques entre les

Schémas des sources est d'abord construit, à l'aide de WordNet, en extrayant les termes utilisés dans les schémas des sources. Une vue intégrée est ensuite créée semi-automatiquement en se basant sur le thésaurus et les mesures d'affinité de concepts. Ces dernières calculent d'abord les similitudes entre deux concepts : affinité de nom, affinité structurelle, et affinité globale. Puis, le résultat de ce calcul est comparé avec un seuil pré-choisi afin d'introduire les concepts plus généraux correspondant à la vue intégrée. Les concepts introduits dans la vue intégrée sont des subsumants communs à plusieurs termes apparus dans plusieurs schémas

I.8.3 Intégration automatique à l'aide d'ontologie conceptuelle

La troisième génération des systèmes d'intégration consiste à associer aux données une ontologie conceptuelle (OC) qui en définit le sens. Une ontologie conceptuelle est "une spécification explicite et formelle d'une conceptualisation faisant l'objet d'un consensus" [33]. En effet, dans une conceptualisation, le monde réel est appréhendé à travers des concepts représentés par des classes et des propriétés. Des mots d'un langage naturel peuvent être associés, mais ce ne sont pas eux qui définissent le sens des concepts. C'est l'ensemble des

[5] http://www.chu-rouen.fr/cismef/
[6] MEDLINE (de l'anglais : Medical Literature Analysis and Retrieval System Online) est une base de données bibliographiques regroupant la littérature relative aux sciences biologiques et biomédicales.

caractéristiques associées à un concept ainsi que ses liens avec les autres concepts qui en définissent le sens. Une OC regroupe ainsi les définitions d'un ensemble structuré de concepts. Ces définitions sont traitables par machine et partagées par les utilisateurs du système. Elles doivent, en plus, être explicites, c'est-à-dire que toute la connaissance nécessaire à leur compréhension doit être spécifiée.

La référence à une telle ontologie est alors utilisée pour éliminer automatiquement les conflits sémantiques entre les sources dans le processus d'intégration de données. L'intégration de données est donc considérée comme automatique. Nous pouvons citer le projet PICSEL [21], OBSERVER [27], OntoBroker [22], KRAFT [30], COIN [1], etc. Comparées aux approches d'intégration utilisant des ontologies linguistiques, ces approches sont :

- plus rigoureuses : car les concepts sont définis avec plus de précision, l'appariement peut être plus argumenté,
- facilement outillables : les différentes ontologies sont basées sur le même modèle, comme OWL [24], PLib [32], etc. Une correspondance entre ontologies peut être exploitée par des programmes génériques pour intégrer les données correspondantes.

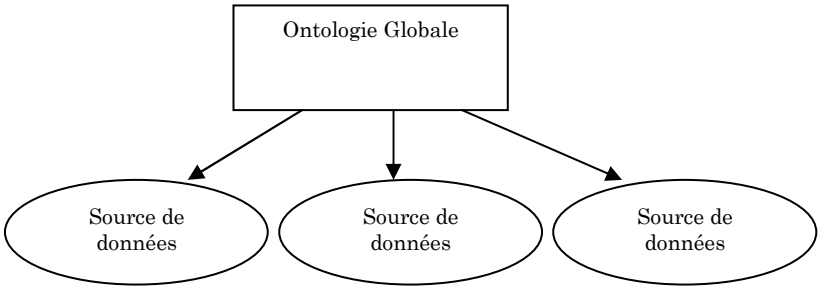


FIG I.8 - Architecture d'intégration avec une seule ontologie

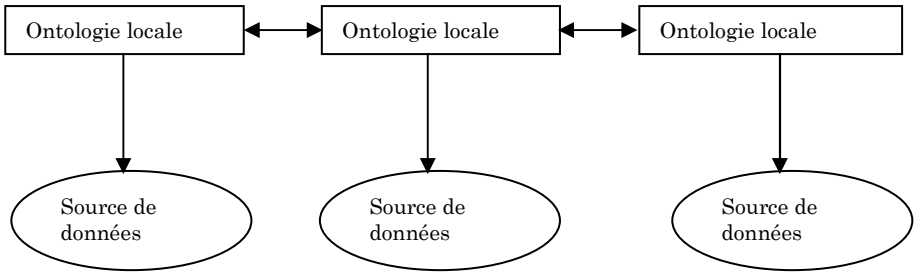


FIG I.9- Architecture d'intégration avec Ontologies Multiples

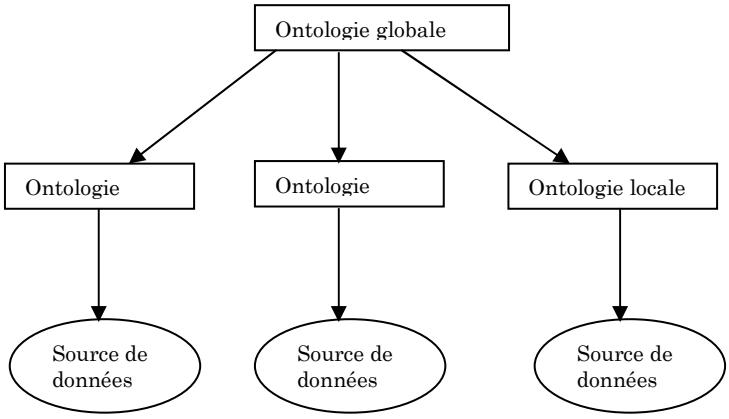


FIG I.10 - Approche hybrides

Plusieurs approches d'intégration à base ontologique ont été développées [25]. Ces dernières peuvent être divisées en trois catégories : approches avec une seule ontologie (FIG I.8), approches avec ontologies multiples (FIG I.9) et approches hybrides (FIG I.10). Dans l'approche avec une seule ontologie, chaque source référence la même ontologie globale de domaine. Les systèmes d'intégration SIMS [26] et COIN [1] sont des exemples de cette approche. En conséquence, une nouvelle source ne peut ajouter aucun nouveau concept sans exiger le changement de l'ontologie globale. Dans l'approche à multiples ontologies (exemple du projet OBSERVER [27]), chaque source a sa propre ontologie développée indépendamment des autres sources. Dans ce cas, les correspondances inter-ontologies sont difficiles à mettre en oeuvre. L'intégration des ontologies est donc faite d'une façon manuelle ou semi-automatique. [27]. Pour surmonter l'inconvénient des approches simples ou multiples d'ontologies, l'approche hybride a été proposée. Dans cette dernière, chaque source a sa propre ontologie, mais toutes les ontologies utilisent un vocabulaire partagé commun (exemple du projet KRAFT [30]).

I.9 Intégration de donnée et Ontologie conceptuelle

Les ontologies conceptuelles représentent un outil intéressant pour résoudre les problèmes liés à l'hétérogénéité sémantique. L'ontologie fournit une représentation explicite de la sémantique des données pouvant servir de base à la mise en correspondance des modèles différents. Elle est utilisée dans la plupart des systèmes à la fois comme un schéma global de données et comme une interface d'interrogation. L'utilisation d'une ontologie spécifique au domaine permet de ramener les concepts à un référentiel unique et de mesurer la distance et le recouvrement entre eux. Les concepts de l'ontologie sont liés aux termes des sources de données dans le méta-modèle global. Ces liens sont utilisés pour identifier les sources de données pertinentes et pour transformer les requêtes en sous-requêtes locales sur les sources originelles.

I.9.1 Ontologie conceptuelle et modèle conceptuel

Il est important d'expliquer la différence entre une ontologie et un modèle conceptuel afin de rendre clair l'intérêt d'une ontologie conceptuelle pour l'intégration de données. La conception du modèle d'une base de données, en particulier, est réalisée selon une approche prescriptive. Cette dernière a plusieurs implications [28] :

- seules les données pertinentes pour l'application cible sont décrites ;
- les données doivent respecter les définitions et contraintes définies dans le modèle conceptuel ;
- la conceptualisation est faite selon le point de vue des concepteurs et avec leurs conventions ;
- aucun fait n'est inconnu : c'est l'hypothèse du monde fermé.

Une telle approche engendre les problèmes d'hétérogénéité cités dans la section I.2. L'ontologie conceptuelle est un composant informatique générique capable de comprendre l'hétérogénéité des sources de données. Donc, ce point nous amène à expliquer la différence entre une ontologie et un modèle.

Une définition de MINSKY clarifie le rôle des modèles et souligne leur multiplicité et leur nécessité [29]**:"Pour un observateur B, un objet A* est un modèle d'un objet A si B peut utiliser A* pour répondre aux questions qui l'intéressent au sujet de A"**. Cette définition montre que le modèle conceptuel dépend du contexte dans lequel a été conçu. Le Problème lorsque l'on conçoit un modèle informatique est que le contexte de modélisation est défini par les buts et l'environnement de ce système. Or, les buts et l'environnement de systèmes ne sont jamais exactement identiques. Au contraire, le but d'une ontologie est d'être indépendant de tout contexte particulier qui prescrit une base de données, l'ontologie décrit ce qui existe et sera ainsi conçue selon une approche descriptive et dans un cadre consensuel sans contexte de conception ou alors en le définissant explicitement.

I.9.2 Utilisation d'ontologies conceptuelles pour l'intégration de données

L'utilisation d'une ontologie conceptuelle (OC) dans un système d'intégration de données passe par trois étapes :

- **la représentation de la sémantique des données** qui vise à interpréter le sens de chaque source en l'associant à une ontologie locale. Ce processus est effectué d'une façon manuelle ou semi-automatique. Cette étape peut être éliminée si chaque source contient a priori une ontologie locale, c'est-à-dire que la sémantique d'une source est déjà sauvegardée au sein de cette source quand on la crée.
- **l'intégration sémantique** qui vise à intégrer les ontologies des sources. Elle consiste à établir les relations sémantiques (équivalence,

subsomption) entre les concepts des ontologies. L'automatisation du processus d'intégration sémantique dépend des méthodes d'utilisation d'ontologies qui sont précisées ci-dessous.

- l'intégration de données qui vise à peupler les données dans un entrepôt pour les systèmes matérialisés ou à construire des interfaces de requêtes pour les systèmes virtuels qui fournissent une vue unique sur les données. Cette étape peut être faite par des programmes génériques qui exploitent la correspondance ontologique établie dans l'étape précédente.

On se basant sur la nature de l'étape d'intégration sémantique, il existe deux classes de système d'intégration à base ontologique: (1) intégration sémantique a posteriori, et (2) intégration sémantique a priori.

I.9.2.1 Intégration sémantique a posteriori

L'approche d'intégration sémantique a posteriori est caractérisée par les principes suivants :

- les ontologies locales sont indépendantes ;
- l'étape d'intégration sémantique est donc effectuée d'une façon manuelle ou semi-automatique. Elle consiste à établir la correspondance entre les concepts primitifs des ontologies.
- dans ce contexte, deux structures d'intégration d'ontologies sont possibles :

A. la structure "multi-ontologies" (FIG I.11-A) :

- la correspondance entre deux ontologies locales est établie directement de l'une à l'autre. Supposons qu'il existe n ontologies locales, il faut alors créer [n * (n − 1)/2] correspondances.

- l'interface utilisateur peut être créée directement à partir d'une ontologie locale quelconque.

B. la structure "ontologie partagée" (FIG I.11-B) :

- la correspondance entre deux ontologies locales est établie indirectement à travers une ontologie référencée. Cette ontologie est appelée également l'ontologie partagée du système. Une ontologie locale n'est mise en correspondance qu'avec l'ontologie partagée. Et seuls les concepts locaux intéressés par le système sont mis en correspondance avec les concepts partagés. Supposons qu'il existe n ontologies locales, il faut alors créer n articulations d'ontologies.
- l'ontologie partagée est soit une ontologie spécifique au système, soit une ontologie normalisée indépendante du système.
- l'interface utilisateur est créée à partir de l'ontologie partagée.

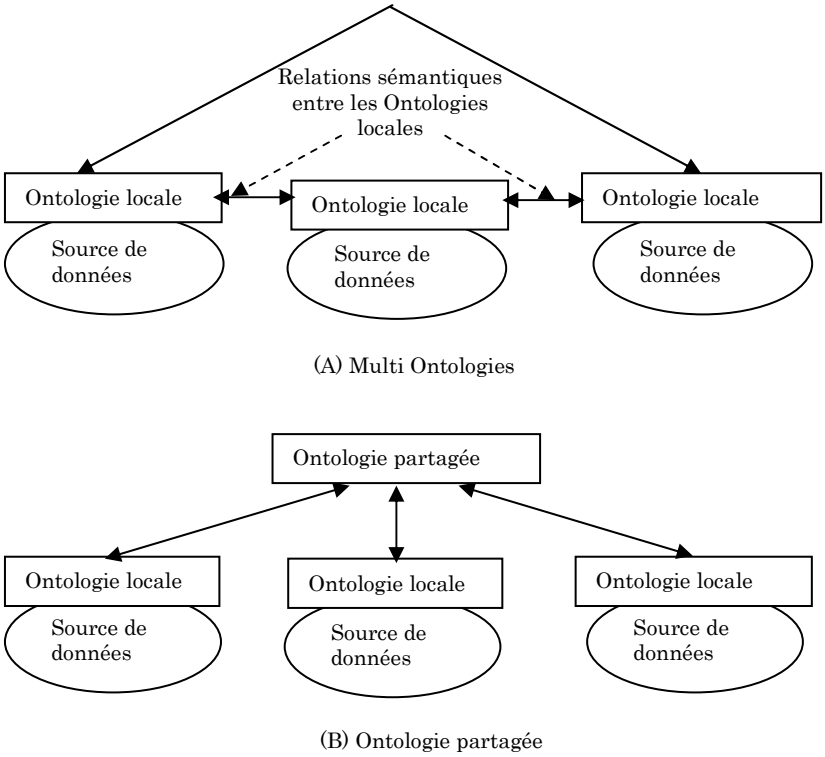


FIG I. 11 - Utilisation d'Ontologies Conceptuelles dans l'approche d'intégration sémantique a posteriori

L'avantage d'une telle approche est l'indépendance des sources de données intégrées. Par contre, l'intégration sémantique n'est pas automatique. Pour la structure "multi-ontologies", le nombre de mise en correspondance entre les ontologies est important. Pour la structure "ontologie partagée", ce nombre est diminué, et l'interface utilisateur correspond à l'ontologie partagée.

Parmi les systèmes d'intégration qui suivent cette approche, nous pouvons citer OBSERVER [27] pour la structure "multi-ontologies", et KRAFT [30] pour la structure "ontologie partagée" :

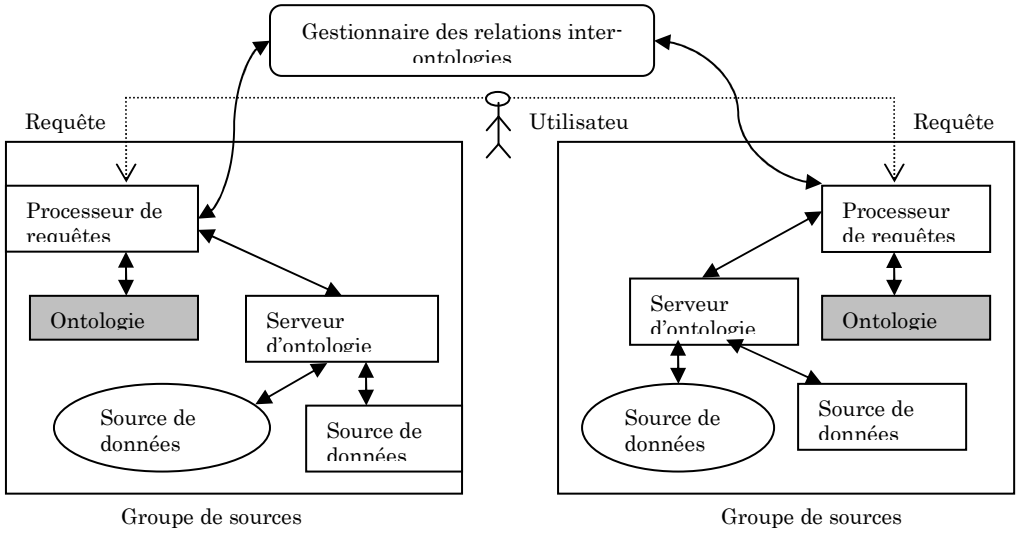


FIG I.12 -Architecture du système OBSERVER

OBSERVER: Cette association ontologie-source de données est matérialisée par des liens entre les concepts de l'ontologie et les termes de la source de données. L'intégration de plusieurs sources de données se fait par les liens sémantiques entre les concepts des ontologies. Les ontologies sont représentées dans OBSERVER en utilisant CLASSIC [27]. Le système d'intégration OBSERVER (FIG I.12) se compose de groupes de sources de données. Chaque groupe possède une ontologie locale, un serveur d'ontologies et un processeur de requêtes.

- **l'ontologie locale** : contient les définitions des concepts locaux qui représentent la sémantique des sources de données de ce groupe.

- **le serveur d'ontologie** : contient les explications sur les termes utilisés dans l'ontologie locale. Il garde également les liens des concepts ontologiques vers les données (ou plutôt vers les structures de données). Cela permet de retrouver les informations à partir des sources de données et de fournir les informations nécessaires à l'administrateur pour établir les relations sémantiques entre les ontologies locales des groupes de sources différents.

- **le processeur de requêtes** : présente à l'utilisateur une interface lui permettant de formuler sa requête en termes des concepts présents au niveau de chaque groupe.

Le lien entre groupes de sources est assuré par le module de gestionnaire des relations inter-ontologies. OBSERVER stocke dans ce module toutes les informations concernant les relations sémantiques (subsomption/équivalence) entre des concepts appartenant aux différentes ontologies. Les requêtes du

système sont d'abord formulées à partir des concepts d'une ontologie choisie par l'utilisateur. Elles sont ensuite analysées par le système en utilisant les mécanismes d'inférence ontologique afin de déterminer les sources pertinentes et de transformer la requête dans le langage de la source. D'après OBSERVER, sa stratégie permet d'éviter de mettre en place une ontologie globale et de développer des ontologies spécifiques aux besoins d'utilisateurs qu'on peut extraire à partir des ontologies locales.

KRAFT : (Knowledge Reuse and Fusion/Transformation) [30], par exemple, est un projet de recherche coopérative entre trois universités britanniques (l'Université de Aberdeen, l'Université de Cardiff et l'Université de Liverpool) avec BT (British Telecommunication). KRAFT propose une architecture d'agents dans le but d'intégration des sources de données hétérogènes. Dans KRAFT, les sources de données possèdent leurs ontologies propres. Ces ontologies sont décrites indépendamment. Elles ne sont pas obligées d'avoir un même format de représentation. Au contraire d'OBSERVER, une ontologie locale est mise uniquement en correspondance avec l'ontologie partagée du système. Cette mise en correspondance est faite d'une façon manuelle. KRAFT utilise le langage CIF (Constraint Interchange Format) pour spécifier la correspondance ontologique.

En résumé, l'intégration de données par l'intégration sémantique a posteriori est proposée pour le cas où les sources de données intégrées possèdent des ontologies locales indépendantes. Une ontologie locale respecte uniquement sa source de données. Ainsi, l'intégration sémantique est faite de façon manuelle. Elle exige donc une bonne compréhension de l'administrateur du système sur chaque source. Ceci limite la scalabilité[7] du système.

I.9.2.2 Intégration sémantique a priori

L'approche d'intégration sémantique a priori est caractérisée par les principes suivants:

- les administrateurs veulent une communication directe entre les sources de données intégrées. Chaque source reprend a priori des concepts dans une ontologie de domaine préexistante pour construire son ontologie locale. Cette ontologie de domaine est considérée comme l'ontologie globale du système. L'ontologie locale peut être vue comme un sous ensemble de l'ontologie globale.
- l'intégration sémantique est donc naturellement automatique grâce aux relations sémantiques a priori entre les ontologies locales (ces relations sont celles entre les concepts dans l'ontologie globale).
- l'interface utilisateur est créée à partir de l'ontologie globale.

[7] La capacité d'un système, ou de ses composants, à être utilisé sur des plates-formes de tailles très inférieures ou très supérieures. Il s'agit d'une extension du concept de "portabilité ".

- un système d'intégration de données suivant cette approche n'intègre que les données dont la sémantique est présentée par l'ontologie globale.

Cette approche permet d'intégrer facilement une nouvelle source dans le système si la sémantique de cette source est couverte par OG. Par contre, cet aspect limite les sources de données au niveau de leurs indépendances. Parmi les projets utilisant cette méthode, nous pouvons citer SIMS [26], PICSEL [35, 39], OntoBroker [22], BUSTER [31], COIN [1].

PICSEL, par exemple, propose une structure médiateur (FIG I.13) qui permet d'interroger des sources d'information multiples, hétérogènes et éventuellement réparties. Les systèmes médiateurs auxquels PICSEL s'intéresse regroupent un ensemble important de sources d'information XML relatives à un même domaine d'application. Dans le PICSEL : "une ontologie est un élément central. Son rôle est double. D'une part, elle fournit aux utilisateurs un vocabulaire de base approprié pour formuler leurs requêtes et interroger les sources auxquelles le médiateur a donné accès. D'autre part, elle permet la description des connaissances contenues dans les sources d'information interrogeables à l'aide d'un même vocabulaire et établit, de ce fait, une connexion entre elles [20]".

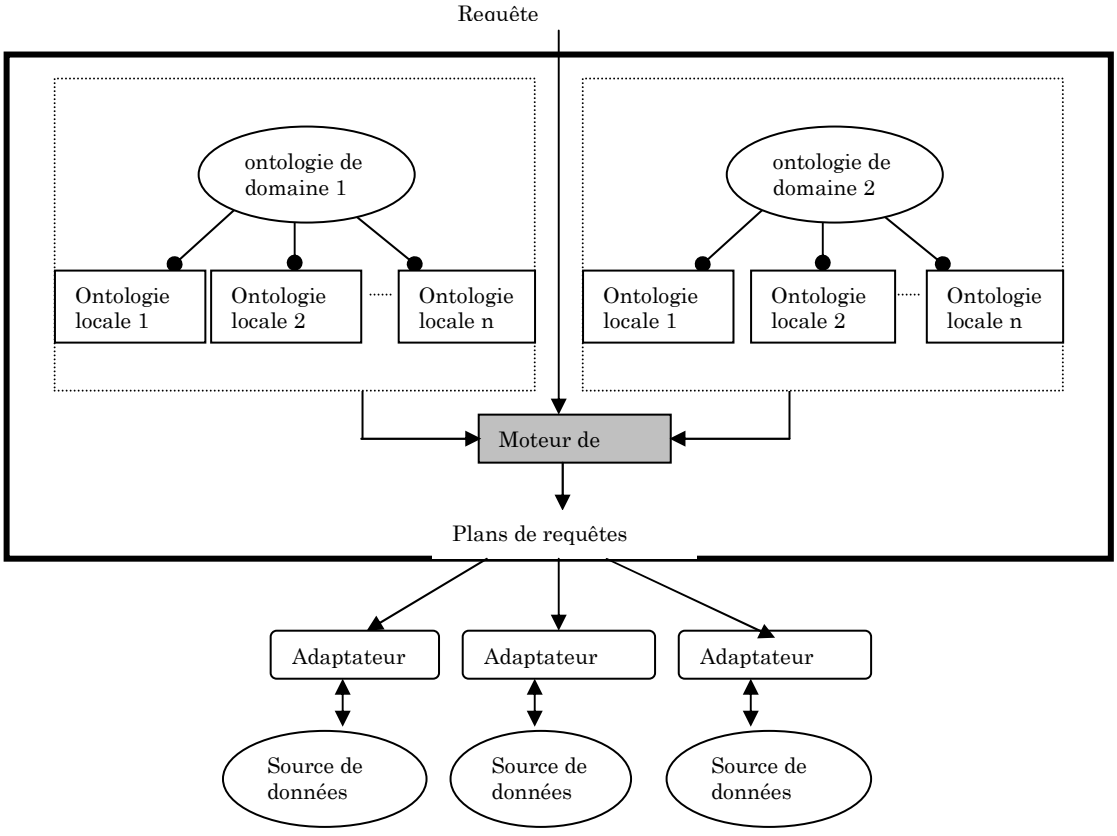


FIG I.13 - Architecture du système médiateur PICSEL

Le système de médiateur PICSEL comporte un moteur de requêtes générique et une partie de base de connaissances spécifique au domaine :

- le moteur de requêtes est conçu d'une façon générique pour être utilisable quel que soit le domaine d'application.

- la base de connaissances est spécifique au domaine appliqué. Elle se compose d'une ontologie du domaine et des ontologies locales. L'ontologie du domaine pré-existante modélise le domaine d'application et fournit ainsi, un vocabulaire structuré servant de support à l'expression des requêtes. Les ontologies locales décrivant le contenu des sources d'information sont formées a priori à partir des termes de l'ontologie du domaine.

PICSEL utilise CARIN [21] comme le langage de représentation de connaissances. Ce langage est un formalisme qui combine dans un cadre logique et homogène un langage de règles et un langage de classes. L'ontologie du domaine est représentée dans le formalisme CARIN-ALN, à l'aide de deux composantes : la composante terminologique et la composante déductive [20]. PICSEL possède également un langage de vues [20] et un langage de requêtes permettant d'exprimer, en termes de l'ontologie du domaine, respectivement, le contenu des sources et les requêtes des utilisateurs. L'ontologie globale (l'Ontologie du domaine) dans PICSEL est créée à travers l'ONTOMEDIA (Ontologie pour un MEDIAteur)[20]. Le système ONTOMEDIA permet de construire de façon semi-automatique une ontologie du domaine. Il s'agit d'un analyseur syntaxique dont la fonctionnalité consiste à appliquer un ensemble d'heuristiques pour identifier, parmi un ensemble de DTDs relatives à un domaine, les termes qui vont composer l'ontologie de ce domaine, puis de les organiser entre eux.

I.10 Conclusion

Dans ce chapitre, nous avons défini la problématique de l'intégration de données, à savoir l'hétérogénéité sémantique de données. Cette hétérogénéité provient des choix différents qui sont faits pour représenter des faits du monde réel dans un format informatique. La question fondamentale lorsque l'on veut faire interopérer des sources de données hétérogènes est d'une part, l'identification de conflits entre les concepts dans des sources différentes qui ont des liens sémantiques, d'autre part, la résolution de ces conflits entre les concepts sémantiquement liés.

Une classification proposée selon trois axes: (1) la représentation de données intégrées, (2) la mise en correspondance entre schéma global et schéma local, et (3) la nature du processus d'intégration. Le premier critère permet de déterminer le type de stockage de données du système d'intégration qui est virtuel (Médiateur) ou matériel (Entrepôt). Le deuxième critère permet d'identifier le sens de mise en correspondance entre schéma global et schéma local (GAV) ou (LAV). Ce sens influence directement la possibilité de passage à l'échelle (Web sémantique) et la complexité de traitement de requête du système d'intégration. Le dernier critère spécifie si le processus d'intégration de données est effectué d'une façon manuelle, semi-automatique ou automatique. Ce critère devient essentiel lorsque l'on veut intégrer un nombre important de sources de données indépendantes. Parmi les approches d'intégration de données, l'utilisation d'ontologies conceptuelles apparaît comme le seul moyen assurant l'automatisation du processus d'intégration sémantique de données. Certaines ontologies visent essentiellement l'inférence quand d'autres visent la caractérisation et le partage de l'information.

L'utilisation d'ontologies conceptuelles pour développer des systèmes d'intégration est récente, mais devient populaire. Nous avons présenté comment les approches existantes utilisent des ontologies conceptuelles pour résoudre les conflits sémantiques de données.Nous classifions ces approches dans deux catégories :(1) l'intégration sémantique a posteriori, et (2) l'intégration sémantique a priori. La première approche permet de garder l'indépendance de chaque source de données, mais l'intégration sémantique est faite manuellement. Au contraire, la deuxième approche n'intègre que les sources de données dont la sémantique est représentée a priori en connexion avec une ontologie de domaine, et en conséquence son processus d'intégration sémantique est automatique.

Dans ce contexte, le processus d'intégration de données se décompose en deux étapes : une intégration des ontologies puis une intégration des données. Notre travail, détaillé dans le chapitre suivant, se situe dans la première à traiter le problème de l'hétérogénéité sémantique par l'intégration des ontologies formelles représentées explicitement dans chaque source de données.

Chapitre II : L'intégration de données et les Ontologies

II.1 Introduction

L'explosion du nombre de sources d'informations accessibles via le Web multiplie les besoins de techniques permettant l'intégration de ces sources. En définissant les concepts associés à des domaines particuliers, les ontologies sont un élément essentiel des systèmes d'intégration, car elles permettent à la fois de décrire le contenu des sources à intégrer et d'expliciter le vocabulaire utilisable dans les requêtes des utilisateurs. La tâche d'alignement d'ontologies (recherche de mappings, appariements ou mises en correspondance) est particulièrement importante dans les systèmes d'intégration puisqu'elle autorise la prise en compte conjointe de ressources décrites par des ontologies différentes.

L'intégration logique de plusieurs ontologies fournit à l'utilisateur une vision unifiée des différentes sources. Il peut s'agir de définir un système permettant d'interroger de manière unifiée des sources de données. Chacune d'entre eux possède son propre référentiel. Un système unique doit respecter l'intégrité des sources locales. Il faut définir une ontologie globale, des règles de dérivation des ontologies locales et des règles de traduction des ontologies locales vers l'ontologie globale. Pour l'utilisateur, tout se passe comme s'il n'y avait qu'une seule ontologie.

Les ontologies sont très souvent représentées par des hiérarchies taxonomiques des classes et des relations, bien qu'elles n'aient pas besoin d'être limitées à ces formes. Les liens entre des classes ou entre des relations dans ces taxonomies sont les liens de subsomption. Une ontologie simple, qui n'offre pas de relations entre classes (autre que ce lien de subsomption) et donc est sans hiérarchie des relations, peut être considérée comme un schéma. Du fait de cette observation, il est possible d'adapter des algorithmes d'alignement des schémas de base de données, qui ont déjà été bien étudiés dans le domaine des bases de données, dans le contexte de l'intégration de données et de la traduction de données, à l'alignement des ontologies dans le contexte de la représentation des connaissances.

Dans ce chapitre, nous allons présenter des notions sur les ontologies (langages, formalismes..), le rôle des ontologies dans le processus d'intégration de données guidé par les ontologies et les différentes méthodes de mesure de similarité pour aligner l'ontologie globale avec les ontologies locales.

II.2 Système de Médiation

Nous avons exposé (dans les sections I.4 et I.5) brièvement différents systèmes intégrés, c'est-à-dire des architectures qu'il est possible de mettre en placer pour faire coopérer de manière relativement transparente (selon le niveau d'intégration) des sources de données initialement indépendantes. Nous avons ainsi vu les systèmes multibases, les systèmes fédérés et les systèmes de médiation. Nous avons également présenté les entrepôts de données car il existe une phase d'intégration de données pour les constituer.

L'approche d'intégration par médiation constitue sans doute aujourd'hui la solution la plus courante pour relier différentes sources qui cette fois,ne correspondent pas nécessairement à des bases de données. La notion de médiateur a été initialement proposée par [59]. Il définit un médiateur comme suit :

« **A mediator is a software module that exploits encoded knowledge about some sets or subsets of data to create information for a higher layer of applications** ».

Un médiateur doit être vu comme une couche logicielle permettant d'accéder de manière transparente pour l'utilisateur à différentes ressources (BD, fichiers) réparties et hétérogènes. Pour cet accès, le médiateur exploite des connaissances (Ontologies) qui sont utiles à différents services (interrogation, localisation des ressources notamment).

L'approche médiation à base ontologie étant adoptée, il convient maintenant d'exposer la manière d'aboutir à un tel système. Quelle est la démarche à suivre pour intégrer les ontologies dans le médiateur ? Dans quel niveau les ontologies peuvent servir à la fois comme support d'interrogation et permettre la location des ressources pertinentes.

Pour répondre à ces interrogations, nous avons décidé de présenter le processus d'intégration proposé dans [60]. Il s'agit d'un processus destiné à la conception d'un système de médiation. Sa présentation nous permettra de bien préciser l'étape que nous traitons dans l'intégration.

II.3 Etapes d'intégration des sources de données

Le processus d'intégration est décomposé en trois phases distinctes :

La pré-intégration : Cette phase vise à préparer l'intégration des schémas en les rendant plus homogènes. Elle consiste principalement à traduire les schémas initiaux dans un modèle de données commun (réduction de l'hétérogénéité syntaxique). Elle s'attache également à enrichir leur sémantique.

L'identification des correspondances:Durant cette phase, les correspondances entre les éléments des schémas sources sont détectés et formalisées de même que les différents conflits.

L'intégration: Cette phase finale fournit les règles de traduction selon (L'approche GAV ou LAV) permettant de passer des schémas source au schéma intégré et inversement (« mapping »).

Ce processus est illustré en figureII.1. Nous exposons plus en détail l'étape **« identification des correspondances »** dans la section suivante parce que notre travail s'inscrit dans cette étape et qui consiste à trouver ces correspondances sémantiques, en utilisant le calcul de similarité, entre les concepts de l'ontologie globale et ceux des ontologies locaux, le résultat est utiliser dans l'étape suivante pour localiser les sources à intégrer ainsi que la réécriture de la requête utilisateur.

II.3.1 Identification des correspondances

Les systèmes de médiation (intégration virtuelle) présentent à l'utilisateur une interface uniforme des différentes sources de données, au travers d'un modèle commun. Les enjeux de l'intégration de schéma, problème largement abordé dans les bases de données, sont la prise en compte de l'hétérogénéité des sources, la définition et la modélisation du schéma global, la définition et la gestion des règles de transformation établissant les correspondances entre les sources et le schéma de médiation, ainsi que la prise en compte de la sémantique des sources, la gestion de la cohérence et l'évolution de schéma.

Une difficulté importante dans l'intégration de sources hétérogènes réside dans Le matching de schémas. Ce dernier consiste à trouver les correspondances sémantiques entres les éléments de deux schémas [38]. Il s'agit d'un problème critique dans le domaine de l'intégration de données. Les sources de données doivent être décrite de manière à faciliter leur compréhension. Cette description amène à utiliser des ontologies qui représentent, dans un cadre formel, les connaissances d'un domaine. Elles peuvent être utilisées dans le processus d'intégration de données pour décrire la sémantique des sources d'information et rendre ainsi leur contenu explicite. Elles permettent alors l'identification et l'association de concepts sémantiquement correspondants. Plusieurs approches ont été étudiées. Elles sont fondées sur l'utilisation soit d'une seule ontologie, soit de plusieurs, ou également d'une approche hybride (voir I.8.3). Dans cette dernière, chaque source possède sa propre ontologie de description mais, afin de permettre de rendre chacune comparable avec une autre, elles se réfèrent toutes à un vocabulaire partagé.

Dans un système d'intégration de données guidé par les ontologies, par l'utilisation d'une approche hybride, Les ontologies permettent, d'une part, utilisées comme schéma global d'interrogation (ontologie globale) et ,d'autre part,décrire le contenu des sources de données (ontologie local). En effet, le

système d'intégration doit posséder la connaissance du contenu de chaque source avec les correspondances sémantiques entre les concepts de l'ontologie globale et ceux de l'ontologie locale pour déterminer de quelle manière il doit effectuer les sous-requêtes. Dans la littérature, la recherche de ces correspondances s'appelle l'alignement des ontologies. Nous montrerons au chapitre III que cet aspect est au coeur de notre travail.

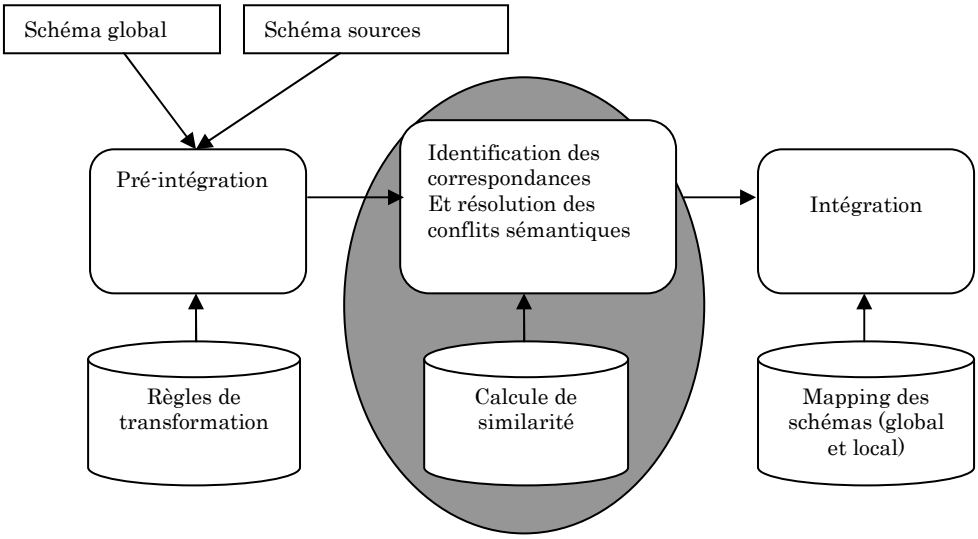


FIG II.1 - Les étapes du processus d'intégration des sources de données

II.4 Notion d'ontologie

Historiquement, l'ontologie est un concept philosophique. Il désigne la science de l'être en général. Elle décrit une théorie à propos de la nature de l'existence [34]. Plus tard, l'ontologie est apparue en pleine lumière dans le domaine de l'intelligence artificielle, afin de résoudre les problèmes de modélisation des connaissances et, plus précisément, en ingénierie des connaissances. Ceci a engendré de nombreuses définitions que nous allons résumer dans les paragraphes suivants en présentant les points de vue philosophique et informatique.

Vue philosophique : dans le domaine de la philosophie, l'ontologie est considérée comme une branche de la métaphysique qui s'intéresse à la nature et l'organisation de la réalité. Elle a une signification plus large, celle de « science de ce qui existe » dans laquelle on ne cherche pas à expliquer le monde, mais à le représenter. Elle s'applique à « l'être en tant qu'être », indépendamment de ses déterminations particulières.

Vue informatique : d'une manière générale, une ontologie est vue comme un ensemble de concepts permettant de modéliser un ensemble de connaissances dans un domaine donné. Un concept peut présenter plusieurs sens thématiques. Les concepts sont liés entre eux par des relations sémantiques,

des relations de composition et d'héritage. Une définition générale a été donnée par Thomas R. Gruber [35] où il décrit une ontologie comme une spécification explicite d'une conceptualisation modélisant des concepts et les relations entre concepts.

II.5 Eléments constitutifs de l'ontologie

Les ontologies sont, à l'heure actuelle, au coeur des travaux menés en ingénierie des connaissances. Elles permettent de représenter les connaissances et les manipuler automatiquement, tout en gardant leur sémantique. Les connaissances sont définies à travers des concepts. Les liens entre concepts sont appelés relations. Afin de relier les concepts, l'ontologie se présente, généralement, sous forme d'une organisation hiérarchique des concepts.

II.5.1 Concepts: Ils représentent des groupes d'individus partageant les mêmes caractéristiques. Ils correspondent aux entités "génériques" d'un domaine d'application. Les concepts d'une ontologie sont organisés hiérarchiquement par une relation d'ordre partiel qui est la relation de subsomption ("est-une") permettant d'organiser sémantiquement les concepts par niveau de généralité: intuitivement, un concept C1 subsume un concept C2 si C1 est plus général que C2 au sens où l'ensemble d'individus représenté par C1 contient l'ensemble d'individus représenté par C2. Par exemple, le concept Personne subsume le concept Femme. Formellement, C1 subsume C2 si dans tout contexte : $x \in C2 \Rightarrow x \in C1$.

II.5.2 Propriétés : Elles permettent de décrire et de caractériser des instances appartenant à une (ou plusieurs) classes de l'ontologie par des valeurs d'éléments caractéristiques ou des associations avec d'autres concepts.

II.5.3 Relations : Elles constituent des types d'associations prédéfinis entre les concepts. La relation commune qui est supportée par n'importe quel formalisme d'ontologie est la subsomption : "est-un". Elle organise les concepts en une hiérarchie, où tout concept se compose d'une description propre définie par des propriétés locales et d'une description partagée avec ses subsumants comme c'est le cas entre classes dans un langage à objets [36].

II.5.4 Instances: Elles représentent des individus du domaine de l'ontologie.

II.5.5 Axiomes : Ils explicitent les énoncés conceptuels toujours vrais dans le contexte de l'ontologie. Ils peuvent être utilisés pour contrôler la correction des concepts ou des relations, ou pour déduire de nouveaux faits.

II.6 Formalismes de représentation

Une ontologie, telle qu'elle est décrite dans la section précédente (II.3), a besoin d'être représentée formellement. Plus encore, elle doit représenter

l'aspect sémantique des relations liant les concepts. A cet effet, de nombreux formalismes ont été développés [37].

II.6.1 Les réseaux sémantiques : un réseau sémantique est une structure de graphe qui encode les connaissances ainsi que leurs propriétés. Les noeuds du graphe représentent des objets (concepts, situations, événements, etc) et les arcs expriment des relations entre ces objets.

II.6.2 Les Frames : un schéma est une structure de données complexe. Il est considéré comme un prototype décrivant une situation ou un objet standard. Il sert de référence pour comparer des objets que l'on désire reconnaître, analyser ou classer. Les prototypes doivent prendre en compte toutes les formes possibles d'expression de la connaissance. Un schéma est caractérisé par des attributs, des facettes et des relations.

- Les attributs définissent la structure de données ;
- Les facettes définissent la sémantique des attributs et décrivent l'ensemble des valeurs possibles pour cet attribut ;
- Les relations expriment la sémantique d'héritage. Elles peuvent être générales (spécialisation, composition) ou spécifiques à une application.

II.6.3 La logique de description : Les Logiques de Description (LD), appelées parfois les logiques terminologiques, ont pour base les frames, la logique des prédicats et les réseaux sémantiques. Ces systèmes s'appuient sur un modèle logique pour bâtir des représentations autour de la notion de concept structuré et pour organiser des concepts dans une hiérarchie de subsomption (notée$\sqsubseteq$); ils manipulent et mettent à jour la hiérarchie de concepts par un algorithme de classification de concepts, le "classifieur".

II.7 Les langages pour les ontologies

Un langage d'ontologie est un langage formel permettant de représenter une ontologie. Nous pouvons utiliser RDF(S) pour représenter des ontologies simples. Les ontologies en RDF(S) peuvent être sérialisées en des langages tels que XML. Cependant, une ontologie est employée pour représenter des connaissances dans un domaine, donc il est nécessaire de disposer d'un langage aussi expressif pour les représenter, et RDF(S) ne répond pas à ces besoins. Par exemple, en utilisant RDF(S), on ne peut pas représenter la cardinalité d'une relation ou exprimer des caractéristiques des relations telles que la transitivité, la symétrie ou la fonctionnalité, ou de faire des restrictions pour certaines classes… Ainsi, le W3C a recommandé un langage standardisé plus puissant au niveau expressivité, qui est spécialement conçu pour représenter des ontologies dans le Web sémantique. Cela permet avec facilité de créer, partager et échanger des connaissances dans le Web sémantique. Le langage d'ontologie recommandé est le langage OWL. Il est

dérivé du langage DAML+ OIL[8]. OWL couvre la plupart des caractéristiques du langage DAML+OIL et renomme la plupart de ses primitives.

Le langage d'ontologie OWL est divisé en trois sous-langages avec une puissance d'expressivité dégressive : OWL Full, OWL DL et OWL Lite. En ce qui concerne de la compatibilité de ces sous-langages, le sous-langage OWL Full peut être considéré comme une extension de RDF, tandis que OWL Lite et OWL DL peuvent être considérés comme des extensions d'une vue restreinte de RDF. OWL Full est une extension de OWL DL, et ce dernier est une extension de OWL Lite. Une ontologie légale en OWL Lite est aussi légale en OWL DL et OWL Full. Tous les documents en OWL (Full, DL, Lite) sont des documents valides en RDF, et un document RDF est un document OWL Full, mais seulement quelques documents en RDF sont des documents légaux en OWL Lite ou OWL DL.

II.8 Les méthodes de base pour mesurer la similarité

II.8.1 La similarité

Dans la plupart des approches dans le contexte d'alignement des ontologies, la notion de similarité sémantique est vue comme celle de la similarité topologique en mathématiques, où on l'associe à une fonction, appelée fonction de similarité. La définition de cette dernière peut changer selon les approches, selon les propriétés souhaitées. La valeur de cette fonction est souvent comprise entre 0 et 1, ce qui permet des possibilités d'interprétation probabiliste de la similarité. Des propriétés ou des caractéristiques communes possibles de la fonction sont des caractéristiques positives, auto similaires ou maximales, symétriques ou réflexives. On peut aussi trouver d'autres caractéristiques telles que la finitude[9] ou la transitivité.

> ***Définition1 :*** *(Similarité). La similarité S: O*O→R est une fonction d'une paire d'entités à un nombre réel exprimant la similarité entre ces deux entités telle que:*
>
> - $\forall a,b \in O, S(a,b) \geq 0$ *(positivité)*
> - $\forall a,b,c \in O, S(a,a) \geq S(b,c)$ *et* $.S(a,a) = S(a,b) \Leftrightarrow a = b$ *(auto similarité)*
> - $\forall a,b \in O, S(a,b) = S(b,c)$ *(symétrie)*
> - $\forall a,b,c \in O, S(a,b) = S(b,c) \Rightarrow S(a,b) = S(a,c)$ *(transitivité)*
> - $\forall a,b \in O, S(a,b) \leq \infty$ *(finitude)*

La dissimilarité est parfois utilisée au lieu de la similarité. Elle est définie de manière analogue à la similarité, sauf qu'elle n'est pas transitive :

[8] http://www.daml.org/2001/03/daml+oil-index
[9] Caractère de ce qui est fini, de ce qui est limité.

Défmition2 : *(Dissimilarité). La dissimilarité DS: O x O→R est une fonction d'une paire d'entités à un nombre réel exprimant la dissimilarité entre ces deux entités telle que:*

- $\forall a,b \in O, DS(a,b) \geq 0$ *(positivité)*
- $\forall a,b,c \in O, DS(a,a) \leq DS(b,c)$ *et* $.DS(a,a) = 0$ *(minimalité)*
- $\forall a,b \in O, DS(a,b) = DS(b,a)$ *(symétrie)*
- $\forall a,b \in O, DS(a,b) \leq \infty$ *(finitude)*

La distance est une mesure utilisée aussi souvent que les mesures de similarité. Elle mesure la dissimilarité de deux entités, elle est inverse de la similarité : si la valeur de la fonction de similarité de deux entités est élevée, la distance entre elles est petite et vice-versa. Elle est donc définie dans [41] comme suit :

Définition3 : *(Distance). La distance D : O x O→R. est une fonction de la dissimilarité satisfaisant la difinitivité et l'inégalité triangulaire:*

- $\forall a,b \in O, D(a,b) = 0 \Leftrightarrow a = b$ (définitivité)
- $\forall a,b,c \in O, D(a,b) + D(b,c) \geq D(a,c)$ (Inégalité triangulaire)

Les valeurs de similarité sont souvent normalisées pour pouvoir être combinées dans des formules plus complexes. Si la valeur de similarité et la valeur de dissimilarité entre deux entités sont normalisées, notées S et DS, alors on a S +DS =1.

Définition4 : *(Normalisation). Une mesure est normalisée si les valeurs calculées par cette mesure ne peuvent varier que dans un intervalle de 0 à 1. Ces valeurs calculées sont appelées valeurs normalisées. Les fonctions du calcul sont appelées fonctions normalisées et notées* $\overline{f}$ *.*

Les mesures de la similarité, de la dissimilarité, de la distance peuvent être classées selon la nature des entités que l'on veut comparer: des termes,des chaînes de caractères, des structures, des instances (des individus des classes)et des modèles théoriques. Une synthèse des travaux présentés dans [40] et [43] résume les différentes mesures de la similarité, catégorisées selon les techniques utilisées.

II.8.2 Les méthodes terminologiques

Ces méthodes se basent sur la comparaison des termes ou des chaînes de caractères ou bien les textes. Elles sont employées pour calculer la valeur de la similarité des entités textuelles, telles que des noms, des étiquettes, des commentaires, des descriptions...Ces méthodes peuvent encore être divisées en deux sous-catégories: l'une contient des méthodes qui comparent des termes en

se basant sur les caractères contenus dans ces termes et l'autre utilise certaines connaissances linguistiques.

II.8.2.1 Les méthodes se basent sur des chaînes de caractères

Ces méthodes analysent la structure des chaînes de caractères, l'ordre des caractères dans la chaîne,le nombre d'apparitions d'une lettre dans une chaîne pour concevoir des mesures de la similarité. Par contre, elles n'exploitent pas la signification des termes. Par exemple, les mesures dans cette catégorie retournent une grande valeur de similarité (jusqu'à 1) si elles comparent les termes « Voiture » et « voitures », mais une petite valeur, voire la valeur 0, si elles comparent les termes « voiture » et «Automobile».

Les résultats de la comparaison des chaînes de caractères seront améliorés si ces chaînes sont « nettoyées » ou traitées avant de les fournir aux formules calculant la similarité. Cette phase est appelée la phase de normalisation ou de normalisation textuelle, qui diffère de la normalisation des valeurs de similarité dans un intervalle de [0, 1] discutée ci-dessus (Définition 4). Les différents types de normalisation textuelle sont ceux empruntés au domaine de traitement automatique de la langue naturelle (TALN) :

- **Normalisation des caractères :** ce type de normalisation convertit toutes les Majuscules dans une chaîne de caractères en leurs formes minuscules ou vice-versa. Par exemple, la chaîne de caractères (AutomobileS) sera convertie à (automobiles) et ensuite, elle est considérée comme égale exactement à l'autre Chaîne de caractères (automobiles)

- **Normalisation des espaces :** ce type de normalisation remplace toutes les séquences consécutives des espaces, des tabulations, des retours de chariot (les caractères CR) trouvées dans une chaîne de caractères par un seul caractère d'espace. Par exemple, l'expression «ma voiture » est normalisée à « ma voiture »

- **Suppression des signes diacritiques ou des accents (aigus, graves…) :** ce type de traitement remplace des caractères avec des signes diacritiques par caractères correspondants sans signes diacritiques. Par exemple, le mot «gâteau » est remplacé par le mot «gateau » sans changer la signification du mot. Cependant, certaines suppressions changeront la signification du terme: « là » (adverbe de lieu) et «la » (article).

- **Suppression des chiffres**.

- **Élimination des ponctuations.**

- **Élimination des mots vides** (les mots contenant peu d'informations tels que « est », « un », « les »…)
- **Suppression des affixes** (préfixes, suffixes).
- **Extension des abréviations.**
- **Tokenisation.**
- **Lemmatisation** (passer au singulier, à l'infinitif pour les verbes, au masculin pour les adjectifs…).

Il existe plusieurs mesures calculant la valeur de similarité ou la distance entre deux chaînes de caractères dans la littérature telles que la similarité de **Jaccard**, la distance de **Hamming** et distance de **Levenshtein**. Nous présentons ici deux mesures les plus utilisées dans les approches d'alignement d'ontologies dans le cadre du Web sémantique.

Si nous considérons une chaîne de caractères **s** comme un ensemble de caractères **S**, la similarité de Jaccard entre deux chaînes est définie ainsi :

> ***Définition5 :*** *(Similarité de Jaccard). Soit s et t deux chaînes de caractères. Soit S et T les ensembles des caractères des s et t respectivement. La similarité de Jaccard est une fonction de la similarité SJaccard : S x S [0, 1] telle que:*
>
> $$\overline{S_{Jaccard}}(s,t) = \frac{|S \cap T|}{|S \cup T|}$$

La métrique Jaro[44] produit la similarité entre deux chaînes de caractères en se basant sur le nombre et l'ordre des caractères communs entre elles.

> ***Définition6 (Distance de Jaro).*** *Soit s et t deux chaînes de caractères. Soit N_c le nombre des caractères communs apparaissant dans les deux chaîne dans une distance de moitié de la longueur de la chaîne la plus courte. Soit N_t le nombre des caractères transposés, qui sont des caractères communs apparaissant dans des positions différentes. La distance de Jaro est une fonction de la dissimilarité DSJaro:S x S→ [0, 1] telle que :*
>
> $$\overline{DS_{Jaro}}(s,t) = 1 - \frac{1}{3}\left(\frac{Nc}{|s|} + \frac{Nc}{|t|} + \frac{Nc - Nt/2}{Nc}\right)$$

II.8.2.2 Les distances basées dur les tokens

Les mesures présentées ci-dessus s'adaptent bien lorsque l'on veut comparer deux termes ou deux courtes chaînes de caractères. Il existe aussi des cas où l'on a besoin de comparer des textes longs ou bien des documents textuels. Dans ces cas, ces entités sont découpées en plusieurs morceaux, appelés tokens. Elles deviennent des ensembles des tokens, et la similarité entre elles est produite grâce aux mesures de similarité basées sur des tokens.

II.8.2.3 Les méthodes linguistiques

La similarité entre deux entités représentées par des termes peut aussi être déduite en analysant ces termes à l'aide des méthodes linguistiques. Ces méthodes exploitent essentiellement des propriétés expressives et productives de la langue naturelle [45].Les informations exploitées peuvent être celles intrinsèques(des propriétés linguistiques internes des termes telles que des propriétés morphologiques ou syntaxiques) ou celles extrinsèques (employant des ressources externes telles que des vocabulaires ou des dictionnaires).

Les méthodes intrinsèques:une même entité ou un même concept peut être référencé par plusieurs termes (synonymie) ou par plusieurs variantes d'un même terme. Les méthodes intrinsèques fonctionnent avec le principe de chercher la forme canonique ou représentative d'un mot ou d'un terme (lemme) à partir de ses variantes linguistiques (lexème). La similarité entre deux termes est donc décidée en comparant leurs lemmes. Par exemple, le résultat de la mesure de similarité exacte de deux mots « ran » et « running » sera égal à 0 (c-a-d. ils sont différents), alors que le résultat de la même mesure pour les lemmes de ces mots sera égal à 1, ce qui indique que « ran » et « running » sont similaires.

La recherche du lemme d'un mot peut être effectuée dans un dictionnaire. Une autre approche qui est automatique et plus légère et plus efficace est d'utiliser des stemmers. Un stemmer est un programme ou un algorithme qui détermine la forme radicale à partir d'une forme infléchie ou dérivée d'un mot donné. Les radicaux (stems) trouvés par les stemmers n'ont pas besoin d'être identiques à la racine morphologique du mot. Il suffit que les mots similaires soient associés à un même radical, même si ce radical n'est pas une racine de mot valide. Un stemmer pour le français, par exemple, devrait identifier les chaînes de caractères « maintenaient », «maintenait»,«maintenant»,ou « maintenir » comme basées sur la racine "mainten".

Les méthodes extrinsèques: Ces méthodes calculent la valeur de similarité entre deux termes en employant des ressources externes telles que des dictionnaires, des lexiques ou des vocabulaires. La similarité est décidée grâce aux liens sémantiques déjà existants dans ces ressources externes

tels que des liens synonymes (pour l'équivalence),des liens hyponymes/ hypernymes (pour la subsomption). Par exemple, à l'aide des ressources des synonymes, «voiture» et « bagnole » sont dites similaires. Typiquement, WordNet[10] un système lexicologique, est employé pour trouver des relations telles que la synonymie entre des termes, ou pour calculer la distance sémantique entre ces termes, en utilisant des liens sémantiques dans WordNet, afin de décider s'il existe une relation entre eux.

II.8.3 Les méthodes structurelles

Ce sont des méthodes qui déduisent la similarité de deux entités en exploitant des informations structurelles lorsque les entités en question sont reliées aux autres par des liens sémantiques ou syntaxiques, formant ainsi une hiérarchie ou un graphe des entités. Nous appelons méthodes structurelles internes les méthodes qui n'exploitent que des informations concernant des attributs d'entité, et méthodes structurelles externes les autres qui considèrent des relations entre des entités.

II.8.3.1 Les méthodes structurelles internes

Ces méthodes calculent la similarité entre deux entités en exploitant des informations des structures internes de ces entités. Dans la plupart des cas, ce sont des informations concernant des attributs de l'entité, telles que des informations du co-domaine des attributs, celles de la cardinalité des attributs, celles des caractéristiques des attributs (la transitivité,la symétrie), ou celles des autres types de restriction sur des attributs. Par exemple, en considérant l'entité :le concept « Humain », nous pouvons exploiter des informations concernant des attributs de ce concept tels que l'intervalle des valeurs de donnée pour l'attribut «Age », à savoir [0, 150] ; la cardinalité de l'attribut «a épouse», à savoir 1 ; ou bien la caractéristique transitive de l'attribut « a Ascendant ».

Dans le domaine des bases de données, plusieurs méthodes ont été proposées pour calculer la similarité entre deux éléments de deux schémas de base de données, en se basant sur les contraintes à propos de ces éléments. Dans la revue [42], nous pouvons trouver l'algorithme Cupid [46] qui cherche des correspondances entre des éléments en se basant entre autres, sur la compatibilité des attributs, des types de données ;l'approche SEMINT [47] qui identifie des correspondances des attributs en se basant sur des informations de schéma (telles que des types de donnée, la longueur, la précision, l'existence des clés,des contraintes de co-domaine ou de valeur, l'autorisation des valeurs nulles…) et sur les statistiques des instances (tel que le maximum, le minimum,la moyenne,le coefficient de variance,

[10] http://wordnet.princeton.edu/

l'existence des valeurs nulles ou des décimales, le groupe, ou le nombre des segments).

II.8.3.2 Les méthodes structurelles externes

Contrairement aux méthodes internes, qui exploitent des informations des attributs d'entité, les méthodes structurelles externes exploitent des relations entre des entités elles-mêmes,quisont souvent des relations de subsomption (is-a ou spécialisation) ou de méréologie[11] (part-whole). Avec ces relations, les entités sont considérées dans des hiérarchies et la similarité entre elles est déduite de l'analyse de leurs positions dans ces hiérarchies. L'idée de base est que si deux entités sont similaires, leurs voisines pourraient également être d'une façon ou d'une autre similaires. Cette observation peut être exploitée de plusieurs manières différentes en regardant des relations avec d'autres entités dans des hiérarchies. Deux entités peuvent être considérées similaires si :

- Leurs super-entités directes (ou toutes leurs super-entités) sont similaires.
- Leurs sœurs (ou toutes leurs sœurs, qui sont les entités ayant la même super-entité directe avec les entités en question) sont déjà similaires.
- Leurs sous-entités directes (ou toutes leurs sous-entités) sont déjà similaires.
- Leurs descendants (entités dans le sous-arbre ayant pour racine l'entité en question) sont déjà similaires.
- Toutes (ou presque toutes) leurs feuilles (les entités de même type, qui n'ont aucune sous-entité, dans le sous-arbre ayant pour racine l'entité en question) sont déjà similaires.
- Toutes (ou presque toutes) les entités dans les chemins de la racine aux entités en question sont déjà similaires.

[11] La **méréologie** est une collection de systèmes formels axiomatiques qui traitent des relations entre la partie et le tout. La méréologie est à la fois une application de la logique des prédicats et une branche de l'ontologie, en particulier de l'ontologie formelle.

II.8.4 Les méthodes sémantiques

Les méthodes sémantiques se basent sur des modèles de logique (tels que la satisfiabilité propositionnelle(SAT[12]),la SAT modale ou les logiques de descriptions) et sur des méthodes de déduction pour déduire la similarité entre deux entités.Les approches dans [48,49,50] emploient des techniques de satisfiabilité propositionnelle(SAT)pour vérifier la validité d'un ensemble de formules propositionnelles qui est construit en traduisant des relations déjà connues et des relations à vérifier entre des entités vers des formules propositionnelles.

L'approche dans [49] étend les méthodes proposées ci-dessus, qui sont pour le modèle de la SAT propositionnelles, vers le modèle de la SAT modale, qui peut aussi contenir des prédicats binaires. La limite du premier modèle est qu'il n'accepte que des prédicats unaires qui sont des entités comme des concepts, des classes. Le dernier permet de calculer en plus avec des prédicats binaires tels que des relations, des attributs ou des propriétés (slots) et d'employer des opérateurs de la logique modale. La validité de l'ensemble de formules formulées en logique modale est aussi vérifiée en utilisant des procédures de recherche de la satisfiabilité (SAT). Si la validité est satisfaite, les relations hypothétiques entre des entités, qui sont des traductions de la requête sur la relation entre ces entités en logique modale, sont confirmées.

Les techniques des logiques de description (telles que le test de subsomption) peuvent être employées pour vérifier des relations sémantiques entre des entités telles que l'équivalence (la similarité est égale à 1), la subsomption (la similarité est de 0 à 1)ou l'exclusion (la similarité est égale à 0),et permettent donc de déduire la similarité de deux entités.

[12] Soit une formule logique sous forme normale conjonctive (CNF). Cette CNF est *satisfaisable* s'il est possible d'associer une valeur logique booléenne à chacune de ses variables de telle manière que cette formule soit logiquement vraie.

II.9 Conclusion

Lors de l'intégration de sources de données hétérogènes, une des tâches les plus importantes est la transformation de ces sources par une phase de recherche de correspondances entre les schémas des sources à intégrer. Le schéma matching apparaît comme une phase essentielle dans le processus d'intégration de données. Dans des sources de données guidées par ontologie, dont chacune contient une ontologie locale qui référence une ontologie de domaine partagée, un schéma peut être considéré comme une partie de l'ontologie locale.

Le processus d'intégration sémantique intègre d'abord les ontologies puis les données. L'approche que nous avons proposée est basée sur l'alignement des ontologies des sources à intégrer avec l'ontologie globale spécifique à un domaine. Le résultat de ce processus d'alignement, qui est un ensemble de correspondances sémantiques, peut être appliqué dans le cas d'intégration de données. Nous avons montré que ces correspondances facilitent le processus d'intégration de données, dans notre cas un médiateur, au niveau de localisation des sources ainsi que la réécriture de requête en sous requêtes adressées au adaptateurs.

Dans cette partie, nous avons examiné les techniques et les méthodes utilisées dans la littérature qui attaquent le problème de recherche de la similarité, de la dissimilarité ou de la correspondance entre deux entités en général, qu'elles apparaissent dans des schémas, ou dans des ontologies représentées en OWL. Ensuite, des approches qui emploient ces techniques sont présentées.

Les deux principales questions que l'on se pose sont :(1) Comment trouver ces correspondances sémantiques? (2)Comment les utiliser dans le processus de la réécriture de requête ? Dans le chapitre suivant, nous proposons un système adaptatif d'aide à l'intégration de données hétérogènes, basé sur les ontologies. Il a pour objectif d'une part de générer ces correspondances au niveau du médiateur et d'autre part d'aider à la réécriture des requêtes pour la confronter aux besoins des utilisateurs.

Chapitre III: L'approche proposée

III.1 Introduction

Parmi les approches méthodologiques traitent l'hétérogénéité sémantique (différence de signification entre concepts) on trouve les ontologies qui sont de plus en plus utilisées aujourd'hui pour résoudre cette hétérogénéité. Une ontologie est donc une description explicite de la sémantique des éléments d'un domaine considéré. De ce fait, l'utilisation d'une ontologie est particulièrement adaptée pour résoudre les conflits d'hétérogénéité sémantique puisqu'elle permet la compréhension d'un vocabulaire.

Parmi Les problèmes liés à l'intégration sémantique des données à base d'ontologie est qu'étant donne un même domaine ou des domaines connexes, il est possible que plusieurs ontologies soient disponibles, car elles sont développées simultanément par plusieurs communautés différentes. Le choix d'une ontologie particulière et/ou l'exploitation de plusieurs ontologies en même temps devient difficile. Le besoin de comparer les termes des ontologies, de passer de l'une à l'autre ou d'échanger les ressources et les instances entre des bases des ressources indexées par des ontologies devient donc nécessaire. Notre approche s'inscrit également dans le contexte d'intégration virtuelle (Médiateur) des données autour d'une ontologie partagée. Elle est basée sur trois principes :

- il existe une ontologie (partagée) de domaine recouvrant la totalité des termes consensuels ;
- chaque source participante au processus d'intégration doit contenir sa propre ontologie qui en définit le sens;
- chaque ontologie locale s'articule a priori avec l'ontologie partagée;
- les différentes ontologies sont basées sur le même modèle OWL.

Notre travail est le premier à traiter du problème d'alignement entre l'ontologie global et les ontologies des sources en proposant qu'une ontologie conceptuelle soit explicitement représentée dans chaque source de données, et que ces sources s'engagent sur le sens et sur le fait d'utiliser les définitions ontologiques de l'ontologie partagée qui ont été acceptées et éventuellement normalisées.

Notre objectif est de permettre un accès unifié via une interface aux documents d'un même domaine d'application. Plus précisément, nous proposons d'aligner la taxonomie d'un portail Web avec celle de documents externes de façon à

augmenter le nombre de documents accessibles à partir de ce portail sans en modifier l'interface d'interrogation. Ce travail vise à :

- Aligner les ontologies (OG, OL) en réduisant l'implication directe de l'utilisateur. Dans ce système les conflits sémantiques entre les sources sont éliminés automatiquement dans le processus d'intégration de données.
- Trouver des correspondances entre l'ontologie globale et les ontologies locales pour permettre au médiateur de l'exploiter pour intégrer les données correspondantes.

III.2 Architecture générale du système d'intégration ISGO

Notre architecture d'intégration (**I**ntégration des **S**chémas de données **G**uidée par les **O**ntologies) part, en entrée, d'un ensemble de sources de données référençant a priori une ontologie partagée, et vise à produire, en sortie, un médiateur avec un schéma global (intégré) ayant comme vocabulaire l'ontologie partagée. Cette architecture est illustrée dans la figure III.1.

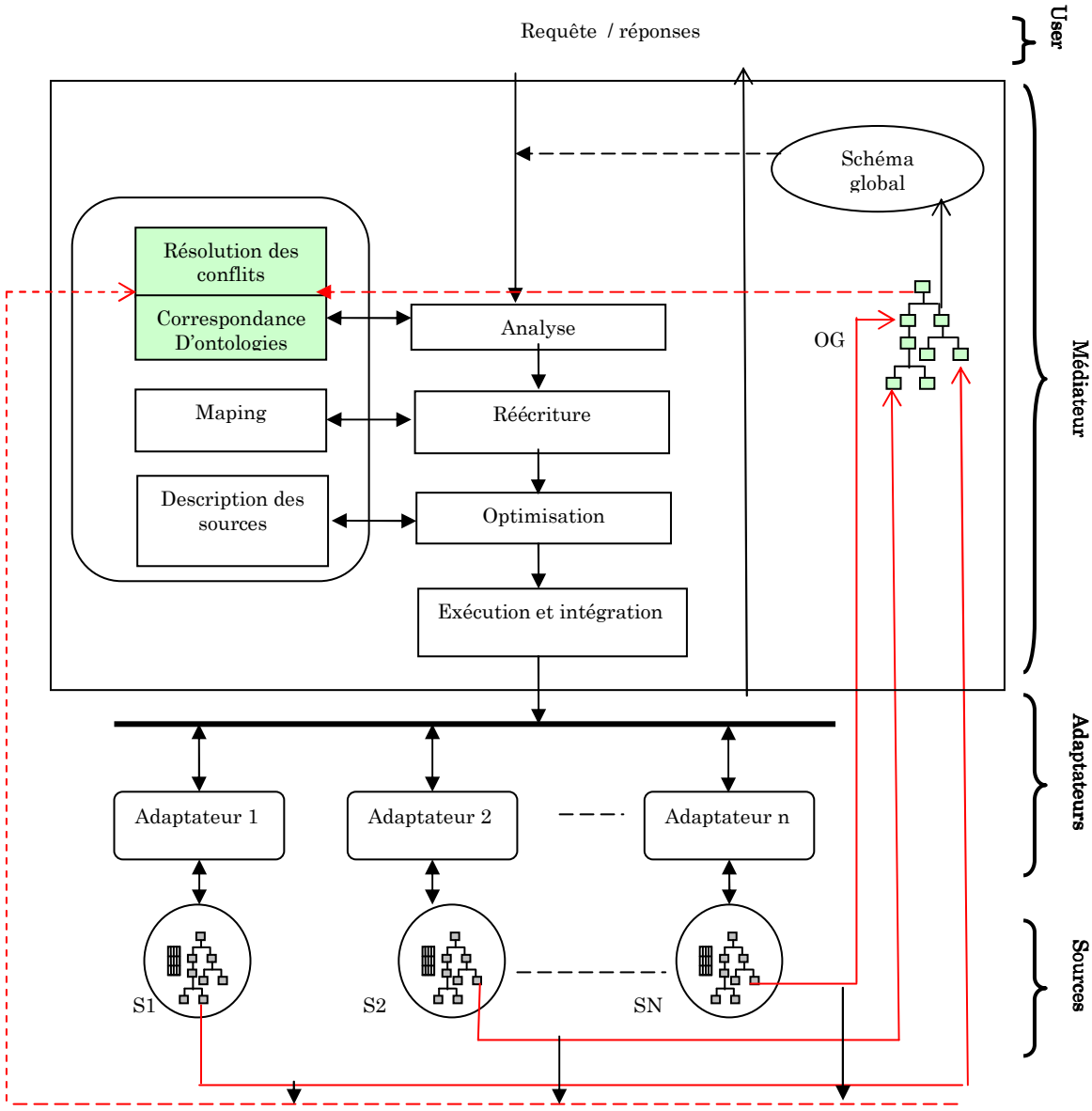


FIG III.1 - Intégration des Schémas de données Guidée par Ontologie

Pour représenter un univers de discours commun et consensuel des systèmes à intégrer, nous fournissons une ontologie globale de domaine qui capture les connaissances consensuelles et inclut un vocabulaire de concepts avec une spécification précise et formelle de leur signification. Elle est également représentée en OWL et adresse l'hétérogénéité sémantique au niveau des schémas. Pour cela, nous utilisons les opérateurs fournis par le langage OWL, notamment pour faire correspondre les concepts de l'ontologie globale avec ceux des ontologies locales.

Le système de médiation, **ISGO** comporte quatre niveaux: le niveau interface, le niveau médiateur, le niveau adaptateur et le niveau des sources locales (FIG III.1.). Dans la suite, nous décrirons le médiateur au niveau du composant **correspondances d'ontologies qui contribue à l'enrichissement sémantique.**

III.2.1 Médiateur

Au niveau médiateur, la requête passe d'abord par la phase d'analyse et de réécriture, puis par les phases d'optimisation, d'exécution et d'intégration et enfin la présentation des résultats. Outre les modules de traitement de la requête, le médiateur dispose d'un catalogue (voir FIG III.1). Notre contribution consiste à doter le médiateur par un module d'alignement pour aligner l'ontologie globale avec les ontologies des sources locales et générer les correspondances entre les concepts pour les utiliser dans la phase d'analyse et de réécriture de la requête.
La requête générée à travers l'interface utilisateur est exprimée en fonction des concepts de l'ontologie globale. Le module de réécriture utilise le mapping pour identifier les sources locales pertinentes. Puis, chaque concept global est remplacé par son correspondant local.

III.2.2 Adaptateur

L'hétérogénéité syntaxique est résolue au niveau des adaptateurs. En effet, ces derniers adaptent la sous requête exprimée en langage de requête de médiateur au langage de requête de la source et envoient la requête à la source en utilisant le protocole adéquat. Les résultats renvoyés par la source sont acheminés vers le médiateur.

III.3 Langage de description d'ontologie

Le langage que nous avons choisi pour représenter les connaissances dans ISGO est le langage OWL. Ce dernier est composé de trois sous langages d'expressivité croissante: OWL Lite, OWL DL et OWL Full. Dans notre système de médiation, nous avons utilisé OWL DL et OWL Lite. Ce dernier est suffisant pour représenter le thésaurus puisqu'il permet d'exprimer des hiérarchies et des contraintes simples. Quant à OWL DL, il comporte toutes les constructions du langage mais avec des restrictions sur la hiérarchie. Il possède en plus la puissance d'expressivité des logiques de description (Description Logic). Nous avons choisi OWL DL pour représenter les ontologies parce qu'il est suffisamment expressif pour décrire le mapping entre ontologies.

III.4 Enrichissement sémantique du médiateur

Avant de décrire plus en détail les composants qui contribuent à l'enrichissement sémantique (résolution des conflits et correspondances entre O_L, O_G), nous présentons les contraintes d'engagement sur une ontologie de référence (ontologie partagée) et un scénario d'intégration de données.

III.4.1 Contraintes d'engagement sur une ontologie de référence

Une ontologie est dite "partagée" entre plusieurs sources, lorsque elle reflète la notion que l'ontologie capture la connaissance consensuelle, c'est-à-dire, elle n'est pas privée, mais admise par un groupe. Afin de garder l'autonomie d'une source, cette dernière peut définir sa propre hiérarchie de classes, et, si besoin, rajouter les propriétés qui n'existent pas dans l'ontologie partagée. Plus précisément, s'engager sur une ontologie partagée signifie respecter la double contrainte suivante (appelée SSCR :Smallest Subsuming Class Reference [10]):

– toute classe locale doit référencer, par la relation de subsomption, la plus petite classe subsumante existante dans la hiérarchie de référence si ce n'est pas la même que celle de sa propre super classe.
– toute propriété nécessaire à l'ontologie locale et existant dans l'ontologie de référence doit être importée à travers la relation de subsomption.

Si une ontologie locale O_i est articulée avec l'ontologie partagée O_p en respectant le principe SSCR, nous disons qu'elle "*référence autant que cela est possible*" l'ontologie O_p.

III.4.2 Scénario d'intégration de données

Soit $S = \{S_1, S_2, ..., S_n\}$ l'ensemble des sources de données participant au processus d'intégration. Notons que dans une source de données, tout élément représenté dans le schéma, classe ou propriété doit appartenir à l'ontologie, de sorte que le schéma est un sous-ensemble de l'ontologie, chaque entité représentée correspondant à une classe et ses attributs correspondant aux propriétés applicables choisies.

Nous pouvons distinguer deux scénarios d'intégration, correspondant à deux articulations entre les ontologies locales et l'ontologie partagée du domaine :

1. les ontologies locales des sources de données sont directement extraites de l'ontologie partagée (chaque ontologie locale est un sous ensemble de l'ontologie partagée).
2. chaque source définit sa propre ontologie. Par contre l'ontologie locale référence l'ontologie partagée en respectant la condition SSCR. Dans ce scénario,on souhaite néanmoins intégrer les instances de chaque source comme des instances de l'ontologie partagée.

Dans notre système intégré, le scénario proposé consiste à donner, au fournisseur de la source de données, une certaine autonomie pour choisir la hiérarchie de classes (C_i, sub_i) où chaque source a sa propre ontologie et ses classes spécifiques. Néanmoins, l'ontologie O_i référence autant que possible l'ontologie partagée O_p en respectant la condition SSCR. Cela montre qu'il est

possible d'offrir aux sources locales une autonomie tout en permettant également une construction automatique du système intégré d'une manière déterministe et exacte.

Pour un tel contexte, l'intégration des sources de données consiste à intégrer (aligner) d'abord les ontologies, puis les données. Une fois l'alignement entre l'ontologie globale et chaque ontologie locale est établi, les utilisateurs peuvent potentiellement poser des centaines de requête de sources de données en utilisant une seule requête qui cache les hétérogénéités sous-jacentes. En utilisant cette démarche, l'interrogation peut être facilement étendue à une nouvelle source de données en alignant une ontologie locale avec la globale. À cette fin, nous avons conçu et mis en œuvre un outil pour aligner des ontologies. La sortie de cet outil est un ensemble de correspondances entre les concepts qui seront utilisés pour produire les requêtes aux sources de données locales, une fois qu'une requête est formulée sur l'ontologie globale. Pour faciliter la tâche de l'utilisateur, nous proposons des méthodes semi-automatiques pour propager le long des mappings de telles ontologies.

III.5 Approche d'alignement

Les schémas en entrée du processus de mise en correspondance sont des taxonomies, correspondant à des ontologies très sommaires avec des définitions de concepts très pauvres. .Une taxonomie (C, H_C) comprend un ensemble de concepts C et une hiérarchie de subsomption entre concepts H_C. Un concept est défini par son label et les relations de sous classes qui le relient à d'autres concepts. Le label est un nom (chaîne de caractères) qui décrit des entités en langage naturel et qui peut être une expression composée de plusieurs mots. Les relations de sous-classes établissent des liens entre concepts. Il s'agit de l'unique association sémantique utilisée dans la classification. Une taxonomie est généralement représentée par un graphe acyclique dont les noeuds sont les concepts et les arcs correspondent aux liens de sous-classes.

III.5.1 Caractéristiques des taxonomies alignées

Taxonomies spécifiques : Les techniques que nous proposons d'exécuter en priorité sont des techniques terminologiques exploitant des mesures de similarité basées sur de comparaisons de chaînes de caractères. Ces techniques sont bien adaptées lorsque les taxonomies sont des descriptions très fines de domaines d'application car, dans ce cas, des concepts très spécialisés dont le label traduit cette spécialisation sont représentés.

Labels de concepts généraux inclus dans les labels de concepts plus spécifiques
L'approche suppose que très souvent le label d'un concept est construit en reprenant le label du concept qu'il spécialise et en lui ajoutant des qualificatifs permettant de décrire ses spécificités.

III.5.2 types de relations

Le processus d'alignement génère des mappings 1-1 qui sont des relations de deux types : des relations d'équivalence et des relations de subsomption.

Relations d'équivalence : Une relation d'équivalence isEq est un lien entre un concept dans $O_L$et un concept dans O_G dont les noms sont considérés comme étant similaires. Cette similarité recouvre des réalités variées. Il s'agit tout d'abord de relier des termes dont les noms sont rigoureusement identiques syntaxiquement. En effet, dans ce cas l'ontologie local c'est un fragment de l'ontologie partagée pour tout concept de l'ontologie local O_L choisi parmi les concepts proposés de l'ontologie global O_G.

Relations de subsomption: Les relations de spécialisation isA sont des liens usuels de sous-classe/super-classe. Quand ce type de lien relie un concept de O_Là un concept de O_G, son degré de généralité est le même que celui reliant ce super-élément à d'autres sous-éléments de O_G. Ce lien existe entre un concept propre à O_L (spécialisation d'autres concepts de O_G) et un concept de O_G.

III.5.3 Une approche basée sur la mesure de similarité $Sim_{Jaro\text{-}Winkler}$

La comparaison de la similarité de deux noms de deux classes est la comparaison entre deux ensembles de tokens[13]correspondant à ces noms. La similarité de deux tokens, qui sont actuellement des chaînes de caractères courtes, est calculée en employant la métrique Jaro-Winkler (Définition1), qui est basée sur le nombre et l'ordre de caractères communs entre deux chaînes des caractères [58]. La Définition1 est la version de la mesure de distance de la métrique Jaro-Winkler, DSJaro-Winkler.

Définition1 (Distance de Jaro-Winkler). *Soit s et t deux chaînes de caractères. Soit P la longueur du préfixe commun le plus long de s et t. Soit n un nombre positif. La distance de Jaro-Winkler est une fonction de la dissimilarité* $DS_{JaroWinkler}$*:* $\mathbf{S} \times \mathbf{S} \rightarrow [0, 1]$ *telle que:*

$$\overline{DS_{JaroWinkler}}(s,t) = \overline{DS_{Jaro}}(s,t) - \frac{\max(P.n)}{10}\overline{DS_{Jaro}}(s,t)$$

Puisque la métrique Jaro-Winkler est une métrique normalisée, donc la mesure de similarité Jaro-Winkler SJaro-Winkler est obtenue par : SJaro-

[13] Token : chaîne de caractère courte indécomposable

Winkler=1-DSJaro-Winkler. La valeur de similarité de nom (Snom) entre deux noms est alors la moyenne des valeurs de similarité entre chaque token d'un ensemble et le tokenle plus similaire dans l'autre ensemble (Algorithme 2).

III.6 Méthode d'alignement

Le processus d'alignement produit un ensemble de correspondances et ne modifie pas l'évolution des ontologies. Etant données deux ontologies O_1 et O_2, effectuer le mapping entre O_1 et O_2 signifie que pour chaque concept (noeud) de l'ontologie O_1, nous essayons de trouver un concept (noeud) correspondant ayant une sémantique identique ou similaire dans l'ontologie O_2 et vice versa.

Une définition relativement formelle d'alignement est formulée par Ehrig & Staab [26] : **" given two ontologies O_1 and O_2, mapping one ontology onto another means that for each entity (concept C, relation R, or instance I) in ontology O_1, we try to find a corresponding entity, which has the same intended meaning, in ontology O_2 ".**

D'une façon formelle, l'alignement de deux ontologies est défini par la fonction **align** comme suit :

$$\textbf{align} : O_1 \longrightarrow O_2 \text{ tel que } \textbf{align}(e_1) = e_2 \text{ si } Sim(e_1, e_2) > t$$

Où O_1 et O_2 sont les deux ontologies à apparier, t désigne un seuil minimal de similarité appartenant à l'intervalle [0,1], $e_1 \in O_1$ et $e_2 \in O_2$. Ce seuil indique le niveau minimum pour que deux entités soient similaires. Chaque entité e_i est alignée à une entité e_j.

La fonction **align** retourne comme sortie les correspondances entre deux ontologies et leurs valeurs de similarité sous forme des triplets (e1, e2, sim) où e1 est une entité (classe, relation ou instance) de la première ontologie, e2 est une entité de même type que e1 de la deuxième ontologie, et sim est la valeur de similarité entre ces deux entités. Deux entités dans un triplet sont considérées similaires l'une à l'autre, suivant la valeur de similarité sim entre elles. Cette valeur est souvent comprise entre 0 et 1, ce qui permet des possibilités d'interprétation probabiliste de la similarité.

Les schémas en entrée du processus de mise en correspondance sont des taxonomies, correspondant à des ontologies avec des définitions de concepts. Les concepts sont principalement définis par référence à leur terminologie. Dans notre contexte[14] pour construire un algorithme d'alignement de deux ontologies en OWL nous considérons des ontologies simples avec les entités suivantes :

[14] Dans les systèmes d'intégration de données, les schémas peuvent être considérés comme des ontologies simples, ne disposant que des concepts (ou classes) organisés dans une hiérarchie de subsomption, on peut donc d'établir des correspondances entre les schémas dans le domaine d'intégration des donnée.

Classe : (appelée aussi concept): Une classe est une représentation d'une collection ou d'un groupe d'objets ayant des caractéristiques similaires.

Relation: Une relation est employée pour décrire un rapport entre deux classes. Ces deux classes sont indiquées par le domaine (domain en anglais) et le co-domaine (range en anglais) de la relation.

Instance: Une instance d'une classe correspond à un objet réel qui est un membre de l'ensemble dénoté par cette classe. L'instance d'une classe possède toutes les caractéristiques définies pour la classe.

Les types de rapport possibles entre deux entités de deux ontologies peuvent être l'équivalence, la subsomption. Le résultat de la tâche d'alignement est un ensemble de paires d'entités, qui sont des correspondances entre les ontologies. Dans notre système **ISGO** ce résultat utilisé par le médiateur au niveau d'analyse et réécriture de requête.

III.6.1 Enchaînement de techniques

Plusieurs techniques sont utilisées : des techniques terminologiques puis structurelles (voir la figure III.2) après un traitement préalable de normalisation des concepts (remplacement des signes de ponctuation et des symboles spéciaux par des espaces, lemmatisation). Elles sont appliquées séquentiellement de façon à ce que le processus d'alignement soit le plus efficace possible.

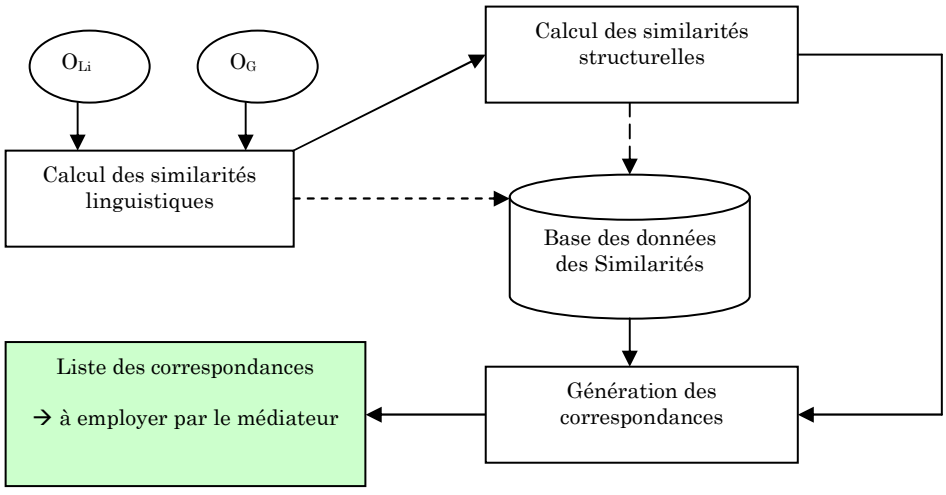


FIG III.2 - Le processus d'alignement

III.7 L'algorithme en détail

L'algorithme consiste à trouver des correspondances entre chaque ontologie locale $\mathbf{O_i}$ de la source $\mathbf{S_i}$ et l'ontologie globale O_G.

Algorithme
Debut
n ← nombre de sources de données à intégrer
Pour i **de** 1 **à** n
Mi_{sim}←MAP_ONTO (O_{Li} , O_G, seuil)
Fin

Algorithme d'alignement

// **Algorithme principal**
MAP_ONTO (O_L , O_G, seuil)
Debut
MSim ← 0 // initialiser la matrice de similarité
pour chaque C_L de O_L **faire**
Debut
Sim_{nom_max} ← 0
Pour chaque C_G de O_G **faire**
Debut
$Sim_{nom}(C_L, C_G)$
si $Sim_{nom}(C_L, C_G) > Sim_{nom\_max}$ **alors**
$Sim_{nom\_max} \leftarrow Sim_{nom}(C_L, C_G)$
Fin si
Fin
Si $Sim_{nom\_max}(C_L, C_G)$ >=seuil **alors** $MSim(C_L, C_G) \leftarrow Sim_{nom\_max}(C_L, C_G)$
Sinom
$Sim(C_L, C_G) \leftarrow (Sim_{structural}(C_L, C_G) + Sim_{nom\_max}(C_L, C_G))/2$

Fin
Retourner MSim
Fin

Algorithme principal MAP_ONTO

III.7.1 La similarité linguistique

La similarité linguistique de deux classes est calculée à partir des composantes linguistiques de la classe. Le calcul de la similarité linguistique est basé sur le nom de classe. Une mesure de similarité est construite pour pouvoir exploiter au maximum des informations contenues dans cette composante.

Normalement, le nom d'une classe est une chaîne des caractères, sans espaces. Un nom de classe peut être un mot, un terme, ou une expression (une combinaison des mots). Ce nom est unique dans une ontologie pour identifier la classe. Le calcul de la valeur de similarité de deux noms est effectué dans deux étapes: la *normalisation* et la *comparaison*.

```
//Algorithme1
Normalisation(nom)
Debut
résultat ← créer un ensemble vide
tokens ←Tokenisation(nom)
n ← nombre de token dans tokens
Pour i de 1 à n
ex ← Expansion(tokens[i])
ex ← Minusculisation(ex)
resultat ←Ajouter(ex, resultat)
Fin Pour
Fin
```

Algorithme de normalisation de nom

L'étape de normalisation (Algorithme1) convertit un nom de classe en un ensemble d'unités lexicales des tokens. Un nom est découpé en plusieurs tokens grâce à la ponctuation, à la casse (majuscule), aux symboles spéciaux, aux chiffres. Par exemple, le nom « Numéro_Etudiant » est convertit à l'ensemble {« Numéro », «Etudiant »}. La normalisation du nom inclut une expansion de token : les abréviations, les acronymes sont élargis, par exemple le token « WS » est élargi à {«Web»,«Sémantique»}.Cette expansion est effectuée grâce à un dictionnaire externe, dont chaque entrée est une paire composée d'un token (abréviation ou acronyme) et d'un ensemble de mots qui correspondent au token. Le dictionnaire est soit construit spécialement pour le domaine où les ontologies à aligner se trouvent soit il s'agit d'un dictionnaire général contenant des termes communs. Les tokens dans l'ensemble de tokens sont enfin rendus minuscules pour être comparés après.

```
//Algorithme2
Debut
Sim_Max (token, tokens)
valeur_max ← 0
n <- nombre de tokens dans tokens
Pour i de 1 à n
valeur_sim ← SJaro-Winkler (token, tokens[i])
Si valeur_sim > valeur_max
valeur_max ← valeur_sim
Fin Si
Fin Pour
Retourner valeur_max
Fin
```

Algorithme de similarité de deux tokens

```
//Algorithme3
Debut
Simnom (OL,OG)
nomOL ← extraire du nom de l'entité OL
nomOG ← extraire du nom de l'entité OG
tokens1 ←  Normalisation(nomOL)
tokens2 ← Normalisation(nomOG)
n1 ← nombre de tokens dans tokens1
n2 ← nombre de tokens dans tokens2
somme1 ← 0
Pour i de 1 à n1
sim ← Sim_Max(tokens1[i], tokens2)
somme1 ← somme1 + sim
Fin Pour
somme2 ← 0
Pour j de 1 à n2
sim ← Sim_Max(tokens2[j], tokens1)
somme2 ← somme2 + sim
Fin Pour
Si n1 = n2 = 0
résultat ← 1
Sinon
résultat ← (somme1 + somme2) / (n1 + n2)
Fin Si
Retourner résultat
Fin
```

Algorithme de similarité de nom

III.7.2 La similarité structurelle

Les classes dans une ontologie ont des rapports de spécialisation entre elles. Si une classe est sous-classe d'une autre, elle spécialise cette dernière et hérite de toutes les caractéristiques de cette dernière. Dans cette hiérarchie de classes, toutes les classes (sauf les classes racines) sont descendantes d'une des classes racines, et une classe hérite toutes les caractéristiques de toutes les classes dans le chemin de la classe en question jusqu'à sa racine dans la hiérarchie. De cette observation, deux classes (de même pour deux relations) de deux ontologies sont considérées similaires si les classes dans deux chemins connectant les classes en question à leurs racines dans leurs hiérarchies de classes sont déjà similaires. À noter que dans le cas où il y a des multi-héritages, c'est-à-dire une entité a plusieurs super-entités (entités parentales), il y a plusieurs chemins d'une entité vers un ou plusieurs racines, le **chemin de la classe** se compose de tous les entités dans tous les chemins possibles partant de la classe en question vers un des racines de la hiérarchie.

Soit S_1, S_2 deux ensembles. Nous définissons **ESL** (S_1, S_2) l'ensemble d'éléments dans S_1 qui sont similaires linguistiquement avec un élément quelconque dans S_2(Algorithme5),et **EDL** (S_1, S_2) l'ensemble d'éléments dans S_1 auquel sont rajoutes les éléments de S_2 qui ne sont pas similaires linguistiquement avec un élément quelconque dans S_1 (Algorithme6).

Définition 2 (ESL et EDL). Soit S_1 et S_2 deux ensembles. ESL et EDL de ces deux ensembles sont définis comme suivant :

$$ESL\ (S_1, S_2) = \left\{ s_i \in S_1 \mid \exists s_j \in S_2 . l - similar\ (s_1, s_2) \right\}$$

$$EDL\ (S_1, S_2) = S_1 \cup \left\{ s_j \in S_2 \mid \forall s_j \in S_1, \neg l - similar\ (s_1, s_2) \right\}$$

```
//Algorithme 4
INCLU_E (concept, ensemble)
Debut
n ← nombre de concepts  dans l'ensemble ensemble
Pour i de 1 à n
Si concept et ensemble[i] sont linguistiquement similaires
Retourner vrai
Fin Si
Fin Pour
Retourner faux
Fin
```

Algorithme4

```
//Algorithme 5
ESL (ensemble1, ensemble2)
Debut
ES ← créer un ensemble vide
n1 ← nombre de concepts dans l'ensemble ensemble1
Pour i de 1 à n1
Si INCLU_E (ensemble1 [i], ensemble2)
ES ←Ajouter (ensemble1[i], ES)
Fin Si
Fin Pour
Retourner ES
Fin
Fin
```

Algorithme5

```
//Algorithme6
EDL (ensemble1, ensemble2)
Debut
ED ← ensemble1
n2 ← nombre de concepts dans l'ensemble ensemble2
Pour i de 1 à n2
Si not INCLU_E (ensemble2[i], ensemble1)
ED ← Ajouter(ensemble2[i], ED)
Fin Si
Fin Pour
Retourner ED
Fin
```

Algorithme6

Définition3 :(Chemin des classes).Soit Hc la hiérarchie de classes de l'ontologie. Soit c une classe dans la hiérarchie Hc. Chemin(c) est défini comme l'ensemble de classes qui sont des classes dans tous les chemins possibles construits par les liens de subsomption, connectant la classe c à une des classes racines dans la hiérarchie Hc.

La similarité entre deux chemins de deux classes (Algorithme7) est donnée dans la Définition 4. Dans le cas où toutes les deux classes à comparer sont les racines dans les hiérarchies, leurs chemins sont donc vides, leur similarité de chemin des classes SChemin est considérée égale à 1.

Définition 4 :(Similarité de chemin des classes). Soit c_1 et c_2 deux classes de deux ontologies. La similarité de chemin des classes S_{Chemin} entre deux classes c_1 et c_2 est définie comme suivant :

$$S_{Che\ \min}(c_1, c_2) = \frac{\left|ESL_{Che\ \min}(c_1, c_2)\right|}{\left|EDL_{Che\ \min}(c_1, c_2)\right|}$$

Où

$$ESL_{Che\min}(c_1, c_2) = ESL(Che\min(c_1), Che\min(c_2))$$

$$EDL_{Che\min}(c_1, c_2) = EDL(Che\min(c_1), Che\min(c_2))$$

```
//Algorithme7
Sim_structural(C_L, C_G)
Debut
CheminL ← extraire des entités dans des chemins du concept C_L
CheminG ← extraire des entités dans des chemins du concept C_G
Si | EDL(Chemin(C_L), Chemin(C_G)) | = 0
résultat ← 1
Sinon
nES ← | ESL(Chemin(C_L), Chemin(C_G)) |
nED ← | EDL(Chemin(C_L), Chemin(C_G)) |
résultat ← nES / nED
Fin Si
Retourner résultat
Fin
```

Algorithme de similarité structurelle

III.7.3 La génération des correspondances

La sortie du processus d'alignement d'ontologies est une liste de correspondances entre deux ontologies d'entrée (locale et globale). Ce sont des correspondances entre les entités de même type des ontologies : entre une classe d'une ontologie locale et une autre classe de l'autre ontologie global ; entre une relation d'une ontologie et une autre relation de l'autre ontologie, entre deux instances de deux ontologies. Les correspondances sont celles du type 1-1, où une entité d'une ontologie a une et seulement une entité correspondante dans l'autre ontologie.

L'algorithme retourne comme sortie les correspondances entre deux ontologies (locale et globale) et leurs valeurs de similarité sous forme des triplets (C_G, C_L, sim) où C_G est une entité (classe, relation ou instance) de l'ontologie local, C_L est une entité de même type que C_L de l'ontologie locale, et **sim** est la valeur de similarité entre ces deux concepts. Deux concepts dans un triplet sont considérés similaires l'un au l'autre, suivant la valeur de similarité **sim** entre

eux. Cette valeur, appelée la valeur de similarité finale, est calculée, par la somme pondérée, à partir de la similarité linguistique et de la similarité structurelle présentées dans les sections 4.7.1 et 4.7.2 respectivement (Algorithme principal).

Pour chaque entité de la première ontologie, l'algorithme cherche l'entité de la deuxième ontologie qui est la plus similaire (la valeur de similarité entre elles est la plus élevé). Si la valeur de similarité linguistique dépasse un seuil de similarité prédéfini, ces deux entités sont ajoutées dans la liste des triplets (une paire d'entités et leur valeur de similarité). Si la valeur de similarité linguistique est inférieure à un seuil prédéfini l'algorithme ajoute la valeur de similarité structurelle à la valeur linguistique divisée par 2.

III.8 Conclusion

Nous avons proposé un système de médiation pour l'intégration de sources hétérogènes et réparties. L'hétérogénéité sémantique (schéma) est traitée au niveau du médiateur de ISGO. La résolution de conflits schématiques est réalisée grâce à la mise en correspondance entre les concepts de l'ontologie globale et ceux des ontologies locales représentant les sources hétérogènes. Notre système est fondé sur un outil de mise en correspondance entre les concepts de l'ontologie globale et ceux des ontologies locales. Il utilise les ontologies OWL pour représenter le domaine d'application. L'utilisation de cet outil permet au médiateur d'analyser et de réécrire la requête de l'utilisateur en utilisant les correspondances fournies.

En vue de faciliter la tache du médiateur, nous avons lui doté d'un outil d'alignement pour la génération automatique d'alignement entre ontologies. Un enchaînement de technique est utilisé pour calculer la similarité entre deux concepts de deux ontologies. Des techniques terminologiques puis structurelles après un traitement préalable de normalisation des concepts (remplacement des signes de ponctuation et des symboles spéciaux par des espaces, lemmatisation). Elles sont appliquées séquentiellement de façon à ce que le processus d'alignement soit le plus efficace possible.

L'algorithme attaque le problème d'alignement des ontologies spécialement pour intégration de données hétérogènes et autonomes. Notre objectif est de construire un outil qui génère automatiquement des mappings entre deux ontologies. Cet outil est basé sur une architecture indépendante de tout domaine d'application qui lui permet d'implémenter avec n'importe quelle ontologie de domaine représentée en OWL.

Chapitre IV: Développement et expérimentations

IV.1 Introduction

Lors de l'utilisation d'un système de médiation, la définition de requêtes de médiation est l'une des taches les plus complexes à effectuer manuellement, surtout lorsque le nombre de sources et le volume de méta-données qui les décrivent sont importants. Cette complexité se multiplie en présence de données hétérogènes dans les sources. Des ontologies décrivant le contexte sémantique des schémas des sources et du schéma de médiation sont utiles, même essentielles dans le processus d'intégration, plus il y a de connaissances sémantiques disponibles, plus facile et plus précise sera l'intégration de données vis-à-vis des besoins applicatifs.

Dans le contexte de la génération de requêtes de médiation, la non prise en compte des conflits liés à l'hétérogénéité des sources peut fausser la sémantique des requêtes de médiation et retourner à l'utilisateur des résultats non conformes à ceux définis dans son schéma de médiation. Pour résoudre le problème de conflit entre attributs (attributs du schéma de médiateur et ceux des schémas locaux) nous avons proposé une architecture où chaque source de données est dotée d'une ontologie locale qui décrit son schéma et une ontologie globale qui décrit le schéma du médiateur.

Le résultat d'alignement de l'ontologie globale avec les ontologies locales permet de trouver ces correspondances sémantiques afin de les utiliser, sous forme de méta-données, dans l'étape de réécriture de requête du médiateur.

IV.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. Dans la suite on utilisera S_v pour désigner le schéma virtuel et S_i schéma source pour l'ensemble de sources de données $\{S_1, S_2\}$.

Exemple 1:

Schéma virtuel (Sv)

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

Schémas des sources (Si)

Source1:
TRANSPORTAERIEN (idT, avion, type_trajet, type_classe)
VOYAGE_VOL (idV, prix, ville_depart, ville_arrivee,nbre_jours, date_depart, heure, type_sejour, idT, idH)
HOTEL (idH, nbre_etoiles, nom, region, ville, restaurant)

Source 2 :
TRANSPORTFERRIES (idT, bateau, type_trajet, comfort),
VOYAGE_MARINIER (idV, prix, port_depart, port_arrivee, nbre_jours, date_depart, heure, type_sejour, idT, idH).

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'exemple1.Les instances de ce schéma sont calculées à partir de deux sources de données sources1 et sources2.

Rappelons que ce travail s'inscrit dans le contexte d'une approche d'intégration sémantique a priori qui vise à intégrer les ontologies des sources. Elle consiste à établir les relations sémantiques (équivalence, subsomption) entre les concepts (attributs) des ontologies. , et est associé à trois hypothèses :

- il existe une ontologie de domaine recouvrant la totalité des termes utilisés dans le schéma global (schéma du médiateur) ;
- chaque source de données contient sa propre ontologie locale ;
- chaque ontologie locale s'articule a priori avec l'ontologie partagée (ontologie associée au schéma du médiateur)

IV.3 Création des ontologies

Face à la problématique de décomposition de requêtes dans un système de médiation, nous avons opté pour une approche de réécriture automatique qui permet de décharger l'utilisateur de l'exploration d'un volume important de méta-données. 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.

Les opérations qui composent une requête de médiation ne sont valides que si les conflits sémantiques liés aux schémas sont détectés et résolus. Par exemple, une sélection sur l'attribut moyen de la relation TRANSPORT (schéma global) peut retourner un résultat vide quand les attributs (avion, bateau) des sources (source1, source2) sont interprétés différemment. La création d'une ontologie pour chaque source de données est une solution au problème; encore faut-il savoir détecter les conflits sémantiques et identifier les fonctions de correspondance appropriées.

Nous proposons un système qui permet de générer automatiquement des méta- données comme résultat d'alignement entre l'ontologie globale, qui décrit les attributs du schéma de médiation, et l'ontologie locale de chaque source.

IV.3.1 Ontologie globale (associé au schéma du médiateur)

Nous considérons des sources de données à base ontologique, dont chacune contient une ontologie locale qui référence une ontologie de domaine partagée. Ces sources sont autonomes ; elles évoluent de façon asynchrone et peuvent étendre ou/et spécialiser, en cas de besoin, l'ontologie de domaine.Cette ontologie consiste à définir pour chaque relation et attribut du schéma global un concept qui le décrire et porte le même nom comme identifiant.

Concept-relation : pour chaque nom de relation on associé un concept dans l'ontologie globale. Exemple (**VOYAGE, TRANSPORT, HOTEL)**

Concept-attribut : de même que pour la description des relations, nous essayons à décrire les attributs de chaque relation par des concepts dans l'ontologie globale.

L'ontologie globale associée au schéma du médiateur (Sv) est présentée par le fichier Global-Ontologie.owl qui est consultable en annexe du mémoire.

IV.3.2 Ontologie des sources de données

L'ontologie globale associée au schéma global est partagée entre les sources, lorsque ces dernières utilisent les définitions ontologiques de l'ontologie partagée pour créer les concepts de l'ontologie locale. Afin de garder l'autonomie d'une source, on peut définir sa propre hiérarchie de classes et, si besoin , rajouter les propriétés qui n'existent pas dans l'ontologie partagée.

Lors de la création des ontologies des sources on doit respecter la double contrainte suivante :
- toute classe locale doit référencer, par la relation de subsumption, la plus petite classe subsumante existante dans la hiérarchie de l'ontologie globale.
- toute propriété nécessaire à l'ontologie locale et existant dans l'ontologie de référence doit être importée à travers la relation subsumption. les ontologies locales associées aux deux sources sont consultables en annexe.

IV.3.3 Règles de création de l'ontologie locale

Règle1 : pour chaque attribut (ou relation) du schéma local on doit créer le concept qui le correspond et porte le même nom de l'attribut.

Règle2 :c'est le nom de l'attribut existe dans le schéma global on importe le concept crée précédemment dons l'ontologie globale et on appel ça point d'articulation linguistique entre l'ontologie globale et locale (les nom des concepts sont identiques). Exemple : on trouve l'attribut **date_depart** dans la relation **VOYAGE** du schéma global et aussi dans la relation **VOYAGE_VOL** du schéma de la source1.

Règle 3 : c'est le nom de l'attribut n'existe pas dans le schéma global on doit le créer dans l'ontologie locale à condition qu'il soit subsumé par un concept de l'ontologie globale point d'articulation structurelle. Exemple l'attribut **ville_depart** de la relation **VOYAGE_VOL (source1)** n'existe pas dans l'ontologie globale mais il est considéré comme sous classe de la classe **lieu_depart** grâce à la relation de subsomption.

IV.3.4 Alignement des ontologies

Les ontologies décrit les concepts intervenant dans un domaine particulier. Ces concepts sont typiquement organisés dans une hiérarchie de généralisation et peuvent être reliés par des relations d'équivalence ou de subsomption.

Soient deux ontologies Og = (Cg; Hg),Oi = (Ci; Hi), décrites en OWL, représentent l'ontologie globale et l'ontologie locale respectivement où Ci est un ensemble de concepts caractérisés par leurs labels et Hi une hiérarchie de subsomption entre les nœuds correspondants aux concepts. Un alignement produit par MAP_ONTO est un ensemble de mises en correspondance, les

mappings, établies entre chaque concept de l'ontologie globale et un unique concept de l'ontologie locale. Les mappings sont exprimés par des relations d'équivalence (isEq) ou subsomption (isA ou isMoreGnl), auxquelles sont associées des mesures de similarité. Ces mappings sont sûrs parce que les ontologies alignées décrivent le même domaine d'application. Les concepts qui ont le même nom ou (ID sous OWL) peuvent être considérés comme équivalents et la similarité proche de 1.

IV.4 Outils de développement

L'application ISGO prend en entrée,d'une part l'ontologie du domaine,associée au schéma global, en format OWL, et d'autre part les ontologies des sources (source1,source2)respectivement associées au schémas (shémas1,schéma2).
Elle renvoie un document XML représentant les correspondances sémantiques, où chaque concept de l'ontologie globale est associé par les concepts des ontologies locales qui lui correspondre. le document XML est structuré par des éléments de la forme:

```
<Concept-Global ID="lieu_arrivee">
<cl>ville_arrivee</cl>
</Concept-Global>
```

IV.4.1 L'exploitation de l'ontologie

L'ontologie de domaine de voyage que nous avons conçus est implémentée en langage OWL (ontology Web Language).les fichiers OWL sont inexploitables en état brut, c'est-à-dire la structure du fichier OWL est très complexe. Donc pour pouvoir l'exploiter il nous a fallu un «traducteur» capable de traduire les balises la sémantique véhiculée par le fichier OWL en objet manipulable par des programmes. L'outil disponible qu'on a peut avoir c'est L'API JENA.

JENA est développée entièrement en Java, elle donne aux programmes la possibilité d'exploiter du contenu des fichiers RDF et OWL (extraction du contenu sémantique de ces derniers).

IV.4.2 Langage de développement

Dans la partie programmation des algorithmes proposés nous avons utilisé le langage JAVA. Ce langage nous a paru beaucoup plus une évidence qu'un choix, vu que les outils que nous utilisant sont entièrement développés en JAVA. Ce dernier s'est révélé particulièrement adapté à cette réalisation. Un des principaux atouts du langage Java est sa portabilité qui assure l'exécution de tout programme développé en Java sur n'importe quelle plate-forme (MS-Windows, UNIX, MAC-OS, etc). De plus, de nombreuses APIs (Application Programming Interface) sont disponibles dans le langage Java pour traiter des documents XML. Le modèle DOM (Document Object Model)a été utilisé pour avoir un accès plus aisé aux documents XML via un ensemble d'objets classés

en arborescence. Pour le chargement et manipulation des ontologies on utilisé L'API JENA, ce qui nous permet la modélisation des documents OWL.

IV.5 L'utilisation du système

Dans ce qui suit nous allons présenter le prototype réalisé à travers des captures d'écran.

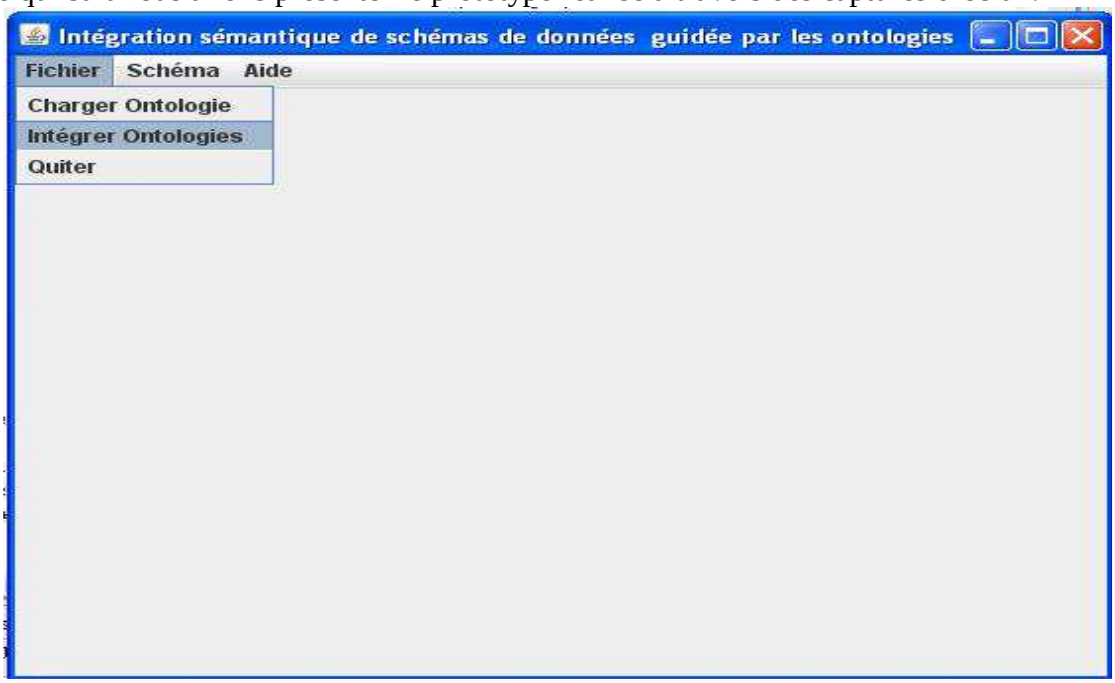


FIG IV.1 – L'interface principale

IV.5.1 Chargement de l'ontologie

Pour exécuter l'application on doit lancer le chargement des ontologies global et locale associées respectivement aux schémas du médiateur Sv et de la source Si comme le montre la figure IV.2

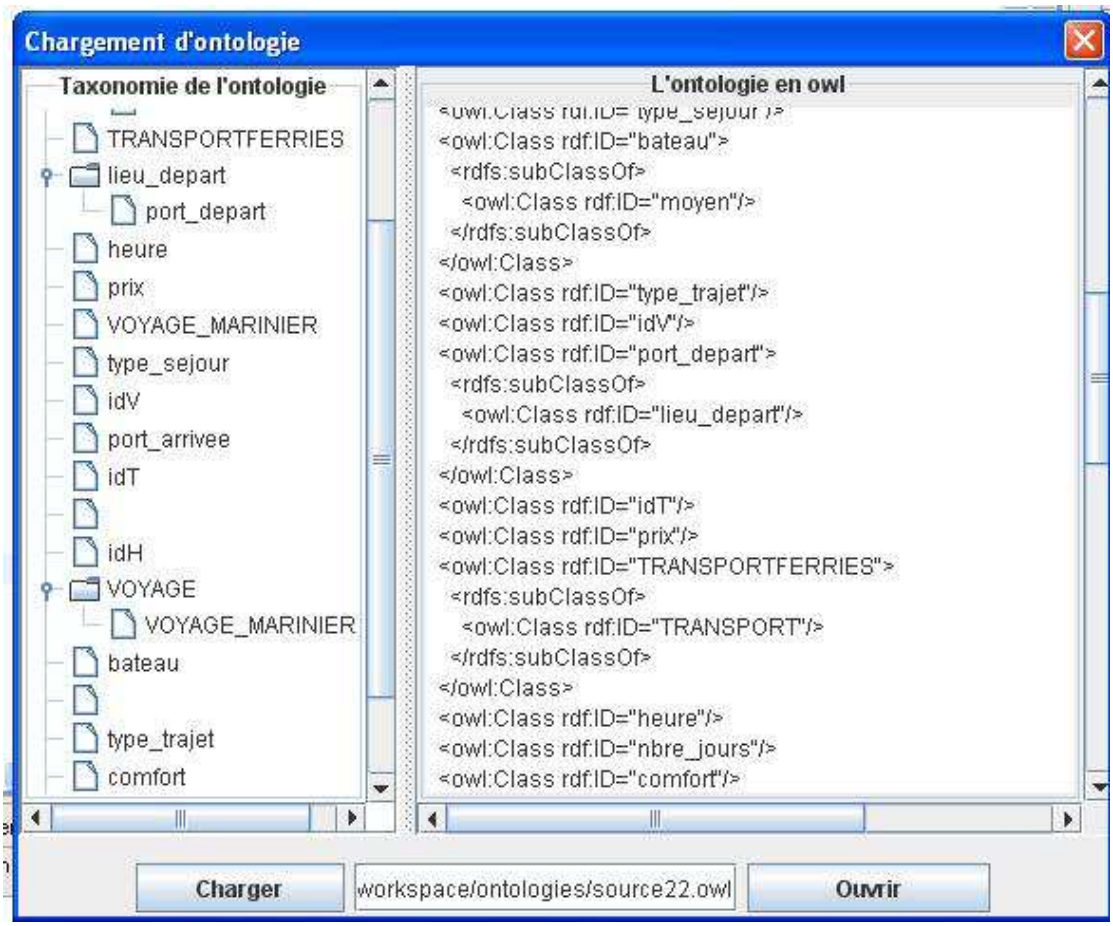


FIG IV.2 - Le chargement d'ontologie

IV.5.2 Intégration des ontologies associées aux schémas des sources

Ce module chargé d'aligner l'ontologie globale, associée au schéma global, avec l'ontologie locale d'une source de données participante au processus d'intégration. Le résultat est un ensemble de correspondances sémantiques utilisées par le médiateur dans l'étape de réécriture de requête. Deux cas possibles pour qu'un concept global puisse être associé à un concept local:

- le concept global porte le même nom que celui du concept local (IV.3.3 règle 2) comme le montre la figure IV.3 pour le concept global *nbre_etoiles*.
- Un concept global associé à un concept local par la relation de subsomption (IV.3.3 règle 3) exemple *VOYAGE_VOL* est subsumé par le concept global *VOYAGE*.

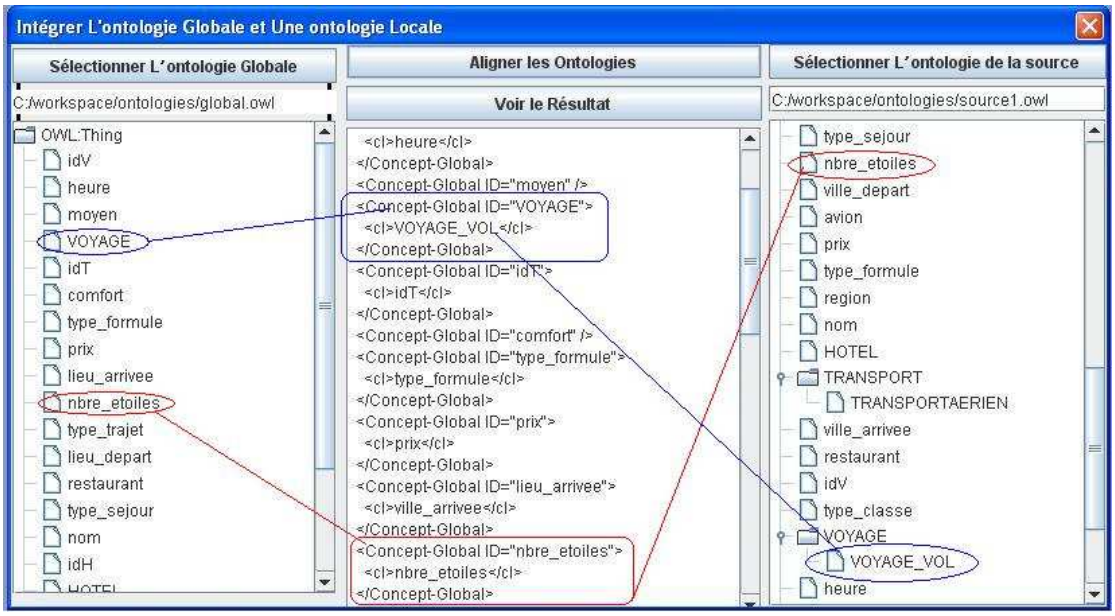


FIG IV.3 – Alignement de l'ontologie globale avec l'ontologie locale

IV.5.3 schéma virtuel et schémas des sources

Ce module permet de visualiser le schéma virtuel d'interrogation des sources

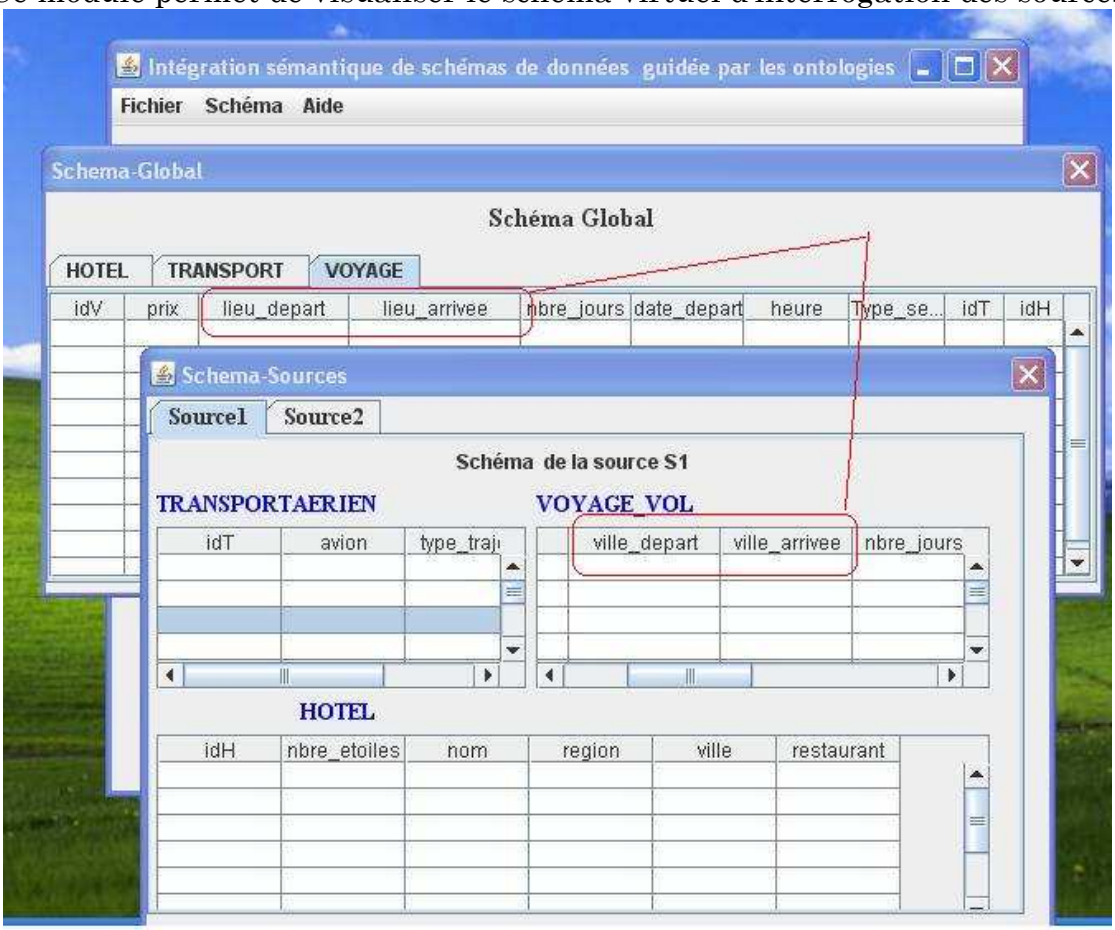


FIG IV.4 –Schéma global et Schémas des sources

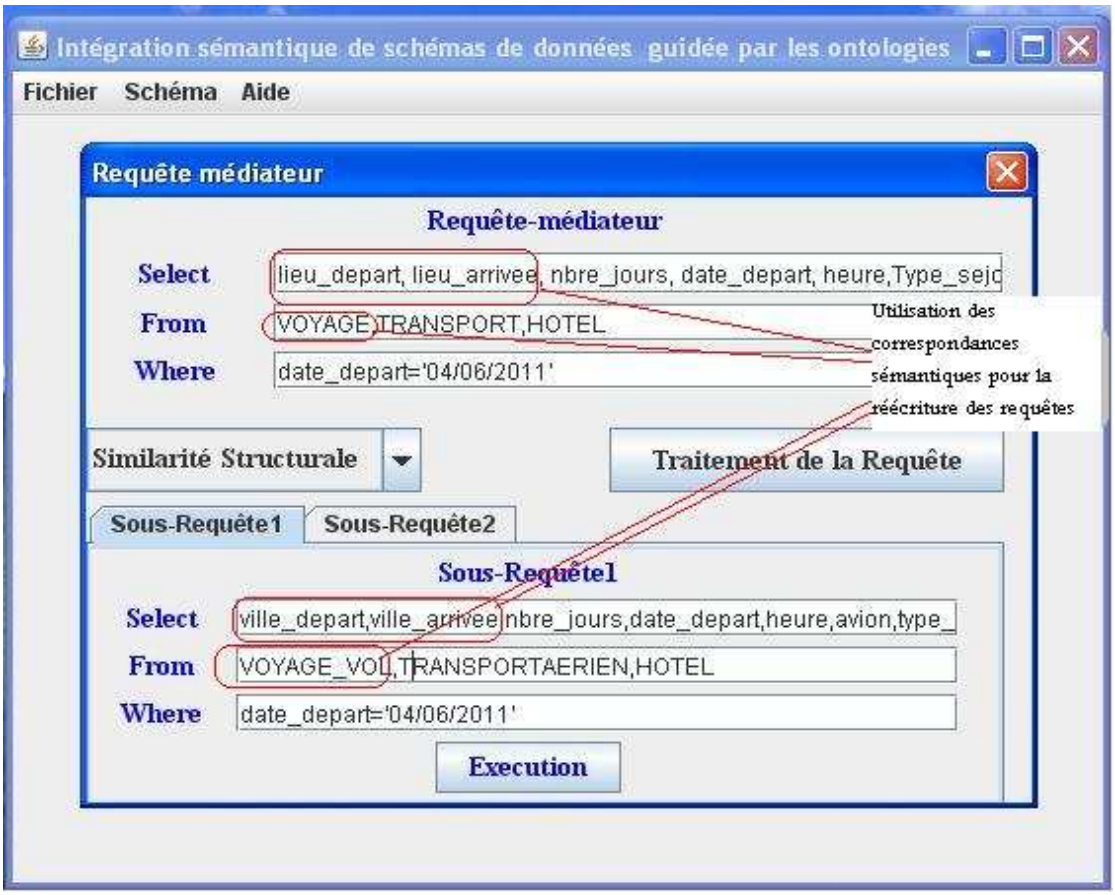


FIG IV.5 –Réécriture de requête médiateur

IV.5.4 Requête –Médiateur

Ce module nous montre comment le système décompose la requête posée par utilisateur ainsi que le traitement de sous requête de chaque source participant à la requête.

IV.6 Expérimentation

Nous présentons les expérimentations que nous avons effectuées pour évaluer notre méthode d'enrichissement sémantique des schémas des données. L'expérimentation a été effectuée sur des concepts représentant les attributs et les relations du schéma global et des schémas des sources de données, le tableau Tab IV.1 récapitule le nombre de concepts pour chaque source.

	Concept-attribut	Concept-relation
Schéma Global	20	3
Schéma Source1	20	3
Schéma Source2	14	2

Tab IV.1 tableau des concepts associés aux schémas des sources

Pour mieux comprendre les expérimentations sur ces travaux, et montrons l'impact de l'enrichissement sémantique par les ontologies sur le résultat ainsi

que sur l'automatisation du processus d'intégration, nous divisons les tests en deux types:

Test basé sur la similarité linguistique: la similarité entre deux entités (concepts) provenant des deux ontologies est calculée à partir de différentes informations disponibles sur leurs noms, qui décrivent textuellement les concepts afin de faciliter leurs compréhensions aux utilisateurs. Les techniques linguistiques basées sur des algorithmes qui utilisant des mesures de similarité entre les termes.

Test basé sur la similarité structurelle: exploite les informations taxonomiques dans les structures des ontologies pour calculer la similarité structurelle entre deux entités. Cette dernière est calculée en combinant la similarité (linguistique) entre leurs voisins dans l'arbre de l'ontologie (super-entités directes, sous-entités directes, et les entités au même niveau).

IV.6.1 Présentation des résultats

Dans cette section nous présentons les résultats que nous avons obtenus lors de l'expérimentation de notre méthode d'enrichissement sémantique. Le tableau Tab IV.2 montre les résultats obtenus suivant le type de test.

L'utilisateur formule une requête sur la vue unifiée en utilisant le vocabulaire retenu dans le schéma global. La requête est envoyée ensuite au médiateur afin d'être décomposée en plusieurs requêtes sur différentes sources de données. Dans notre exemple ci-dessous, le médiateur décompose la requête en trois parties, à savoir les données à afficher qui sont entre les mots-clés SELECT et FROM, les relations interrogées qui se trouvent entre les mots-clés FROM et WHERE ainsi que les relations et les attributs qui sont contenus dans la condition et qui viennent après le mot-clé WHERE. Le traitement des requêtes repose sur les étapes suivantes:

Etape1: pour chaque attribut/relation (concept dans l'ontologie globale) appartient à la première partie on cherche l'attribut/relation et sa source qui lui correspond en exploitant le résultat des correspondances sémantiques trouvé par l'alignement de l'ontologie globale avec les ontologies des sources.

Etape2:décomposition de la requête en sou requête suivant les sources pertinentes en remplaçant les attributs/relations (concepts globaux) par les attributs/relation (concepts locaux) trouvés dans l'étape1.

Etape3:exécution des sous requête sur les sources de données

Soit la requête médiateur (posée par un utilisateur)

SELECT lieu_depart, lieu_arrivee, nbre_jours, date_depart, heure,type_sejour,moyen ***FROM*** VOYAGE, TRANSPORT
WHERE VOYAGE .idT= TRANSPORT.idT

IV.6.1.1 Test basé sur la similarité linguistique: après l'analyse de cette requête et l'application des algorithmes de similarité linguistique le résultat retourné, sous forme de correspondances entre les concepts globaux et les concepts locaux, est récapitulé dans le tableau IV.2 :

Concepts du schéma global (attribut/relation)	**Concept correspondant du schéma source1**	**Concept correspondant du schéma source2**
lieu_depart	/	/
lieu_arrivee	/	/
nbre_jours	nbre_jours	nbre_jours
date_depart	date_depart	date_depart
heure	heure	heure
type_sejour	type_sejour	type_sejour
moyen	/	/
VOYAGE	/	/
TRANSPORT	/	/
idT	idT	idT

Tab IV.2 correspondances linguistique entre les concepts du schéma global et les concepts des schémas des sources

La réécriture de la requête suivant les étapes ci-dessus en exploitant les correspondances du tableau IV.2 donne les deux sous requête:

Sous- requête1: SELECT nbre_jours, date_depart, heure, type_sejour
FROM VOYAGE, TRANSPORT
WHERE VOYAGE .idT= TRANSPORT.idT

Sous- requête2: SELECT nbre_jours, date_depart, heure, type_sejour
FROM VOYAGE, TRANSPORT
WHERE VOYAGE .idT= TRANSPORT.idT

Le test, basé sur la similarité linguistique, montre que ces méthodes donnent un résultat incomplet de correspondances sémantiques et par conséquent le traitement de la requête est incorrect par exemple: l'existante des attributs lieu_depart et lieu_arrivee dans la requête médiateur et l'absence de ses

correspondances au niveau de sous requêtes. De ce fait, l'information délivrée au utilisateur soit incorrecte soit incomplet.

IV.6.1.2 Test basé sur la similarité structurelle:repose principalement sur la structure locale de l'entité pour déduire la similarité. Les informations linguistiques dans la définition d'une entité sont aussi exploitées avec les primitives rdf:id. Les valeurs de similarité linguistique sont prises comme points d'articulation dans la recherche des correspondances basées sur la structure comme la relation de subsomption qui permet d'exploiter les avantages et la capacité d'expression puissante du langage OWL en déduisant la similarité entre des entités des ontologies. Exemple le concept **lieu_depart** de l'ontologie globale correspond **ville_depart** dans l'ontologie locale de la sources S1. Le résultat des correspondances basé sur la similarité structurelle présenté par le tableau IV.3

Concepts du schéma global (attribut/relation)	**Concept correspondant du schéma source1**	**Concept correspondant du schéma source2**
lieu_depart	**ville_depart**	**port_depart**
lieu_arrivee	**ville_arrivee**	**port_ arrivee**
nbre_jours	nbre_jours	nbre_jours
date_depart	date_depart	date_depart
heure	heure	heure
type_sejour	type_sejour	type_sejour
moyen	**avion**	**bateau**
VOYAGE	**VOYAGE_VOL**	**VOYAGE_MARINIER**
TRANSPORT	**TRANSPORTAERIEN**	**TRANSPORTFERRIES**
idT	idT	idT

Tab IV.3 correspondances structurelle entre les concepts du schéma global et les concepts des schémas des sources

Ce test montre que cette méthode fournit les résultats attendus des correspondances sémantiques pour l'intégration de schémas et elle prend en compte la structure hiérarchique des ontologies. Les concepts d'une ontologie sont organisés hiérarchiquement par une relation d'ordre partiel qui est la relation de subsomption (”est-une”) permettant d'organiser sémantiquement les concepts par niveau de généralité: intuitivement, un concept C1 subsume un concept C2 si C1 est plus général que C2 au sens où l'ensemble d'individus représenté par C1 contient l'ensemble d'individus représenté par C2. Par exemple, le concept **lieu_depart** subsume le concept **ville_depart**. Cela permet de trouver toutes les correspondances de ce type et compléter les méthodes linguistiques. Les résultats renvoyés par les algorithmes sont très bons et

toutes les correspondances correctes sont bien trouvées comme le montre le tableau IV.3.

Rappelons que la génération des ces correspondances sert à la décomposition et à la découverte des sources pertinentes. Le traitement de la requête de l'exemple ci-dessus donne les deux sous requêtes:

Sous- requête1: **SELECT ville_depart**, **ville_arrivee**,nbre_jours, date_depart, heure, type_sejour
FROM VOYAGE_VOL, **TRANSPORTAERIEN**
WHERE VOYAGE_VOL.idT= TRANSPORTAERIEN.idT

Sous- requête2: **SELECT port_depart**, **port_ arrivee**, nbre_jours, date_depart, heure, type_sejour
FROM VOYAGE_MARINIER, **TRANSPORTFERRIES**
WHERE VOYAGE_MARINIER.idT= TRANSPORTFERRIES.idT

IV.6.2 Mesures d'évaluation

Pour évaluer la précision et l'efficacité des l'algorithmes, nous employons les mesures de précision et de rappel. Ce sont les mesures bien utilisées dans le domaine de recherche d'information et ensuite appliquées dans le domaine d'alignement d'ontologies pour permettre une analyse fine des performances de système. Le calcul de ces mesures est basé sur la comparaison entre les correspondances produites par un système automatique d'alignement et un ensemble de correspondances de référence produit par un expert de domaine.

Le **rappel** est la proportion de correspondances correctes renvoyées par l'algorithme parmi toutes celles qui sont correctes (en incluant aussi des correspondances correctes que l'algorithme n'a pas détectées). Le rappel mesure l'efficacité d'un algorithme. Plus la valeur de rappel est élevée, plus le résultat de l'algorithme couvre toutes les correspondances correctes.

La **précision** est le nombre de correspondances pertinentes retrouvé rapporté au nombre de correspondances total proposé par l'algorithme.

La **F-mesure** est un compromis entre le rappel et la précision, qui combine la précision et le rappel est leur pondération, nommée F-mesure (soit F-measure en anglais) ou F-score : **F=2.(précision* rappel / précision + rappel)**.

F – le nombre des correspondances renvoyées par l'algorithme
T – le nombre des correspondances correctes déterminées manuellement par des experts (celles dans le fichier de référence)
C – le nombre des correspondances correctes trouvées (appelé aussi vrais positifs)

I=F-C – nombre des correspondances incorrectes trouvées (appelé aussi faux positifs)
M =T-C – nombre des correspondances correctes mais pas trouvées (appelé aussi faux négatifs).

Méthode	F	T	C	I=F-C	M=T-C	Précision P=C/F	Rappel R=C/T	F-mesure
Linguistique	23	23	13	0	10	0.56	0.56	0.55
Structurelle	23	23	23	0	0	1.0	1.0	1.0

Tab IV.4 Résultats de l'expérimentation des deux méthodes Linguistique et structurelle

IV.6.3 Validation des résultats

Le tableau IV.4 montre les résultats en termes de précision, de rappel et de F-Mesure de notre méthode d'enrichissement sémantique. La première observation importante est que, dans le cas où nous nous contentons la méthode linguistique (basée sur le traitement des chaînes de caractères) nous obtenons alors seulement une valeur de 0.56 pour le rappel alors que dans le cas où nous considérons la méthode structurelle, qui exploite l'enrichissement sémantique notamment la relation de subsomption, nous obtenons alors une valeur proche de 1 pour le rappel. Ce résultat montre l'intérêt de la description structurelle des concepts dans l'augmentation du rappel.

Notre principal objectif est précisément d'obtenir un rappel satisfaisant pour notre méthode qui est complètement automatique. En effet, nous avons choisi de garder toute information linguistique ainsi que les informations structurelle qui ne sont pas exploitées comme la relation entre les concepts. Un autre résultat important est l'augmentation de la F-Mesure grâce au fait que la valeur du rappel augmente de 44 % entre la première méthode et la deuxième méthode.

IV.7 Conclusion

Dans ce chapitre, nous avons présenté les résultats de l'expérimentation de notre méthode d'enrichissement sémantique. Cette expérimentation a été effectuée sur une ontologie globale associée au schéma d'interrogation avec des ontologies locales associées aux schémas des sources de données, traitant du domaine du voyage. À travers cette expérimentation nous avons pu montrer l'intérêt de doter les sources de données par des ontologies conceptuelles pour le processus d'intégration de ces sources d'une part, et pour les schémas d'autre part.

Les résultats obtenus pour le rappel de la méthode ont pu confirmer l'intérêt de conserver à la fois ces informations syntaxiques et des mesures de similarité utilisant les structures arborescentes de la taxonomie. Nous avons également montré que les informations structurelles, qui sont les propriétés du langage OWL, peuvent être exploitées pour enrichir un schéma de données par des ontologies pour compléter des réponses aux requêtes de l'utilisateur ou pour obtenir des réponses approchées en cas d'absence de réponses.

Cette approche, appelée intégration par articulation a priori d'ontologies, suppose l'existence d'une ontologie(s) de domaine, mais elle laisse chaque source autonome quant à la structure de sa propre ontologie. Au lieu de réaliser l'intégration des ontologies a posteriori, comme c'est le cas dans toutes les approches classiques, notre approche exige de l'administrateur de chaque source à intégrer que :

1. sa source de données contienne une ontologie, et
2. qu'il s'engage sur l'ontologie de domaine, c'est-à-dire qu'il ajoute a priori à cette ontologie les relations (articulations) existantes entre celle-ci et l'ontologie de domaine.

Cette hypothèse est réaliste dans tous les secteurs où des ontologies de domaines existent ou apparaissent, et où chaque administrateur qui publie sa source de données souhaite à la fois lui conserver sa structure propre, et la rendre accessible à des usagers de façon homogène à travers une ontologie de domaine.

Chapitre V: Conclusion Générale

V.1 Conclusion générale

L'intégration de sources de données, hétérogènes, autonomes et évolutives pose à la fois des problèmes structurels et des problèmes sémantiques. Les travaux concernant l'hétérogénéité structurelle sont relativement anciens et ont abouti à diverses approches permettant de la traiter dans le contexte des fédérations de bases de données ou des multi-bases de données. L'hétérogénéité sémantique, par contre, reste la plus importante difficulté .Plusieurs systèmes récents d'intégration de sources de données hétérogènes utilisent les ontologies afin de résoudre les conflits sémantiques. Les principales limitations de ces systèmes sont soit leur absence d'automatisation en l'absence d'une ontologie partagée, soit l'absence d'autonomie (ou la faible autonomie) des sources dans le cas où une ontologie partagée est utilisée, et, dans tous les cas, l'absence de prise en compte des besoins d'évolution asynchrone tant des différents sources de données que des ontologies.

Nous avons présenté dans ce mémoire notre travail sur l'intégration de schémas de données guidée par une ontologie. Le rôle central de l'ontologie dans notre système d'intégration présente un intérêt majeur pour la généricité de l'approche, puis qu'elle permet de stocker toute la connaissance du domaine d'application étudié et qu'il suffit de modifier l'ontologie pour que notre approche soit applicable à d'autres domaines. Nous précisons dans cette conclusion générale, différentes contributions portant sur des aspects plus spécifiques de notre approche :

Première contribution : Une architecture d'intégration guidée par d'ontologies: Plusieurs travaux sur l'intégration de données utilisant des ontologies ont été reportés dans la littérature. Les ontologies sont utilisées dans les systèmes issus de ces travaux, exclusivement pour représenter le schéma global de médiation et/ou les schémas des sources locales. Eventuellement, plusieurs ontologies peuvent être utilisées à cet effet. L'architecture à base de médiateurs que nous avons proposé repose sur l'utilisation des technologies du Web Sémantique (OWL) et de deux types d'ontologies (globale pour le schéma du médiateur et locale pour la source). Les ontologies servent à définir le schéma global d'intégration (ontologie globale) et les différentes sources à intégrer. Les ontologies qui représentent les sources à intégrer sont appelées schémas virtuels de sources ou ontologies locales, et les correspondances sont établies entre l'ontologie globale et les différentes ontologies locales.

Deuxième contribution : le degré d'automatisation du processus d'intégration:
Dans cette approche d'intégration nous avons proposé que chaque source doit contenir à la fois sa propre ontologie et les relations sémantiques qui l'articulent a priori avec l'ontologie partagée. Ceci permet une intégration automatisée des différentes sources de données.

Troisième contribution: correspondances sémantiques entre plusieurs ontologies:
Nous avons proposé, au niveau du médiateur, un module d'alignement d'ontologies basé sur un enchaînement d'algorithmes pour calculer la valeur de similarité et génère les correspondances entre les concepts de l'ontologie globale et ceux de l'ontologie locale. Ces correspondances servent à la réécriture de sous-requêtes et à la découverte des sources pertinentes.

V.2 Perspectives

De nombreuses perspectives tant à caractère théorique que pratique peuvent être envisagées. Nous présentons dans cette section celles qui nous semblent les plus prometteuses.

- Les ontologies utilisées dans le cadre de notre mémoire ont été créées dans le but de valider nos prototypes. Nous les avons créées de telle manière qu'elles soient les plus proches possibles de celles qui peuvent contenir des schémas des sources à intégrer. Afin de rendre plus général notre système, il est préférable de l'utiliser dans d'autres domaines telles que les banques, les catalogues électroniques etc.

- La réalisation d'un système d'intégration de sources de données basé sur les ontologies et entièrement automatique dans un environnement décentralisé.

- Les requêtes envoyées à notre médiateur sont écrites en SQL. Nous devrons créer une interface d'interrogation qui permette à l'utilisateur de créer sa requête en mode graphique. Il n'aura plus à apprendre un langage de requêtes mais simplement à sélectionner les relations et attributs à interroger.

- Une description détaillée des concepts des ontologies associées aux sources rendre la localisation de ces sources pertinentes est plus efficace.

Table des figures

Liste des tableaux

Bibliographies

[1] Goh.C, Bressan.S, Madnick.E, and Siegel.M. Context interchange: New features and formalisms for the intelligent integration of information. ACM Transactions on Information Systems,17(3) :270–293, 1999.

[2] Inmon W.H.,"Building the Data Warehouse", John Wiley&Sons, ISBN 0471-14161-5.

[3] Jarke M., Jeusfeld M.A., Quix C., Vassiliadis P., "Architecture and quality in data warehouses", Proceedings of the 10th Conference on Advanced Information Systems Engineering (CAiSE '98), Pisa, Italy, June, 8-12, 1998.

[4] Jeusfeld M.A., Quix. C, Jarke. M.,"Design and Analysis of Quality Information for Data Warehouses", Proceedings of the17th International Conference on Conceptual Modelling R'98, Singapore, Nov 16-19, 1998.

[5] Vavouras A., Gatziu S., Dittrich K.R., "The SIRIUS Approach for Refreshing Data Warehouses Incrementally", BTW'99, pp 80-96, 1999.

[6] Widom J., "Research problems in data warehousing", Proceedings of the 4th
International Conference on Information and Knowledge Management - ACM CIKM'95, November 29-December 2 1995, Baltimore (Maryland, USA).

[7] Hammer J., Garcia-Molina H., Widom J., Labio W. J. and Zhuge Y., "The Stanford Data Warehousing Project", IEEE Data Engineering Bulletin, June 1995.

[8] Wiener J. L., Gupta H., Labio W. J., Zhuge Y., Garcia-Molina H.,Widom J., "A System Prototype for Warehouse View Maintenance", In Proceedings of the ACM Workshop on Materialized Views: Techniques and Applications, pp. 26-33, Montreal (Canada), June 7 1996.

[9] Wiederhold G..Mediators in the architecture of future information systems, computer, Vol. 25(3). pp.38-49, 1992.

[10] Chawathe S., Garcia-Molina H., Hammer J., Ireland K.,Papakonstantinou Y., Ullman J., Widom J., "The TSIMMIS Project: Integration of Heterogeneous Information Sources", In Proceedings of IPSJ Conference, pp. 7-18, Tokyo, Japan, October 1994.

[11] Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y.,Ullman J., Widom J., "Integrating and Accessing Heterogeneous Information Sources in TSIMMIS", In proceedings of the AAAI Symposium on Information Gathering, pp. 61-64, Stanford (California, USA), March 1995.

[12] hawathe.S, Garcia-Molina.H. Hammer.J, Ireland.K, Papakonstantinou.Y, J. D. Ullman, and J. Widom. The tsimmis project : Integration of heterogeneous information sources. Proceedings of the 10th Meeting of the Information Processing Society of Japan, pages 7–18, Mars 1994.

[13] Beneventano.D, Bergamaschi.S, Castano.S, Corni.A, Guidetti.R, Malvezzi.G, Melchiori.M, and Vincini.M. Information integration: The MOMIS project demonstration. In The VLDB Journal,pp 611–614, 2000.

[14] Genesereth M.R., Keller A.M.,and Duschka O. M.. Infomaster : an information integration system. In ACM SIGMOD International Conference on Management of Data, pp 539–542, 1997.

[15] Gomez. G.-L. G. Construction automatisée de l'ontologie de systèmes médiateurs. Application à des systèmes intégrant des services standard accessibles via le Web. PhD thesis, Université Paris XI Orsay, 2005.

[16] Levy. A.Y, Rajaraman.A, and J. Ordille.J. The world wide web as a collection of views: Query processing in the information manifold. Proceedings of the International Workshop on Materialized Views : Techniques and Applications (VIEW'1996), pp 43–55, June 1996.

[17] Breitbart.Y, Silberschatz.A, and Thompson.G.R. Reliable transaction management in a multidatabase system.In Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, pages 215–224, 1990.

[18] Sheth.A.P. and Larson.J.A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys,22(3) :183–236, 1990.

[19] Miller.A. Wordnet : A lexical database for english. Communications of the ACM, 38(11) :39–41, November 1995.

[20] Beneventano D. and Bergamaschi.S. The momis methodology for integrating heterogeneous data sources. In IFIP World Computer Congress, Toulouse, France., August 2004

[21] Fran.F¸ Goasdoué.C, Lattés.V, and Rousset M. C. The use of carin language
and algorithms for information integration : The picsel system. International Journal of Cooperative Information Systems (IJCIS), 9(4) :383–401, December 2000.

[22] Decker.S, Erdmann.M, Fensel.D, and Studer.R. Ontobroker : Ontology based
access to distributed and semi-structured information. In DS-8, pages 351–369,
1999.

[23] Hacid.M-H and Reynaud.C. L'intégration de sources de données. La revue I3 :Information - Interaction - Intelligence,Vol5 n°1, 2005.

[24] McGuinness.D.L and Harmelen.F. Owl web ontology language overview. W3C Recommendation, 10 February 2004.

[25] Wache.H, ogele T. V, U. Visser, Stuckenschmidt.H, Schuster.G, Neumann.H, and ubner S.H. Ontology-based integration of information - a survey of existing approaches. Proceedings of the International Workshop on Ontologies and Information Sharing, pages 108–117, August 2001.

[26] Arens.Y and Knoblock.C.A. Sims :Retrieving and integrating information from multiple sources. Proceedings of the International Conference on Management of Data (SIGMOD'1993), pages 562–563, May 1993.

[27] Mena.E, Kashyap.V, Sheth A. P, and Illarramendi.A. OBSERVER : An approach for query processing in global information systems based on interoperation across pre-existing ontologies. In Conference on Cooperative Information Systems, pages 14–25, 1996.

[28] Jean.S. Langage d'exploitation de base de données ontologiques. Mémoire pour l'obtention du DEA T3IA, Université de Poiters, juin 2004.

[29] Minsky.M. Matter, mind and models. International Processing Congress, 1:45–49, 1996.

[30] Visser P. R. S., Beer.M, BenchCapon.T, Diaz .B. M, and. Shave M. J. R. Resolving ontological heterogeneity in the kraft project. 10th International Conference on Database and Expert Systems Applications (DEXA'99), pages 668–677, September 1999.

[31] Vögele T., Hübner S., Schuster G. Busteran information broker for the semantic web. Knstliche Intelligenz, 3 :31–34, July2003.

[32] Pierra G. Context-explication in conceptual ontologies : Plib ontologies and their use for industrial data. to appear in Journal of Advanced Manufacturing Systems (JAMS), 2006.

[33] Pierra G. Un modèle formel d'ontologie pour l'ingénierie, le commerce électronique et le web sémantique : Le modèle de dictionnaire sémantique plib. Journées Scientifique WEBSEMANTIQUE, Paris, 2002.

[34] Busse S., Kutsche R.-D., Leser U. and Weber H. Federated Information Systems:Concepts, Terminology and Architectures,TechnicalReport n°99-9,Technical University of Berlin, 38 p; 1999.

[35] Litwin W., Abdelattif A., Zeroual A., Nicoals B & Vigier P.. MSQL : A multidatabase Language, Information Science, 48(1-3), pp. 59-101; 1989.

[36] Laurini R. and Millert-Raffort F. Les bases de données en géomatique. Paris : Hermès, 340 p; 1993.

[37] Parent Ch. et Spaccapietra S.. Intégration de bases de données : panorama des problèmes et des approches, Revue ISI : Ingénierie des Systèmes d'Information, 4(3), pp. 333-359; 1996;.

[**38]** Sheth A. and Larson J. Federated database systems for managing distributed, heterogeneous and autonomous databases, ACM Computing Surveys, 22(3), pp. 183-236; 1990.

[39] Christophe R. Terminologie et ontologie. Revue Langages, numéro 157,Mars 2005.

[40] Thomas G. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220, 1993.

[41] Napoli. A. Une brève introduction aux logiques de descriptions. Cours de la Logique de Description, 2005.

[42] Lortal G État de l'art ontologies et intégration/fusion d'ontologies, 2002.

[43] Do H-H, Rahm E, COMA A system for flexible combination of Schema Matching Approaches,2002.

[44] Choi NAMYOUN, Song IL-YEOL et Han HYOIL. A survey on ontology mapping. SIGMOD Rec., 35(3):34–41, September 2006.

[45] Erhard R et Philip B. A survey of approaches to automatic schema matching. The VLDB Journal, 10(4):334–350, 2001.

[46] Euzenat, J. Bach, T.L., Barrasa, J .,Bouquet,P.,Bo,JD. , Dieng Kuntz, R., Ehrig, M,Hauswirth ,M, Jarrar ,M.,Lara .R.Maynard D., Napoli, A.Stamou, G.Stuckenschmidt, H.shvaiko,,P., Tessris, S, Acker,S.V et Zaihraveu, I,State of the art on ontology alignment, deliver abele 2.2.3IS Knowledge web NoE ,80p, 2000

[47] Rahm , E. et Bernstein , P. Asurvey of approaches to automatic schema matching .VLDB Journal ,10(4):334-350,2001

[48] Shviako, P., Euzenat, J. A Survey of Schema-based Matching Approaches. Journal on Data Semantics, 2005

[49] Jaro .M. A. Probabilistic linkage of large public health data files disc: pp687-689). Statistic in Medecine14:491-498;1995

[50] Maynard, D.G. et Ananiadou, S. Term extraction using a similarity-based approach.In Recent Advances in Computationl Terminology. John Benjamins, 1999.

[51] Madhaven J, Bernstein P. et Rahm, E. Generic schema matching with cupid. Dans Proceedings of the 27 th International Conference on Very large Data Bases, pages 49-58.Morgan Kaufmann publishersInc., 2001

[52] Li W-S et Clifton C. Semint: a tool for identifying attribute correspondences in heterogeneaus databases using neural networks. Data Knowl.Eng. 33(1):49-84, 2000.

[53] Giunchiglia F et Shvaiko P Semantic matching. Dans Proc. IJCAI 2003 Workshop on ontologies and distributed Systemes, Acapulco (MX), pp 139-146, 2003.

[54] Giunchiglia F, Shvaiko P. et Yatskevich, M.S-Match: an algorithm and an implementation of semantic matching. Dans Proceedings of ESWS 2004, Heraklion (GR), pages 61-75, 2004.

[55] Bouquet P, Giunchiglia F, vanHarmelen F, Serafini L., et Stuckenschmidt, H.Contextualizing Ontologies. Dans Journal of Web Semantics, 2004

[56] Ehrig M. & Staab S. . Qom - quick ontology mapping. In International SemanticWeb Conference 2004, pp. 683–697; 2004.

[57] Bellatreche L, Pierra G, Nguyen-Xuan D, and Dehainsala H. Intégration de sources de données autonomes par articulation a priori d'ontologies. In Ictes du XXIIème Congrés INFORSID, Biarritz, France, pages 283–298, 2004.

[58] Winlles, W.E The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04, http://www.census.gov/srd/www/byname.html.1999.

[59] Wiederhold G. Mediators in the Architecture of Future Information Systems, IEEE Computer, 25(3), pp. 38-49;1992.

[60] Parent C. et Spaccapietra S. Database Integration the Key to Data Interoperability, In Papazoglou M., Spaccapietra S. and Tari Z. (Eds.) : Advances in Object-Oriented Data Modeling. MIT Press; 2001.

Annexe

Ontologie Globale associée au Schéma Médiateur

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://www.owl-ontologies.com/Global-Ontology2011.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/Global-
Ontology2011.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="comfort"/>
  <owl:Class rdf:ID="VOYAGE"/>
  <owl:Class rdf:ID="idT"/>
  <owl:Class rdf:ID="nbre_etoiles"/>
  <owl:Class rdf:ID="idV"/>
  <owl:Class rdf:ID="type_sejour"/>
  <owl:Class rdf:ID="moyen"/>
  <owl:Class rdf:ID="restaurant"/>
  <owl:Class rdf:ID="region"/>
  <owl:Class rdf:ID="prix"/>
  <owl:Class rdf:ID="idH"/>
  <owl:Class rdf:ID="date_depart"/>
  <owl:Class rdf:ID="lieu_depart"/>
  <owl:Class rdf:ID="HOTEL"/>
  <owl:Class rdf:ID="type_trajet"/>
  <owl:Class rdf:ID="ville"/>
  <owl:Class rdf:ID="heure"/>
  <owl:Class rdf:ID="nom"/>
  <owl:Class rdf:ID="nbre_jours"/>
  <owl:Class rdf:ID="lieu_arrivee"/>
  <owl:Class rdf:ID="TRANSPORT"/>
</rdf:RDF>
```

Ontologie1 associée à la source1

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
    xmlns="http://www.owl-ontologies.com/source1.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/source1.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="restaurant"/>
  <owl:Class rdf:ID="type_classe"/>
  <owl:Class rdf:ID="TRANSPORT"/>
  <owl:Class rdf:ID="HOTEL"/>
  <owl:Class rdf:ID="VOYAGE_VOL">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="VOYAGE"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="TRANSPORTAERIEN">
    <rdfs:subClassOf rdf:resource="#TRANSPORT"/>
  </owl:Class>
  <owl:Class rdf:ID="ville"/>
  <owl:Class rdf:ID="nom"/>
  <owl:Class rdf:ID="idH"/>
  <owl:Class rdf:ID="avion"/>
  <owl:Class rdf:ID="date_depart"/>
  <owl:Class rdf:ID="idV"/>
  <owl:Class rdf:ID="nbre_etoiles"/>
  <owl:Class rdf:ID="ville_arrivee"/>
  <owl:Class rdf:ID="prix"/>
  <owl:Class rdf:ID="region"/>
  <owl:Class rdf:ID="idT"/>
  <owl:Class rdf:ID="type_formule"/>
  <owl:Class rdf:ID="type_sejour"/>
  <owl:Class rdf:ID="nbre_jours"/>
  <owl:Class rdf:ID="heure"/>
  <owl:Class rdf:ID="type_trajet"/>
  <owl:Class rdf:ID="ville_depart"/>
</rdf:RDF>
```

Ontologie2 associée à la source2

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
   xmlns="http://www.owl-ontologies.com/source2.owl#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://www.owl-ontologies.com/source2.owl">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="date_depart"/>
 <owl:Class rdf:ID="VOYAGE_MARINIER">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="VOYAGE"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="port_arrivee">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="lieu_arrivee"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="idH"/>
 <owl:Class rdf:ID="type_sejour"/>
 <owl:Class rdf:ID="bateau">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="moyen"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="type_trajet"/>
 <owl:Class rdf:ID="idV"/>
 <owl:Class rdf:ID="port_depart">
  <rdfs:subClassOf>
   <owl:Class rdf:ID="lieu_depart"/>
  </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:ID="idT"/>
 <owl:Class rdf:ID="prix"/>
 <owl:Class rdf:ID="TRANSPORT"/>
 <owl:Class rdf:ID="TRANSPORTFERRIES">
  <rdfs:subClassOf rdf:resource="#TRANSPORT"/>
 </owl:Class>
```

```
<owl:Class rdf:ID="nbre_jours"/>
  <owl:Class rdf:ID="heure"/>
  <owl:Class rdf:ID="comfort"/>
  <owl:DatatypeProperty rdf:ID="idende">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#idV"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="attributde">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#prix"/>
          <owl:Class rdf:about="#port_arrivee"/>
          <owl:Class rdf:about="#port_depart"/>
          <owl:Class rdf:about="#nbre_jours"/>
          <owl:Class rdf:about="#date_depart"/>
          <owl:Class rdf:about="#heure"/>
          <owl:Class rdf:about="#type_sejour"/>
          <owl:Class rdf:about="#idH"/>
          <owl:Class rdf:about="#idT"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```