الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma –

Faculté des Mathématique, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Master Académique

Thème :

Intégration sémantique de sources de données hétérogènes : Une approche à base d'ontologie

Encadré Par :

M^me^ Aicha AGGOUNE

Présenté Par :

Asma ABADLIA

Zeineb BOUMAHRA

Juin 2015

Remerciements

En préambule à ce mémoire nous remerciant ALLAH qui nous aide et nous donne la patience et le courage durant ces langues années d'étude.

Ces remerciements vont tout d'abord au corps professoral et administratif de la Faculté 08 Mai 1945 des Mathématique, d'Informatique et des Sciences de la matière pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.

La première personne que nous tenons à remercier est notre encadreur M^me^ « Aggoune Aicha » pour l'orientation, la confiance, la patience qui a constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Nous voudrions remercier les membres de jurés.

Nous tenons à remercions tous les enseignants qui nous ont suivis durant notre formation pour leurs valeureux conseils. Nous les remercions pour leur intérêt à ce travail et la bonne formation que nous avons eu au département d'informatique.

Nous n'oublions pas nos parents pour leur contribution, leur soutien et leur patience.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Dédicaces

Je voudrais dédier cet humble travail

A mon cher père Rachid et ma Chère maman Dalila.

A mes frères Chihab, Charaf, Didine et Ali.

A mon encadreur Aggoune Aicha.

A toute ma famille.

A mon binôme Zaineb Boumahra

A mes amis Ahlem, Selma, Zeineb, Sara, Afef, Safa. Et tous les étudiants d'informatique.

A tous ccux qui m'aiment.

Asma Abadlia

Dédicaces

Je voudrais dédier cet humble travail

À mes parents, mes sœurs et mes frères à toute ma famille ;

À mon mari et toute sa famille ;

À mon binôme Asma et toute sa famille ;

Et finalement à tous mes amis ;

Zeineb Boumahra

Plan de travail

Chapitre3 : Conception de système de médiation sémantique

Chapitre 4 : Approche proposée pour l'intégration sémantique à base d'ontologie

Chapitre 5 : Mise en œuvre de système de médiation sémantique

Liste des figures

Chapitre 1 : Web sémantique et ontologies

Chapitre 2 : Systèmes d'intégration de données hétérogènes

Chapitre 3 : Conception de système de médiation sémantique

Chapitre 5: mise en œuvre de système de médiation sémantique

Liste des tables

Résumé

La disponibilité croissante de sources de données hétérogènes et dispersées pose le problème de leur accès de façon efficace. L'intégration de ces sources en particulier les bases 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. Deux approches d'intégration ont été proposées dans la littérature : Approche de médiation et approche d'entrepôt de données. Cependant, ces approches sont confrontées aux problèmes d'hétérogénéité, tant en ce qui concerne la structure et la syntaxe, qu'en ce qui, concerne la sémantique. L'objectif de ce travail est de proposer une approche de médiation sémantique à base d'ontologie pour traiter ces problèmes et assurer un accès efficace.

Introduction Générale

Contexte :

L'explosion du nombre de sources d'information accessibles via le Web multiplie les besoins de techniques d'intégration des sources de données hétérogènes. L'intégration des données est le processus par lequel plusieurs sources de données autonomes, réparties et hétérogènes ayant chacune un schéma local, sont intégrées de telle sorte qu'elles apparaissent comme une source unique et donnent aux utilisateurs l'illusion de n'interagir qu'avec cette seule source.

Les systèmes d'intégration de données fournissent une vue unifiée de diverses sources hétérogènes, autonomes et réparties, facilitant ainsi l'accès à l'information. Ceci est réalisé par l'utilisation d'un schéma global ou d'une ontologie globale, qui fournit une vue réconciliée des sources locales.

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 : *approche matérialisée* ou *entrepôt de données et l'approche virtuelle, appelée aussi système de médiation.*

La première approche consiste à créer un entrepôt de données à partir des sources locales, dupliquant ainsi les données. Cette approche a l'avantage de fournir de temps d'accès rapide mais nécessite un support de stockage volumineux et fiable et des outils spécifiques pour le traitement préalable des données.

La seconde approche d'intégration où la vue unifiée est virtuelle et les données restent stockées dans les sources d'origine. Un système de médiation repose sur deux niveaux : le médiateur et les adaptateurs. Le médiateur a pour fonction d'offrir une vue unifiée des différentes sources de données grâce au schéma global, cachant en cela leur hétérogénéité et leur répartition. Il offre un protocole d'accès et un langage de requêtes commun à toutes ces sources. Quant à l'adaptateur, il adapte la requête exprimée dans le langage commun au langage de la source, tout en utilisant le bon protocole d'accès.

Dans le cadre de notre travail nous avons adopté cette approche de médiateur car elle permet d'avoir une vue *fraîche* des données, sans avoir à les dupliquer ni à les transformer.

Par ailleurs, la requête utilisateur est exprimée en fonction du schéma global, et exécutée par les sources locales via un mapping ou mise en correspondance entre ce schéma global et les schémas locaux des sources. Ce mapping constitue un traitement clé dans le processus général. Il sera utilisé pour réécrire la requête initialement exprimée en fonction du schéma global, en des sous-requêtes exprimées, chacune, en fonction du schéma local de la source qui l'exécutera.

Deux principales approches existent pour définir le mapping entre le schéma global et les schémas des sources: *Local As View* (LAV) et *Global As View* (GAV).

Dans l'approche GAV, le schéma global est exprimé à l'aide de vues sur les schémas locaux, à l'inverse de l'approche LAV qui nécessite la description des sources locales en fonction du schéma global.

D'une manière générale, la représentation unifiée des données nécessite la détection et la résolution d'éventuels conflits de schémas et de données, tant du point de vue structurel que sémantique.

Problématiques abordées:

Le problème de l'intégration de données a été posé depuis un certain nombre d'années. De nos jours, aussi bien les entreprises que les communautés scientifiques et les gouvernements éprouvent un large besoin de rendre accessible leurs données, mais aussi d'accéder à des données d'autres structures. Ce problème est essentiellement celui de l'hétérogénéité entre différentes représentations des mêmes entités du monde réel. En effet, les sources d'informations sont nombreuses et diversifiées (bases de données relationnelles et objets, Pages Web, fichiers, etc.) ainsi que les modes de consultation (façon de *formuler* la requête, manière de *présenter* les résultats). Alors, l'hétérogénéité concerne les données, les modèles et les langages. L'hétérogénéité des sources de données est généralement classée en deux catégories :

Hétérogénéité sémantique:

Elle est due aux conflits sémantiques dans les termes, les expressions, etc, lorsque différents vocabulaires sont utilisés pour représenter les données, qui sont adoptées par différents schémas locaux. Autrement dit, elle est due aux différentes interprétations pour les objets du monde réel.

Hétérogénéité structurelle ou des schémas:

Elle provient quand les sources adoptent différents modèles de données (bases de données relationnelles ou orientées objets), structures de données ou schémas décrits dans un même modèle sont différents.

Contribution :

Notre objectif est de réaliser un système d'intégration sémantique des données structurées (bases de données relationnelles), en se basant sur la médiation (intégration virtuelle) via les ontologies pour résoudre les conflits sémantiques et structurels et ceci, tout en préservant l'autonomie des sources de données.

Pour traiter les deux problèmes d'hétérogénéité (structurelle et sémantique), nous proposons une approche de médiation dans laquelle l'ontologie à un rôle central. Cette approche consiste à utiliser deux catégories d'ontologies : linguistiques et formelles. Dans la première catégorie, nous utilisons Wordnet qui vise à définir le sens des mots et les relations entre ces mots. Pour la seconde catégorie d'ontologie, nous construisons une ontologie de domaine afin

de définir formellement les concepts d'un domaine et les relations entre ces
avons choisi le domaine biomédical et plus précisément les risques alimentaires

Il existe différentes utilisations possibles des ontologies dans un système d
plus récente est de représenter le schéma de chaque source locale par sa pr
Dans ce cas, le problème d'intégration sémantique devient un problème d'intégration d'ontologies. Alors, une mise en correspondance entre l'ontologie globale de médiateur et les ontologies locales, est nécessaire pour le fonctionnement du système d'intégration.

Par conséquent, et afin de réduire les problèmes d'hétérogénéités abordés, les principales contributions sont résumées dans les points suivants :

- Proposition d'une solution au problème d'intégration sémantique des sources hétérogènes.
- Proposition d'une méthode de traitement de requêtes complexes.
- Proposition des solutions pour le traitement des conflits sémantique, syntaxique et structurel.
- Proposition d'une méthode d'Alignement entre l'ontologie globale et les ontologies locales.
- Développement d'un système d'intégration sémantique dédié au domaine des risques alimentaire, nommé MS4AR.

Plan du manuscrit :

Ce manuscrit est constitué de deux parties principales. La première partie comporte deux chapitres sur l'état de l'art en rapport avec les thèmes qu'aborde notre travail. La seconde partie décrit notre contribution concernant l'intégration sémantique.

Première partie :

Nous allons présenter un état de l'art relatif à nos objectifs de recherche, où dans le premier chapitre **« Web sémantique et ontologies » :** nous présenterons les différents concepts essentiels sur le web sémantique et l'ontologie. Et le deuxième chapitre **« Système d'intégration de données hétérogènes » :** nous présenterons une description générale des systèmes d'intégration et un aperçu plus détaillé de plusieurs méthodes spécifiques qui sont utilisées pour l'intégration de données. Nous décrivons également les principaux problèmes d'hétérogénéité rencontrés.

Deuxième partie :

Dans la deuxième partie nous allons présenter en détail notre plateforme **« Système de médiation sémantique pour l'intégration de données hétérogènes »**, où dans le troisième chapitre **« Conception de système de médiation sémantique »** nous présenterons la conception de notre système de médiation sémantique à base d'ontologie. Et le quatrième chapitre **« Approche proposée pour l'intégration sémantique à base d'ontologie »** nous consacrons à la représentation de l'approche que nous proposons pour le traitement du problème d'hétérogénéité.

Finalement dans le dernier chapitre nous décrivons notre système d'intégration sémantique en se basant sur l'étude conceptuelle dans les deux chapitres précédents.

En conclusion, nous dressons nos travaux réalisés. Nous dégageons ensuite les perspectives envisageables pour ces travaux.

Mots-clefs: système d'intégration, données hétérogènes, web sémantique, ontologie, médiateur, approche GAV, similarité sémantique

après le résumé

Chapitre 1

« Web sémantique et ontologies »

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation

Tim - Berners-Lee

Introduction :

Le Web sémantique (WS), est une extension du Web, a comme objectif majeur d'apporter de la sémantique à l'information manipulée par l'utilisateur. Il a été représenté par Tim Berners-Lee. Le WS fournit aux machines un meilleur accès à l'information grâce à des métadonnées qui décrivent les informations disponibles. Ces métadonnées sont fournies par les ontologies. Ces dernières sont utilisées comme un élément de référence auquel nous nous intéressons dans nos travaux pour l'intégration des sources de données hétérogènes.

Dans ce chapitre, nous allons présenter les différents notions et concepts liés aux ontologies et leur ingénierie.

1. Le web sémantique :

1.1.Historique du Web sémantique :

Le document initial du projet « *Information Management: A Proposal* » où Tim Berners-Lee décrit en mars **1989** le système d'hypertexte distribué qui deviendra le web. Sa structure est un graphe étiqueté et orienté comme le montre la figure 1.1 issue de ce document de proposition de projet [1].

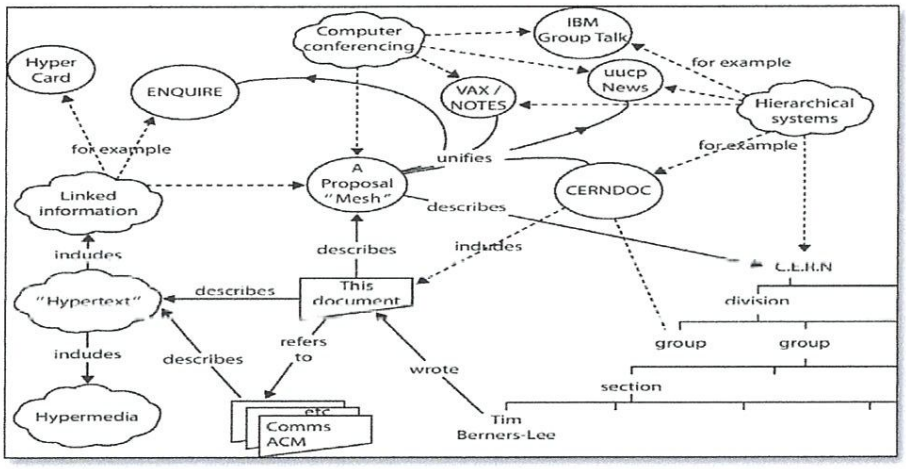


Figure 1.1 **:** Schéma explicatif du système d'hypertexte distribué qui deviendra le web (« Information Management: A Proposal », Tim Berners-Lee, mars 1989) [1] .

En 1991, le premier serveur web est installé hors d'Europe au *Stanford Linear Accelerator Center*. Le travail de portage et les débuts de la bibliothèque standard Libwww pour développer des clients web sous le langage de programmation C qui permettent notamment le développement du premier navigateur web textuel sur Sun/Unix et sur MS/DOS.

Début **1992**, on recense une dizaine de serveurs web et de nouveaux navigateurs apparaissent dans le courant de l'année (Erwise, ViolaWWW, MidasWWW, Samba pour Macintosh, etc.)[1].

En 1993, la technologie du web devient gratuite et de droits libre. Cette étape engendre la pénétration virale de ces technologies dans toutes les organisations. En début d'année, on dénombre une cinquantaine de serveurs. De nouveaux navigateurs apparaissent (Lynx, Cello, Arena) mais le plus important est Mosaic, alors disponible sous Unix, Windows et Mac OS. Il permet de visualiser les images directement dans le texte d'une page. Avec le navigateur Mosaic, le web va réellement se répandre mondialement, laissant derrière lui ses ancêtres Gopher, WAIS et FTP. À Mosaic succèderont, dans la suite généalogique, Netscape puis Mozilla et enfin FireFox [1].

En 1994, lors de la première conférence WWW à Genève, plus précisément au CERN, a lieu l'annonce de la création du W3C(Le World Wide Web Consortium). C'est d'ailleurs à cette période que Tim Berners-Lee dresse les objectifs du W3C et montre les besoins d'ajouter de la sémantique au Web futur [web1].

Après cette conférence, la publication d'un une première ébauche de recommandations sur le Web sémantique en octobre **1997** et d'une seconde en avril **1998**. Cette même année, Tim Berners-Lee publie un article sur les versions du web. Ces versions consistent à mettre en place les différentes technologies du Web sémantique [web1].

L'idée est de parvenir à un Web intelligent, où les informations ne seraient plus stockées mais comprises par les ordinateurs, pour apporter à l'utilisateur ce qu'il cherche vraiment. Le Web sémantique permettra donc de rendre le contenu sémantique du Web interprétable non seulement par l'homme, mais aussi par la machine [web1].

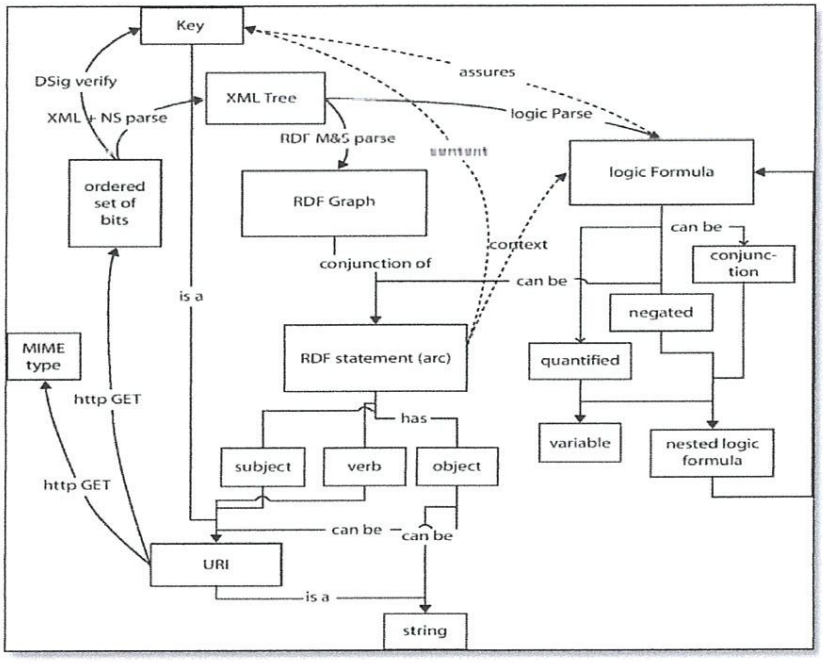


Figure 1.2 : *Schéma de synthèse issu de la feuille de route du web sémantique (« Semantic Web Roadmap », Tim Berners-Lee, 1998) [1].*

Par ailleurs, en **1999**, Tim Berners-Lee publie le livre «Weaving the Web » dans lequel il dresse un portrait du Web et les pistes pour son avenir.
« I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web the content, links, and transactions between people and computers. A "Semantic Web", which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize»[web1].

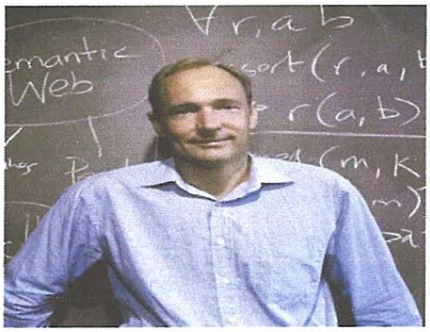


***Image 1**: Tim Berners Lee.*

1.2.Définition :

Web sémantique (WS), est une extension du Web qui a comme objectif majeur d'apporter de la sémantique à l'information manipulée par l'utilisateur. Il a été représenté par Tim Berners-Lee. Le WS fournit aux machines un meilleur accès à l'information grâce à des métadonnées qui décrivent les informations disponibles. Ces métadonnées sont fournies par les ontologies.

Le Web sémantique a pour but de rendre l'information disponible sur le Web plus visible, plus accessible. Comme le rappelait Tim Berners-Lee en 2007 dans le magazine La Recherche, "il existe un énorme gisement de données enfouies dans tous les ordinateurs du web, en les reliant, le Web sémantique permettra d'exploiter cette mine pour améliorer nos connaissances et éviter les incompréhensions suscitées par l'utilisation [3].

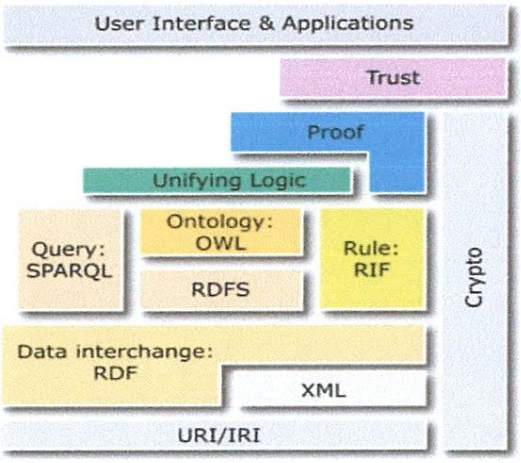


***Figure1.3**: Les briques technologiques du Web sémantique [1].*

Le résultat de ces travaux (qui sont toujours en cours) est constitué d'un ensemble de briques technologiques se reposant les unes sur les autres. Le schéma d'ensemble forme ainsi une sorte de pyramide, le fameux "Semantic Web Layer Cake".

Chaque technologie se base sur les couches inférieures et elle est utilisée par les couches supérieures et chaque nouvelle couche est plus expressive et plus riche que le substrat sur lequel elle repose.

-Couche XML : Base syntaxique.
-Couche RDF : Modèle de données de base RDF pour les faits, Langage ontologique simple RDF Schema.
-Couche ontologique : Langages plus expressifs que RDF Schema, Standard web actuel OWL
-Couche logique : améliore les langages ontologiques, connaissance déclarative propre à l'application.
-Couche de contrôle : génération de contrôle, échange, validation.
-Couche de sécurisation : signatures numériques, avis, agences de notation [16].

Nous allons les découper en 3 catégories :

- La catégorie des technologies qui existent indépendamment du Web Sémantique (comme Unicode, XML ou les URI). Il s'agit du "bas niveau".
- La catégorie des technologies liées au Web Sémantique qui existent et qui sont standardisées (RDF, RDFS, OWL, SPARQL).
- La catégorie des technologies qui sont en cours de développement. Il s'agit du "haut niveau" et nous ne nous étendrons pas sur cette partie. De toute façon, il y a déjà pas mal de chose à raconter sur les couches antérieures [1].

2. Développement du Web :

Depuis son existence jusqu'au nos jours le Web a connu beaucoup de développements on va les présenter dans ce qui suit :

2.1.Le web 1.0 :

Le Web 1.0 représente les sites de première génération, où le contenu est produit et hébergés par une entreprise, propriétaire du site. Ces sites sont les systèmes d'information du début de l'histoire de l'Internet.
Ils sont statiques, le contenu des pages est rarement mis à jour et l'utilisateur n'est qu'un lecteur de l'information. Une première révolution était réalisée par des solutions se basant sur un web dynamique, où des systèmes de gestion de contenu servaient des pages web dynamiques, créées à la volée à partir d'une base de données en constant changement. À ce stade, le web était considéré principalement comme un outil de diffusion et de visualisation de données [2].

2.2.Le web 2.0 :

C'est un terme qui définit une certaine évolution technique et sociale, là où les internautes n'étaient que des simples consommateurs et lecteurs des pages web, ils ont désormais la possibilité de commenter, noter, de partager, etc., le Web 2.0 devient de plus en plus l'épicentre du marché, le foyer de la création et de l'acquisition des connaissances et le principal milieu de la communication et de la vie sociale.

Le web 2.0 permettrait vraiment d'encourager la collaboration entre les utilisateurs, d'offrir des formes d'expression plus authentiques et d'aider les organisations à partager les connaissances [2].

Les limites de Web 2.0 :

Depuis quelque temps on a constaté que le Web est changé, et devenu plus ouvert à la participation de tout le monde, le contenu est devenu aussi riche que parfois inutile C'est le partage de tout et n'importe quoi qui pose les limites suivantes pour le Web 2.0 :

- Mauvaise qualité de l'information.
- Faible taux de participation.
- Durée de vie de l'information très courte.
- Manque de sécurité et droit d'auteur.
- Manque de sémantique [7].

2.3.Vers le Web 3.0 :

Après le passage d'un Web plutôt statique à un Web plus dynamique, on se prépare à un Web plus intelligent, un Web (dit sémantique) qui nous comprendra mieux et pourra mieux nous répondre. Le Web 3.0 est un concept en pleine évolution qui a fait son apparition dans les années 2008, il doit être mobile, indépendant de toute plateforme ou support, et que les pages qui composent le site doivent être gérées par une base de données relationnelle intelligente ou du moins ayant un minimum de travail d'ontologie en amont [web2].

3. Web sémantique, quoi de nouveau ?

Les gens toujours surfent sur le Web, achètent des choses sur des sites Web, etc. Il serait beaucoup plus efficace et moins fastidieux si les applications Web peuvent donner plus d'assistance aux utilisateurs, peut-être que les nouveautés du Web sémantique font un grand pas vers la résolution de ce problème. Ces nouveautés sont :

- Les données sont lisibles par la machine :
Les données définies et liées de manière qu'il puisse être utilisé par machines non seulement à des fins d'affichage, mais pour l'automatisation, l'intégration et la réutilisation des données entre différentes applications [10].

- Les agents intelligents :
Un agent intelligent est un personnel sur le web sémantique a pour but de recevoir certaines tâches et préférences d'une personne, de chercher les informations de sources web,

communiquer avec d'autres agents. De plus, il vise à comparer les informations selon des critères et préférences de l'utilisateur, effectuer certains choix et donner des réponses à l'utilisateur [16].

-Le serviteur de l'humanité :
La vision du Web sémantique est de permettre aux logiciels de nous soulager de beaucoup de charges pendant la localisation des ressources sur le Web qui sont pertinentes à nos besoins [12].

Le Web sémantique est une vision de l'Internet de la prochaine génération, qui permet aux applications Web à collecter automatiquement des documents Web provenant de sources diverses, d'intégrer et de traiter l'information et interagir avec d'autres applications afin d'exécuter les tâches complexes pour les humains [11].

-L'amélioration de la recherche :
Le but principal du Web sémantique et de rendre possible d'accéder aux ressources Web par le contenu plutôt que par mots-clés [11].

-Les services Web :
Le Web sémantique promet d'élargir les services pour le web existant en permettant à des agents logiciels d'automatiser les procédures actuellement exercées manuellement et en introduisant de nouvelles applications qui sont irréalisables aujourd'hui [13].

4. Ontologie: un élément clé du web sémantique:

4.1.Définitions :

Les ontologies représentent un composant pivot du web sémantique. Elles servent de vocabulaire standardisé pour le partage de connaissances. Le W3C soutient les activités liées au WS à travers le Web Ontology Working Group.
Plusieurs définitions du terme ontologie ont été proposées selon la philosophie et l'ingénierie des connaissances.

En philosophie, l'ontologie est l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe [4].

Dans le cadre de l'intelligence artificielle, Nooohooot ooo oollèguoo étaient leo premiero à proposer une définition:« Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire » [5].

En ingénierie des connaissances, la définition communément admise d'une ontologie est énoncée par **T. Gruber** En 1993, comme la « spécification explicite d'une conceptualisation» [4].

Cette définition a été modifiée légèrement par **Borst** comme « spécification formelle d'une conceptualisation partagée ».

Ces deux dernières définitions sont regroupées dans celle de **Studer** comme « *spécification formelle et explicite d'une conceptualisation partagée* ».

Formelle : l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel.

Explicite : la définition explicite des concepts utilisés et des contraintes de leurs utilisations.

Conceptualisation : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.

Partagée : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs [5].

-La même notion est également développée par **Gomez** comme : « une ontologie fournit les moyens de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances ».

-Welty et Nancy en 1999: une ontologie est la définition des classes, relations, contraintes et règles d'inférence qu'une base de connaissances peut utiliser [15].

-Guarino en 1998: une ontologie est un vocabulaire partagé, plus une spécification du sens de ce vocabulaire [15].

-Le W3C : une ontologie définit les termes utilisés pour décrire et représenter un domaine de la connaissance [15].

-Aussenac-Gilles et ses collèges en 2000 soulignent la dépendance entre la formalisation de l'ontologie et l'application dans laquelle elle va être utilisée : « Une ontologie organise dans un réseau des concepts représentant un domaine. Son contenu et son degré de formalisation sont choisis en fonction d'une application » [17].

-Jean et ses collèges 2007 qui caractérisent une ontologie comme une représentation formelle, référençable et consensuelle de l'ensemble des concepts partagés d'un domaine sous forme de classes, de propriétés et de relations qui les lient [17].

Référençable, c'est-à-dire que toute entité ou relation décrite dans l'ontologie peut être directement référencée par un symbole, à partir de n'importe quel contexte, afin d'expliciter la sémantique de l'élément référençant [17].

D'une manière générale, La définition de Grüber est la plus largement adoptée.

4.2.Les composants d'une ontologie :

Typiquement, une ontologie contient une description hiérarchique des concepts importants d'un domaine et décrit les liens sémantiques entre eux. Nous présentons dans ce qui suit les différents composants d'une ontologie :

Concept: Les concepts, aussi appelés termes ou classe de l'ontologie, notion exprimée en général par un terme ou par un symbole littéral ou autre. Il représente un ensemble d'objets, d'êtres, et leurs propriétés communes, par exemple : bactérie.

Terme : exprimant le concept en langage naturel à titre d'exemple : Voiture, Vache, Violon

Notion ou intension du concept : la signification du concept.

Les objets dénotés par le concept : appelés également « réalisations » ou « extensions » du concept.

Les classes : elles représentent le centre d'intérêt de l'ontologie et décrivent les concepts d'un domaine. Une classe peut avoir des sous-classes qui représentent des concepts plus spécifiques que la super classe (ou classe supérieure).

Instances : Une classe peut avoir des instances ou individus. Ces instances sont des entités réelles de cette classe, elles sont une représentation des extensions du concept. Exemple : salmonella est une instance de la classe bactéries.

Les attributs : Les attributs décrivent les propriétés des classes et des instances de l'ontologie Exemple : Age, nom, prénom, etc [4].

Les relations : désignent les associations prédéfinies qui existent entre les concepts de l'ontologie :

•**Relation binaire** entre un concept général et un concept plus spécifique.

•**Une relation de type « est-un »:** une classe A est une sous-classe de B si chaque instance de A est également une instance de B. Par exemple, femme est une sous-classe de Personne.
Une autre manière de voir la relation taxonomique est de la voir comme relation du genre «une-sorte-de» : femme est une sorte de personne, La viande est une sorte d'aliment.

•**Une relation de sous-classe est transitive** : Si B est une sous-classe de A et C est une sous-classe de B, alors C est une sous-classe de A.

•**Relation d'Equivalence** : une relation sémantique qui existe entre deux concepts jouant le même rôle dans des RC (représentations conceptuelles) différentes. Exemple : Equivalence (Usine, Fournisseur). L'équivalence vérifie la propriété suivante: La relation Equivalence est une relation symétrique : SI Equivalence (C1, C2) ALORS Equivalence (C2, C1) [18].

Les axiomes : désignent les assertions acceptées comme vraies dans le domaine étudié. Les axiomes et les règles permettent de vérifier la cohérence d'une ontologie, et aussi d'inférer de nouvelles connaissances [4].

Exemple d'ontologie:
L'ontologie de la figure 1.4 présente un petit exemple d'ontologie sur l'arbre généalogique. Cette ontologie contient un ensemble de concepts (gender, person, daughter, father, mother,offspring, parent, son) et un ensemble de relation, comme 'Has child' entre 'parent'

et 'son', et d'attributs, comme 'Has_Name', des instances, comme 'Tom' et enfin des types de données, comme String.

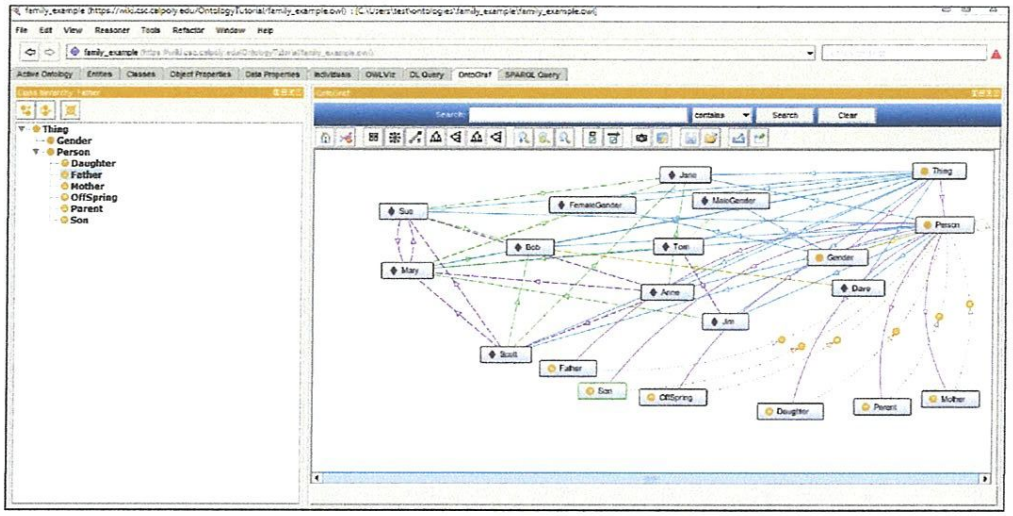


***Figure1.4**: exemple d'une familiale ontologie.*

4.3.Thésaurus :

Selon la norme internationale ISO 2788, Le mot thésaurus a été employé en documentation à cause du Thesaurus de Peter Mark Rodget en 1852, qui était un dictionnaire anglais de synonymes et de notions connexes, organisé systématiquement [19].

Un thésaurus est l'outil le plus performant pour structurer systématiquement l'information car il garantit que la même terminologie est utilisée de manière cohérente au travers d'un système de recherche d'information. De ce fait, en RI, il existe trois points fondamentaux liés au thesaurus [20]:

- *La construction*: Il existe deux types de thésaurus, manuel et automatique.
- *L'accès:* Étant donné une requête particulière, le thésaurus doit être accessible et utilisable pour améliorer ou étendre la requête.
- *L'évaluation :* Après la construction d'un thésaurus, il est important de savoir comment il est efficace. Les thésaurus Manuels sont évalués en termes de la solidité, la couverture de classification et la sélection des éléments du thésaurus. L'évaluation des thésaurus automatique se fait généralement grâce à l'expansion de requête pour voir si les performances de recherche sont améliorées.

Le thésaurus en ligne du BDSP (Banque de Données de la Santé Publique) illustre très bien les nombreux avantages liés à cette nouvelle utilisation des thésaurus. Il propose deux moyens d'accès aux documents, le premier consiste à se déplacer dans la représentation graphique du thésaurus, en exploitant les liens dynamiques reliant les rubriques entre elles, pour sélectionner le concept qui permettra d'accéder aux documents pertinents (voir la figure 1.5). Le second est un moteur de recherche classique, où l'utilisateur saisit le terme qu'il recherche

dans un champ. Il peut être guidé pendant sa démarche par une liste d'autorité qui reprend uniquement les descripteurs du thésaurus (voir la figure 1.6).

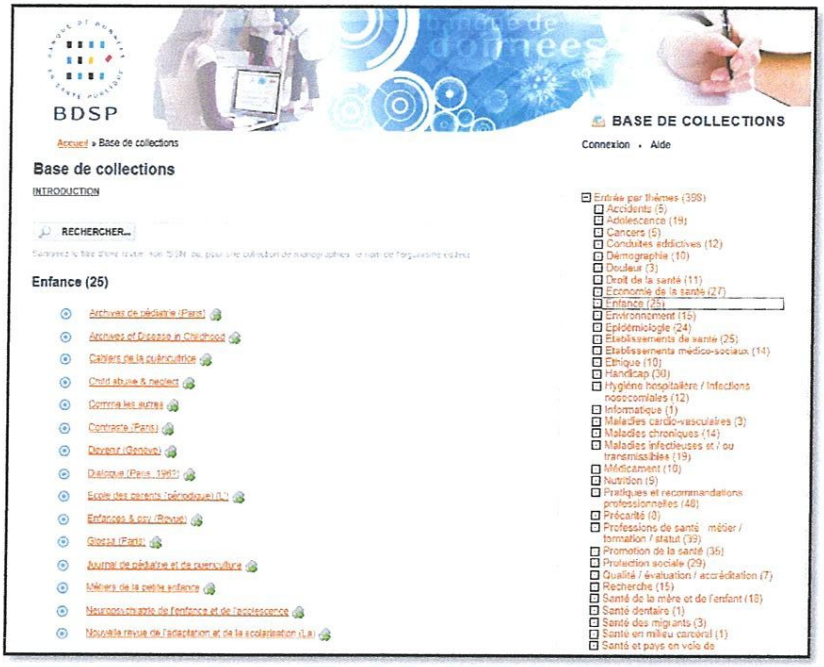


Figure 1.5 : *Interface graphique de Thésaurus médical « BDSP » [20].*

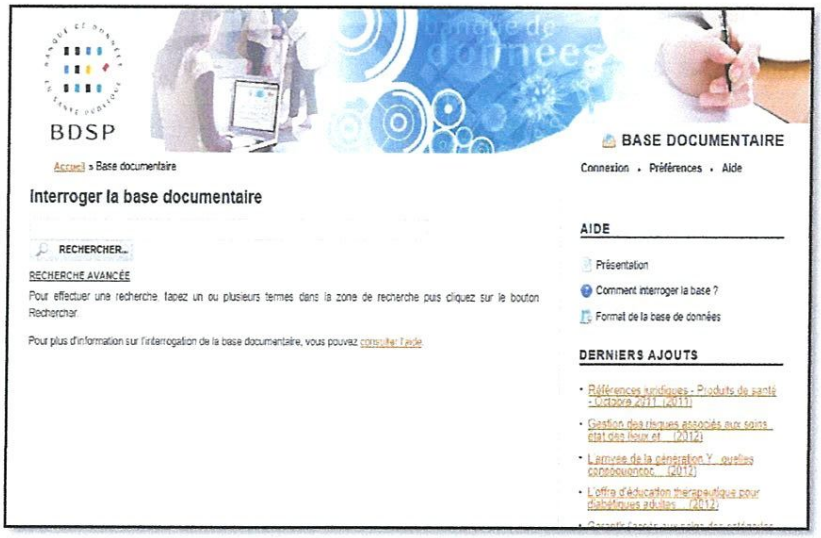


Figure 1.6 : *Moteur de recherche du Thésaurus médical « BDSP » [20].*

4.3.1. Différence entre un thésaurus et une ontologie :

Les ontologies, les thesaurus, si ils partagent l'organisation hiérarchique de concepts entre eux, n'ont pas les mêmes usages, ni les mêmes objectifs. L'ontologie est faite pour décrire le monde tel qu'il est; le thesaurus est fait pour faciliter l'accès à des contenus. Les systèmes d'accès aux contenus, aux données ou aux connaissances, combinent et articulent ces deux systèmes d'organisation pour décrire le monde et indexer les contenus [web3].

	ontologie	Thésaurus
Composants	Des classes, des propriétés, et des règles logiques formelles. Eventuellement des instances de classe.	Des concepts et des termes, organisés entre eux, avec leurs libellés, leurs traductions, leurs synonymes, et leurs descriptions/définitions.
Utilisation	Sert à instancier et à raisonner	Sert à indexer des contenus ou des ressources avec des mots-clés et à les rechercher (avec les mêmes mots-clés)
Niveau de formalisme logique	Très formel (formalisme mathématique)	Peu formel
Niveau de proximité avec la langue naturelle	Très éloigné de la langue naturelle (utilise des identifiants techniques pour s'abstraire du langage naturel)	Proche de la langue naturelle (donne des équivalents linguistiques de chaque entrée, des traductions dans d'autres langues)
Types de relations utilisées	-Inclusion (classe / sous-classe); -Opérations ensemblistes : union, intersection, exclusion; -Caractéristiques des propriétés : domaine, ensemble d'arrivée, transitivité, propriétés inverses, etc.	Hiérarchiques et associatives; éventuellement relations d'alignement
Standard de représentation sur le web de données	OWL	SKOS
Un exemple	L'ontologie FOAF définit des classes et des propriétés pour décrire des personnes	Le thesaurus GEMET(General Multilingual Environmental Thesaurus) est un thesaurus multilingue (+ de 30 langues) développé par l'agence européenne de l'énergie et disponible gratuitement.
A utiliser si	vous avez besoin de décrire les choses telles qu'elles sont, et pas simplement d'indexer des contenus, et que vous avez besoin de décrire précisément les caractéristiques de chaque chose.	vous voulez mettre des mots-clés sur des contenus, et pouvoir rechercher avec ces mêmes mots-clés.Vous pouvez vouloir utiliser un thesaurus en combinaison avec un moteur de recherche plein-texte pour améliorer sa pertinence.

Tableau 1.1 : *Différence entre ontologie et thésaurus [web3].*

5. Construction et types d'ontologies:

La construction d'une ontologie n'est pas une activité facile, d'autant plus qu'il n'existe pas aujourd'hui une méthodologie précise, comme c'est le cas dans le domaine des bases de données. Le processus de construction intègre souvent un expert du domaine. Une fois construite, l'ontologie doit être claire, cohérente, compréhensible, facile à utiliser et extensible.

5.1. Conception d'une ontologie :

La conception d'ontologies est une tâche difficile nécessitant la mise en place de procédés élaborés afin d'extraire la connaissance d'un domaine, manipulable par les systèmes informatiques et interprétable par les êtres humains.

Deux types de conception existent : la conception entièrement manuelle et la conception reposant sur des apprentissages.

-la construction manuelle : Plusieurs principes et méthodologies ont été définis pour faciliter la construction manuelle. Ces principes se basent sur des fondements philosophiques et suivent des procédés de modélisation collaboratifs. Ils mènent à la conception d'ontologies dites légères et d'ontologies dites lourdes (ces ontologies se distinguent par la présence ou non d'axiomes). Cependant, ce procédé de génération est très coûteux en temps et pose surtout des problèmes de maintenance et de mise à jour [9].

-La conception automatique d'ontologies commence à émerger comme un sous-domaine de l'ingénierie des connaissances. Face à la masse croissante de documents présents sur le Web et aux avancées technologiques dans le domaine de la recherche d'information, de l'apprentissage automatique et du traitement automatique des langues, de nouveaux travaux portent sur la recherche de procédés plus automatiques de génération d'ontologies. Ces mécanismes mènent généralement à la conception d'ontologies dites légères [9].

-La réingénierie d'ontologies est le processus qui consiste à reconstruire et lier un modèle conceptuel d'une ontologie déjà implémentée à une autre en cours d'implémentation. Une méthode représentative de cette approche est celle de Gomez-Perez et Roza en 1999, qui ont adapté une technique de réingénierie de schémas à une ontologie de domaine [15].

-Conception Mixte : construite par des techniques automatiques mais elles permettent d'étendre les ontologies ayant été construites manuellement [9].

5.2.Méthodologie de construction :

Méthodologie de construction le Processus en V permet de construire une ontologie selon les étapes suivantes. «The reality is that the construction of ontologies is an art rather than a science (Fernandez, METHONTOLOGY) »:

1. **Spécification des besoins** : identifier le domaine et le but de l'ontologie.

2. **Acquisition des connaissances** : par l'expert de domaine, textes d'articles (text mining), méta-données de bases de données etc. dresser une liste de questions de compétences.
3. **Conceptualisation** : identifier les concepts-clés du domaine, leurs propriétés et leurs relations; identifier les termes du langage naturel; structurer le savoir du domaine.
4. **Intégration** : utiliser ou spécialiser une ontologie existante.
5. **Encodage** : choisir un langage de représentation formel.
6. **Documentation** : produire des définitions formelles, informelles, complètes, pour préciser la signification des termes de l'ontologie; donner des exemples.
7. **Evaluation**: déterminer l'adéquation de l'ontologie pour l'application visée; évaluation à faire de façon pragmatique [5].

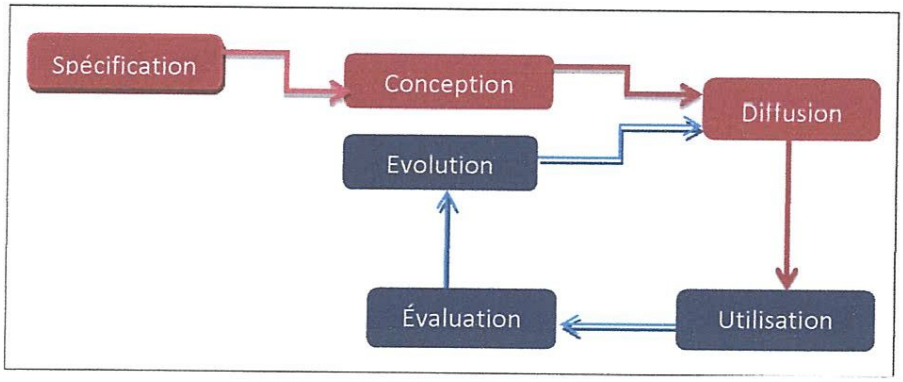


Figure1.7 : *Cycle de vie d'une ontologie* [5].

D'une manière sommaire, les travaux sur la construction des ontologies sont classés en trois :

1. les méthodes et méthodologies pour la construction d'ontologies en partant de zéro ;
2. les méthodes pour la réingénierie d'ontologies ;
3. les méthodes de construction coopérative d'ontologies.

Par ailleurs, il existe maintenant de nombreuses ontologies en ligne qu'il est possible de réutiliser [1].

5.3.Types d'ontologies :

Van Heijst et al en 1997 définissent deux grandes typologies d'ontologies : (i) une typologie Fondée sur la structure de la conceptualisation et (ii) une typologie fondée sur le sujet de la conceptualisation.

Dans la première typologie, ils distinguent trois catégories à savoir
- les ontologies terminologiques (lexiques, glossaires…).
- les ontologies d'information (schéma d'une BD).

- les ontologies des modèles de connaissances.

Dans la deuxième typologie, qui est la plus citée, ils distinguent quatre catégories :

- **les ontologies d'application** : elles contiennent toutes les informations nécessaires pour modéliser les connaissances pour une application particulière.

- **les ontologies de domaine** : elles fournissent un ensemble de concepts et de relations décrivant les connaissances d'un domaine spécifique.

- **les ontologies génériques** (dites aussi de haut niveau) : elles sont similaires aux ontologies de domaine, mais les concepts qui y sont définis sont plus génériques et décrivent des connaissances tels que l'état, l'action, l'espace et les composants. Généralement, les concepts d'une ontologie de domaine sont des spécialisations des concepts d'une ontologie de haut niveau.

- **Les ontologies de représentation** (dites aussi méta-ontologies) : elles fournissent des primitives de formalisation pour la représentation des connaissances. Elles sont généralement utilisées pour écrire les ontologies de domaine et les ontologies de haut niveau. Exemples : Frame Ontology (Gruber) et RDF Schema Ontology (McBride, 2004)[17].

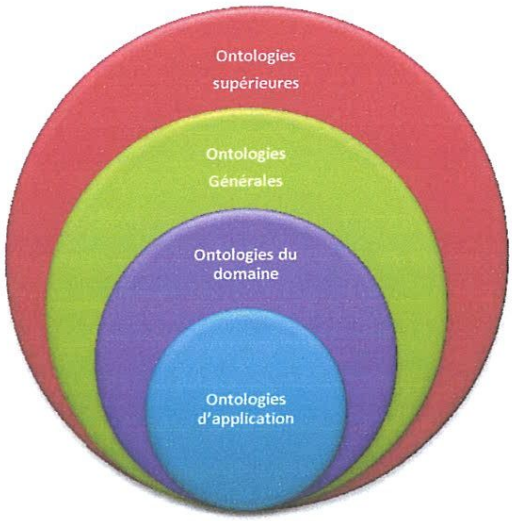


Figure 1.8 : *Type d'ontologies [18].*

5.4.Rôles d'ontologies :

L'intégration d'une ontologie dans un système d'information vise à réduire, voir éliminer, la confusion conceptuelle et terminologique à des points clefs du système, et à tendre vers une compréhension partagée pour améliorer la communication, le partage, l'interopérabilité et le degré de réutilisation possible, ce qui permet de déclarer formellement un certain nombre de connaissances utilisées pour caractériser les informations gérées par le système, et de se baser sur ces caractérisations et la formalisation de leur signification pour automatiser des taches de traitement de l'information

L'ontologie se trouve maintenant dans une large famille de systèmes d'information. Elle Est utilisée pour :

- Décrire et traiter des ressources multimédia.
- Assurer l'interopérabilité d'applications en réseaux.
- Piloter des traitements automatiques de la langue naturelle.
- Construire des solutions multilingues et interculturelles.
- Permettre l'intégration des ressources hétérogènes d'information.
- Vérifier la cohérence de modèles.
- Permettre les raisonnements temporel et spatial.
- Faire des approximations logiques ; etc.

Ces utilisations des ontologies se retrouvent dans de nombreux domaines d'applications tel que :

- Intégration d'information géographique ;
- Gestion de ressource humaine ;
- Aide à l'analyse en biologie, suivi médicale informatisé ;
- Commerce électronique ;
- Enseignement assisté par ordinateur ;
- Bibliothèque numérique;
- Recherche d'informations [5].

5.5.Outils d'édition d'ontologies :

Les outils d'édition constituent un aide pour l'implémentation d'une ontologie dans laquelle les principaux choix ont déjà été faits.

5.5.1. Protégé 2000 :

Est une interface modulaire permettant l'édition, la visualisation, le contrôle d'ontologie, l'extraction d'ontologies à partir de sources textuelles, et la fusion semi-automatique d'ontologies. Le modèle de connaissances sous-jacent de protégé 2000 est issu du modèle des frames et contient des classes, des slots (propriétés) et des faces (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. Il autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de construire une ontologie [5].

L'environnement d'édition offre deux possibilités de construction d'ontologies :

1. avec l'éditeur à base de frames, il permet la construction et l'instanciation d'ontologies basées sur le modèle des frames, en conformité avec le protocole OKBC; dans ce modèle, une ontologie est composée d'un ensemble de classes, organisé sous la forme d'une hiérarchie de subsumption ; d'un ensemble de slots associés aux classes, décrivant leurs propriétés et relations ; et enfin d'un ensemble d'instances des classes [15].

2. avec l'éditeur OWL, il permet la construction d'ontologies pour le Web Sémantique et particulièrement la construction d'ontologies OWL. Il offre avec cet éditeur tous les outils nécessaires pour l'édition des différents éléments d'une ontologie OWL (concepts, propriétés, instances), avec la possibilité de spécifier des contraintes et d'utiliser des moteurs d'inférence externes tels que Pellet pour vérifier la consistance de l'ontologie et d'inférer de nouvelles connaissances [15].

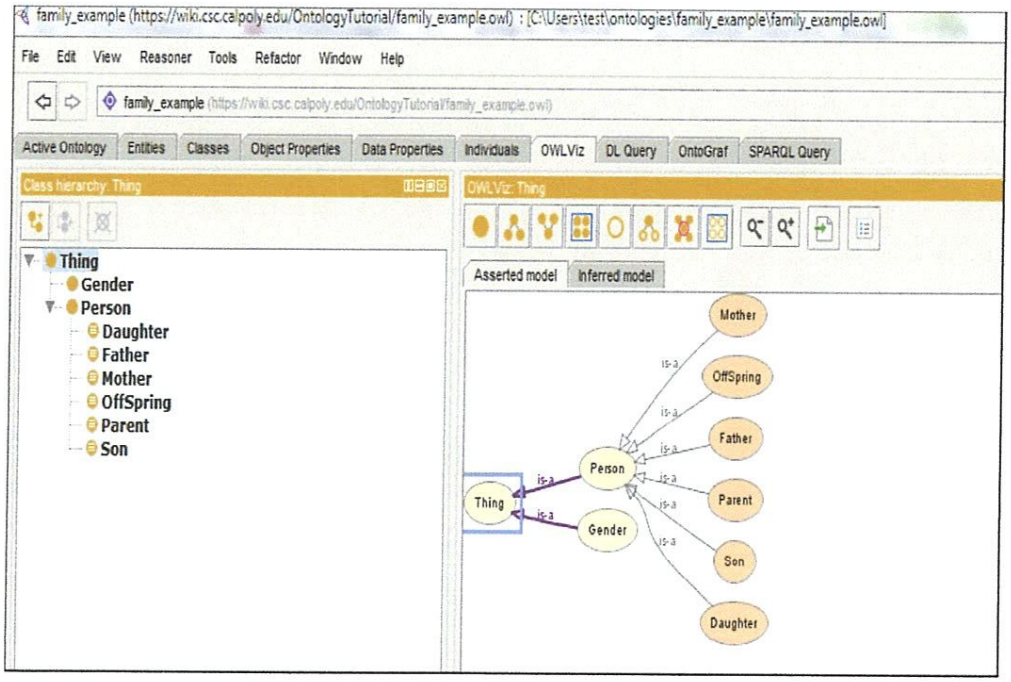


Figure1.9 : *Interface du protège 2000.*

5.5.2. ODE et WebOde :

L'outil ODE (Ontology Design Environment) permet de construire des ontologies au niveau connaissance, comme le préconise la méthodologie METHONTOLOGY. L'utilisateur construit son ontologie dans un modèle de type frame, en spécifiant les concepts du domaine, les termes associés, les attributs et leurs valeurs, les relations de subsomption [5].

5.5.3. OntoEdit :

OntoEdit (Ontology Editor) est également un environnement de construction d'ontologies indépendant de tout formalisme. Il permet l'édition des hiérarchies de concepts et de relations et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la généricité d'un concept. Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement. OntoEdit intègre un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs. Un contrôle de la cohérence de l'ontologie est assuré à travers la gestion des ordres d'édition [5].

6. Langages d'ontologies :

6.1.RDF (Ressource Description Framework) :

RDF (Resource Description Framework) est une recommandation du W3C développée pour décrire les ressources du web. Pour ce faire, RDF procède par une description de savoirs (données ou métadonnées) à l'aide d'expressions de structure fixée. En effet, la structure fondamentale de toute expression en RDF est une collection de triplets, chacun composé d'un sujet, un prédicat et un objet (ou {ressource, propriété, valeur}). Un ensemble de tels triplets est appelé un graphe RDF. Ceci peut être illustré par un diagramme composé de nœuds et d'arcs orientés, dans lequel chaque triplet est représenté par un lien nœud-arc-nœud (d'où le terme de "graphe").

A ce modèle est associée une syntaxe écrite en XML et basé sur les triplets :

- Ressource (Sujet) : une entité d'informations pouvant être référencée par un identificateur. Cet identificateur doit être une URI.

- Propriété (prédicat) : l'attribut ou la relation utilisée (e) pour décrire une ressource.

- Valeur (objet) : la valeur d'une propriété associée à une ressource spécifique [17].

Par exemple : En utilisant ce modèle, il est possible de modéliser le fait que 'le livre identifié par l'ISBN 2- 10-006300-6 est écrit par l'équipe ACACIA' comme suit :

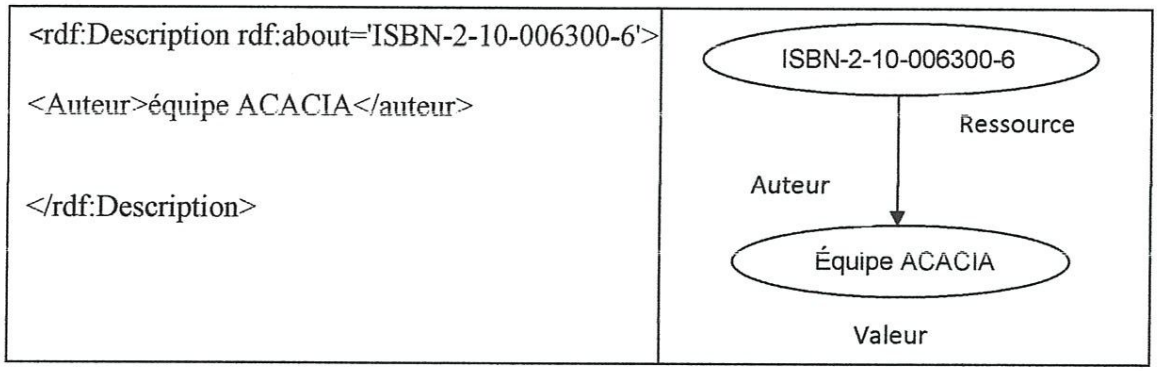


Figure1.10 : Modélisation des ressources avec un graphe RDF [17].

6.2.RDFS (Ressource Description Framework Schema):

RDFS (Resource Description Framework Schema) ajoute à RDF la possibilité de définir des hiérarchies de classes en utilisant des relations de subsomption entre classes (subClassOf) et des hiérarchies de propriétés en utilisant des relations de subsomption entre propriétés (subPropertyOf). De plus, RDFS offre le moyen de spécifier le typage des propriétés en indiquant leurs domaines par rdfs : domaine et leur co-domaine via rdfs : range.

RDFS est indiqué pour la description de ressources, cependant il présente assez rapidement des limites lorsqu'il s'agit de l'utilisation comme langage de représentation d'ontologies

ayant de fortes contraintes telles que : l'expression de la combinaison booléenne de classes ou de la disjonction de classes,...etc. [8].

6.3.OWL (Web Ontology Language):

Le développement rapide des applications basées sur les ontologies et la nécessité de modéliser des connaissances de plus en plus complexes, ont fait émerger quelques limites de RDF/RDFS, RDFS offre un vocabulaire simple, limité à une hiérarchie de classes, une hiérarchie de relations et des définitions des domaines d'application (« Domain » et « range»).

OWL a été recommandé par le W3C (en particulier par le groupe WebOnt) afin d'enrichir RDFS en définissant un vocabulaire plus complet pour la description d'ontologies complexes. Cette richesse par rapport à RDFS se matérialise par l'ajout de nouvelles notions telles que : l'équivalence des classes, l'équivalence des relations, la symétrie et la transitivité des relations, la cardinalité, etc.
Ce nouveau langage est divisé en trois sous-langages définis par une syntaxe expressive avec une sémantique formelle et rigoureuse [17].

-OWL Lite : c'est la version légère de OWL qui reprend RDFS et l'enrichit avec de nouvelles primitives.

-OWL DL : contient toutes les primitives de OWL (y compris OWL Lite) avec des contraintes particulières sur leur utilisation qui assurent la décidabilité du langage.

- OWL Full : plus flexible que OWL DL ce qui le rend vraisemblablement indécidable[15] .
OWL est basé essentiellement sur le formalisme des logiques de descriptions et tire profit des inférences et des mécanismes de raisonnements associés à ces formalismes. Pour ses trois sous langages, seuls les deux premiers maintiennent les tâches d'inférence principales à savoir la satisfiabilité et la classification [1]

6.4.SPRQL (Protocol and RDF Query Language):

SPRQL (***Protocol and RDF Query Language)*** fournit le langage d'interrogation du web, il a été conçu par le W3C et la recommandation date de janvier 2008.
Il y a plusieurs formes de requête SPRQL, mais la forme la plus classique est celle du Select where. L'entête préfix permet de déclarer les préfixes et les namespaces utilisés, il peut être absent, la partie select permet de préciser les variables que l'on désire voir figurer dans le résultat. La partie where permet de décrire le corps de requête, c'est-à-dire ce que l'on recherche dans le graphe cible.

Une requête peut comporter plusieurs triplets séparer par des «. » il s'agit alors d'une conjonction (et) entre ces triplets.

Les variables sont des éléments dont on recherche la valeur dans le graphe cible. Elles ont un nom préfixé par « ? » ou bien « $ » [1].

Les autres formes de requête sont **construct, describe et ask.**

-Requête construct retourne un résultat qui se présente sous la forme d'un graphe RDF construire pour l'occasion. La clause construct décrit le patron de graphe à construire en remplaçant les variables par les valeurs trouvées dans les solutions. A chaque solution trouvée pour la clause where, le motif de graphe donné par la clause construct est ajouté au résultat.
Le graphe résultat retourné contenant toutes les solutions trouvées, peut ensuite être délivré au format RDF/XML [1].

-Une requête de la forme describe retourne une description de ressource avec un contenu approprié, dépendant de l'implémentation et de l'application, le résultat se présente sous forme d'un graphe rdf, que peut ensuite être délivré au format RDF/XML.
Enfin la forme ask permet de poser une requête dont la réponse est une valeur booléenne, true s'il existe au moins une réponse et false sinon.
Le format de résultat est normalisé et se présente soit en RDF (requête describe construct) soit en XML (select, ask) [1].

-Modifier une base en SPRQL : SPRQL introduit également un langage de type CRUD (create, read, update, delete) pour la gestion des bases de graphe RDF.

-L'instruction clear permet de vider le contenu d'un graphe.
-L'opération create crée un nouveau graphe dans la base et accepter les graphes vide.
-L'opération drop supprime un graphe et l'ensemble de son contenu.
-l'opération copy modifie un graphe pour qu'il contienne une copier de l'autre.
-l'opération move déplacer toutes les données d'un graphe dans un autre.
-l'opération add reproduit et ajoute tous les données d'un graphe a un autre graphe.
-La suppression et ajout de données dans les graphes : il est possible de retirer (delete data)et d'ajouter (insert data) des données dans le graphe ,il est possible d'ajouter et de retirer des arcs sélectionnés par une requête where [1].

Alors que RDFS et OWL permettent de définir des ontologies sur le Web sémantique et RDF de modéliser des assertions en se basant sur celles-ci, il est nécessaire pour en tirer parti de disposer d'un langage de requête adapté [8].

7. Domaines d'application du web sémantique:

Depuis le début de développement du Web sémantique, les ontologies sont devenu très utiliser dans diverses domaines informatique.
-Recherche d'informations : Le Web sémantique cherche à atteindre une certaine maîtrise des contenus, afin de fournir des réponses pertinentes aux utilisateurs.

-L'adaptation et personnalisation :À travers l'internet, un nombre potentiellement infini de services et de documents est accessible à tous les utilisateurs. La plupart des services et documents fournis actuellement sur Internet proposent une organisation, un contenu, une mode d'interaction et une présentation uniques pour tous, ceci peut être suffisant dans certains cas, mais tous les utilisateurs ne sont pas intéressés par les mêmes informations et n'ont pas les mêmes attentes, connaissances, compétences, centres d'intérêt, etc.

-Intégration des sources de données hétérogènes : Le but d'intégration des so données hétérogènes est d'offrir aux utilisateurs une vision homogène du système. sémantique peut intégrer les sources hétérogènes, tout en ajoutant une ontologie qui va fournir un vocabulaire commun, et qui sera utilisé pour interroger le système [37].

Conclusion :

Dans ce chapitre nous avons présenté les notions principales auxquelles nous faisons appel, comme support, pour la modélisation de nos propositions. Il s'agit de la notion de l'ontologie et celle de la sémantique.

Les ontologies sont utilisées dans de nombreux domaines pour apporter la sémantique à l'information manipulée par l'utilisateur. Ces dernières années ont vu l'apparition de plusieurs travaux de rapprochement entre bases de données et ontologies visant soit à faciliter la conception des bases de données, soit à faciliter leur intégration dans des environnements ouverts tels que le Web Sémantique.

Dans le chapitre suivant on va présenter l'importance du l'ontologie dans le domaine de l'intégration, afin d'expliciter le contenu des sources et fournir un vocabulaire partagé.

Chapitre 2

« Système d'intégration de données hétérogènes »

Mediator is the architecture of future information systems

Gio Wiederhold

Introduction :

La diversité des sources d'information distribuées et leur hétérogénéité (SGBD-R ou O ou OR, Pages Web, Données semi-structurées, etc.) est une des principales difficultés rencontrées par les utilisateurs du Web aujourd'hui.

L'infrastructure du Web sémantique doit permettre leur intégration donnant ainsi l'impression à l'utilisateur qu'il utilise un système homogène. Les solutions à l'intégration d'information proposées dans le cadre du Web sémantique tireront parti des recherches concernant les approches médiateurs et les entrepôts de données.

L'intégration des données issues de sources hétérogènes et distribuées est devenue un besoin crucial pour un nombre important d'applications, comme l'ingénierie, le commerce électronique, l'intelligence économique, la bio-informatiques, etc. Cette intégration permet à ces applications d'exploiter et analyser cette mine d'informations.

Avec l'explosion du nombre de sources de données, des solutions d'intégration automatiques sont nécessaires. Ces solutions sont confrontées aux problèmes liés aux hétérogénéités structurelle et sémantique des sources.

Dans ce chapitre nous décrivons dans un premier temps, les problèmes que pose la diversité de plusieurs sources de données et dans un second temps, la solution de ces problèmes qui vise à intégrer ces sources. Nous allons présenter également les différents systèmes d'intégration existants dans la littérature en se focalisant sur l'importance des ontologies dans le processus d'intégration.

1. Problématique de l'intégration de données:

Du fait du Développement important de l'internet, la recherche d'information issue des sources de données réparties sur le réseau devient de plus en plus difficile, en effet, grâce à la révolution de nouvelles technologies de l'information, ces données sont stockées dans des sources hétérogènes et autonomes.
Chaque source est décrite par sa localisation, type de donnés qu'elle gère, ces possibilités de l'interrogation et le format des résultats.

- La localisation d'une source englobe toute la référence du site sur lequel se situe (URL, adresse IP+ PORT), ainsi que le Protocol de communication utilisé (TCP, UDP), les moyennes d'accès à la base (ODBC, JDBC).
- Le type de données géré par une source peut être structuré (base de données relationnel), semi structurés (source XML), ou nom structuré (image, texte).
- Les possibilités d'interrogation définissent les langages de requête évolué et standardisés (SQL, OQL).
- Enfin le format des résultats (XML, HTML, relationnel) [31].

L'intégration de données hétérogènes a conduit à de nombreux problèmes qui peuvent se classer en deux catégories : problème d'hétérogénéité de données et celui de l'évaluation des requêtes.

1.1.Hétérogénéité des données :

La question de l'hétérogénéité de données a longtemps été étudiée dans la communauté des bases de données. En général, le problème de l'hétérogénéité peut concerner deux catégories : hétérogénéité structurelle qui due à l'utilisation de modèles différents dans la modélisation des sources de données et l'hétérogénéité sémantique résulte des conceptions et différentes interprétations de même données.

1.1.1. Hétérogénéité structurelle ou schématique:

Elle provient quand les sources adoptent différents modèles de données (bases de données relationnelles ou orientées) et diverse structures de données, tels que les noms des relations et le nombre des attributs, des types de données et le degré de décomposition des attributs, diffèrent d'une source à une autre [22].

Par exemple (l'adresse peut être représentée sur un seule champ, ou plusieurs « rue, code postal, ville) [31].

1.1.2. Hétérogénéité sémantique :

Elle est due aux conflits sémantiques dans les termes, les expressions, etc., qui sont adoptés par différents schémas de données mais exprimés de diverses manières. Autrement dit, elle est due aux différentes interprétations pour les objets du monde réel. Elle apparaît lorsque différents vocabulaires et référentiels sont utilisés pour représenter les données, lorsque certains attributs ne sont pas renseignés, ou représenter par des abréviations, des concaténations de plusieurs noms d'attributs, des synonymies et des homonymes [30].

En effet, les sources de données ont été conçues indépendamment par des concepteurs différents ayant des objectifs applicatifs différents. Chacun peut donc avoir un point de vue différent sur le même concept. Ce désaccord sur la signification des données ne peut donc pas être résolu simplement, c'est pour cela que rendre explicite la sémantique des données intégrées est essentiel pour avoir une intégration sémantiquement correcte des données.

1.2.Evaluation de requêtes :

Le traitement de requêtes est un mécanisme absolument obligatoire dans l'intégration des sources de données et plus précisément les bases de données. Il commence par une formulation de requête à partir du schéma global aux schémas à l'exportation des sources de données. Les algorithmes pour la formulation des requêtes dépendent de la manière dont la relation entre le schéma global et les schémas locaux a été définie comme GAV ou LAV (on verra plus tard ces approches)[22].

Quand une requête est formulée au médiateur, elle est posée indépendamment de la localisation des différentes données intervenant pour calculer le résultat. Cela introduit les difficultés suivantes :

• **La décomposition d'une requête :** il s'agit à partir d'une requête posée sur une vue intégrée, de localiser les données intervenant dans sa résolution, de produire des sous requêtes spécifiques à chacune des sources, de les ordonner et éventuellement d'introduire des opérateurs au niveau du composant de médiation afin de compléter cet ensemble de sous-requêtes. La localisation des sous-requêtes nécessite des structures spécifiques de gestion de méta-données.

• **La recomposition des résultats :** une fois les sous-requêtes soumises à chacune des sources, il s'agit de savoir recomposer les différents résultats entre eux.

• **Au niveau de l'optimisation :** le médiateur a rarement une vision sur la façon dont sont traitées les sous-requêtes au niveau des sources (placement des données, type de stockage, indexation, stratégie d'évaluation). De plus la distribution des données sur des sources [22].

2. Les Systèmes d'intégration de données:

Le problème de combiner des sources de données hétérogènes et de les interroger via une seule interface de requête ne date pas d'aujourd'hui. Depuis l'arrivée et l'adoption des bases de données dans les années soixante, leur partage et leur combinaison sont naturellement devenus indispensables. Cette combinaison peut être effectuée de différentes façons et à différents niveaux de l'architecture du système.

2.1.Classification des systèmes d'intégration:

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 : approche de médiateur ou intégration virtuelle qui préconise la définition d'un schéma global virtuel sur lequel sont posées les requêtes et l'approche d'entrepôt de données dont l'intégration est matérialisée qui préconise la centralisation de toutes les données des sources au niveau d'une localisation unique.

2.1.1. L'intégration virtuelle de données:

Dans ce type d'intégration, l'accès aux sources hétérogènes est basé sur schéma global virtuel et les données restent stockées dans les sources d'origine. L'architecture type pour l'intégration virtuelle de données est l'architecture de médiateur. L'intégration de données est fondée sur la définition d'un schéma global unifiant les schémas hétérogènes des sources à intégrer et sur la description homogène et abstraite du contenu des sources par des vues.

L'architecture de médiateur est représentée par trois couches, comme nous le montrons en Figure 1 : *médiateur, adaptateurs et sources.*

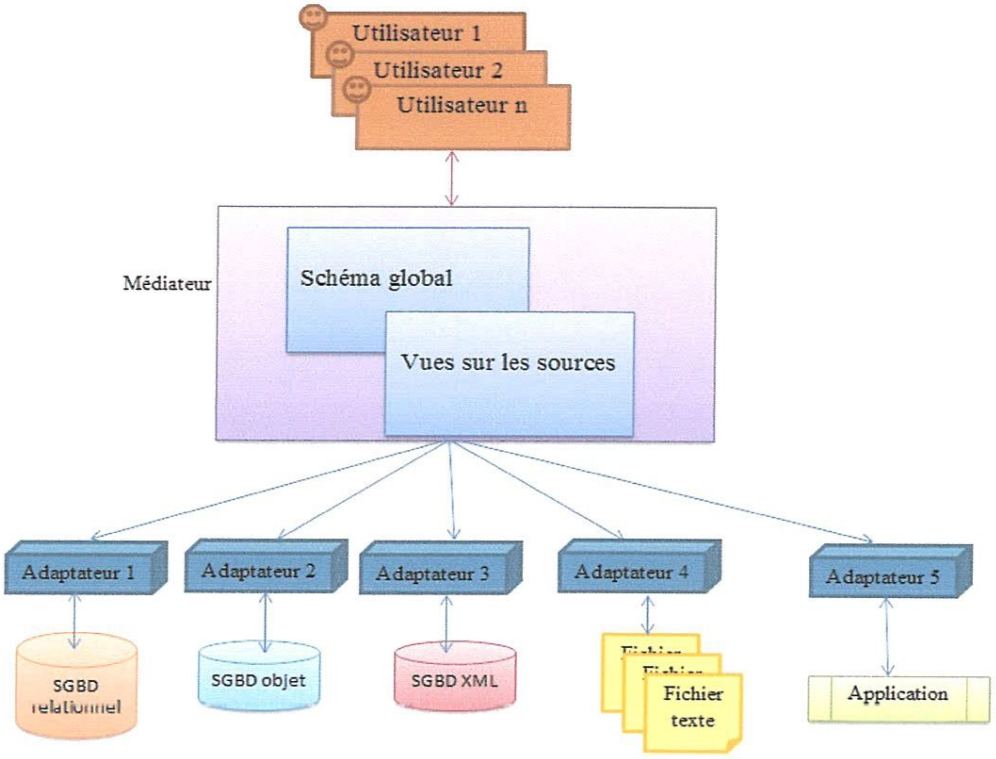


Figure 2.1 : *Architecture d'un système médiateur [21].*

Au niveau du médiateur, le schéma global fournit un vocabulaire unique pour l'expression des requêtes des utilisateurs et pour la description des contenus des sources par un ensemble de vues abstraites sur les sources.

Les adaptateurs ou wrappers traduisent les requêtes exprimées en termes du vocabulaire du schéma global en des requêtes exprimées dans le langage des sources. De plus, les adaptateurs transforment les réponses aux requêtes en des réponses conformes au schéma global du médiateur. Pour l'interrogation des données dans un médiateur, des mises en correspondance doivent être définies entre les relations du schéma global et les relations du schéma local.

De manière formelle, un système classique d'intégration de données I est un triplet < G, S, M> [31] où :

- G est le schéma global, un ensemble de relations globales;
- S est le schéma source, un ensemble de relations locales (disjoint de G) qui constituent la représentation des données contenues dans les sources ;
- M représente les correspondances entre G et S. Les correspondances sont constituées par un ensemble d'assertions qui établissent le lien entre les relations du schéma global et les relations du schéma source.

L'interrogation du système de médiation constitué se fait à l'aide des relations globales définies dans G. L'établissement des correspondances entre les schémas locaux et le schéma global s'effectue en tenant compte des problèmes liés à l'hétérogénéité des sources en présence.

2.1.2. l'intégration matérialisée de données :

L'intégration matérialisée où la vue unifiée des données est matérialisée et les données sont ramenées des sources d'origine et stockées dans une base de données réelle appelée entrepôt de données. Un système d'intégration suivant une approche entrepôt de données est constitué de trois niveaux, comme le montre la Figure 2 : *le niveau de l'entrepôt de données, le niveau des sources et le niveau des magasins de données.*

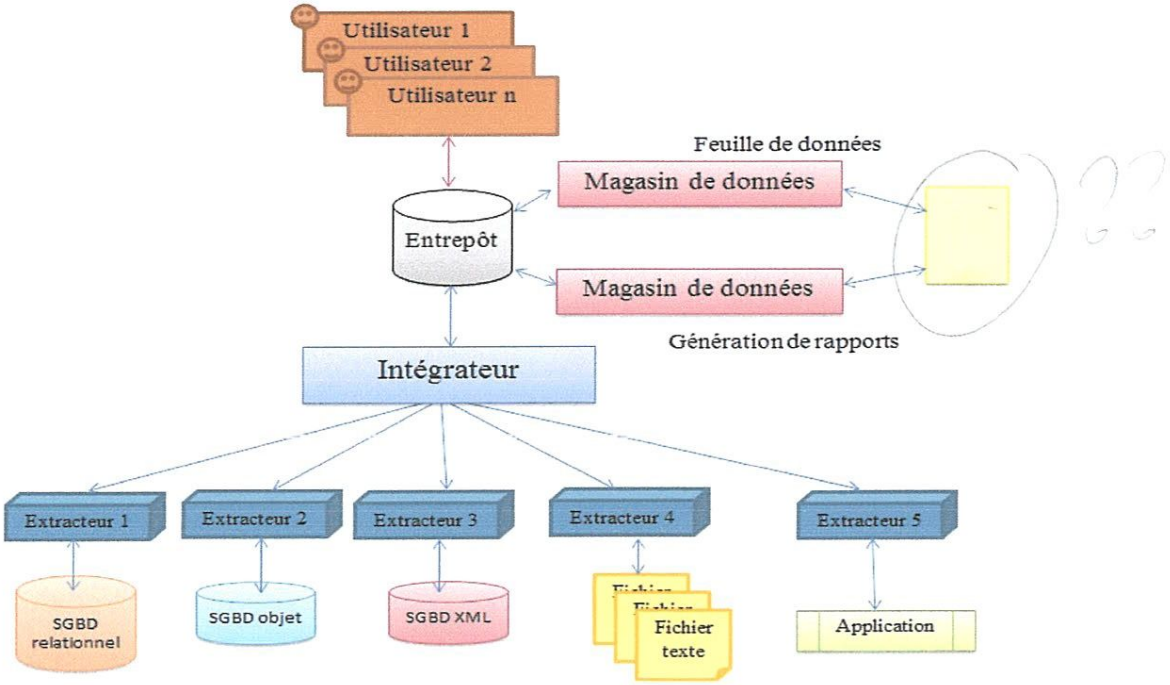


Figure 2.2 : *Architecture d'un entrepôt de données[21].*

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.

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.

Malgré que l'entrepôt simplifie l'optimisation et le traitement de requêtes en triant les données localement en fonction d'un seul schéma global, le principal problème posé est la difficulté d'assurer la synchronisation entre les copies des données stockées dans les entrepôts et les données originales stockées dans les bases de données [21].

Une approche mixte d'intégration de données est une approche qui co
l'approche de médiateur pour l'intégration des sources externes et l'appro
données pour l'intégration de leurs données [32].

Une étude comparative entre ces deux types d'intégration est illustrée dans le tab
suivant.

	Intégration virtuelle	**Intégration matérialisée**
Principe	Interface unifiée	Copie des sources
Actualisation temps réal	Données fraîches	Données historisées et non volatiles
Performance	Défis principal	Bonne
Opérations sur les données	Rien	Extraction, transformation, Nettoyage, filtrage
Opérations sur les sources	Accès aux sources	Alimentation des entrepôts et accès direct à l'entrepôt
Opérations sur les requêtes	Adaptation aux schémas locaux, fragmentation	Rien
Opérations sur les résultats	Adaptation au schéma global	Rien
Evolution	Rien	Souvent et largement
Type de requêtes	Simple mais coûteuses	Complexes et transactions longues
Personnalisation	Reformulation de requête	Données multidimensionnelles
Complexité	Maximale	Minimale

***Tableau2.1** : Comparaison entre les approches d'intégration [33].*

Dans ce sens, nous avons situé dans la problématique de traitement d'hétérogénéité sémantique de sources de données par la construction d'un système de médiation sémantique en exploitant ses avantages par rapport le système d'intégration matérialisé.

3. Approches d'intégration de données :

Un autre critère de classification est le mapping dans l'intégration de données qui sert à définir la relation entre le modèle global et celui des sources. Pour ce faire, plusieurs approches et systèmes d'intégration ont été proposés dans la littérature. L'intégration de données nécessite que chaque schéma local de chaque source de données subisse une transformation (mapping) en termes du schéma global de l'interface commune.

La classification des systèmes d'intégration de données ce fait selon la relation entre les schémas des sources locales et le schéma global du médiateur c-à-dire le mapping[22].

3.1.L'approche GAV (Global-As-View):

Dans l'approche GaV (Global-as-View) est une approche ascendante depuis les sources vers le médiateur où les relations du schéma médiateur (schéma global) sont exprimées en termes

de vocabulaire des schémas locaux. Les requêtes sont reformulées simplement par le remplacement des éléments changeant [27].

Exemple de requête en GAV	**Traduction de la requête**
Quelles sont les critiques des films réalisés par W. Allen ? SELECT **Films.titre,Critiques.critique** FROM **Films, Articles** WHERE **Films.Realisateur** = 'W. Allen'ANDFilms.titre=Articles.titre	SELECT S1.Film.titre, S2.Film.titre, S3.Critiques.critique FROM S1.Film, S2.Film, S3.Critiques WHERE (S1.Realisateur='W.Allen' AND S1.Film.Fid=S3.Critiques.Fid) OR (S2.Directeur='W.Allen' AND S2.Film.Fid=S3.Critiques.Fid).

***Tableau 2.2** : Exemple de requête en GAV [24].*

3.2.L'approche LAV « Local-As-View » :

Contrairement à l'approche GAV, dans l'approche LAV les vues sur les sources sont utilisées dans le sens opposé. Ces vues définissent comment l'information locale est liée au schéma global en exprimant une correspondance entre chaque relation dans le schéma local et une relation dans le schéma global.

- Définir les schémas locaux en fonction du schéma global : les relations des schémas locaux (sources) sont définies comme des vues (requêtes) sur le schéma global.
- Chaque source locale est définie comme une vue locale du schéma global.
- Une requête sur le schéma global doit être traduite en termes des schémas locaux (réécriture des requêtes).
-Approche descendante depuis le médiateur vers les sources[22].

Exemple de requête en LAV	**Traduction de la requête**
Quelles sont les critiques des comédies depuis 1950 ? SELECT **Films.titre,Critiques.article** FROM **Film, Critiques** WHERE Film.Fid=Critiques.Fid AND Film.genre='comedie'ANDFilm.annee>=1950	SELECT S1.Film.titre,S3.Critiques.article FROM S1.Film, S3.Critiques WHERE S1.Film.Fid=S3.Critiques.Fid

***Tableau 2.3** : Exemple de requête en LAV [24].*

3.3.L'Approche GLAV « Global-and-Local-as-Vie» :

Les avantages des approches LaV et GaV ont été combinés dans des approches mixtes. C'est ainsi qu'a été proposée l'approche GLaV (Generalized-Local-as-View), qui dispose de vues au niveau global et local. Une correspondance entre les vues au niveau global et local est requise. Ces correspondances n'ont pas de direction et s'appliquent dans les deux sens puisque les concepts dans le schéma global sont considérés comme des vues.

En fait, l'approche hybride permet d'avoir la correspondance entre les schémas locaux via le vocabulaire partagé. Donc la correspondance peut être calculée via le schéma global. Dans le cas de l'approche hybride modélisée selon GLaV, en gardant une indépendance entre les

sources (permet l'ajout et la suppression de sources), et en calculant indirectement les correspondances entre les sources. Cette caractéristique est impérative si nous voulons avoir des résultats cohérents [27].

3.4.Approche BGLaV « Brigham Young University Global Local as View»:

Cette approche est venue juste après l'approche GLaV, L'approche BGLaV se présente comme un point de vue alternatif qui n'est ni GaV ni LaV. L'approche emploie des mapping de la source à la source cible basée sur un schéma conceptuel prédéfini de la source cible, qui est spécifié ontologiquement et indépendamment des sources les unes des autres.

Il est plus facile de maintenir le système d'intégration de données avec BGLaV contrairement à GaV et LaV. La reformulation de requêtes est réduite au déploiement de règles. Comparera d'autres approches d'intégration de données, BGLaV combine les avantages de GaV et LaV, réduit leurs inconvénients, et fournit une alternative pour l'intégration de données flexibles et extensibles [27].

3.5.Approche BAV« Both as View» :

L'approche BaV (Both-as-View),qui utilise des transformations incrémentales afin de lier deux schémas. L'approche BaV a été introduite dans le cadre du projet d'intégration AutoMed[27].

Dans le tableau suivant, nous résumons les avantages et les inconvénients des approches vues précédemment :

Approche	Avantages	Inconvénients
GAV	-La réécriture d'une requête est simple -Conception naturelle et intuitive.	-N'est pas flexible pour l'ajout ou la suppression des sources de données, - N'est pas flexible pour l'ajout ou la suppression de certaines contraintes sur les sources.
LAV	-Flexible pour l'ajout ou la suppression des sources de données, - Flexible pour l'ajout ou la suppression de certaines contraintes sur les sources.	- La réponse aux requêtes est difficile, - La réécriture des requêtes est difficile. - changement du schéma global nécessite une reconsidération globale des vues
GLAV	- Donne la possibilité de mappings plus expressifs, - L'ajout/suppression est facile	-la réécriture des requêtes est complexe.
BGLAV	- Utiliser la Source et La Cible. - Mapping basé sur un schéma cible prédéfini. -Donne une algèbre relationnelle étendue.	-Non documentés.
BAV	Permet l'expression du mapping dans les deux sens. - Supporte l'évolution du schéma global et des schémas locaux.	-Susceptible d'être plus coûteux de raisonner et de traiter avec BAV qu'il ne serait avec les définitions des vues correspondantes dans LAV, GAV ou GLAV.

***Tableau 2.4 :** Les avantages et les inconvénients de différentes approches d'intégration [22].*

Une étude comparative entre ces différentes approches d'intégration est illustrée dans le tableau suivant.

	Automaticité de Mapping			Scalabilité passage à l'échelle	Extension	Traitement
	Sans ontologie Manuelle	Ontologie linguistique	Ontologie Formelle			
GaV	Manuelle	Semi-automatique	Automatique	Etude	Maj manuelle	Requête facile
LaV	Manuelle	Semi-automatique	Automatique	Oui	Maj manuelle	Réecriture difficile(vues)
GLaV	Manuelle	Semi-automatique	Automatique	Oui	Maj manuelle	Réécriture facile (vues)
BGLaV	Manuelle	Semi-automatique	Automatique	Oui	Maj manuelle	Réécriture facile (vues)
Bav	Manuelle	Semi-automatique	Automatique	oui	Maj manuelle	A prouver

Tableau 2.5 : *Comparaison entre les différentes approches d'intégration [27].*

D'une manière sommaire, nous nous focalisons dans notre travail sur la construction d'un système de médiation basé sur l'approche LAV. Ce système doit assurer une intégration sémantique à base d'ontologie pour traiter les différents problèmes d'intégration.

Dans cette optique, la section suivante présente la notion de sémantique dans le système d'intégration de type Médiateur.

4. Processus d'intégration à base de Médiateur :

L'intégration de sources de données via l'architecture de Médiateur peut effectuer manuellement ou semi-automatique ou bien automatique [21].

4.1.Approches manuelles :

Les premières approches proposées étaient des approches *manuelles*. Ces approches permettent d'automatiser l'intégration des données au niveau syntaxique. Les conflits sémantiques sont gérés manuellement et nécessitent la présence d'un expert humain pour interpréter la sémantique des données.

Plusieurs systèmes ont été développés selon cette approche comme les systèmes multi-bases de données, la fédération des bases de données, le système Tsimmis. Ces approches manuelles deviennent impraticables lorsque le nombre de sources de données à intégrer est important, ou lorsque les sources évoluent fréquemment [27].

4.2.Les approches semi-automatiques :

Afin d'apporter plus d'automatisation dans la résolution des conflits sémantiques, plusieurs travaux se sont tournés vers les ontologies. Une ontologie permet de fournir la sémantique des concepts d'un domaine de manière formelle.

Les approches *semi-automatiques* reposent sur des ontologies linguistiques et permettent d'automatiser partiellement la gestion des conflits sémantiques. Les ontologies linguistiques traitent des termes, et non des concepts. Ceci peut créer des conflits de noms. Momis est un exemple de projet reposant sur l'ontologie linguistique Wordnet pour l'intégration des sources[27].

4.3.Les approches automatiques :

Les approches *automatiques* consistent à associer aux données des sources une ontologie qui en définit le sens. Une ontologie traite les concepts d'un domaine donné. La sémantique du domaine est ainsi spécifiée formellement à travers des concepts, leurs propriétés ainsi que les relations entre les concepts. La référence à une ontologie permet d'éliminer *automatiquement* les conflits sémantiques entre les sources exemple Projet Buster annotation des données, Projet kraft [27].

5. Système de médiation sémantique :

Les systèmes de médiation sont de plus en plus développés et connus. Leurs composants essentiels sont : le schéma global, les mappings du schéma global avec les sources, les fonctions de réécriture de requêtes et les fonctions de composition des résultats. Tous ces composants prennent en compte l'hétérogénéité qui est un des principaux problèmes pour lesquels les systèmes de médiation sont construits.

L'objectif est de donner l'impression d'interroger un système centralisé et homogène alors que les sources interrogées sont réparties, autonomes et hétérogènes.

L'approche médiateur présente l'intérêt de pouvoir construire un système d'interrogation de sources de données sans toucher aux données qui restent stockées dans leurs sources d'origine. Ainsi, 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. L'interrogation effective des sources se fait via des adaptateurs, appelés des wrappers en anglais, qui traduisent les requêtes réécrites en termes de vues dans le langage de requêtes spécifique accepté par chaque source [22] [26].

Avant de présenter qu'est-ce qu'un système de médiation sémantique, il est nécessaire de décrire l'architecture détaillée d'un système de médiation classique (sans sémantique).

5.1.Architecture d'un système de médiation:

L'architecture de médiation fut conçue en 1992 par Gio Wiederhold dans l'article fondateur «*Mediator in the architecture of future information systems*». Il y développe une nouvelle

vision de l'architecture du traitement de l'information en entreprise, il tente de régler la problématique de l'accès et de l'intégration de l'information en introduisant la notion de médiateur [22].

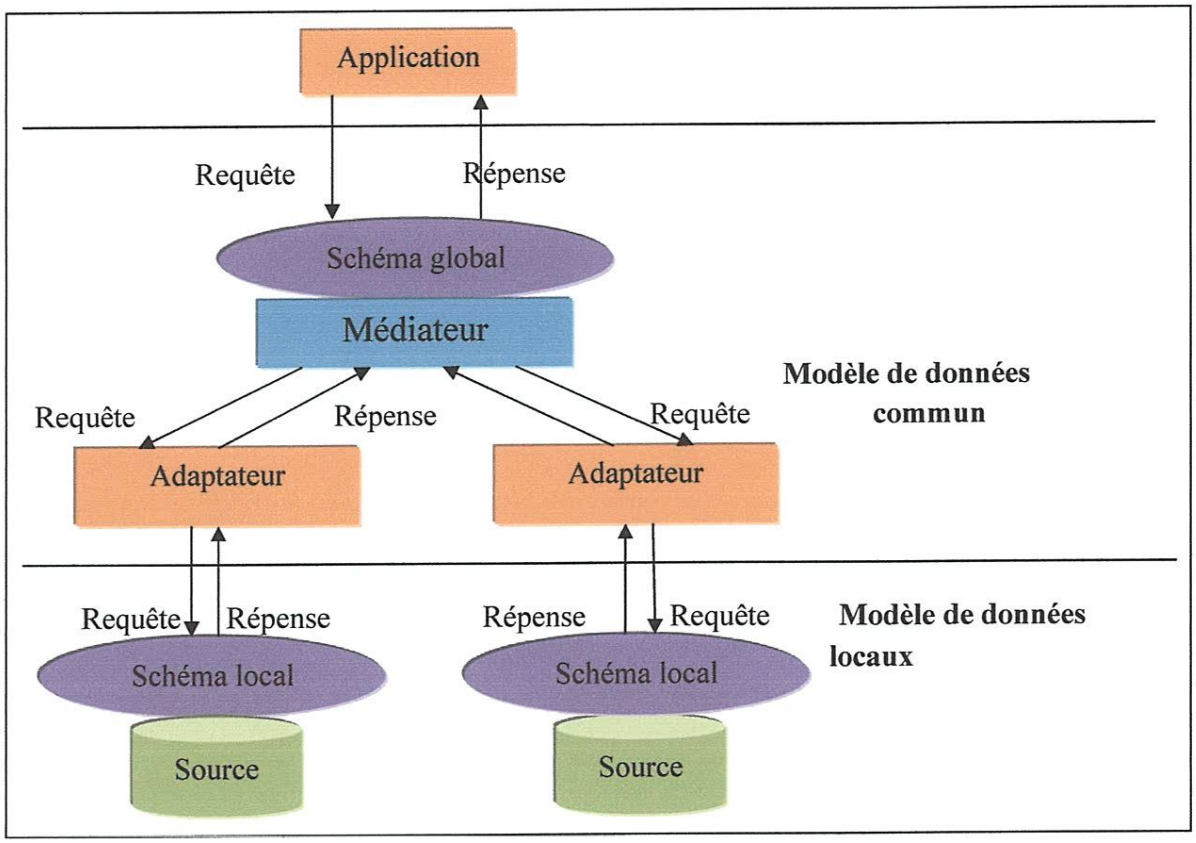


Figure 3.3:*Architecture d'un système de médiation [22].*

Comme nous pouvons le voir sur la figure 3, les systèmes de bases de données hétérogènes actuels adoptent une architecture distribuée qui consiste en plusieurs composants spécialisés.

Un médiateur exporte un schéma médiateur ou (schéma global) qui est une représentation intégrée des sources de données ; son rôle est central. Les applications accèdent aux sources de données hétérogène et distribué à travers les médiateurs. Les composants essentiels d'un système de médiation sont :

5.1.1. Adaptateur (Wrapper):

Un adaptateur prend en compte les dimensions Distribution, Autonomie et Interopérabilité. Son objectif est d'accéder à une source, d'extraire les données appropriées et de les présenter dans un format spécifié. Un adaptateur est donc un composant qui :

• Réalise la transformation entre le modèle de données dans lequel sont représentées les données de la source et le modèle choisi au niveau médiateur.

• Effectue la transformation entre les expressions de requêtes du niveau global en expressions compréhensibles par les sources. Inversement, il traduit les réponses des sources au format du niveau global.
• Assure les traitements spécifiques non disponibles au niveau local et nécessaires pour le niveau global (augmentation du pouvoir d'expression et de traitement de la source locale) [22].

5.1.2. Médiateur (Mediator):

Le rôle d'un médiateur est de collecter, nettoyer et combiner les données attendues par le système de médiation. C'est un module logiciel recevant directement la requête d'un usager et devant la traiter. Celui-ci doit localiser l'information nécessaire pour répondre à la requête, résoudre les conflits schématiques et sémantiques, interroger les différentes sources et intégrer les résultats partiels dans une réponse homogène et cohérente [22].

5.2.Utilisation des ontologies pour l'intégration de données :

L'ontologie comme représentation consensuelle et explicite d'une conceptualisation permet de fournir un vocabulaire partagé pour les différentes sources.

D'un point de vue général, Ushold et Gruninger divisent l'espace d'utilisation des ontologies en trois parties :

1. la communication entre personnes ayant des points de vue et des besoins différents ;
2. l'interopérabilité entre utilisateurs qui ont besoin de s'échanger des données et qui emploient des outils différents ;
3. l'ingénierie des systèmes, où la capacité des ontologies à faire partager et réutiliser des connaissances est exploitée dans la construction et l'utilisation des systèmes à base de connaissances.

L'utilité des ontologies dans le domaine de l'intégration de données peut se retrouver dans les deux premières catégories identifiées par Ushold et Gruninger [34].

Les ontologies peuvent jouer plusieurs rôles dans le processus d'intégration. Elles peuvent être utilisées pour établir les liens sémantiques entre des éléments dans des sources différentes. Elles peuvent aussi servir de modèle d'interrogation pour le système intégré lorsqu'elles sont utilisées pour décrire le schéma global.

Classiquement, nous distinguons trois architectures en fonction de la façon dont les ontologies sont utilisées au sein d'une infrastructure d'intégration [35]:

5.2.1. Architecture utilisant une ontologie unique :

Dans cette approche, chaque source à intégrer est liée à une seule ontologie de domaine globale (exp. Projet PICSEL). Ceci implique qu'une nouvelle source ne peut être décrite par sa propre ontologie. Tout ajout de source peut entraîner la modification de l'ontologie globale[27].

5.2.2. Architecture utilisant plusieurs ontologies :

Dans cette approche, chaque source à intégrer est décrite par sa propre ontologie, indépendamment des autres sources. Des correspondances entre les ontologies doivent être définies (exp. OBSERVER). Ces correspondances peuvent se révéler parfois très complexes, notamment à cause de niveaux de granularité différents entre les ontologies [35].

5.2.3. Architecture utilisant une approche hybride :

Dans cette approche, la sémantique de chaque source est décrite par sa propre ontologie. Cependant, les différentes ontologies sont connectées entre elles par un vocabulaire commun partagé.

La figure suivante illustre les trois architectures d'utilisation d'ontologie dans un système d'intégration [27].

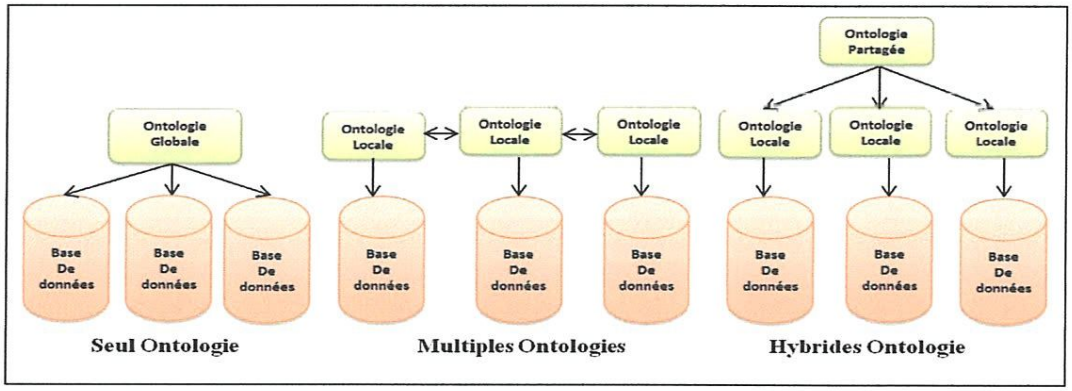


Figure 2.4 : *Trois architectures de système de médiation à base d'ontologie [27].*

5.3.L'alignement d'ontologies :

L'alignement d'ontologies est le processus de mise en correspondance (ou Matching) sémantique des entités qui les composent. Le processus est exécuté selon une stratégie ou une combinaison de techniques de calcul de mesures de similarité et utilise un ensemble de paramètres (ex: paramètres de pondération, seuils …) et un ensemble de ressources externes (ex: thésaurus, lexique…). Au final, nous obtenons un ensemble de liens sémantiques reliant les entités qui composent les ontologies. Ces derniers comprennent des relations d'équivalence, de généralisation/spécialisation, de chevauchement ou encore d'incompatibilité. De nombreux travaux ont été développés dans le domaine de l'alignement d'ontologies et portent sur les techniques de recherche de similarité et sur les outils ou sur les Framework qui les intègrent [28].

5.3.1. Les techniques de mesures de similarité sémantique :

On retrouve plusieurs méthodes de calcul de la similarité entre les entités de plusieurs ontologies.

5.3.1.1.Techniques terminologiques:

Elles sont souvent utilisées afin de déterminer le matching des noms et de leur
Ces méthodes se basent sur la comparaison des termes ou des chaînes de caractères ou les textes.

Il y a plusieurs manières d'évaluer la similarité entre deux entités. La manière la plus commune est de définir un seuil à une mesure de cette similarité. En premier lieu on applique des techniques qui considèrent les labels comme une suite de caractères et permettent de relier les concepts dont les labels sont rigoureusement identiques syntaxiquement, l'application de la mesure de similarité Edit distance (distance de Levenstein), cette mesure est basée sur la même hypothèse : deux termes sont similaires s'ils partagent assez d'éléments importants.

$$\mathbf{Sim_{syn}(t1, t2) = max(0, \frac{min(|t1|,|t2|) - ed(t1,t2)}{min(|t1|,|t2|)})}$$

Si Simsyn(c1,c2)=val >**seuil** alors on peut déduire une correspondance d'équivalence de la forme (c1, c2 , = , val). Après expérimentation un seuil est défini pour accepter les couples de termes syntaxiquement rapprochés.

5.3.1.2.Techniques linguistiques :

Utilisant des ressources externes (dictionnaires, taxonomies,…) la similarité entre deux entités représentées par des termes est calculée à partir des liens sémantiques déjà existants dans les ressources externes par exemple (WordNet) [28].

5.3.1.3.Les méthodes structurelles internes:

Elles calculent la similarité entre deux concepts en exploitant les informations relatives à leur structure interne (restrictions et cardinalités sur les attributs, valeurs des instances,…). Dans la plupart des cas, ce sont les informations concernant des attributs de l'entité, telles que la cardinalité des attributs, les caractéristiques des attributs ou les autres types de restriction sur les attributs.

5.3.1.4.Les méthodes structurelles externes:

Exploitent les relations entre les entités elles-mêmes, qui sont souvent des relations de subsomption (« is-a »). 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 également similaires [29][36].

5.3.2. Les outils d'alignement d'ontologies :

Différents outils ont été développés dans le but d'aligner plusieurs ontologies.

5.3.2.1.PROMPT :

Est un système interactif constituant une aide pour la comparaison, l'alignement, la fusion et l'évolution de plusieurs formalismes de représentation des connaissances. Son module

d'alignement appelé Anchor-Prompt permet de rapprocher des ontologies de la façon suivante: d'abord, des 'matchers' linguistiques permettent de déterminer un ensemble initial de concepts similaires. Ensuite, à partir de cette liste, un algorithme d'analyse les chemins dans les sous graphes délimités par ces concepts et détermine quelles classes apparaissent fréquemment dans les mêmes positions sur des chemins similaires. Cette analyse permet de guider l'utilisateur pour choisir les meilleurs mappings [28].

5.3.2.2.OLA :

(OWL Lite Alignment) est un système implémentant un algorithme d'alignement des ontologies décrites en OWL. OLA mesure la similarité entre deux entités à partir des calculs de similarité entre leurs caractéristiques (leurs types : classe, relation ou instance, leurs liens avec d'autres entités : sous-classes, domaine, ...). La valeur de similarité finale est la somme pondérée des valeurs de similarité de chaque caractéristique. Les poids sont associés suivant le type d'entité à comparer et ses caractéristiques [28].

5.3.2.3.AROMA :

(Association Rule Ontology Matching Approach) est une approche d'alignement pour des ontologies représentées en OWL. Elle permet de découvrir des liens sémantiques de type « subsomption » ou « équivalence » entre deux entités (classes ou propriétés). Le processus d'alignement se déroule en quatre étape étapes : (1) déduire des relations d'équivalence ; (2) trouver des incohérences (cycles) et de les éliminer (3); supprimer les relations redondantes ; (4) sélectionner le meilleur alignement pour chaque entité [28].

5.3.2.4.ASMOV :

(Automated Semantic Mapping of Ontologies with Validation) est un système d'alignement d'ontologies conçu pour l'intégration de sources de données hétérogènes représentées dans des ontologies. ASMOV permet de produire des mappings entre des concepts et/ou des propriétés et/ou des instances de deux ontologies. L'algorithme implémenté est automatique, il calcule de façon itérative, la similarité entre deux entités appartenant à deux ontologies différentes suivant quatre caractéristiques : (1) les éléments lexicaux (labels, commentaires); (2) les relations structurelles (ancêtres/descendants dans la hiérarchie) ; (3) la structure interne (restrictions sur les propriétés pour les concepts ;types, domaines et intervalles pour les propriétés ; valeurs pour les instances) ; (4) et les extensions (instances de classes et valeurs des propriétés). La similarité finale est calculée à partir de la somme pondérée des quatre mesures et permet d'obtenir un alignement. Le système vérifie ensuite cet alignement afin de s'assurer qu'il ne contienne pas d'incohérence sémantique [28].

6. Approches de mesure de similarité sémantique :

Dans cette section nous présentons les approches principales pour mesurer la similarité sémantique entre concepts.

6.1.Approches basées sur les arcs :

Le principe de cette approche consiste à calculer le nombre d'arcs entre deux objets d'une ontologie. Cette similarité est évaluée par la distance qui sépare les objets dans l'ontologie. Plusieurs types de similarité sémantique ont été proposés dans la littérature, les plus utilisées sont :

6.1.1. Similarité de « Wu & Palmer 1994 » :

Ils ont défini une mesure de similarité entre concepts pour la traduction automatique entre l'anglais et le chinois. Leur mesure s'applique à un domaine conceptuel qui correspond à un point de vue donné. La similarité est définie par rapport à la distance qui sépare deux concepts c.à.d. par rapport à leur plus petit généralisant PPG ainsi que la racine de la hiérarchie. La similarité entre C1 et C2 est :

$$\boldsymbol{ConSim}(\mathbf{C1},\mathbf{C2}) = \frac{\mathbf{2} * \mathbf{N3}}{\boldsymbol{N1} + \boldsymbol{N2} + \mathbf{2} * \boldsymbol{N3}}$$

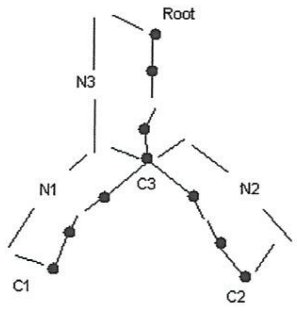


***Figure 2.5** : Les relations conceptuelles [23].*

Plus formellement cette mesure devient :

$$\boldsymbol{ConSim}(\mathbf{C1},\mathbf{C2}) = \frac{\mathbf{2} * \mathbf{depth}(\mathbf{C})}{\mathbf{depth}(\boldsymbol{C1}) + \mathbf{depth}(\boldsymbol{C2})}$$

Où C est le PPG de C1 et C2 (en nombre d'arcs), depth(C) est le nombre d'arcs qui séparent C de la racine et depth(Ci) avec i le nombre d'arcs qui séparent Ci de la racine en passant par C.

Cette mesure a l'avantage d'être simple à implémenter et d'avoir aussi des bonnes performances que les autres mesures de similarité [23].

6.1.2. Similarité du « Path » :

Il est basé sur deux observations. La première est que le comportement de distance conceptuelle. Le deuxième que la distance conceptuelle entre deux noeuds est proportionnelle au nombre d'arêtes séparant les deux noeuds dans la hiérarchie

$$sim_{path}(c_1, c_2) = 2 * deep\_\max - len(c_1, c_2)$$

-Len (c1, c2) : la longueur du plus court chemin de synset c_1 à synset c_2 dans WordNet
-deep_max : La profondeur maximum (c_i) de la taxonomie [23].

6.2.Approches basées sur les nœuds :

Ces approches adoptent une nouvelle mesure en termes de la mesure entropique de la théorie de l'information.

La probabilité P pour l'identification de l'utilisation d'une classe ou de ses descendants dans un corpus désigne l'information de la classe. On définit l'entropie d'une classe par la formule suivante :

$$\mathbf{E(c) = -log(P(c))}$$

Où P est la probabilité de trouver une instance du concept c. La probabilité d'un concept c est calculée en divisant le nombre des instances de c par le nombre total des instances.

6.2.1. Similarité de « Resnik 1995 »:

La notion du contenu informationnel (CI) a été initialement introduite par Resnik qui a prouvé qu'un objet (mot) est défini par le nombre des classes spécifiées et que la similarité sémantique entre deux concepts est mesurée par la quantité de l'information qu'ils partagent.

Pour évaluer la pertinence d'un objet il faut calculer le contenu informationnel. Le contenu informationnel est obtenu en calculant la fréquence de l'objet dans Wordnet. La formule proposée par Resnik est définie par:

$$\mathbf{Sim(X,Y)=Max[E(CS(X,Y))]=Max[-log(p(CS(X,Y))])}$$

Où CS(X, Y) représente le concept le plus spécifique (qui maximise la valeur de similarité) qui subsume (situé à un niveau hiérarchique plus élevé) les deux concepts X et Y dans l'ontologie. Cette mesure est un peu sommaire car elle ne dépend que du concept le plus spécifique [25].

6.2.2. Mesure de « Lin » :

Lin a défini une mesure de similarité légèrement différente de celle de Resnik :

$$Siml(X,Y) = \frac{2 * \mathbf{IC}(\mathbf{lso}(c1, 2c))}{\mathbf{I\,C}(c1) + \mathbf{I\,C}(c2)}$$

IC : contenu informationnel du concept commun le plus spécifique à deux sens.
Lso : le concept commun le plus spécifique de C1 et C2 est C3notée lso (c_1, c_2) = c_3 [45].

Cette mesure utilise une approche hybride qui combine deux sources de connaissances différentes (Thesaurus, corpus). En plus, elle représente la similarité comme degré probabiliste de chevauchement des concepts descendants de X et Y [25].

6.3.Approches hybrides :

Ces approches sont fondées sur un modèle qui combine entre les approches basées sur les arcs (distances) en plus du contenu informationnel qui est considéré comme facteur de décision.

6.3.1. Similarité de «Jiang et Conrath 1997 » :

Pour remédier au problème présenté au niveau de la mesure de Resnik, Jiang et Conrath a apporté une nouvelle formule qui consiste à combiner l'entropie (contenu informationnel) du concept spécifique à ceux des concepts dont on cherche la similarité (combine entre les techniques basées sur les arcs et les techniques basées sur les noeuds qui consistent à compter les arcs afin d'améliorer les résultats par des calculs basés sur les noeuds). La mesure adoptant cette méthode est basée sur la combinaison d'une source de connaissance riche (thesaurus) avec une source de connaissance pauvre (corpus) [25].

Sachant que la distance entre C1 et C2 est calculée par la formule suivante:

$$\mathbf{SimJCN = IC(C1) + IC(C2) + 2 \cdot IC(lso(C1 , C2))\ [45].}$$

-Lso : le concept commun le plus spécifique de C1 et C2 est C3 notée lso $(c_1, c_2) = c_3$ [45].

6.4.Approches basées sur l'espace vectoriel :

Ces approches utilisent un vecteur caractéristique, dans un espace dimensionnel, pour représenter chaque objet et calculent la similarité en se basant sur la mesure de cosinus ou la distance euclidienne. Le modèle de l'espace vectoriel est employé pour un arrangement des objets complexes en les représentants comme des vecteurs de k-dimensions. La définition de la similarité entre deux vecteurs d'objets est obtenue par leurs contenus internes. Par miles mesures à base d'espace vectoriel on peut citer :

6.4.1. Similarité de « Jaccard » :

La mesure de similarité de Jaccard (Jaccard, 1912) est définie par le nombre des objets communs divisé par le nombre total des objets moins le nombre d'objets communs:

$$\boldsymbol{Simj} = \frac{\mathbf{A \cap B}}{\mathbf{A \cup B}}$$

En prend un exemple de deux mots : mot1= « information », mot2= « informatique » :
-l'union de deux chaine : {a, e, f, i, m, n, o, q, r, t, u} : langueur 11.
-l'intersection de deux chaine : {a, f, i, m, n, o, r, t} : langueur 8.
-**Simj**=8/11=0.72 [44].

6.4.2. Similarité de « Cosinus » :

Cette mesure utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots). Deux objets sont similaires si leurs vecteurs sont confondus. Si deux objets ne sont

pas similaires, leurs vecteurs forment un angle (A, B) dont le cosinus représente la valeur de la similarité. La formule est définie par le rapport du produit scalaire des vecteurs A et B et le produit de la norme de A et de B.

$$Simc(A,B) = \cos(\emptyset) = \frac{\mathbf{A.B}}{||A||||B||.} = \frac{\sum_1^n A.B}{\sqrt{\sum_1^n A} * \sqrt{\sum_1^n B}}$$

La mesure de Cosinus quantifie donc la similarité entre les deux vecteurs comme le cosinus de l'angle entre les deux vecteurs (en verra dans le chapitre 4 un exemple) [25] [44].

6.4.3. Similarité « Sorensen » :

Cette mesure est très similaire à la mesure Jaccard, et était d'abord utilisé par Czekanowski en 1913 et redécouvert par Sorensen (1948):

$$S_S = \frac{2 * A \cap B}{||A|| + ||B||}$$

Exemple :
- le langueur de mot « informatique »:12.
- le langueur de mot « information »: 11.
- l'intersection de deux chaine : {a, f, i, m, n, o, r, t} : langueur 8.
- $S_S = \frac{2*8}{12+11} = 0.69$ [44].

7. Exemples de projets de systèmes d'intégration sémantique:

La multiplicité des sources biomédicales et la nécessité d'intégrer des résultats provenant de plusieurs appareils de mesures font que des travaux sur l'intégration sont de plus en plus appliqués au domaine biomédical. Par exemple, pour étudier un phénomène donné, les biologistes sont obligés de tenir compte des aspects physiologique, génétique, anatomique, biochimique, etc.
Nous présentons ici les systèmes d'intégration de domaine bio-informatique.

7.1.TAMBIS :

TAMBIS (Transparent Access to Multiple Bio-informatiques Information Sources) est un système de médiation basé sur une ontologie. Les requêtes dans TAMBIS sont formulées à travers une interface graphique où l'utilisateur navigue à travers les concepts définis au niveau du schéma global et choisit ceux qui l'intéressent pour la requête courante. Le système utilise la logique de description GRAIL, qui est aussi utilisée pour exprimer des requêtes sur le système. Toute requête exprimée en GRAIL est traduite en QIF (Query Internal Format), puis dans un plan d'exécution dépendant des sources [15].

7.2.Neurobase :

Neuro base est un projet commun entre plusieurs laboratoires français (2003-2005), dédié à la gestion de données et de connaissances réparties en neuro imagerie. Neuro base implémente

un système fédéré suivant l'approche médiateur/adaptateurs utilisant un référentiel sémantique commun (une ontologie médicale dénie pour les besoins du projet). Il est basé sur le médiateur Le Select25, qui permet de partager des données et des programmes hétérogènes et distribuées à travers un langage de requêtes de haut niveau [15]

7.3.PICSEL :

PICSEL, Production d'Interface à base de Connaissances pour les Services En Ligne est un projet développé au sein de l'équipe IASI du LRI pour le compte de France Télécom R&D. Le système intègre des bases de données relationnelles et des documents XML. PICSEL utilise un schéma global (qui modélise le domaine concerné par l'intégration) exprimé dans un langage appelé CARIN combinant le pouvoir d'expression d'un formalisme à base de règles et d'un formalisme à base de classes [15].

7.4.KRAFT :

(KRAFT) 164, 1531. Ce projet a été lance entre les universités d'Aberdeen, Cardiff, Liverpool et Britsh Telecom vers la fin des années 90. II vise à fusionner des données et des contraintes de différentes sources, dans un environnement, distribue. Les bases de données et les solveurs sont contrôles par un médiateur, Le principe de cette architecture consiste en premier lieu, A localiser et a contrôler la validité des contraintes et des données dans un environnement distribue et hétérogène. Ensuite, ces connaissances sont convertis dam une syntaxe spécifique et homogène pour générer un problème composite. Avant de fusionner les contraintes, ii faut réécrire les contraintes conforin6nent a un schéma d'intégration afin que toutes les contraintes utilisent les mêmes variables et les mêmes valeurs. Dana rebut, une procédure ("Wrapper") effectue une réécriture déclarative des données en contraintes homogènes. Seules les contraintes qui peuvent être réécrites dans une syntaxe composite propre A l'intégration sont exportées et partagées. Puis les contraintes sont placées sur différents noeuds du réseau (architecture Kraft) et résolues par un solveur de contraintes [43].

Conclusion :

L'intégration de données a pour objectif de combiner des sources de données autonomes, distribuées et hétérogènes afin d'obtenir une vue homogène et uniforme des données intégrées. Une façon pour y parvenir, est de représenter les données selon un même schéma global et selon une sémantique unifiée. Deux approches ont été présentées ; la première basée médiateur et la seconde basée entrepôt.

Dans le cadre de notre travail qui vise à proposer un système d'intégration sémantique de sources de données, nous avons présenté l'utilité des ontologies dans le domaine de l'intégration de données et plus précisément l'association d'ontologie dans le système de médiation afin de traiter les problèmes d'hétérogénéités.

Chapitre 3

« Conception de système de médiation sémantique »

"Ontology is an explicit specification of a conceptualization."

Gruber

Introduction :

Dans les chapitres précédents nous avons fait un état de l'art sur les travaux exis l'intégration des sources de données hétérogènes et plus précisément l'importance des ontologies pour assurer une intégration sémantique de qualité.

Ce chapitre est consacré à la conception du système de médiation sémantique guidé par l'ontologie sur les risques alimentaire. Cette conception est basée sur l'approche LAV pour effectuer le lien entre le schéma global et les schémas locaux et sur l'approche hybride pour l'utilisation d'ontologie en tant que, d'une part, support d'interrogation unifiée et de l'autre part, comme une ressource de connaissance au niveau de chaque source locale.

Nous avons utilisé les bases de données relationnelles comme sources de données structurées pour réaliser notre système.

1. Objectifs de notre système d'intégration sémantique:

Le système d'intégration que nous avons conçu, nous a permet non seulement d'assurer l'accès unifié aux sources hétérogènes mais de traiter les conflits sémantiques et syntaxiques via ontologie de domaine tout en préservant l'autonomie des sources de données.

Nous avons utilisé l'approche de médiation LAV comme méthode d'intégration qui consiste à définir les schémas des sources locales comme des vues du schéma global. Ce dernier est représenté par une ontologie de domaine.

Nous avons choisi le domaine bio-informatique et plus précisément l'étude des risques microbiologique dans les aliments. Ce domaine particulier a été choisi car il s'agit d'un bon exemple pour représenter les sources hétérogènes et montrer les conflits sémantiques et syntaxiques tels que les conventions et les vocabulaires utilisés dans chaque source.

Les objectifs de notre système d'intégration sémantique sont :

- Offrir à l'utilisateur une vue uniforme et une interrogation transparente des sources de données hétérogènes ;
- Traiter les problèmes d'ambigüité des mots d'une requête par l'utilisation d'une ontologie globale comme support d'interrogation ;
- Associer à chaque source locale une ontologie locale pour définir leur sémantique ;
- Traitement de requêtes au niveau des adaptateurs ;

A cet effet, il est nécessaire de présenter l'architecture conceptuelle pour concevoir notre système de médiation sémantique.

2. Architecture conceptuelle du système :

Pour concevoir notre système d'intégration sémantique des sources de données hétérogènes, nous présentons d'une façon générale l'architecture conceptuelle en préservant l'autonomie des sources à intégrer.

Source locale → B.D.R

De ce fait, il est nécessaire de détailler dans un premier temps le niveau médiateur qui comporte cette ontologie par la suite le niveau sources de données et enfin le niveau adaptateur qui se trouve entre les deux niveaux précédents.

3.1.Niveau Médiateur :

Le niveau médiateur représente le cœur de l'architecture et le point d'entrée du système à partir d'un schéma global définit par l'ontologie '**ONTARIS**'.
Le médiateur doit offrir les fonctionnalités suivantes :
1. Localiser les sources de données pertinentes.
2. Accepter les requêtes des utilisateurs.
3. Réécrire (décomposer) et optimiser les requêtes (optimisation répartie).
4. Envoyer les plans d'exécution à faire et les exécuter par les adaptateurs sur des différentes sources.
5. Combiner (recomposer) les résultats des adaptateurs.

Le médiateur comporte un moteur de requêtes et une base de connaissance spécifique au domaine du médiateur. Cette base se compose de l'ontologie '**ONTARIS**' et des descriptions du contenu des sources de données. La sortie de médiateur est un ensemble de plans de requêtes qui seront traduits par les adaptateurs afin d'obtenir la réponse à la requête de l'utilisateur.

La conception de l'ontologie '**ONTARIS**' est présentée dans les prochaines sections.

3.2 Niveau Sources de données :

Ce niveau est constitué des sources concernées par l'intégration. Nous avons utilisé le modèle relationnel pour la modélisation des sources locales. Ces sources sont construites d'une façon automatique via les techniques de fragmentation d'une base de données globale utilisée comme outil d'aide à la construction des sources.

Par ailleurs, nous avons suivi l'approche LAV pour définir les schémas des sources à partir de l'ontologie globale. Cette dernière nous a permet de construire la base de données globale que nous utiliserons pour construire les bases locales. Pour cela, nous avons proposé deux approches:

La première approche : consiste à garder la même structure de l'ontologie, c.à.d, la base de données est composée d'une seule table relationnelle qui contient tous les concepts de l'ontologie avec leurs instances, les data propreties et les Object propreties. La clé primaire de cette table sera l'URI de l'ontologie.

La deuxième approche : est plus simple que la première. La base de données globale est construite à partir les instances de l'ontologie et elle est composée de plusieurs tables relationnelles de sorte que :

-Toutes les supers classes deviennent des table ;
-Les sous classes deviennent des attributs dans la table de sa super classe ;

-Les data type propreties de chaque classe deviennent des attributs de cette table.
-Les Object propreties deviennent des relations entre les tables.
-Enfin les instances de chaque classe sont des données de la table correspondante.

A partir de ces deux propositions, nous avons choisi la seconde parce qu'elle est plus claire pour comprendre son contenu et bien organisée en termes des classes et des sous classes et cela d'une façon automatique à l'aide d'un programme JAVA et l'utilisation de deux types de requêtes : SPARQL pour l'extraction de contenu de l'ontologie et SQL pour la construction des tables.

Nous obtenons par la suite une base de données contenant cinq (5) relations : Factors, Food, Microbe, Symptoms, Therapy (voir la figure 3.2)

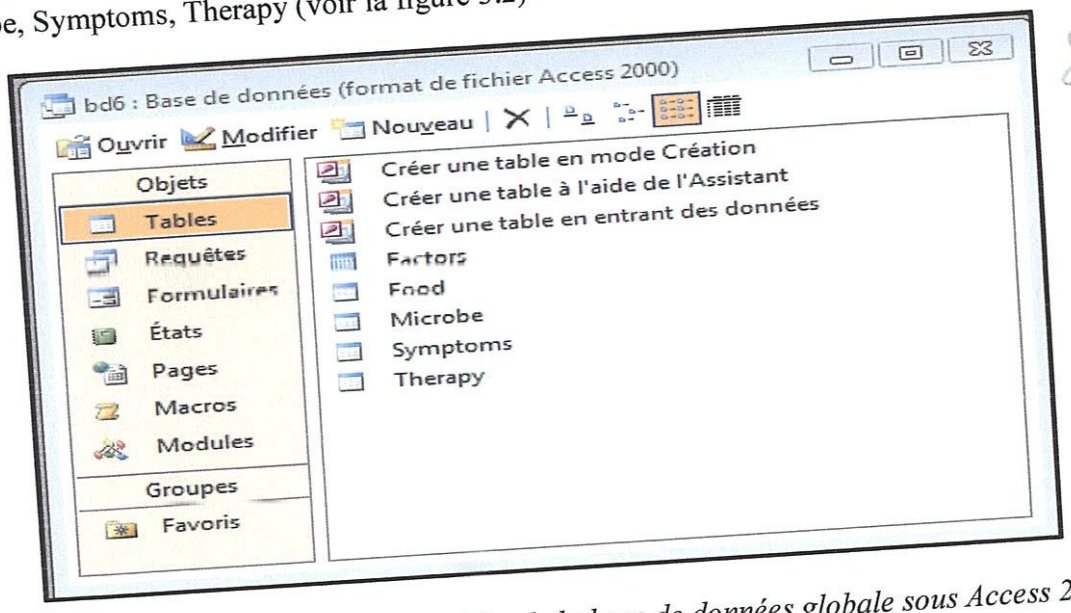


Figure 3.2 : *Imprime écran sur les tables de la base de données globale sous Access 2003.*

Les cinq tables sont :
Factors (Num, Name, Type, Lien).
Food (Num, Name, Type1, Type2, Kingdom , family , order, Binomial_name , date_of, Taste, Vitamin, minerals, Link).
Microbe (Num, Name, Type1, Type2, Scientific_call, unconverring_by, be_dated, Kins, Land, ordering, distance, diam, link_withontology) .
Symptoms (Num, Name, Type, Lien).
Therapy (Num, Name, Type1, Type2, Lien).

Par la suite les sources locales sont obtenues par la réalisation des fragmentations verticales et horizontales.

3.2.1. Fragmentation horizontale :

La fragmentation est le processus de décomposition d'une base de données en un ensemble de sous bases de données. Cette décomposition doit être faite sans perte d'information.

La fragmentation horizontale consiste à segmenter la base selon un certain nombre de conditions en gardant les mêmes attributs de la table initiale. L'opérateur de fragmentation est la sélection (σ) via la clause WHERE de SQL. La recomposition est faite par l'union (U).

Exemple : on appliquant la fragmentation horizontale sur la table food et l'attribut Type 2 afin de construire trois fragments : Feed, Nutriment et Aliment.

Num	Name	Type1	Type2
1	Watermelons	Fruits	medium_food_spoilage
2	Walnut	Nuts	slow_food_spoilage
3	Sausage	Red_Meats	Perishable_foods
4	water_contaminated	Null	Water

Tableau 3.1 : *Exemple des tuples de quatre fragments de la table food*

3.2.2. Fragmentation verticale :

La fragmentation verticale consiste à segmenter la base selon un certain nombre d'attributs tous en gardant la clé primaire comme attribut en commun dans chaque fragment. L'opérateur de fragmentation est la projection (π) via la clause SELECT de SQL. La recomposition est faite par la jointure.

Exemple : Dans la table factors, nous effectuons trois fragmentations verticales : fragment 1 contient deux attributs Num (clé primaire) et Name, le fragment 2 est composé des attributs Num et type et le fragment 3 comporte le Num et le Lien. Le tableau suivant décrit une partie de la table factors.

Num	Name	Type	Lien
4	Insects_and_rodents	Factors	47#Insects_and_rodents
2	Moisture	Factors	47#Moisture
5	The_Absence_Of_Sterilization	Factors	47#The_Absence_Of_Sterilization

Tableau 3.2 : *Exemple des tuples de la table factors.*

La combinaison de ces deux types de fragmentations (horizontale et verticale) est appelée fragmentation Mixte.

3.2.3. Sources de données utilisées :

Nous avons utilisé la fragmentation horizontale et verticale pour la construction de quatre sources de données. Ces sources sont hétérogènes selon la sémantique, la syntaxe et la structure:

Sémantique	Syntaxe	St
noms des tables et des attributs complètement différents.	- Abréviation - erreurs syntaxiques (mots incomplet).	la structu des sourc
Exemple : nom de la table feed de la source 1 est nommé Nutriment dans la source 2 ainsi que le nom d'attribut kind signifie type dans une autre source.	-Veggietbale et Veggie -Vitamin et Vitami,Date_of-descovery et Date-of	- source 1 contient cinq tables. - source 4 comporte seulement 2 tables.

Tableau 3.3 : *les types des hétérogénéités utilisées.*

Pour plus de détail sur les sources hétérogènes utilisées, une récapitulation des sources à intégrer est représentée dans le tableau suivant.

Num de la base	Tables relationnelles	Fragmentation	Description
Base 1 comporte 5 tables	feed	horizontale selon le type du food	Continent food de type medium food spoilage (vegetabele, fruits).
	Microorganism	horizontale selon le type du Microbe	Continent Microbe de type Virus (Rota Virus, Echovirus…Etc.)
	Factorize	Vertical	continent les deux premiers attributs (num, name) (PH, Moisture, Heat…Etc.)
	indication	Vertical	Continent les deux premiers attributs (num, name) (hypersensitivity, Fever, Death...Etc.).
	treatment	horizontale selon le type du Therapy	Continent Therapy de type Vitamin
Base 2 comporte 2 tables	Nutriment	horizontale selon le type du food	Continent food de type slow food spoilage (Cereals, Legumes, Nuts).
	Offer	Vertical	continent les deux attributs (num, lien) (47#hypersensitivity, 47#Fever, 47#Death...Etc.).
Base 3 comporte 5 tables	Aliment	horizontale selon le type du food	Continent food de type perishable food (Meats, Milk and Milk Products).
	Bug	horizontale selon le type du Microbe	continent Microbe de type Mycotoxin (Aflatoxin, Citrinin Etc.)
	Cause	Vertical	continent les attributs (num, type1) (factors.)
	Review	Vertical	continent les attributs (num, type1) (Symptoms).

	Training	horizontale selon le type du Therapy	continent Therapy de type Antibiotic (Polymyxin, Penicillin,...Etc.).
Base 4 comporte 4 tables	Lunch	horizontale selon le type du food	continent food qui n'a pas un type exemple water contaminated et Egges.
	Organism	horizontal selon le type du microbe	Continent Microbe de type Bacterium (Salmonella,...Etc.)
	Impulse	Vertical	contient le num et lien du impulse (47#PH).
	Remedy	horizontal selon le type du Therapy	contient les treatment du type Antibiotic (Polymyxin, Penicillin, Cephalosporin, ... Etc).

Tableau 3.4 : *Sources de données utilisées.*

Enfin et après avoir construit les sources à intégrer, nous associons pour chacune sa propre ontologie locale qui doit référencer l'ontologie globale.

3.3.Niveau Adaptateur :

Les adaptateurs représentent l'interface de communication entre les sources et le médiateur. Ils possèdent les composants suivants :
- Schéma de l'ontologie locale propre à chaque source.
- Module de traitement de requêtes (évaluation des sous requêtes en termes de schéma de la source).
- Module de traitement sémantique à l'aide du dictionnaire Wordnet et les ontologies locales.

De plus, les adaptateurs transforment les réponses aux requêtes en des réponses conformes au schéma global du médiateur.

Ils s'agissent donc d'une interface permettant l'interrogation d'une base de données grâce à un langage normalisé. Il seralt également possible d'utiliser une ontologie pour réaliser l'interopérabilité sémantique et traiter les requêtes complexes

L'adaptateur cache l'hétérogénéité au médiateur et il peut être intelligent et donc effectuer des optimisations spécifiques aux sources.

4. Conception de l'ontologie globale 'ONTARIS' :

Pour la conception d'une telle ontologie, il est nécessaire de choisir un domaine dans lequel on réalise notre travail. Comme on a indiqué dans la section 2 que nous avons choisi le domaine des risques microbiologique dans les aliments. Alors, il est indispensable de présenter dans un premier temps ce domaine d'étude par la suite nous décrivons la conception dc l'ontologic dc cc domainc.

4.1. Présentation de domaine des risques alimentaires :

L'intoxication alimentaire est incluse dans le domaine du risque alimentaire. Elle n'est habituellement pas d'une grande gravité, mais certaines intoxications alimentaires peuvent avoir des conséquences sérieuses, et même mortelles, pour quelques personnes.

Un risque alimentaire se produit le plus souvent après la consommation d'aliments ou d'eau contenant des bactéries, des toxines bactériennes, des parasites, ou des virus. Il peut également se produire quand des poisons non infectieux (comme des champignons vénéneux).

Les aliments qui sont le plus souvent impliqués dans les intoxications alimentaires sont les œufs, les laitages, les viandes et certains légumes. Ils sont représentés dans la figure suivante :

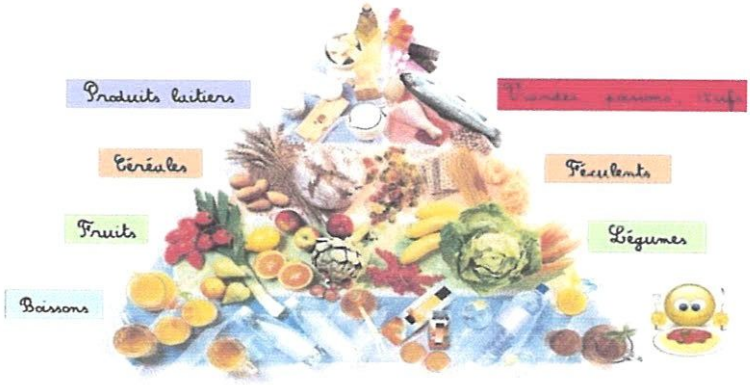


Figure 3.3. : *Domaine de risques alimentaires.*

Le choix du domaine des risques alimentaires est motivé par le fait qu'il représente un bon exemple pour traiter l'hétérogénéité des sources et nous a permis de protéger notre vie contre les virus et les bactéries.

Par conséquent, la sous-section suivante décrit les différents concepts de ce domaine ainsi que les relations entre eux via la construction de l'ontologie de domaine ONTARIS.

4.2. Construction de l'ontologie 'ONTARIS' :

Il existe une multitude de méthodes d'ingénierie ontologique. Cependant, il n'y pas de consensus sur les principes qui doivent guider la modélisation ontologique. La plupart de ces méthodes visent soit à réutiliser une ontologie existante ou bien de la construire à nouveau à partir de documents du domaine, ou de questions posées aux experts.

Dans le cadre de notre travail, nous avons essayé de construire à nouveau notre ontologie baptisée 'ONTARIS' (ONTology for Alimentation RISks) sur les risques alimentaires.

Pour concevoir notre ontologie, nous suivons deux principales phases : Spécification des besoins et la conceptualisation.

4.2.1. Spécification des besoins :

Dans cette phase on va déterminer le domaine et la portée de l'ontologie :

- Le domaine que va couvrir l'ontologie est le domaine des risques alimentaires.
- Le but de l'utilisation de notre ontologie ONTARIS est de traiter le problème d'hétérogénéité sémantique dans un système d'intégration.
- L'ontologie doit répondre aux requêtes des utilisateurs en prenant en compte la sémantique de ces dernières.
- L'ontologie sera utilisée par toute personne qui veut s'informer sur ce domaine.
- L'ontologie doit permettre un utilisateur qui est infecté par une intoxication alimentaire lui donner les traitements nécessaires.
- L'ontologie est une bibliothèque de connaissance qui donne à l'utilisateur la possibilité de connaitre les types des aliments, leurs microbes, la valeur nutritive et les facteurs qui causé la détérioration des aliments.
- L'ontologie sera maintenue par des experts en microbiologie.

4.2.2. Conceptualisation :

Nous allons construire entièrement notre ontologie conceptuelle en effectuant des interviews avec des experts de domaine des risques alimentaires et en s'aidant de la documentation (documents de la FAO, articles, thèses, livres etc.). Nous avons utilisé la langue anglaise pour étudier ce domaine, car elle nous a permet d'utiliser le dictionnaire Wordnet en anglais qui est très riches des termes par rapports le Wordnet des termes français.

Etape 1 : énumérer et identifier les termes important de l'ontologie. L'étude faite sur le domaine des risques alimentaires, nous a permis de dégager une liste importante de termes: Alimentation risks est le concept le plus général. Factors, microbe, food, Symptom, Therapy sont des concepts généraux de niveau supérieur.

Etape 2 : définir les classes et la hiérarchie des classes.

Prenons par exemple les deux concepts de l'ontologie globale : Microbe et Therapy

Concepts	Description	Sous concepts
Microbe	Décrit les microbes qui sont nés dans un aliment selon un certain nombre des facteurs.	-Bacterium, -Virus -Mycotoxine.
Therapy	Décrit le traitement à prendre lorsqu'une personne est infectée par un microbe	- Antibiotic - Vitamins

Tableau 3.5. *Classes et hiérarchie de classes de l'ontologie.*

Etapes 3 & 4 : définir les propriétés des classes, les relations.

Exemple : la Description des Propriétés de classe Salmonella de super classe Microbe et la classe Apple de la classe Food.

Concepts	Propriété	Type
Salmonella	Family: Enterobacteria	String
	Biominalname: Salmonella	String
	Discovered by: Daniel_Elmer_Salmon	String
	Date of discovry: 1900	Datetime
	Length: 2_to_5_µm	String
	Order: Enterobacteriales	String
	Kingdom: Bacteria	String
	Diameters: 0comma7_to_1comma5_µm	String
Apples	Family: Rosaceae	String
	Biominalname:Malus_domestica	String
	Date of discovry: 1600	Datetime
	Order: Rosales	String
	Kingdom: Plantae	String
	Minerales: magnesuim_and_potassium	String
	Vitamin: C_and_ A_ and_B1	String
	Calories: 81_cal	String
	Species : 24	String

Tableau 3.6. *Extrait des propriétés du modèle de l'ontologie.*

Description des relations : prenons trois relations suivantes :

Relation	Concept source	Concept cible	Description
Has_microbe	Minced_Meat	Salmonella	Est une relation entre le nutriment et la bactérie qui vive dans ce nutriment.
Has_Symptoms	Escherichia_coli	Diarrhea	Est une relation entre microbe et les symptômes qui suivre l'intoxication de ce microbe

Tableau 3.7 : *Relations entre concepts du modèle de l'ontologie.*

Etape 5 : créer les instances des classes dans la hiérarchie.

Exemple: la classe Factors comporte les instances suivantes : CO2, O2, Heat, Insects_and_rodents, Moisture…etc.

Etape 6 : création du modèle ontologique

La dernière étape de conceptualisation consiste à lier les différentes informations obtenues dans les étapes précédentes sous un modèle conceptuel. En effet, la figure suivante représente la hiérarchie des concepts et les différentes relations entre eux.

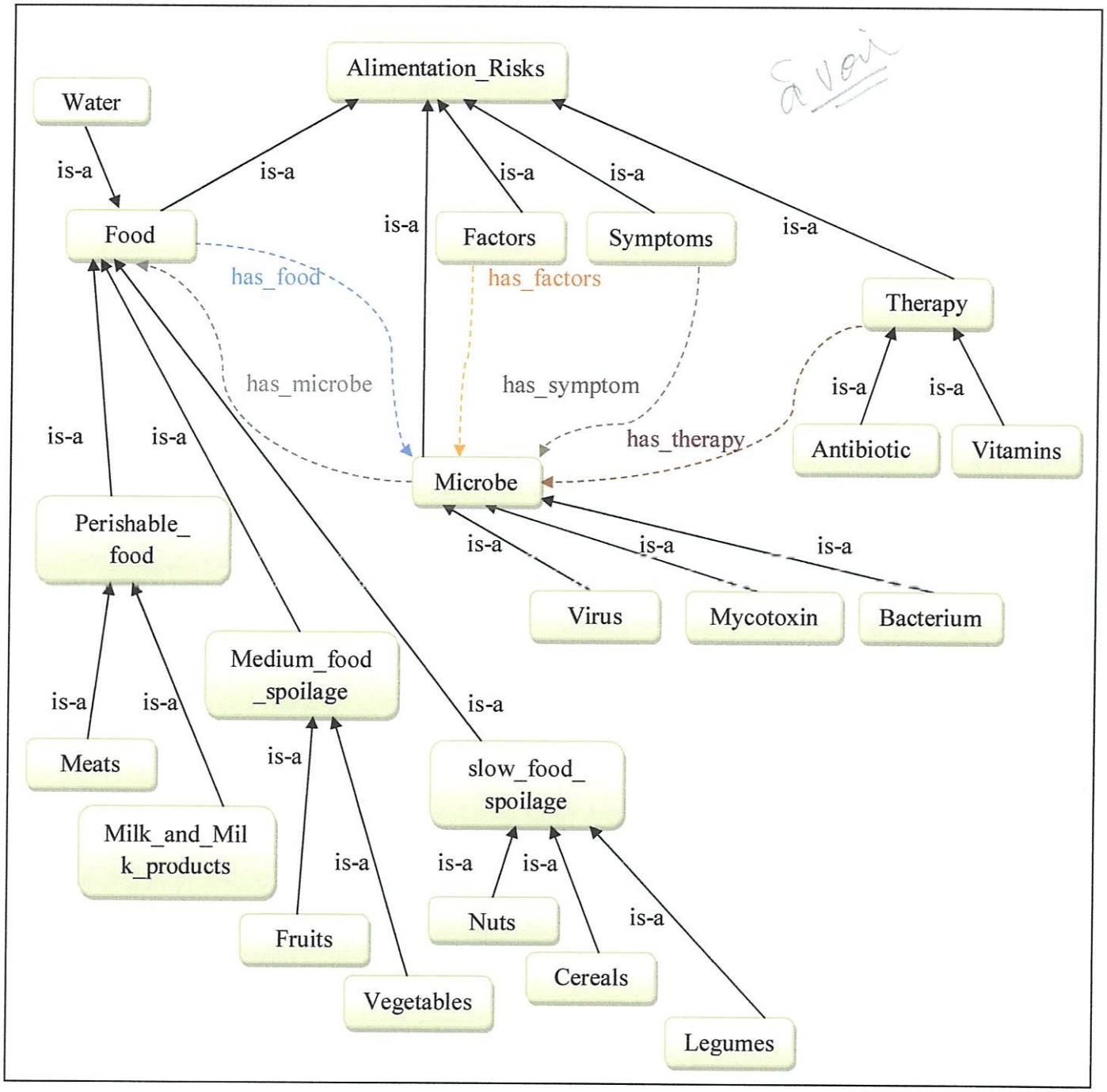


Figure 3.4 : *le modèle conceptuel de l'ontologie ONTARIS.*

5. Ontologies pour la représentation des sources à intégrer :

Chaque source concernée par le processus d'intégration est vue à travers une ontologie locale qui décrit le contenu de la source en question et de la manière d'y accéder. Chaque ontologie locale est décrite dans le langage OWL et elle doit référencer l'ontologie globale (ONTORIS) de médiateur.

Après avoir construit automatiquement les sources locales (quatre bases de données) via les techniques de fragmentation des bases de données, on crée automatiquement les ontologies locales via un programme JAVA qu'on verra ça dans le dernier chapitre En d'autres termes, les ontologies locales sont vues comme des fragments de l'ontologie partagée (globale).

Associer à chaque source locale une ontologie a pour but de décrire sa sémantique ce qui permet de traiter les conflits sémantiques lors de soumission d'une requête en exploitant les liens sémantiques entre concepts de l'ontologie locale. Cette dernière doit être connectée avec l'ontologie commune (ou globale) du médiateur.

La mise en correspondance (ou l'alignement) entre les ontologies locales et l'ontologie globale, permet d'identifier les relations entre les entités des différentes ontologies. Cet alignement peut se faire par plusieurs techniques :

- ***Terminologiques*** *:* basées sur les chaînes de caractères ou sur des connaissances linguistiques
- ***Linguistiques*** *:* basées sur les propriétés morphologiques et syntaxiques des termes ou sur des ressources externes comme les dictionnaires
- ***Structurelles*** : basées sur les attributs des entités ou sur les relations de subsomption entre les entités.
- ***Extensionnelles*** *:* basées sur les instances des concepts.

Le fait que les ontologies locales sont construites automatiquement à partir les bases de données locales, on n'a pas besoins de représenter le modèle conceptuel correspondant de chaque ontologie.

Conclusion :

Dans ce chapitre, nous avons présenté la conception de notre système de médiation sémantique à base d'ontologie. Nous avons présenté également les trois niveaux d'architecture d'intégration et le modèle conceptuel de l'ontologie globale ONTRIS sur les risques alimentaires.

Dans le chapitre suivant, nous allons présenter en détail le processus d'intégration sémantique avec les algorithmes de traitement de requêtes d'utilisateurs, les mesures de similarité sémantique utilisées et le matching entre l'ontologie globale et les ontologies locales.

Chapitre 4

« Approche proposée pour l'intégration Sémantique à base d'ontologie »

"An algorithm must be seen to be believed. "

Donald Knuth

Introduction :

Après avoir présenté dans le chapitre précédent la conception de notre système sémantique à base d'ontologie. Nous abordons dans ce chapitre l'approche l'intégration sémantique.

Ce chapitre décrit les étapes d'intégration sémantique et les algorithmes du traitement de requêtes selon deux types: requête simple et requête composée.

1. Principe de l'approche proposée :

L'intégration sémantique des sources de données hétérogènes a pour but d'offrir une interface d'interrogation unifiée permet de cacher la distribution des sources et leur hétérogénéité. Nous avons proposé notre approche selon les techniques suivantes :

1. *Intégration automatique des sources hétérogènes :* sans intervention de l'utilisateur, l'élimination des conflits sémantiques et syntaxiques est faite via l'ontologie ONTARIS.
2. *Le mapping entre le schéma global et les schémas locaux ;* est effectué selon l'approche LAV où les sources de données sont définies comme un ensemble de vues sur l'ontologie globale. L'avantage principal de l'approche LAV est la facilité offerte pour l'ajout d'une nouvelle source. Cependant l'évaluation des requêtes peut s'avérer complexe.
3. *Architecture hybride d'utilisation d'ontologie ;* chaque source est associée à sa propre ontologie, ces dernières sont connectées entre elles à travers une ontologie globale.
4. *Sources de données structurées :* pour faciliter notre travail, nous avons choisi les bases de données relationnelles comme modèle de source hétérogènes.
5. *L'hétérogénéité :* nous avons étudié trois types d'hétérogénéités : hétérogénéité *sémantique* (la signification des termes), *syntaxique* (les mots incomplets, abréviation) et *structurelle* (schéma relationnel différent).
6. *Module de traitement sémantique :* est basé sur l'application d'un certain type de mesures de similarité sémantique via l'utilisation de l'ontologie linguistique Wordnet et l'ontologie du domaine ONTARIS.

L'approche d'intégration que nous étudions est basée sur les hypothèses suivantes :

- Il existe une ontologie de domaine (ontologie partagée) recouvrant la totalité des termes consensuels, nommée ONTARIS.

-Chaque source locale peut se définir en termes d'une ontologie qui lui est propre (sources de données à base ontologique).
- Chaque ontologie locale référence à priori autant que cela est possible l'ontologie de domaine.

Nous présentons dans les sections suivantes le processus d'intégration sémantique de l'approche proposée avec ses différents algorithmes.

2. Processus d'intégration sémantique :

Le processus d'intégration vise à intégrer d'abord les ontologies locales ensuite les données. L'ontologie partagée peut jouer le rôle d'un schéma global et chaque source pourrait être vue comme un sous schéma de cette ontologie. Cette situation est quasi similaire aux bases de données réparties, où une base de données centralisée pourrait être décomposée en plusieurs fragments qui seront placés sur des sites répartis.

Notons que cette automaticité concerne la résolution ou l'élimination des conflits sémantiques (Wordnet) entre les sources durant le processus d'intégration.

Selon l'architecture conceptuelle de notre système de médiation sémantique ; nous détaillons le fonctionnement de ses trois niveaux :

- Médiateur : c'est le cœur du système qui est composé des modules de traitement de requête utilisateur et la reconstitution de résultat final.
- Adaptateur : adapte la requête du médiateur selon les termes du vocabulaire de schéma local et traite les conflits existants.
- Sources de données : exécution de requête et envoie de résultat à l'adaptateur concerné.

Nous commençons d'abord de détailler le fonctionnement du médiateur qui représente la première étape du processus d'intégration.

2.1.Fonctionnement de médiateur :

Nous pouvons représenter le médiateur selon deux modules qui effectuent l'homogénéisation de recherche. Ces deux modules sont : Module de formulation de requête et l'autre la reconstitution de résultats.

La première étape d'intégration des sources de données hétérogènes consiste à lancer la requête d'utilisateur à l'ontologie globale via l'interface unifiée.

2.1.1. Module de formulation de requêtes :

Ce module consiste à formuler les requêtes en termes de schémas globaux « ontologie globale » et elle est aussi la responsable de l'envoi de cette requête aux adaptateurs.

L'algorithme :

```
ALGORITHME Formulation_R;
ENTREES  ED : L'ensemble des données;
SORTIES Q :Requête;
DEBUT
1. Q←Ecrire-Requête (ED) ;
2. Decomposition (REQUETE (SG)) ←REQUETE (S1) ∧ REQUETE (S2) ∧  ... ∧
   REQUETE (Si);
3. Pour i=1 : n faire
4. Envoie-aux-Adaptateurs (Qi, Adpi) ;
FIN.
```

L'explication du l'algorithme :

•**Etape 1 :** Réception de la requête par le médiateur.

• **Etape 2:** Cette étape concerne la décomposition par le médiateur de la requête globale en sous-requêtes.

• **Etape 3 :** L'envoie des sous requêtes (ou requête atomique) aux adaptateurs pour effectuer leurs travaux.

2.1.2. Module de reconstruction de résultat :

Ce module a pour but de reconstruire un ensemble de résultats adapté aux besoins d'utilisateur à partir les sous résultats qui sont envoyés par les adaptateurs.

```
ALGORITHME Reconstruction _Rs;
ENTREES  ER : L'ensemble des sous résultats;
SORTIES R :Résultat unifié;
DEBUT
1. Pour i=1 : n faire
2. ER←ER∪Résu-Env-Adpi (si)  ;
3. Fin pour ;
4. R ← Reconstruire_Résultat (ER) ;
5. Envoyer (R) ;
FIN.
```

L'explication d'algorithme :

• **Etape 1 :**les adaptateurs envoient les sous résultats au médiateur.

• **Etape 2 :** les sous résultats sont reconstitués selon le vocabulaire du schéma global de médiateur.

• **Etape 3 :** finalement le résultat final du médiateur est envoyé vers l'utilisateur via l'interface unifiée.

2.2.Fonctionnement de l'Adaptateur :

L'adaptateur ou Wrapper est le niveau intermédiaire entre le médiateur et les sources d'information. Chaque adaptateur est associé à chaque source locale, il traduit le schéma des sources en termes du schéma global et les requêtes du médiateur en termes compréhensibles par les sources. Ce processus est effectué via *le module du traitement de requête*.

De plus, l'adaptateur reçoit à partir de propre source les réponses des sous requêtes et les envoie par la suite au médiateur. Cette opération est réalisée via *le module de réception des résultats*.

2.2.1. Module du traitement de requêtes :

Ce module est la base de bon fonctionnement du système de médiation sémantique. Il permet de traiter tous les conflits des sources de données à intégrer. A partir des sous requêtes du médiateur, ce module réécrit ces sous requêtes en termes des vues des sources locales.

Le traitement de requête est lie au degré de composition de requête c'est-à-dire requête simple ou atomique (non décomposable) et requête composée. Dans cette sous-section nous présentons en général le fonctionnement d'adaptateur. Pour plus de détail sur le type de requête est décrit dans la section 3.

```
ALGORITHME Traitement-R (Adpi);
ENTREES EQ: ensemble des sous requêtes;
SORTIES Q :Requête réécrite;
DEBUT
1. Réception (EQ);
2. Trouver_ Sources (Sous-REQUETE (Ri), Adaptateur (Adpi));
3. Q←Traitement -conflits (EQ) ;
4. Envoie_sourcei (Q) ;
FIN
```

Explication de l'algorithme : pour un adaptateur Adpi

•**Etape 1**: la Réception des sous requêtes de médiateur.

• **Etape 2:** la Recherche de sources contributives: Cette étape consiste à trouver pour un adaptateur concerné toutes les sources sous-jacentes, pertinentes qui sont de même modèle que lui, et qui peuvent contribuer au calcul dc la sous-rcquête correspondante.

• **Etape 3:** Par ailleurs, la prise en compte de traitement des conflits sémantique et syntaxique est indispensable afin de s'assurer une meilleur correspondance entre l'ontologie globale et les schémas locaux.

• **Etape 4:**Envoie de requête « réécrite » vers la source concernée.

La procédure Traitement –conflits (EQ) permet de traiter les conflits sémantique et syntaxique. Dans ce cadre, nous avons guidé par l'ontologie linguistique Wordnet pour traiter les problèmes de synonymies et polysémie lors de réécriture de requête. De ce fait, nous avons choisi la mesure de similarité de Wu et Palmer comme mesure de base.

Comme nous avons décrit dans le chapitre 2, que la similarité de Palmer est basée sur la profondeur des concepts dans la hiérarchie de l'ontologie globale. La formule suivante est la similarité de Palmer.

$$\boldsymbol{ConSim}(\mathbf{C1},\mathbf{C2}) = \frac{\mathbf{2} * \mathbf{depth}(\mathbf{C})}{\mathbf{depth}(\boldsymbol{C1}) + \mathbf{depth}(\boldsymbol{C2})}$$

Nous avons utilisé un paquage Java qui calcule automatiquement la similarité de Palmer (plus de détail dans le chapitre 5).

Cette mesure présente l'avantage de la rapidité du temps d'exécution, mais l'inconvénient de la production d'une valeur de similarité de deux concepts voisins qui dépassent la valeur de deux concepts dans la même hiérarchie. Pour cette raison, nous avons combiné d'autre mesure de similarité qui est fondée sur la représentation vectorielle de requête en utilisant une technique algébrique simple. La mesure la plus utilisée dans ce modèle est la valeur du cosinus de l'angle entre le vecteur du concept X et le vecteur de concept Y. Cette mesure a l'avantage d'être facile à mettre en œuvre. Si les vecteurs sont normalisés, le cosinus se calcule par la formule suivante :

$$\boldsymbol{Simc}(\boldsymbol{X},\boldsymbol{Y}) = \mathbf{cos}(\boldsymbol{X},\boldsymbol{Y})\frac{\mathbf{X}.\mathbf{Y}}{||\mathbf{x}||_2.||\mathbf{y}||_2}$$

Sachant que X.Y est le produit scalaire de deux vecteurs et ||X|| est la magnitude (ou l'amplitude) du vecteur X. Deux objets sont similaires si leurs vecteurs sont confondus. Si deux objets ne sont pas similaires, leurs vecteurs forment un angle (X, Y) dont le cosinus représente la valeur de la similarité. Le résultat de ce calcul est toujours une valeur comprise entre 0 et 1, où 0 signifie 0% similaire, et le 1 est 100% similaire. [Web4]

Nous présentons dans ce qui suit l'algorithme de calcul de similarité cosinus.

Algorithme :

```
ALGORITHME Cosinus;
ENTREES a, b : mot du type String;
SORTIES S :similarité entre 0 et 1;
DEBUT

1. Si (nombre-caractère(a)>= nombre-caractère(b))
2. N= nombre-caractère(a)
3. Sinon N= nombre-caractère(b)
4. FIN SI
5. POUR i de 1  jusqu'à N
6. A[i]=Calculer _occurrence(a) ;
7. B[i]=Calculer _occurrence(b) ;
8. P=Produit-scalaire (A[i], B[i]) ;
9. MA=Magnitudes (A[i]) ;
10. MB=Magnitudes (B[i]) ;
11. FIN POUR;
12. MM=Multiples-magnitudes (MA,MB) ;
13. S= Deviser(P,MM) ;
14. FIN.
```

Pour appliquer cet algorithme à notre proposition, nous commençons de convertir les deux mots a et b en vecteurs selon les lettres composées «a», «b», «c», etc. Avec son occurrence binaire si le caractère existe en mai 1 sinon en mai 0, en suite on calculer la fréquence d'apparition d'un caractère dans un mot.

Le produit scalaire des vecteurs A et B est le produit usuel entre les éléments du vecteur. Le résultat de ce produit une valeur.

$$(A * B) = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

- La Formule pour Calculer la magnitude de Vecteur.

$$\|X\| = \sqrt{x^2{}_1 + \cdots + x^2{}_n}$$

Exemple d'un concept de notre ontologie:

1-Considérons le terme «apple»:
-Clés de dimension dans l'ordre :{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}.
-le vecteur du occurrence binaire pour "apple"
{1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0}.
-le vecteur de Fréquence du Occurrence pour "apple"
{1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,2,0,0,0,0,0,0,0,0,0,0}.
-Conversion de nos mots « apple» et «applet» dans la fréquence des vecteurs d'occurrence:
2-A (apple)= (1, 1, 1, 2, 0), B (applet) (1, 1, 1, 2, 1).
3-Le produit scalaire des vecteurs A et B = (1 * 1) + (1 * 1) + (1 * 1) + (2 * 2) + (0 * 1) = 1 + 1 + 1 + 4 + 0 = 7
4- Calculer la magnitude pour appel= $\sqrt{(1^2 + 1^2 + 1^2 + 2^2 + 0^2)} = \sqrt{(1 + 1 + 1 + 4 + 0)} = \sqrt{(7)}$
5-Calculer la magnitude pour applet = $\sqrt{(1^2 + 1^2 + 1^2 + 2^2 + 1^2)} = \sqrt{(1 + 1 + 1 + 4 + 1)} = \sqrt{(8)}$
6- Multiples les magnitudes de A et B= $\sqrt{(7)} * \sqrt{(8)} = \sqrt{(56)} = 7.48331477$
7-Devlser le produit scalaire de A et B par le produit des magnitudes A et B = 7 / 7.48331477 = .935414347 (94% similaire).

L'algorithme Cosinus similarité est utile pour déterminer si la composition des deux chaînes est similaire, et ne prend pas l'ordre des chaînes en compte [Web4].

2.2.2. Module de réception de résultats :

Les réponses à une requête posée sont obtenues en évaluant les réécritures sur les extensions des vues.

```
ALGORITHME Réception-Rs (Adp_i);
ENTREES Qr : requête réécrite ;
SORTIES R :Résultat;
DEBUT
1. R←Réception-Résultat  (Qr, S_i);
2. Envoie (R, médiateur) ;
FIN
```

- **Etape 1:** réception des résultats R des sous-requêtes exécutées Qr par les sources qui concernent S_i.

- **Etape 2:** l'adaptateur Adp_i envoie le résultat R au médiateur.

2.3.Fonctionnement des sources :

Deux modules ont été effectués qui sont : l'exécution de requêtes réécrites et l'envoie de sous résultats à l'adaptateur.

2.3.1. Module d'exécution de requêtes :

Ce module consiste à exécuter les sous requêtes réécrites.

L'algorithme :

```
ALGORITHME Exu_Si (Si);
ENTREES Qr : requête réécrite;
SORTIES R :Résultat de Requête;
DEBUT
1. Qr ← recevoir(Adpi) ;
2. R ← Exécution _ Requête (Qr) ;
FIN
```

L'explication du l'algorithme :

- **Etape 1:**Réception de requête réécrite Qr de la part de l'adaptateur
- **Etape 2:**Résultat de l'Exécution de requête par la source concernée.

2.3.2. Module d'envoi de résultats :

Ce module consiste à envoyer les sous résultats aux adaptateurs.

```
ALGORITHME Rés_Si;
ENTREES R :Résultat;
DEBUT
1. Envoi_Résultat (R, Adpi) ;
FIN
```

3. Type des requêtes :

Pour expliquer le fonctionnement de l'approche proposée, nous avons distingué deux types de requête initiale: requête simple et requête composée.

Dans cette section nous détaillons comment la requête d'utilisateur est exécutée dans notre système de médiation sémantique.

3.1.Requête simple :

L'interrogation en mode requête simple (requête non décomposable) permet de faire des interrogations atomiques sans se soucier des champs sur lesquels va porter la recherche. La réécriture de requête est faite à partir du schéma global en une requête écrite en SQL (langage des sources locales) sans utilisation des ontologies locales. Dans ce cas-là, nous appliquons

aisément les similarités de wu palmer du Wordnet et de cosinus pour résoudre les conflits existants. L'algorithme suivant montre le déroulement de réécriture d'une requête simple.

```
ALGORITHME  Requête _simple
ENTREES  ED : L'ensemble des données;
SORTIES R : Résultat du Requête;
DEBUT
1. Réception (ED) ;
2. T ← Extrait_table(ED) ;
3. A ← Extrait_attribu(ED) ;
4. TL ← Extrait_table (Liste des sources s1.....sn) ;
5. TS ← Wu-Palmer_Similarité(T, TL);
6. A1 ← Extract_attribute(TS) ;
7. Si (Type(A)=Simple) alors AS←Wu-Palmer_Similarité (A,
   A1);
8. Sinon AS ← Cosinus _Similarité (A, A1);
9. NR [1..n] ← Formulation_des_ Requête (TS, A1);
10.   R ← Exécution (NR) ;
11.   Envoi_Réslt (M, R) ;
End
```

•**La première étape** : la réception de la donnée dans le terme d'ontologie globale. Ces donnes contient nom du table et nom d'attribue (T, A). Ex : T=Food et A=Name

•**La deuxième étape :** consiste à construire toutes les tables à partir des 4 sources TL.

•**La troisième étape :** consiste à calculer la similarité sémantique de Wu-palmer entre les table TL et T et met les résultats dans un tableau des double et gardé le nom de la table qui a la grande similarité et le met dans un String TS.

•**La quatrième étape :** consiste à extrait les attribues des tables de TS.

•**La cinquième étape :** en faisant un test sur le type d'attribue : si le type simple (atomique) on utilise la similarité de wu-palmer pour déterminer l'attribue A1 qui a la plus grande similarité avec A et on le met dans une String AS, la même chose si le type de l'attribue est complexe en appliquant la similarité de cosinus.

•**La sixième étape** : la reformulation de la requête selon le vocabulaire des tables locales TS et leurs attributs A1.

•**La septième étape** : l'exécution de la requête reformulée dans la précédente étape R, et finalement envoyer le résultat R de cette requête au médiateur M.

3.2.Requête composée :

Dans ce type de requêtes, l'utilisation de l'ontologie linguistique Wordnet est insuffisante, il faut utiliser l'ontologie formelle de chaque source et exploiter les liens sémantiques entre les concepts composant la requête. En effet, un matching entre l'ontologie globale et l'ontologie

locale de la source concernée pour réécrire la requête composée en termes de vocabulaire de l'ontologie locale. Par la suite, un mapping est fait entre l'ontologie locale et les sources de données puisque les données ne résident qu'au niveau des sources.

Exemple :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX lo:<http://www.semanticweb.org/hp/ontologies/2014/10/untitled-ontology-47#>
SELECT ?x
        WHERE {
                Lo: Apples lo: has_microbe ?x.
              }
```

Le problème dans cette requête est que le concept Apples se trouve dans la source 1(ontologie local 1) et il est lie sémantiquement avec aflatoxine qui lui-même n'existe pas dans la source 1. Hors, L'aflatoxine se trouve dans la source 3 (ontologie local 3). Pour résoudre ce problème, on fait un matching entre l'ontologie locale 1 et l'ontologie globale et on trouve un lien sémantique entre appeles et microbe. Enfin, l'affichage des résultats à l'utilisateur via l'interface unifiée.

L'algorithme suivant montre le déroulement de réécriture d'une requête composée :

```
ALGORITHME  Requête _Composée
ENTREES  ED : L'ensemble des données;
SORTIES R :Résultat du Requête;
DEBUT
1. Réception (ED) ;
2. OLx=Exist (ED,OL1...OLN)
3. CM ← concpet_matching (OG,OLx) ;
4. SM ← concept_Structural_matching (OG,OLx);
5. DM ← datatype_ matching (OG,OLx);
6. OM ← objectdata_ matching (OG,OLx);
7. F ← Formulation_des SPRQLRequête ();
8. R ← Exécution _ Requête (F, OLx, OG) ;
9. Envoi-Réslt(M, R) ;
10.  End
```

Explication de l'algorithme :

- **La première étape** : la réception des données en termes de vocabulaire de d'ontologie globale.

- **La deuxième étape :** consiste à chercher dans quelles ontologies locales existent les données ED. Si on trouve cette ontologie locale on le nommé OL_x.

- **La troisième étape :** extraire l'ensemble des correspondances entre OL_x et l'ontologie globale OG selon 4 techniques de matching.

-Concpet_matching: on va extraire tous les concepts de l'ontologie globale et de l'ontologie locale OL_x, après on va calculer la similarité sémantique entre les concepts. Il existe des similarités qui sont calculées à l'aide du Wordnet (wu palmer) exemple : food et feed et d'autres qui sont calculées par Cosinus. Les résultats de ce matching on les met dans CM pour les utiliser dans le matching suivant « structurale matching».

-concept_Structural_matching : le but de ce type de matching est de voir est ce que la structure d'un individu par exemple Apples dans l'ontologie locale est similaire à sa structure dans l'ontologie globale, cette étape est un peu complexe, Alors on a essayé de l'expliquer étape par étape :

1- On a CM contient le résultat de concept-matching, on vérifie est ce qu'il existe des concepts qu'on a entrés dans la première étape, si oui on extrait tous les individus de ces concepts et on vérifie est ce que Apples existe dans ces individus. S'il existe, on garde les concepts qui ont vérifié cette condition. Le résultat de cette étape est la structure de l'individu Apple dans l'ontologie globale.

2-Après, on construit la structure du Apples dans ontologie locale, on va calculer la similarité sémantique entre les résultats de l'étape précédente et les concepts de cette ontologie locale. Par la suite, on va extraire les individus des concepts qui sont similaires et on construit systématiquement la structure de l'individu.

Exemple : L'ensemble des donnés qui on a entré est:

- Concept: [Food, medium_food_spoilage, Fruits].
- Individus: [Apples].
- Relation :[has_microbe].

-Apples existe dans l'ontologie locale 1 avec la liste des concepts représentés dans le tableau suivant :

Concept dans l'ontologie globale	**Concept dans l'ontologie locale**	**Type de similarité**	**Résultat**
Microbe	Microorganism	Wu Palmer	{0.94, Microbe}
Food	Feed	Wu Palmer	{0.92, Food}
factor	Factorize	Wu Palmer	{0.72, factor}
Therapy	treatment	Wu Palmer	{0.90, Therapy}
Symptom	indication	Wu Palmer	{0.46, Symptom}
medium_food_spoilage	medium_food	CosineSimilarit	{0.88, medium_food_spoilage}
Vegetables	Veggie	CosineSimilarit	{0.71, Vegetables }
Fruits	Fruits	CosineSimilarit	{1, Fruits}
vitamins	Vita	CosineSimilarit	{0.63, vitamins}
Virus	Virus	CosineSimilarit	{0.99, Virus}

Tableau 4.1 : *Liste des concepts similaires entre l'ontologie globale et l'ontologie locale1.*

•**On va faire le test** : est-ce que Concept: [Food, medium_food_spoilage, Fruits] existe dans le résultat du concept matching. Selon le tableau ci-dessus, la réponse est Oui.

-On va extraire les individus de Food, medium_food_spoilage, Fruits et on a trouvé Appels dans :

```
Apples---->Food.
Apples---->medium_food_spoilage.
Appels---->Fruits.
```

•L'étape suivante consiste à trouver les concepts de l'ontologie locale qui sont similaires du {Food, medium_food_spoilage, Fruits}. On trouve {Feed, medium_food, Fruits}, ensuite, on extrait les individus de ces concept est vérifiés l'existence de l'individu Apples dans ces concepts.

```
Apples---->Feed
Apples---->medium_food
Apples---->Fruits
```

Alors, nous remarquons que la structure de l'Apples dans l'ontologie locale est similaire à sa structure dans l'ontologie globale.

-datatype_ matching : dans ce type de matching, nous essayons de vérifier le type de propriétés de l'individu Apple dans l'ontologie globale via une requête SPARQL et faire la même chose avec l'ontologie locale.

Exemple :

Apples dans l'ontologie globale	Apples dans l'ontologie locale
Binomial_name : Malus_domestica	Binomial_call : Malus_domestica

-objectdata_ matching : la dernière étape consiste à réaliser le matching au niveau de relation [has_microbe] et les Object proprety dans ontologie locale.

Object proprety dans l'ontologie globale	Object proprety dans ontologie locale	Type de la similarité	Résultat
has_microbe	hasMicroorganism	WuPalmer	0.94

•On remarque que dans cette ontologie locale 1, il y a comme type du microbe les virus, mais Apples dans l'ontologie globale a des relations avec d'autre type du microbe mycotoxine, alors :

•Dans l'ontologie locale : `Apple→ hasMicroorganism→Rota_Virus.`

•Dans l'ontologie globale :

```
Apple→has_microbe →Aflatoxin.
Apple→has_microbe → Altenuene.
Apple→has_microbe → Rota_Virus.
Apple→has_microbe →Expansin.
```

-finalement le résultat de la requête `{Apple has_microbe}` est `Aflatoxin, Altenuene, Rota_Virus, Expansin.`

4. Système expert :

Dans notre système de médiation sémantique, nous avons ajouté une fonction supplémentaire qui consiste à diagnostiquer les intoxications des malades sous forme des questions/réponses. Cette fonction est faite par un système expert.

Si l'utilisateur est infecté par une intoxication alimentaire, lui donner les traitements nécessaires.

Le système expert est fondé sur les bases de connaissance qui sont représentées par les ontologies locales et l'ontologie globale et l'application de matching pour trouver les liens sémantiques entre elles.

L'algorithme suivant montre le déroulement de réécriture du Système expert :

```
ALGORITHME Système _Expert
ENTREES  ED : L'ensemble des données;
SORTIE   T:Résultat de Requête;
DEBUT

1. Réception (ED) ;
2. F←Extrait-Food(ED) ;
3. S←Extrait-Symptom(ED) ;
4. M←Extrait-microbe (F) ;
5. S1←Extrait-Symptom(M) ;
6. Si identique (S, S1) = vrais alors
7. T←Extrait-therapy(S1) ;
8. Fin Si
9. Affichage _Résultat(T) ;
10.  FIN
```

Explication de l'algorithme :

- **La première étape** : L'utilisateur entre l'ensemble des données ED qui contient la liste des aliments qui il mange F et les symptômes S

- **La deuxième étape** : le système cherche les microbes M qui sont existé dans l'aliment F.

- **La troisième étape** : extrait les symptômes de ces microbes M et on le met dans S1.

- **La quatrième étape** : en vérifiés est ce que ces symptômes S1 sont égaux aux les symptômes S, si oui le système donné les traitements à suivre à partir de la fonction Extrait-therapy(S1). Toutes ces opérations sont faites via notre système de médiation sémantique et l'approche proposée.

Exemple :

- Un utilisateur mange sausage (saucisse en français).

- Après quelque heure, il est senti les symptômes comme Vomit, Spasm, Diarrrhea, Nausea.

- A partir de ces deux informations, le système cherche les bactéries qui ont une relation avec sausage. Il trouve la bactérie salmonella.

- Le système extrait tous les symptômes de salmonella et les compare avec ceux de l'utilisateur.

- S'il y'a des symptômes en commun, alors il affiche les traitements nécessaires à partir du concept Therapy.

- Finalement, le résultat de ce diagnostique est de prendre le médicament «Sulfamethoxazole"

Conclusion :

Ce chapitre a été consacré à la présentation de l'approche proposée pour assurer un accès unifiée aux sources de données hétérogènes via notre système de médiation sémantique à base d'ontologie. Nous avons opté une architecture modulaire pour concevoir et implémenter ce système. De plus, nous avons donné pour chaque module l'algorithme concerné avec un exemple illustratif et une étude de deux types de requêtes (simple et composée).

Nous avons ajouté une fonction du système expert pour diagnostiquer les symptômes des malades ou d'un utilisateur veut connaitre les risques alimentaires.

Finalement, dans le prochain chapitre, nous allons implémenter et mettre en œuvre ce que nous avons proposé dans l'étude conceptuelle. En d'autres termes, la réalisation d'un système de médiation sémantique à base d'ontologie dédié au domaine des risques alimentaires.

Chapitre 5

« Mise en œuvre de système de médiation sémantique »

"I've failed over and over and over again in my life and that is why I succeed".

Michael Jordan

Introduction :

Nous abordons dans ce chapitre la mise en œuvre de notre système de médiation sémantique des sources de données hétérogènes. Dans ce contexte, nous nous appuyons sur, d'une part, l'étude conceptuelle que nous avons fait dans le chapitre 3 et de l'autre part, sur l'approche proposée à base d'ontologie pour assurer une intégration sémantique de qualité en chapitre 4.

L'implémentation de notre système est faite en utilisant le langage de programmation JAVA avec l'IDE Eclipse LUNA et le logiciel Protégé 2000 pour la construction des ontologies. Nous avons choisi le domaine des risques alimentaires pour valider notre travail.

Le présent chapitre décrit dans un premier temps les outils logiciels nécessaires pour la mise en œuvre et dans un second temps les détails d'implémentation.

Nous terminons ce chapitre par une évaluation de notre approche selon différentes mesures de similarité sémantique.

1. Les outils de développement utilisés :

Afin de mettre en place notre système de médiation sémantique à base d'ontologie, nous avons utilisé les outils logiciels suivants :

- Protégé 2000 version 4.3 : un environnement graphique de développement d'ontologies.
- Eclipse luna : l'environnement de développement fondé sur le langage Java.
- Wordnet 2.1 : une ontologie linguistique, est un dictionnaire des mots en anglais.
- Microsoft Picture Manager pour traiter la taille des images (agrandir, redimensionner,...).
- Microsoft Access : un SGBD des bases de données relationnelles.
- PhotoFiltre7 : logiciel de traitement d'images (éclairage, contraste, ...).

1.1.PROTÉGÉ-2000 :

Protégé est un éditeur d'ontologies distribué en open source par l'université en informatique médicale de Stanford. Protégé n'est un outil spécialement dédié à OWL, mais un éditeur hautement extensible, capable de manipuler des formats très divers. Le support d'OWL, comme de nombreux autres formats, est possible dans protégé grâce à un plugin dédié [38].

Protégé est un outil employé par les développeurs et des experts de domaine pour développer des systèmes basés sur les connaissances (knowledge).

Des applications développées avec Protégé sont employées dans la résolution des problèmes et la prise de décision dans un domaine particulier. Protégé est aussi une plate-forme extensible, grâce au système de plug-ins, qui permet de gérer des contenus multimédias, interroger, évaluer et fusionner des ontologies, etc.

L'outil Protégé possède une interface graphique (GUI) lui permettant de manipuler aisément tous les éléments d'une ontologie : classe, méta-classe, propriété, instance,...etc. Protégé peut être utilisé dans n'importe quel domaine où les concepts peuvent être modélisés en une hiérarchie des classes [38].

Protégé permet aussi de créer ou d'importer des ontologies écrites dans les différents langages d'ontologies tel que : RDF-Schéma, OWL, DAML, OIL, ...etc. Cela est rendu possible grâce à l'utilisation de plugins qui sont disponibles en téléchargement pour la plupart de ces langages.

La figure suivante représente l'interface principale de protégé 2000. Trois onglets sont nécessaires [38] :

1. L'onglet « OWLClasses » permet de voir et d'éditer des classes et leurs propriétés.
2. L'onglet « Properties » permet d'éditer toutes les propriétés.
3. L'onglet « Individuals » permet d'éditer les instances des classes.

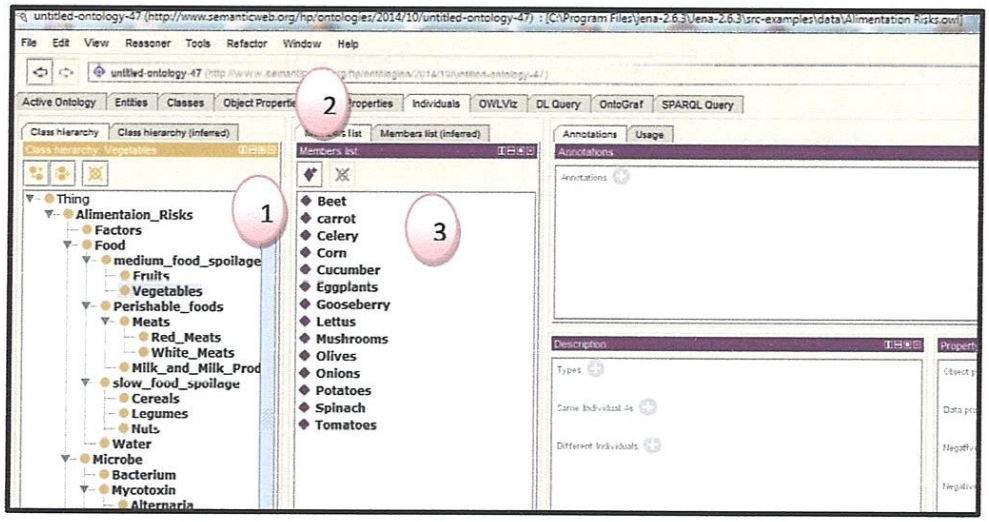


***Figure 5.1** :Interface du protégé 2000.*

1.2.Microsoft Access 2003 :

Microsoft Office Access (Ms Access) est un Système de Gestion de Bases de Données (SGBD) relationnel édité par Microsoft. Il fait partie de la suite burcautiquc.

Ms Access est composé de plusieurs programmes: le moteur de base de données Microsoft Jet, un éditeur graphique, une interface de type Query By Exemple pour manipuler, traiter les ensembles de données formant la base [Web5].

Ms Access est un logiciel utilisant des fichiers au format Access (extension de fichier *mdb* pour Microsoft DataBase (extension *.accdb depuis la version 2007)). Il est compatible avec les requêtes SQL (sous certaines restrictions) et dispose d'une interface graphique pour saisir les requêtes. Il permet aussi de configurer, avec des assistants ou

librement, des formulaires et sous-formulaires de saisie, des états imprimables (avec regroupements de données selon divers critères et des totalisations, sous-totalisations, conditionnelles ou non), des pages html liées aux données d'une base, des macros et des modules VBA.

D'un point de vue concret Access (avec ses versions 2000 à 2003) convient bien à des applications faisant intervenir jusqu'à une centaine de tables avec un maximum pratique de 100 000 enregistrements pour les tables principales et de 1 000 000 d'enregistrements pour les tables de jointures (appelées aussi tables de liaisons ou de relations). En pratique la taille maximum d'une base Access 2003 est de 2Go.

Les bases de données produites par Access restent accessibles à tous les langages de programmation qui permettent une connexion à une base ODBC, c'est le cas par exemple sous Java en se servant de la passerelle JDBC-ODBC d'Access [Web5].

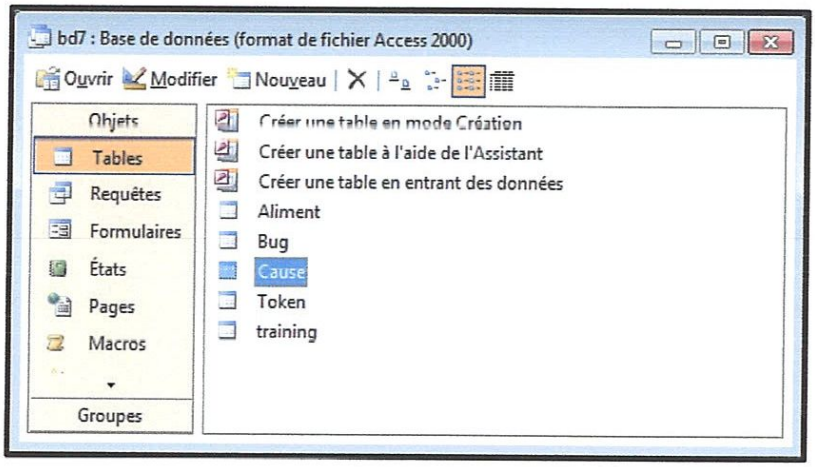


***Figure 5.2:** interface de l'Access 2003.*

1.3.Eclipse Luna :

Nous avons choisi le langage de programmation orienté objet JAVA pour implémenter notre travail. La particularité centrale de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que : UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plates-formes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java [Web6].

Plusieurs environnements de développement ont été apparus pour faciliter la programmation java (Netbeans, eclipse, ..). Dans le cadre de notre travail nous avons utilisé l'Eclipse Luna.

Éclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques. Éclipse utilise énormément le concept de modules nommés « plug-ins » dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé, tout le reste de la plate-forme est développé sous la forme de plug-ins.

Les principaux modules fournis en standard avec Éclipse concernent Java, mais d'autres modules sont développés pour d'autres langages notamment C++, **PHP**, **JavaScript**, etc [41].

Eclipse Luna (June 25, 2014) : Inclut le support officiel pour Java ™ 8 dans les outils de développement Java, Plug-in Outils de développement, les équipes de l'objet, framework Eclipse Communication, intégration Maven, xtext, Xtend, Web Tools Platform et Memory Analyzer. Le compilateur Eclipse comprend des améliorations linguistiques, la recherche et la refactorisation [Web7].

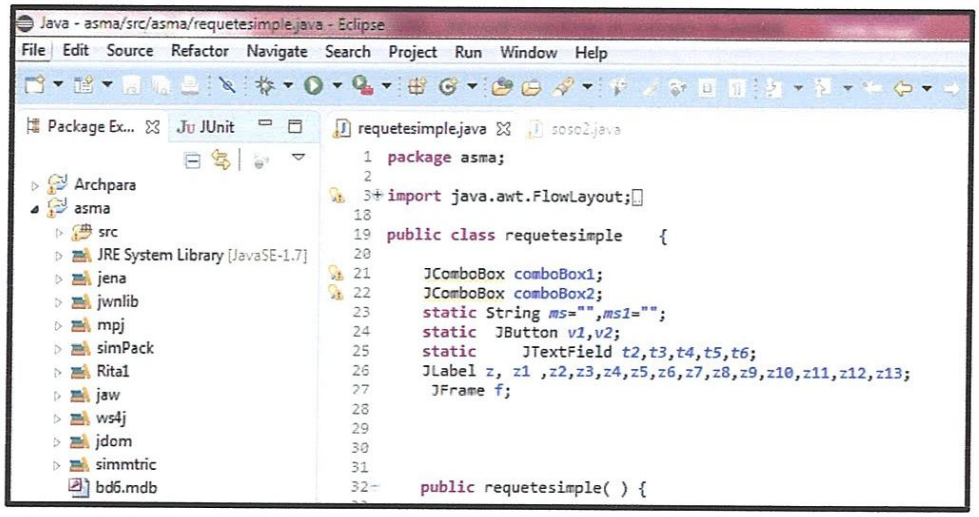


Figure 5.3:*interface de l'Eclipse Luna.*

1.4.Wordnet :

Wordnet (Miller, 1995) est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise.

Le système se présente sous la forme d'une base de données électronique qu'on peut télécharger sur un système local. Des interfaces de programmation sont disponibles pour de nombreux langages [42].

Informations manquantes Wordnet ne précise pas l'étymologie, la prononciation, les formes de verbes irréguliers et ne contient que des informations limitées sur l'usage des mots. La contrepartie de son importante couverture est que Wordnet est très précis dans le sens des définitions, par exemple, le verbe to give (« donner ») n'a pas moins de 44 sens. Une telle profusion ne facilite pas une tâche de désambiguïsation lexicale. Wordnet reste l'une des ressources de TAL les plus populaires [42].

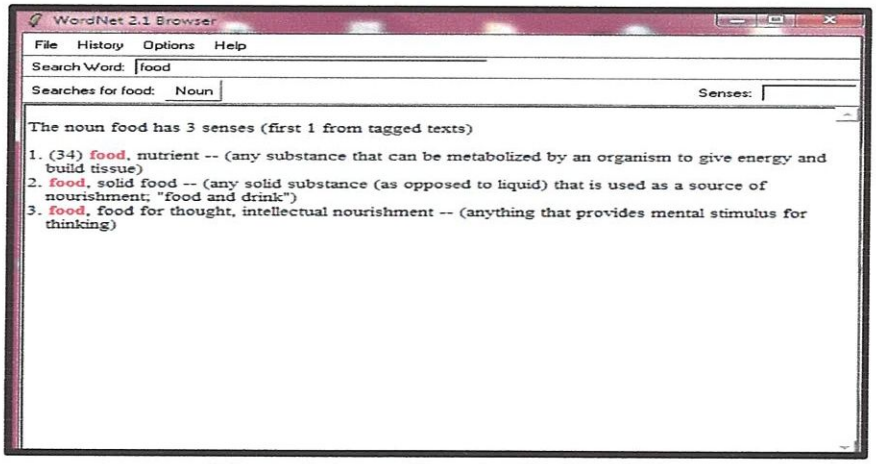


Figure 5.4:*interface du Wordnet 2.1.*

1.5.PhotoFiltre7 :

PhotoFiltre7 est un logiciel de retouche d'images très complet. Il permet d'effectuer des réglages simples ou avancés sur une image et de lui appliquer un large éventail de filtres. Son utilisation simple et intuitive offre une prise en main rapide. La barre d'outils, proposant l'accès aux filtres standards par simple clic de souris, lui donne un côté convivial.

PhotoFiltre7 possède également un gestionnaire de calques (avec couche Alpha), des pinceaux personnalisables, un module d'automatisation et des tas d'autres outils puissants [web11].

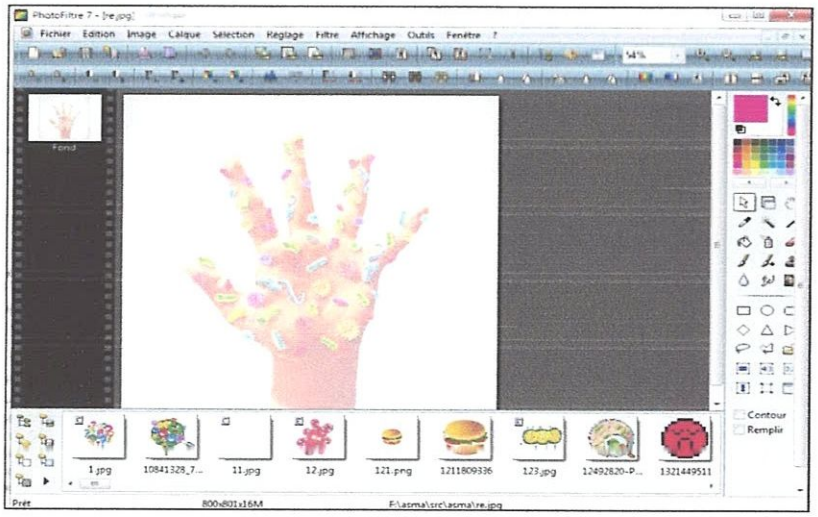


Figure 5.5:*interface du PhotoFiltre7.*

1.6.Microsoft Picture Manager :

Microsoft Picture Manager est un logiciel de la famille Microsoft office, permettant de gérer, modifier, partager et visualiser des images à partir de l'emplacement où elles sont stockées.

individus. Pour les petites ontologies, il est facile de comprendre leur structure et contenu mais, dans les grandes ontologies rendent leur compréhension une tâche fastidieuse. Des packages ou plu gins sont développés pour faciliter la visualisation des ontologies sous formes des graphes, par exemple : Graphviz, OntoGraf, Jambalaya.

OWLViz est une partie de la distribution Protégé-OWL qui se base sur Graphviz. L'installation de Graphviz sera probablement nécessaire. Après l'installation, le chemin de l'application Dot (de Graphviz) doit être réglé en conséquence; configuration se trouve sur l'onglet OWLViz dans le dialogue des préférences Protégé ('Fichier → Préférences).

OWLViz visualise la structure de classe d'une ontologie comme graphe orienté où bords représentent parent-enfant (is-a) relations. Le plug-in ne tire pas la représentation des propriétés ou des individus. La mise en page du graphique généré peut être configurée mais les nœuds ne peuvent pas être déplacés [39].

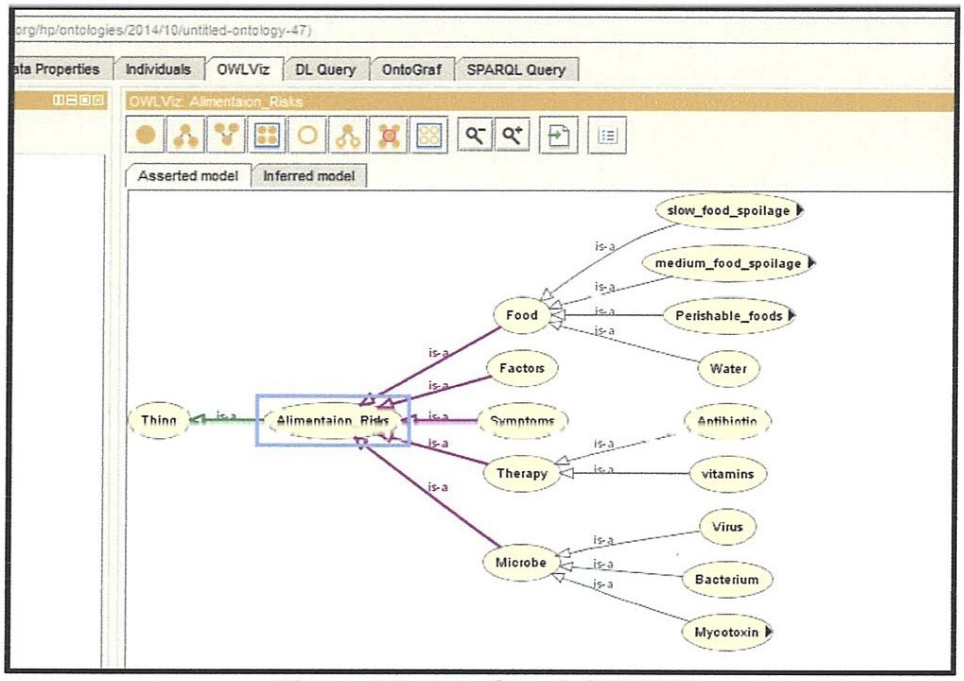


Figure 5.7 : *interface de l'OWLVis.*

2.2.Jambalaya 2.7.1 :

Jambalaya se compose de nombreux outils différents pour l'affichage d'une ontologie. Il fournit une Treemap vue qui est très utile pour acquérir un sens de la structure globale d'une ontologie. Par exemple, lorsque la grille est dimensionnée par nombre d'enfants, les concepts avec sous-concepts plus seront plus grands, et plus facile à voir [40].

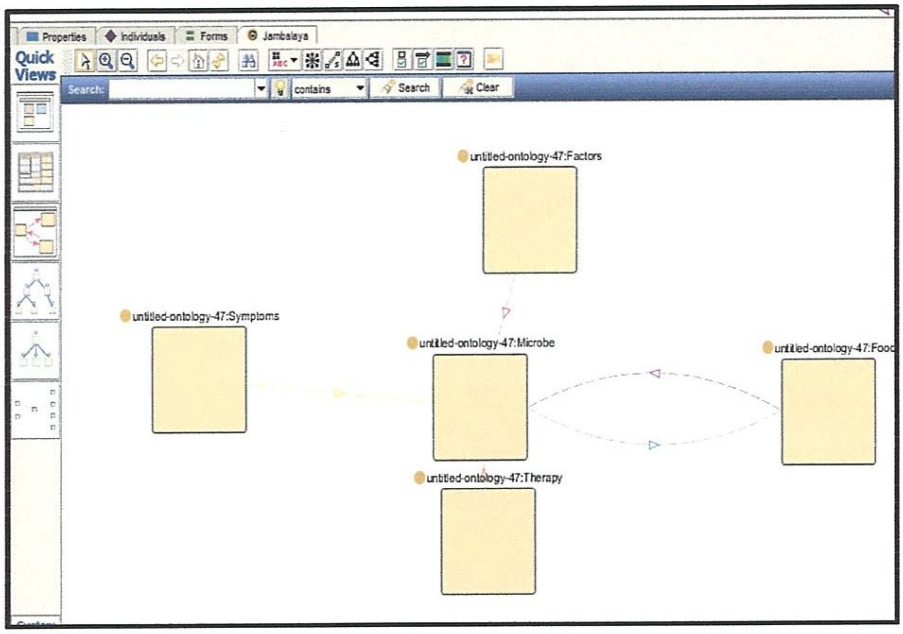


Figure 5.8 : interface du Jambalaya.

2.3.Jena:

Jena est une bibliothèque de classes Java développée par HP qui facilite le développement d'applications pour le web sémantique.

-Manipulation de déclarations RDF.
-Lecture et écriture RDF/XML, Notation
-Stockage en mémoire ou sur disque de connaissances RDF.
- Jena est fourni avec un interpréteur SPARQL qui pouvant être utilisé en ligne de commande.
-Gestion d'ontologies : RDF-Schema, OWL.

Jena peut être téléchargé à partir du site http://jena.apache.org (Apache jena - Binary distribution). Tous les fichiers jar nécessaires à la compilation et à l'utilisation de Jena sont Dans lib. Ils doivent tous être présents dans le CLASSPATH pour la compilation ou l'exécution de programmes utilisant Jena [Web8].

2.4.WS4J

WS4J (Wordnet similarity for Java) fournit une API Java pur pour plusieurs types de similarité sémantiques. [Web9]

La liste des Algorithmes:

ID	publication	Description
HSO	(Hirst & St-Onge, 1998)	Deux concepts lexicalisés sont sémantiquement similaires si les synsets sont reliés par un chemin qui n' est pas trop long et que "ne change pas de direction".
LCH	(Leacock&Chodorow, 1998)	S'appuie sur la longueur du plus court chemin entre deux synsets.
LESK	(Banerjee & Pedersen, 2002)	Lesk (1985) a proposé que le degré de similarité entre de deux mots est proportionnelle à l'étendue de chevauchement de leurs définitions du dictionnaire. Banerjee et Pedersen (2002) ont étendu cette notion par le calcul de similarité entre synset.
WUP	(Wu & Palmer, 1994)	Détail dans le chapitre 2.
RES	(Resnik, 1995)	Détail dans le chapitre 2.
JCN	(Jiang & Conrath, 1997)	Détail dans le chapitre 2.
LIN	(Lin, 1998)	Détail dans le chapitre 2.

Tableau 5.1 : *La liste des Algorithmes de WS4J (Wordnet similarity for Java).*

2.5.JFreeChart 1.0.19 :

JFreeChart est une bibliothèque open source qui permettent d'afficher des données statistiques sous la forme de graphiques. Elle possède plusieurs formats : les barres ou les lignes et propose de nombreuses options de configuration pour personnaliser le rendu des graphiques. Elle peut s'utiliser dans des applications ou des applets et elle permet également d'exporter le graphique sous la forme d'une image.

Pour l'utiliser, il faut télécharger le fichier jfreechart-0.1.19.zip et le décompresser. Son utilisation nécessite l'ajout dans le classpath des fichiers jfreechart-0.1.19.zip et des fichiers .jar présents dans le répertoire lib décompressé [web10].

3. Architecture générale de MS4AR :

Notre système de médiation sémantique sur le domaine des risques alimentaires baptisé **MS4AR (Mediation System for Alimentary Risks)** a été planifié selon l'architecture suivante: L'architecture générale illustrée dans la figure ci-dessous, montre les principaux composants de notre système **MS4AR.** Le premier composant représente l'interface principale du système de médiation sémantique dans laquelle on effectuant la formulation des requêtes et la représentation des résultats.

Ce composant offre à l'utilisateur la possibilité d'interroger plusieurs sources hétérogènes à partir de l'ontologie globale ONTARIS (ONTology for Alimentation RISks). De ce fait,

l'utilisateur peut choisir l'une des options suivantes : *Simple Query* (requête atomique ou simple), *Compound Query* (requête composée).

Trois options supplémentaires qui sont: *Expert System* (système expert) qui permet à l'utilisateur de diagnostiquer les symptômes d'intoxication, *Statistics* qui donne les statistiques d'utilisation du système. *Conceptuel model* qui représente la technique de fragmentation automatique utilisée pour la création des sources locales (bases de données locales).

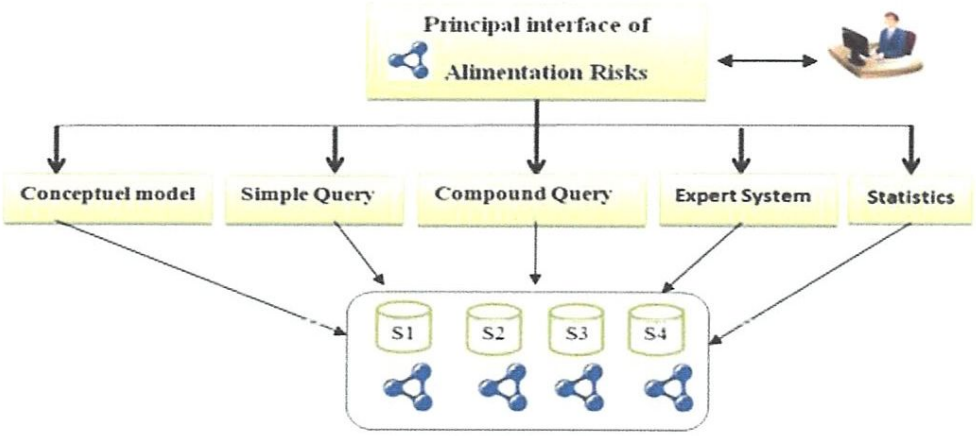


Figure 5.9 : *Architecture générale de l'application.*

4. Les interfaces de MS4AR :

Notre système MS4AR est implémenté avec quatre (4) sources de données hétérogènes concernant le domaine des risques alimentaires. Le MS4AR est composé d'un certain nombre d'interfaces organisées. La fenêtre principale du système est apparue après le lancement de l'écran de démarrage (splash screen) qui illustre le but du système (voir la figure 5.10).

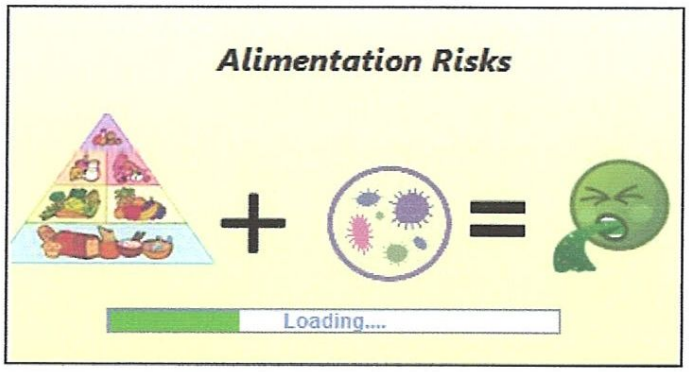


Figure 510 : Ecran de démarrage de MS4AR.

La fenêtre principale décrit l'interface d'accès unifiée aux sources hétérogènes. Elle dispose quatre items suivants: simple Query, compound Query, expert system et Statistics. Dans la section suivante, nous présentons les scénarios de l'exécution de ces items.

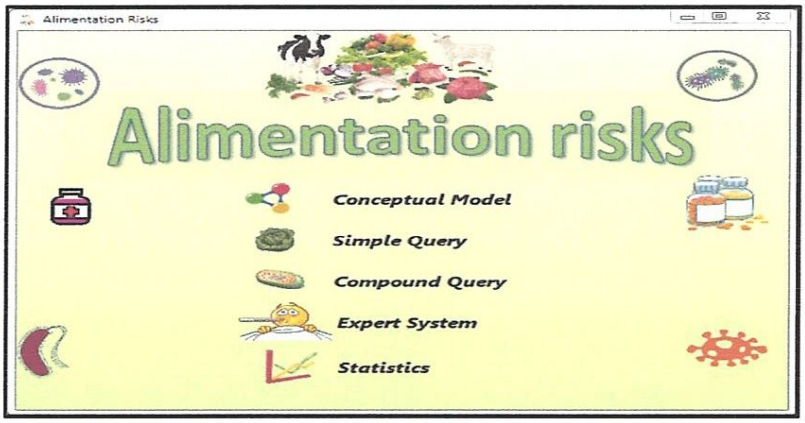


Figure 5.11 *: Fenêtre d'accueil de l'application.*

Par ailleurs, nous avons implémenté une interface graphique supplémentaire (figure 12) qui montre aux concepteurs de logiciels comment on crée automatiquement les sources locales (bases de données relationnelles).

A partir de l'étude conceptuelle que nous avons faite dans le chapitre 3 concernant la création automatique des sources hétérogènes, nous présentons dans ce qui suit son implémentation JAVA. Cette interface est invisible par l'utilisateur final. Nous nous expliquons cette partie seulement pour les utilisateurs type (informaticien, concepteur des logiciels, enseignant en informatique, programmeur,…).

L'interface nommée « **Conceptuel model** » dispose trois options:
-création de la base globale à partir de l'ontologie globale.
-fragmentation automatique de la base globale (fragmentation horizontale et verticale) en quatre bases locales.
-finalement la possibilité de supprimer tous les bases locales et la base globale par la commande Drop.

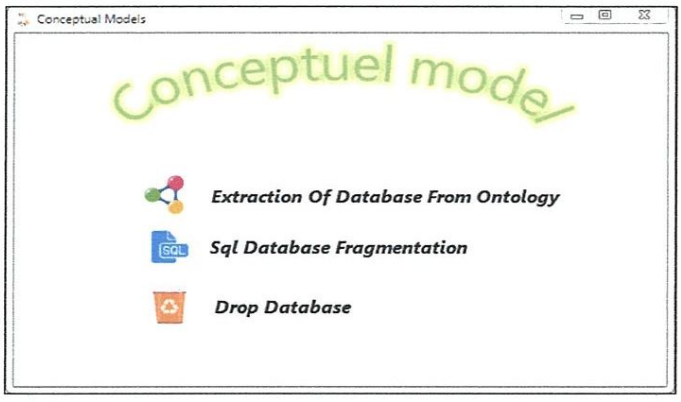


Figure 5.12 : *Fenêtre de la conception.*

Un extrait de code pour le concept food de l'ontologie globale ONTARIS sur les risques alimentaires est le suivant :

```
String q=PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX lo :<http://www.semanticweb.org/hp/ontologies/2014/10/untitled-ontology-47#>
SELECT ?X    WHERE { ?X rdfs:subClassOflo:Food }
Query qo8 = QueryFactory.create(q)
QueryExecution qp8 = QueryExecutionFactory.create(qo8, model);
```

4.1.Interface d'interrogation via Simple Query :

L'interrogation via Simple Query permet de faire des interrogations simples sur un concept donné par une simple sélection des combobox sur lesquels va porter la recherche.

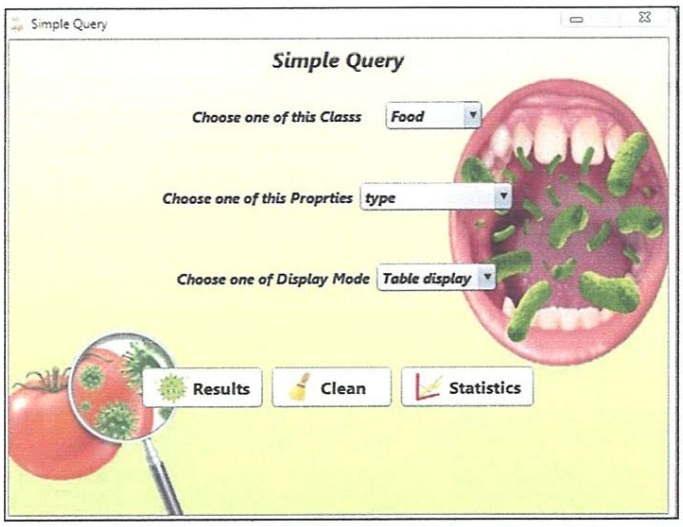
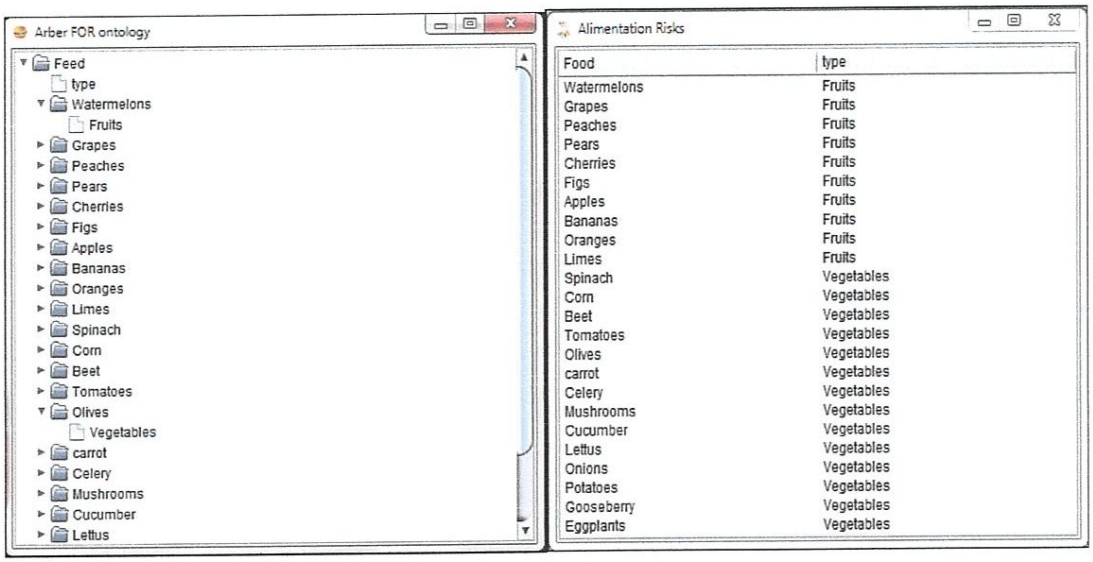


Figure 5.13 *: Exemple d'interrogation via* Simple Query.

Dans la figure ci-dessus, l'utilisateur sélectionne un concept (ou classe) de l'ontologie globale par exemple food, puis l'une des propriétés du concept choisi (dans notre exemple la propriété Type).

De plus, l'interface offre à utilisateur deux modes de visualisation de résultat : visualisation arborescente (partie (a) de figure 14) et autre sous forme d'un tableau (partie (b) de figure 14).



Partie (a) ***Partie (b)***

***Figure 5.14**: Deux modes de visualisation de résultats.*

Le traitement des problèmes d'hétérogénéité est effectué en mesurons les similarités de wu palmer du Wordnet et de cosinus.

L'extrait de résultat de calcul de la similarité de wu palmer à base de depth (profondeur) est:

```
Trying to find a relationship between "microbe" and
"organism".
Looking for relationship of type hypernym.
The depth of this relationship is: 2
Here is how the words are related:
Start: microbe
0: microbe, bug, germ
1: microorganism, micro-organism
2: organism, being
Trying to find a relationship between "microbe" and
"organism".
Looking for relationship of type hyponym.
I could not find a relationship between these words!
```

4.2.Interface d'interrogation via Compound Query :

Le principe d'interrogation via compound query (requete composée) est le même que l'interrogation simple en se basant sur la sélection des critères de recherche plus les relations sémantiques entre concepts.

Dans ce mode d'interrogation, le traitement d'hétérogénéité est réalisé non seulement via le Wordnet mais aussi en faisant appel aux ontologies locales propres à chaque sources de données. Par conséquence, un matching entre l'ontologie globale et ontologie locale de la source à intégrer est nécessaire pour trouver le lien sémantique entre les quatre sources. Cette

mise en correspondance entre ontologie est assurée par la fonction de l'adaptateur de chaque source.

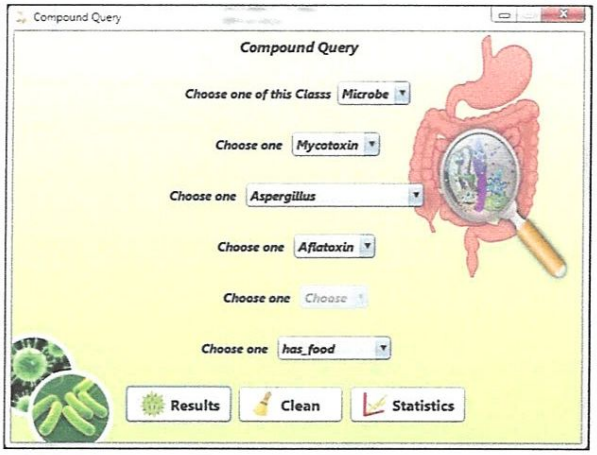


***Figure 5.15** : Exemple d'interrogation via Compound Query.*

Exemple :

Pour trouver les aliments avec lesquels la Bactérie aflatoxine est existée.
L'utilisateur sélectionne dans un premier temps le concept Microbe puis il cherche dans son sous concept Mycotoxine et ensuite il sélectionne la relation sémantique Has_food .

L'exécution de cette requête est effectuée via notre approche que nous avons la détaillée dans le chapitre précédent. En grosso modo, le traitement d'une telle requête implique l'application de processus d'alignement des ontologies. Nous avons fixé quatre types de matching :

1. ***Concept matching :*** pour chercher les concepts de l'ontologie locale proche aux concepts l'ontologie globale :

```
WuPalmer : Bug - Mycotoxin = 0. 87557142857142857
WuPalmer : Bug - Alternaria = 0.0
WuPalmer : Bug - Symptom = 0.2
WuPalmer : Bug - Therapy = 0.16666666666666666
WuPalmer : Bug - Alimentaion_Risks = 0.0
WuPalmer : Bug - Nuts = 0.0
WuPalmer : Bug - Food = 0.33333333333333330.875
0.875----> Mycotoxin
[CosineSimilarit] Aspergill and Aspergillus are 0.9341987329938275% similar
[CosineSimilarit] Aspergill and Penicillium are 0.5533715710928597% similar
[CosineSimilarit] Aspergill and Perishable_foods are 0.5142594772265799% similar
[CosineSimilarit] Aspergill and medium_food_spoilage are 0.45226701686664544% similar
0.93----> Aspergillus
```

2. ***Concept structural matching :*** à partir de résultat de concept matching, nous calculons maintenant la correspondance au niveau de la structure de l'individu aflatoxine. Plusieurs étapes ont été faites, pour plus de détail consulter le chapitre 4. Le tableau suivant illustre le résultat de ce type de matching.

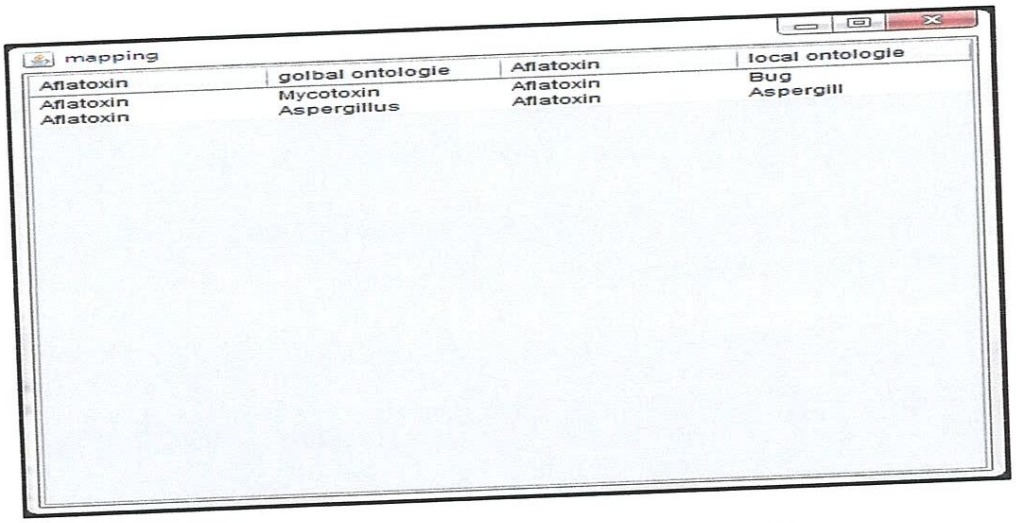


***Figure 5.16**: Résultat de concept_Structural matching.*

3. Data type matching : consiste à déterminer si l'individu aflatoxine ayant les mêmes valeurs de data type. La figure suivante décrit le résultat.

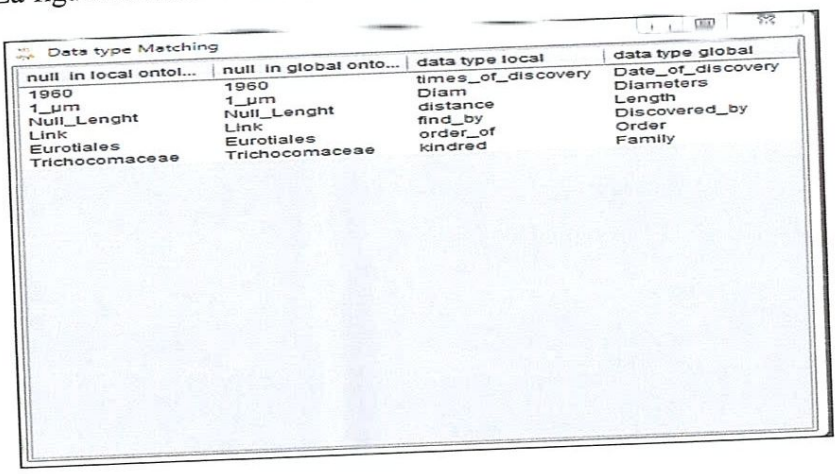


***Figure 5. 17 :** Data type matching.*

4. *Data object matching* : consiste à calculer la similarité entre les data Object dans l'ontologie locale et ceux dans l'ontologie globale. Dans ce qui suit l'extrait de résultat.

```
WuPalmer  : has_food -  hasAliment = 0.9230769230769231
WuPalmer  : has_food -  hasToken = 0.3076923076923077
WuPalmer  : has_food -  hastraining = 0.2857142857142857
WuPalmer  : has_food -  hasBug = 0.3333333333333333
WuPalmer  : has_food -  hasCause = 0.26666666666666666
0.375---->hasAliment
```

Et finalement, le résultat de l'alignement entre l'ontologie partagée et les ontologies locales des sources les plus pertinentes est représenté dans le tableau suivant :

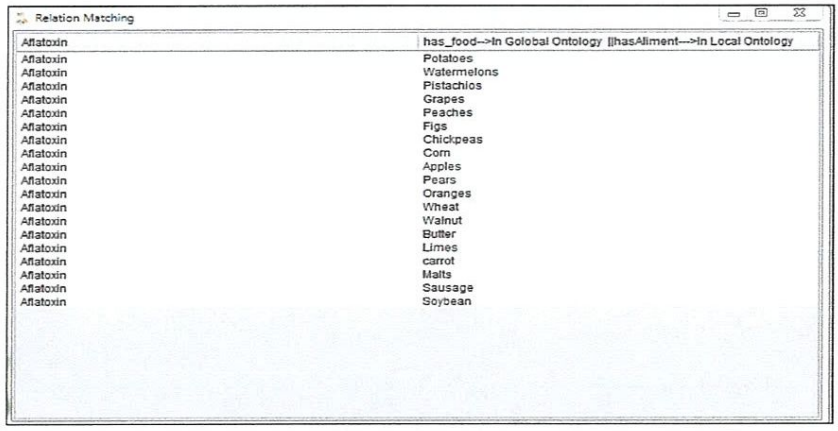


***Figure 5.18**: Relation matching.*

4.3.Interface d'Expert System:

Nous avons ajouté une fonction supplémentaire à notre travail pour représenter l'intérêt et le bon fonctionnement de système de médiation sémantique. Cette fonction permet à l'utilisateur de diagnostiquer les maladies et la proposition de traitement nécessaire. L'interface de cette fonction représente le fonctionnement d'un système expert, si l'utilisateur est infecté par une intoxication alimentaire lui donner les traitements nécessaires.

Selon la figure suivante, l'utilisateur effectue les étapes suivantes :

1. L'utilisateur sélectionne l'aliment qu'il mange dans la dernière 24 heures qui est Sausage de sous concept Meats de concept Perishable food.
2. il indique les symptômes d'intoxication. Dans cet exemple on a : Spasm
3. Affichage de résultats via le bouton Results. Ces résultats représentent la bactérie lie à cette intoxication (c'est Salmonella) et les traitements nécessaires à prendre qui sont Trimethoprim, Sulfamethaxazole et Ampieillin.

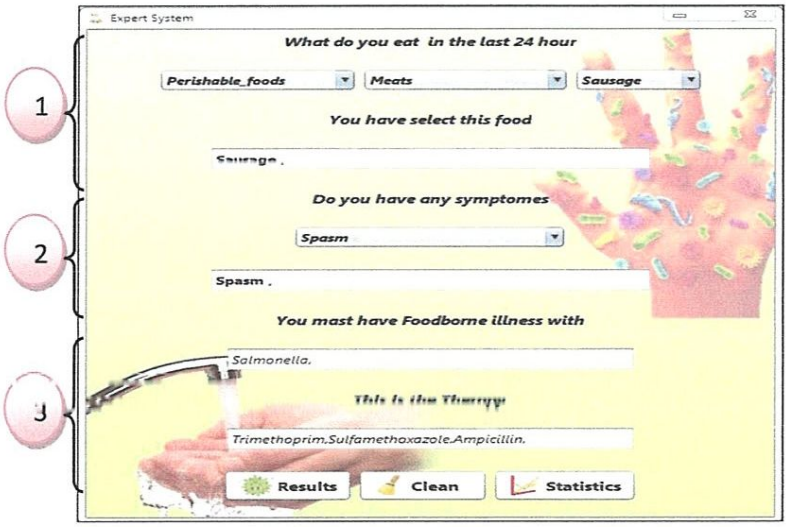


***Figure 5.19**: Interface d'Expert System.*

4.4.Interface de Statistic :

Cette interface offre à l'utilisateur la possibilité de connaitre les statistiques d'utilisation du système en termes de mesure de similarité utilisée.

Pour chaque itération nous gardons le nombre d'utilisation des similarités du wu Palmer et de cosinus. Dans la figure ci-dessous, nous avons accédé à notre système de médiation à travers trois requêtes suivantes : i) name of Food, ii) type of Food et iii) Binomial-name of food.

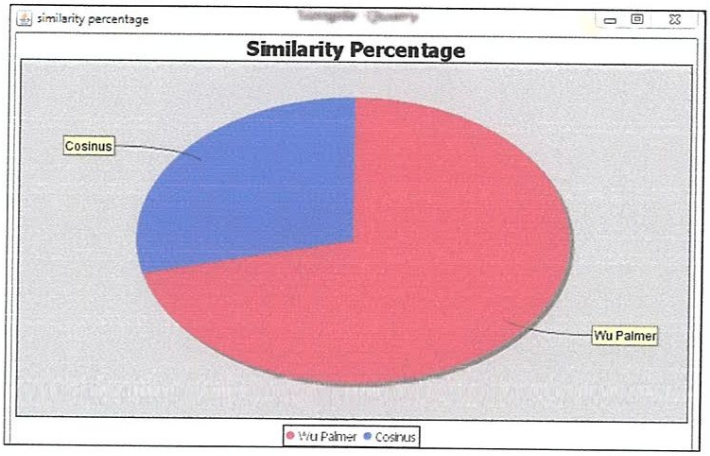


***Figure 5.20**: Interface des* Statistique.

On remarque que le taux d'utilisation de similarité de palmer est plus grand que celui de cosinus. Ce résultat est justifié par le fait que la similarité de Palmer est appliquée dans le cas où le nom des attributs d'un concept est atomique c.à.d. il contient un seul mot (type, name,...), par contre la similarité vectorielle de cosinus s'applique dans le cas contraire (le nom composé des attributs), par exemple: Binomial-name, binomial-terme, Binomial_call ...etc.

5. Evaluation de performance :

Pour évaluer et comparer les systèmes informatiques, on utilise les critères de performance habituels, c'est-à-dire les mesures du temps de réponse et d'espace mémoire utilisé: plus le temps de réponse est court et l'espace occupé par le système est faible, meilleur est le système. Cependant, d'autres mesures de performances ont été introduites, dans le but d'évaluer l'efficacité de système.

Parmi elles, nous pouvons citer :

- la capacité du système à atteindre ses objectifs.
- Comparaison des mesures de similarité utilisée avec d'autres mesures.
- Comparaison avec des approches similaires selon plusieurs critères.

Afin de correctement illustrer le comportement de notre approche, nous distinguons deux cas d'évaluation :

1. Evaluation des mesures de similarités
2. Evaluation par rapport les approches existantes.

5.1.Evaluation des mesures de similarités :

Dans cette évaluation, nous effectuons une comparaison des mesures de similarités existantes par rapport à notre choix de métriques. Dans le cadre de notre travail, nous avons utilisé deux types de similarité ; la première de Palmer qui calcule la similarité entre deux concepts atomiques (contient un seul mot) et la deuxième de Cosinus pour les concepts complexes.

5.1.1. Comparaison avec la similarité de Wu palmer :

Nous avons effectué la comparaison de similarité de Wu palmer avec deux autres mesures suivantes : similarité de Path et similarités de JiangConrath. Le choix de ces deux similarités est justifié par le fait qu'elles ont basé sur le calcul du depth (**Approches basées sur les arcs**) et leur valeur est comprise entre 0 et 1.

Le tableau suivant donne le résultat d'exécution avec trois requêtes.

Requêtes	Wu palmer	path	JiangConrath
Food - Feed	0.92	0.5	0.40
microbe - bug	0.66	0.125	0.0
Factor-Factorize	0.72	0.25	0.0

La représentation graphique de résultat est la suivante :

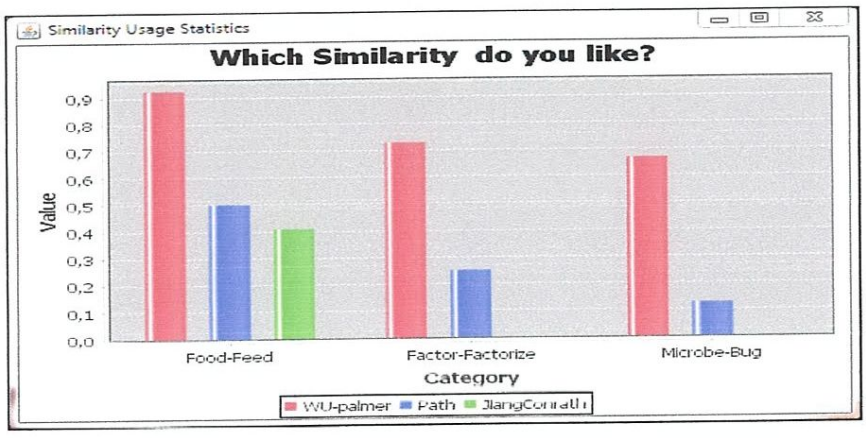


***Figure 5.21**: Evaluation des similarités (Wu palmer, Path, `JiangConrath`,).*

A partir des résultats obtenus, nous remarquons que la similarité de Wu palmer donne un meilleur résultat par rapport de celui obtenu par Path et JiangConrath. Notamment, en ce qui

concerne la similarité entre les concepts microbe et bug, on trouve que Sim $_{palmer}$ (microbe, bug) =0.66 par contre avec path est de 0.125 et une valeur nulle avec JiangConrath et aussi que Sim $_{palmer}$ (Factor-Factorize) =0.72 par contre avec path est de 0. 25 et une valeur nulle avec JiangConrath. Ces deux similarités donnent des valeurs parfois loin de réalité. La comparaison des résultats permet de montrer globalement l'apport de similarité de Palmer à notre travail par rapport aux autres mesures.

5.1.2. Comparaison avec la similarité de Cosinus :

Nous avons effectué la comparaison de similarité de CosineSimilarit avec deux autres mesures suivantes : similarité de SorensenSimilarit et similarité de JaccardSimilarit. Le choix de ces deux similarités est justifié par le fait qu'elles ont basé sur l'espace vectoriel et leur valeur est comprise entre 0 et 1.

Le tableau suivant donne le résultat d'exécution avec trois requêtes:

Les mots	CosineSimilarit	JaccardSimilarit	SorensenSimilarit
Wh_Meat- White_Meats	**0.87**	**0.77**	**0.77**
Veggie-Vegetables	**0.71**	**0.33**	**0.37**
Binomial_call -Binomial_name	**0.819**	**0.8**	**0.61**

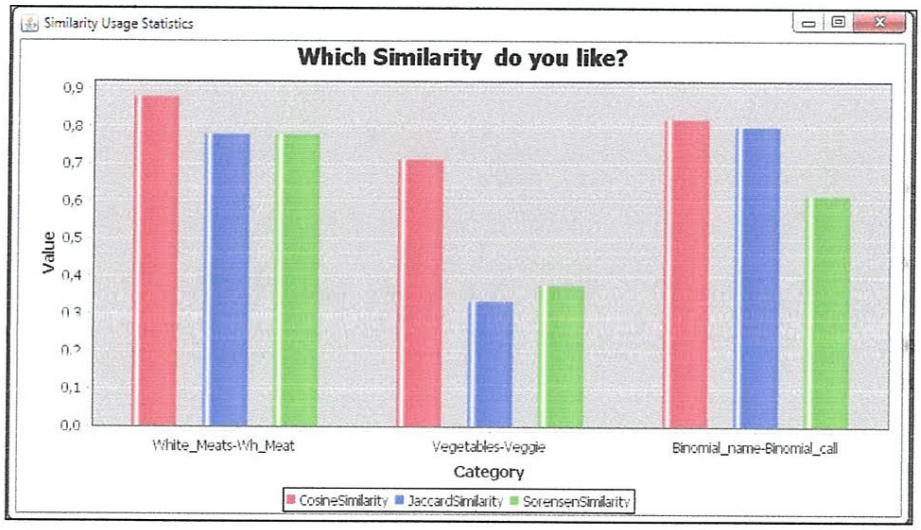


Figure 5.22: Evaluation des similarités (CosineSimilarity, JaccardSimilarity, SorensenSimilarity).

A partir des résultats obtenus, nous remarquons que la similarité de CosineSimilarit donne un meilleur résultat par rapport de celui obtenu par JaccardSimilarit et SorensenSimilarit. Notamment, en ce qui concerne la similarité entre les concepts Vegetables et Veggie, on trouve que Sim $_{cosinus}$ (Vegetables, Veggie) =0.71 par contre avec JaccardSimilarit est de 0.33 et avec SorensenSimilarit 0.37. Ces deux similarités donnent des valeurs parfois loin de

réalité. La comparaison des résultats permet de montrer globalement l'apport de similarité de cosinus à notre travail par rapport aux autres mesures.

5.2.Comparaison avec les approches existante :

Afin de montrer l'originalité de notre travail par rapport les travaux similaires pour le traitement sémantique de différents conflits d'hétérogénéités des sources de données, nous avons réalisé une étude comparative avec trois travaux suivants :

5.2.1. Comparaison avec le Travail de Fatiha SAÄIS :

	Notre travail	Travail de Fatiha SAÄIS
Domaine	Risque alimentaire	Risque alimentaire
Schéma global	Ontologie de domaine	Ontologie de domaine
Sources locales	Des bases de données (obtenir à partir de la fragmentation horizontale et verticale de la base de donnés globale)	des tableaux (ensemble de lignes et de colonnes) XML
Dictionnaire sémantique	Wordnet	-créer son propre dictionnaire -SML (Semantic Markup Language)
Type d'hétérogénéité	- Informations incomplètes et erronées. -abréviation d'une information. -synonyme d'une information.	-Informations incomplètes et erronées. -Informations ambiguës -Informations non identifiables
Système d'intégration	System de médiation sémantique	Entrepôts de Données
Utilisation de l'ontologie	Suivant l'architecture Hybride	Une Seule ontologie
Alignement de l'ontologie	quatre techniques d'alignement	Aucune
similarité sémantique	wu palmer et de cosinus	Jaro-Winkler utilisation des clusters
Les algorithmes	-Des algorithmes au niveau de médiateur -Des algorithmes au niveau des adaptateurs -Des algorithmes au niveau des sources locaux -algorithme pour la requête simple -algorithme pour la requête complexe -algorithme pour système expert	- Algorithme d'enrichissement sémantique de tableaux. - Algorithme d'enrichissement sémantique d'informations structurées -Algorithme de réconciliation de références fondé sur la résolution. -Algorithme de calcul itératif de la similarité des références
Langage de programmation	java	Delphi

Tableau 5.2 : *Comparaison entre notre approche et l'approche de Fatiha SAÄIS*

5.2.2. Comparaison avec le travail de Fatima LAHMAR:

	Notre travail	**Travail de Fatima LAHMAR**
Domaine	Risque alimentaire	Cinéma
Schémas globale	Ontologie de domaine	Rien
Sources locales	Des bases de données (obtenir à partir de la fragmentation horizontale et verticale de la base de donnés globale)	Des sources (2 Schéma relationnel, 2 Schéma Objet 2 Schéma XML)
Dictionnaire sémantique	Wordnet	N'est pas précisé
Type d'hétérogénéité	-Informations incomplètes et erronées. -abréviation d'une information. -synonyme d'une information.	N'est pas précisé
Système d'intégration	System de médiation sémantique	Entrepôt de donnés
Approche d'intégration	LAV	GLAV
Architecture logicielle	-Module Interface -Module médiateur -Module adaptateur - Module de Sources de Données (4 bases de données hétérogène)	- module Base de données de l'entrepôt -Module extracteur -Module traitement - Module de Sources de Données (6 bases de données)
Utilisation de l'ontologie	Suivant l'architecture Hybride	Rien
Alignement de l'ontologie	quatre techniques d'alignement	Aucune
Similarité sémantique	wu palmer et de cosinus	N'est pas précisé
Algorithmes	-Des algorithmes au niveau de médiateur -Des algorithmes au niveau des adaptateurs -Des algorithmes au niveau des sources locaux -algorithme pour la requête simple -algorithme pour la requête complexe -algorithme pour système expert	Algorithmes de Traitement de requêtes dans GLAV
Langage de programmation	java	C

Tableau 5.3 : *Comparaison entre notre approche et l'approche de Lahmar Fatima.*

5.2.3. Comparaison avec le travail de Amel BOUSSIS:

	Notre travail	**Travail de Amel BOUSSIS**
Domaine	Risque alimentaire	sécurité sociale d'une Usine
Schémas globale	Ontologie de domaine	Ontologie de domaine
Sources locales	Des bases de données (obtenir à partir de la fragmentation horizontale et verticale de la base de donnés globale)	Des bases de données (obtenir à partir de la fragmentation verticale de l'ontologie globale)
Dictionnaire sémantique	Wordnet	Rien
Type d'hétérogénéité	-Informations incomplètes et erronées. -abréviation d'une information. -synonyme d'une information.	Rien à mentionner
Système d'intégration et	System de médiation sémantique	System de médiation
Approche d'intégration	Lav	GaV
Architecture logicielle	-Module Interface -Module médiateur -Module adaptateur - Module de Sources de Données (4 bases de données hétérogène)	-Module Interface -Module médiateur - Module de Sources de Données (3 bases de données)
Utilisation de l'ontologie	Architecture Hybrides	Une Seule ontologie
Alignement de l'ontologie	Quatre techniques d'alignement	Rien
Similarité sémantique	wu palmer et de cosinus	Rien
Les algorithmes	-Des algorithmes au niveau de médiateur -Des algorithmes au niveau des adaptateurs -Des algorithmes au nivcau des sources locaux -algorithme pour la requête simple -algorithme pour la requête complexe -algorithme pour système expert	-Algorithme pour la décomposition de la requête, -Algorithme pour la localisation de données, -Algorithme pour la réécriture de requêtes, -algorithme pour la reconstitution de résultats.
Langage de programmation	java	PHP

Tableau 5.4 : *Comparaison entre notre approche et l'approche de Amel BOUSSIS.*

5.3.Discussion de résultats :

Nous avons évalué notre travail par une étude comparative selon deux cas : les mesures de similarité sémantique et les travaux similaires. Le but du premier cas de comparaison étant d'étudier dans quelle mesure notre choix donne un meilleur appariement entre concepts. Alors, nous déduisons que la similarité de Wu palmer est pertinente pour des concepts atomiques par rapport les autres mesures basées arc, de plus, la mesure cosinus est utile pour des concepts complexes.

Par ailleurs, le second cas de comparaison avec trois travaux similaires selon un certain nombre des critères. Les résultats préliminaires montrent d'une part, l'originalité de notre approche par rapport les travaux précédents et d'autre part, l'impact de notre méthode d'alignement entre ontologies pour traiter les problèmes de requêtes composées.

A cette échelle d'évaluation, nous constatons qu'il est nécessaire d'utiliser d'autres critères d'évaluation des performances pour montrer par des chiffres l'importance de l'approche d'intégration sémantique des bases de données hétérogènes

6. Quelques extrait de codes source :

Nous présentons dans cette section quelques lignes de codes sources.

Le premier extrait de code source pour établir la connexion de l'ontologie avec l'IDE Eclipse.

```
Public void ontconx{
String filename1= ("C: … Jena-2.6.3\\src-examples\\data\\Alimentation Risks.owl");
if (filename1== null) {
System.out.println("No input file Found");
} else {
try
{
File file = new File(filename1);
File InputStreamreader = new FileInputStream(file);

model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM_MICRO_RULE_INF);

       model.read(reade$r,null);
                                  } catch (Exception ex) {}  }
```

Le second extrait de code source qui représente l'exécution d'une requête SPARQL

```
String sparqlQuery =
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"+
"PREFIX owl: <http://www.w3.org/2002/07/owl#>"+
"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>"+
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"+
"PREFIX lo:<http://www.semanticweb.org/hp/ontologies/2014/10/untitled-ontology-47#>"+
"SELECT ?x WHERE {lo:Apples lo:has_microbe ?x.}    ";

Query qo3 = QueryFactory.create(sparqlQuery);
QueryExecution qp3 = QueryExecutionFactory.create(qo3, modele);
ResultSetRewindable results = ResultSetFactory.makeRewindable(qp3.execSelect());
System.out.println("---- XML ----");
ResultSetFormatter.outputAsXML(System.out, results);
results.reset();
System.out.println(ResultSetFormatter.asText((com.hp.hpl.jena.query.ResultSet) results));
```

Les deux extraits de code source décrivent le calcul de similarité de wu palmer via le package Ws4j 1.0.1 et JawJaw 1.0.2 et celui de Cosinus:

```
Private static ILexicalDatabasedb = new NictWordNet();
Private static double compute(String word1, String word2) {
              WS4JConfiguration.getInstance().setMFS(true);
              doubles =   newWuPalmer(db).calcRelatednessOfWords(word1, word2);
              returns;}
       double distance = compute(words1, words2);
```

```
Public double calculate(String stringOne, String stringTwo) {
Collection<Character> unionOfStringsOneAndTwo = union(stringOne, stringTwo);
Collection<Integer> stringOneOccurrenceVector = createFrequencyOfOccurrenceVector(stringOne,
unionOfStringsOneAndTwo);
Collection<Integer> stringTwoOccurrenceVector = createFrequencyOfOccurrenceVector(stringTwo,
unionOfStringsOneAndTwo);
              Int dotProduct = 0;
              try {
                     dotProduct = dotp(stringOneOccurrenceVector,
stringTwoOccurrenceVector);
              } catch (VectorMathExceptione){
                     e.printStackTrace();

              }
              Double vectorOneMagnitude = magnitude(stringOneOccurrenceVector);
              Double vectorTwoMagnitude = magnitude(stringTwoOccurrenceVector);
              Return dotProduct / (vectorOneMagnitude * vectorTwoMagnitude);
       }
```

Conclusion :

Nous avons présenté dans ce chapitre, la mise en œuvre de notre système de médiation sémantique à base d'ontologies MS4AR validant notre approche proposée.

Le système a été évalué selon deux cas : le premier, l'évaluation selon quelques mesures de similarité sémantique et le second cas, une étude comparative avec les approches existantes.

Conclusion et perspective

L'intégration de données a pour objectif de combiner des sources de données autonomes, distribuées et hétérogènes afin d'obtenir une vue homogène et uniforme des données intégrées. Une façon pour y parvenir, est de représenter les données selon un même schéma global et selon une sémantique unifiée. Deux approches ont été présentées ; la première basée médiateur et la seconde basée entrepôt.

Dans le cadre de notre travail qui vise à proposer un système d'intégration sémantique de sources de données, nous avons présenté l'utilité des ontologies dans le domaine de l'intégration de données et plus précisément l'association d'ontologie dans le système de médiation afin de traiter les problèmes liés à l'hétérogénéité des sources.

La construction de ce schéma se fait en utilisant l'approche de base LAV pour effectuer le lien entre le schéma global et les schémas locaux et sur l'approche hybride pour l'utilisation d'ontologie en tant que, d'une part, support d'interrogation unifiée et de l'autre part, comme une ressource de connaissance au niveau de chaque source locale.

Notre technique d'alignement des ontologies est basée sur un certains nombre de matching et l'utilisation de mesures de similarité existantes. Nous avons présenté en détail le processus d'intégration sémantique avec les algorithmes de traitement de requêtes d'utilisateurs.

Enfin, pour valider notre approche, nous avons implémenté un système de médiation sémantique dédié au domaine des risques alimentaires. De plus, nous avons effectué une évaluation des performances selon les mesures de similarité et une comparaison avec trois travaux existantes. Les résultats obtenus sont prometteurs. Ils montrent d'une part, l'importance des mesures de Palmer et de cosinus pour traiter les requêtes et de l'autre part, l'originalité de notre approche par la proposition d'une méthode d'alignement des ontologies.

Néanmoins, notre travail ouvre des perspectives que nous envisageons de réaliser :

- L'amélioration de l'approche par la possibilité d'ajouter d'autre source de données avec sa propre ontologie et d'ajouter son adaptateur qui reste une tâche difficile.
- Utilisation des sources de données hétérogènes selon d'autres critères tels que : le modèle (BD O.O, BD semi structurées,..), domaines (médecine, technologie,…).
- Application aux documents multimédia.
- Optimisation de requêtes au niveau médiateur et adaptateur.
- Amélioration de l'approche dans un environnement P2P décentralisé.

Bibliographie:

[1] Fabien Gondon, « Le web sémantique, Chargé de recherche à INRIA », Paris 2012.

[2] Zammar Nisrine, « Conception et implémentation d'un système de recherche à base d'annotations sociales », Mémoire de fin d'études, 2012.

[3] Amel Yessad, « Construction d'un Environnement Pédagogique Adaptatif basé sur les Modèles et Techniques du Web Sémantique », l'université Badji-Mokhtar d'Annaba, Thèse 2009.

[4] Christine Froidevaux, « Ontologie », Equipe Bio-informatiques, Université Paris Sud.

[5] H-Benhmidi, « Les ontologies», Chargé de cour à université du Tlemcen

[6] Véronique Giudicelli, « Application à la formalisation des concepts de description d'ontologie avec l'éditeur Protégé », 23 mai 2013.

[7] Boubker Sbihi, « WEB 2+ : une nouvelle version du web2.0 », 2008.

[8] Alexandre Passant, « Technologies du Web Sémantique pour l'Entreprise 2.0 », Thèse du doctorat de l'Université Paris IV, Sorbonne, 2009.

[9] Florence Amardeilh, « Web Sémantique et Informatique Linguistique », 2007.

[10] Thomas B. Passin, « Explorer's Guide to the Semantic Web», by Manning Publications. 2004.

[11] Anutariya Chutiporn, « Semantic Web Modeling and Programming with XDD. SWWS'01': The First Semantic Web Working Symposium », 2001.

[12] Cranefield, Stephen, « UML and the Semantic Web », 2001.

[13] Tallis, Marcello, Neil Goldman, and Robert Balzer « the Briefing Associate, a Role for COTS Applications in the Semantic Web », 2001.

[14] Gharbi itidal, « Ontologies & Web sémantique », Institut supérieur de gestion de Tunis.

[15] Gayo Diallo, «Une Architecture à base d'Ontologies pour la Gestion Unifies des Données Structurées et non Structurées », Thèse pour obtenir Docteur, Université Joseph Fourier, le 11 décembre 2006.

[16] Grigoris Antoniouet Frank van Harmelen, « Chapitre 1 introduction au web sémantique ».

[17] Mohamed Khaled Khelif, « Web sémantique et mémoire d'expériences pour l'analyse du transcriptome », thèse du doctorat université de Nice Sophia Antipolis, Le 4 avril 2006.

[18] Natalya F. Noy et Deborah L. McGuinness, « Développement d'une ontologie 101 : Guide pour la création de votre première ontologie », Université de Stanford.

[19] Manuel Zacklad, « Classification, thesaurus, ontologies, folksonomie : comparaisons du point de vue de la recherche ouverte d'information », L'archive ouverte pluridisciplinaire HAL.

[20] Jing ET W. Bruce Croft, « An association thesaurus in information retrieval, Proceedings of RIAO », 1994.

[21] Fatiha SAÄIS, « Intégration Sémantique de Données guidée par une Ontologie », Doctorat de l'Université Paris-Sud, 2007.

[22] Lahmar Fatima, « Une approche Hybride d'intégration de sources de données hétérogènes dans les datawarehouses », thèse du doctorat, l'Université Mentouri de Constantine, 2011.

[23] Haïfa Zargayouna et Sylvie Salotti, « Mesure de similarité dans une ontologie pour l'indexation sémantique de documents XML », Université Paris.

[24] Anne Doucet, « Médiateurs », Enseignement dans Laboratoire d'informatique de paris VI Pôle Intelligence Artificielle.

[25] Thabet Slimani, Boutheina Ben Yaghlane, Khaled Mellouli, « Une extension de mesure de similarité entre les concepts d'une ontologie », International Conférence (Sciences of Electronic, Technologies of Information and Telecommunications), Tunisia, March 2007

[26] Dimitre Kostadinov – Verónika Peralta – Assia Soukane – Xiaohui Xue, « Intégration de données hétérogènes basée sur la qualité », Laboratoire PRiSM, Université de Versailles, France.

[27] Amel BOUSSIS, «intégration de source de donnée à base ontologique dans un Environment P2P », Mémoire de Magistère, L'Institut National d'Informatique, 2007 / 2008.

[28] Mina Ziani, Danielle Boulanger, Guilaine Talens, « Système d'aide à l'alignement d'ontologies Métier », Université Jean Moulin, Equipe MODEME, Marseille, mai 2010.

[29] Fatiha DJAHAFI, Abdelkader HAOUAS, « Matching des documents XML par la mesure de similarité à base Wordnet », Université des Sciences et de la Technologie d'Oran USTO-MB Algérie.

[30] Hélène Gagliardi, Ollivier Haemmerlé, Nathalie Pernelle, and Fatiha Saïs. « An automatic ontology-based approach to enrich tables semantically. In Workshop AAAI on Context and Ontology », July 2005.

[31] Wahiba Sedjelmaci, «Détection des opérations de changement des sources de données et leur impact sur un système de médiation », mémoire de magister 2012.

[32] Reynaud, Pernelle, Rousset, Safar, Saïs, « Data Extraction, Transformation and Integration guided by an Ontology», 2009.

[33] Aicha Aggoun, Bouramoul Abdelkrim, Kholladi Mohamed Khiereddine. «Personnalisation d'accès aux sources de données hétérogènes pour l'organisation des grands systèmes d'information d'entreprise ». International Conference on Information Technology for Organization Development, Tebessa, 2014.

[34] Uschold ET Grüninger, « Ontologies: principles, methods, and applications, Knowledge Engineering Review ».

[35] Wache, Vögele, Visser, Stuckenschmidt, Schuster, Neumann, ET Hübner, « Ontology-based integration of information a survey of existing approaches», 2001.

[36] Aissa Fellah, Mimoun Malki, Ahmed ZAHAF, « Alignement des ontologies : utilisation de WordNet et une nouvelle mesure structurelle », Université Djillali Liabes de Sidi Bel abbés, département d'informatique, Algérie.

[37] Amina Chettibi, « Conception d'une ontologie pour une plateforme d'enseignement à distance », université de Jijel, ingénieur informatique, 2005.

[38] H BENHMIDI, « Présentation Protégé est un éditeur d'ontologies, univ-tlemcen », 2011.
[39] Adam Jurcık, « Development of Visualization Plug-in for Protégé», MASTER'S THESIS.
[40] Vincent Wolowski, « OWL Ontologies and Visualization », University Hagen,2006
[41] Jean Michel DOUDOUX « Développons en Java avec Eclipse », 2004.
[42] François-Régis Chaumartin, « Wordnet et son écosystème : un ensemble de ressources linguistiques de large couverture », Montréal, Canada, Apr. 2007.
[43] Belaid Saad, « intégration des problèmes de satisfaction de contrainte distribue et sécurité dans un système d'aide a décision à base de connaissance », thèse du doctorat, université de Paule varailne.
[44] kahluola Boubaker, « Development et implémentation d'un algorithme de schémas matching semi-automatique pour le chargement automatique de donnée XML ver base de données relationnel», thèse du doctorat, université d'Oran, 2013.
[45] Andon Tchechmedjiev, « État de l'art : mesures de similarité sémantique locales et algorithmes globaux pour la désambiguïsation lexicale à base de connaissances Laboratoire d'Informatique de Grenoble-Group, Université de Grenoble, 2012.

Webliographie:

[Web1] http://jplu.developpez.com/tutoriels/web-semantique/introduction/ (Visité le 21/12/2014 à 20 :30).

[Web2] http://web.developpez.com/actu/2469/Debat-Quelles-sont-les-limites-du-Web-2-0-Que-sera-le-Web-3-0-selon-vous/ (Visité le 21/12/2014 à 21 :00).

[Web3] http://blog.sparna.fr/ontologie-thesaurus-taxonomie-web-de-donnees/ (Visité le 15/02/2015 à 21:00).

[Web4]: http://www.gettingcirrius.com/2010/12/calculating-similarity-part-1-cosine.html (Visité le 10/03/2015 à 13:20)

[Web5] www.office.microsoft.com (Visité le 19/04/2015 à 13:20)

[Web6] www.java.com (Visité le 19/04/2015 à 13:30)

[Web7] www.projects.eclipse.org/releases/luna.com (Visité le 19/04/2015 à 14:00)

[Web8] www.info.univ-angers.fr/pub/genest/fichiers/m1_ws/ws_chap6 (Visité le 20/04/2015 à 21:00)

[Web9] https://code.google.com/p/ws4j/ (Visité le 20/04/2015 à 22:00)

[web10] http://jmdoudoux.developpez.com/cours/developpons/java/chap-bibliotheques-free.php#(Visité le 8/05/2015 à 13:20)

[Web11] www.photofiltre-studio.com(Visité le 20/04/2015 à 23:20)