" C'est en marchant que se fait le chemin."

Paulo Coelho.

Typeset in LaTeX 2_ε.

A ma maman

A ma grand-mère et mes deux oncles

A mes frères et sœurs et toute la famille

A mon professeur Mr Belalem Ghalem et mon enseignante, amie

Melle Benguedach Asmaa

A tous mes amis, ainsi que toute la promotion du Master 2 en

informatique 2010/2011 de l'université de Tlemcen

Je dédie ce travail

Saidi Imène

Remerciements

Merci mon dieu.

Je tiens tout particulièrement à remercier mes encadreurs Monsieur SETTOUTI Lotfi et Monsieur MIDOUNI Djallal qui sans eux, ce mémoire n'aurait jamais pu voir le jour. Merci Messieurs pour votre aide, votre patience et tous vos conseils pour mener à bien mon travail.

Je remercie Monsieur M. A ABDERRAHIM qui m'a fait l'honneur de présider le jury, Mesdames S. KHITRI, Z. El YEBDRI, A. HALFAOUI d'avoir accepté d'évaluer ce travail, Messieurs B. BENMAAMAR, S. CHOUITI d'avoir accepté de juger ce travail.

Je tiens aussi à adresser mes remerciements à tous les enseignants et les étudiants du département d'informatique de l'université de Tlemcen. Je remercie toutes les personnes que je n'ai pas citées, bien qu'elles me soient chères et qui ont contribué de prés ou de loin à l'accomplissement de ce travail.

Résumé

Ce travail s'inscrit dans le cadre des travaux de recherches concernant la composition des services web pour l'interrogation des sources de données hétérogènes et distribuées de nature médicale (rapports médicaux, imagerie médicale annotée, ...).

L'objectif de ce mémoire est de proposer des techniques d'indexation sémantique automatiques (indexation textuelle liée à une ontologie) et manuelles (annotations faites par des médecins à base d'une ontologie).
Il s'agit également de spécifier sous forme de services Web une interface permettant l'exploitation des index sémantiques proposés.

Nous proposerons dans un premier temps un scénario d'utilisation sur la base des sources d'information puis spécifier (formellement) les différents éléments permettant la mise en œuvre (e.g. index issu de l'indexation sémantique, recherche dans cet index, etc.).

Mots-clés :

Recherche d'informations, indexation sémantique, web services.

Table des matières

Table des figures

Liste des tableaux

Glossaire

AFNOR Association Française de Normalisation.

CF Concept Frequency.

HTTP HyperText Transfer Protocol.

IDF Inverted Document Frequency.

OWL Ontology Web Language.

RDF Ressource description framework.

RDFS RDF Schema.

RDQL RDF Data Query Language.

RI Recherche d'Information.

SaaS Software as a Service.

SGBD Système de Gestion de Base de Données.

SPARQL SPARQL Protocol and RDF Query Language.

SQL Structured Query Language.

SRI Systèmes de Recherche d'Information.

TF Term Frequency.

URI Uniform Resource Identifier.

W3C World Wide Web Consortium.

XML Extensible Markup Language.

Introduction générale

La diversité des sources d'information distribuées et leur hétérogénéité est l'une des principales difficultés rencontrées par les utilisateurs du Web aujourd'hui. Cette hétérogénéité peut provenir du format ou de la structure des sources (sources structurées : bases de données relationnelles, sources semi-structurées : documents XML, ou non structurées : textes) [1]. D'ailleurs, le futur web, dénommé *web sémantique*, compte parmi ses principaux objectifs la résolution de cette problématique : fournir des mécanismes d'accès à des sources de données distribuées et hétérogènes de manière normalisée et intelligible pour les machines et les humains [2].

Le web sémantique décrit une infrastructure permettant à des agents logiciels de faciliter l'accès des différents utilisateurs aux ressources du web (sources d'information et services). C'est dans cette infrastructure que l'intégration des informations d'une variété de sources peut être réalisée et facilitée. Cette infrastructure requiert la construction de systèmes devant offrir à l'utilisateur une vue uniforme et centralisée des données distribuées, une vue pouvant correspondre à une vision plus abstraite, condensée des données et donc, plus significative pour l'utilisateur. De tels systèmes sont connus sous le nom de systèmes de médiation. Ces systèmes sont, par ailleurs, très utiles, en présence de données hétérogènes, car ils donnent l'impression d'utiliser un système homogène [1].

Une des techniques récemment utilisée pour intégrer des données dans les systèmes de médiation, est l'utilisation des services web [3]. En effet, les services web sont particulièrement adaptés et abondamment exploités pour fournir un accès uniforme aux

informations pouvant être distribuées. Les services Web permettent de disposer d'un système basé sur un ensemble de services logiciels distribués, fonctionnant indépendamment les uns des autres afin de réaliser une fonctionnalité globale [4]. Ceci facilitera la découverte des sources d'informations pertinentes étant donnée une requête posée, et simplifie l'accès aux sources pertinentes, évitant ainsi à l'utilisateur d'interroger lui-même chacune d'elles et combiner automatiquement les réponses partielles obtenues de plusieurs sources de façon à obtenir une réponse globale à ses besoins.

Contexte du travail

Ce travail s'inscrit dans le cadre d'un projet de recherche[1] visant la construction de systèmes de médiation orientés services. L'originalité de ce projet est la considération de deux classes de services web pour l'intégration des sources de données hétérogènes et distribuées :

- Des services web permettant l'interrogation de données (Querying As Service) (Figure 1.1) pour l'intégration de sources de données structurées. (e.g. base de données etc.)
- Des services web permettant la recherche d'information (Retrivieng As Service) (Figure 1.1) pour l'intégration de sources de données peu structurées (textes,images annotées, etc.)

Le cadre de ce travail de recherche est principalement motivé par le cadre applicatif des systèmes d'information médicaux. Il s'agit notamment de mobiliser des services web permettant d'exploiter des sources d'information hétérogènes peu structurées ou semi structurées (rapports médicaux, imagerie médicale annotée, etc.) moyennant des index issus d'une indexation sémantique. Mais également de fournir un accès uniforme aux sources de données structurées (e.g., base de données relationnelles, entrepôts RDF, XML, etc.) en se basant sur les langages de requête qu'ils offrent (e.g. SQL, SPARQL, XQuery, etc.). Nous tenons à signaler que le travail présenté dans ce mémoire est conditionné et contraint par le contexte de recherche et vise ainsi la définition de

[1] Initié par l'Institut National des Sciences Appliquées (INSA) de Lyon et l'Université Abou bakr bekaid Tlemcen

services d'indexation sémantique non pas universels (dans le style de google, fast[2] etc.) mais pouvant être combinés avec d'autres services (d'indexation ou d'interrogation) dans le cadre d'un système de médiation.

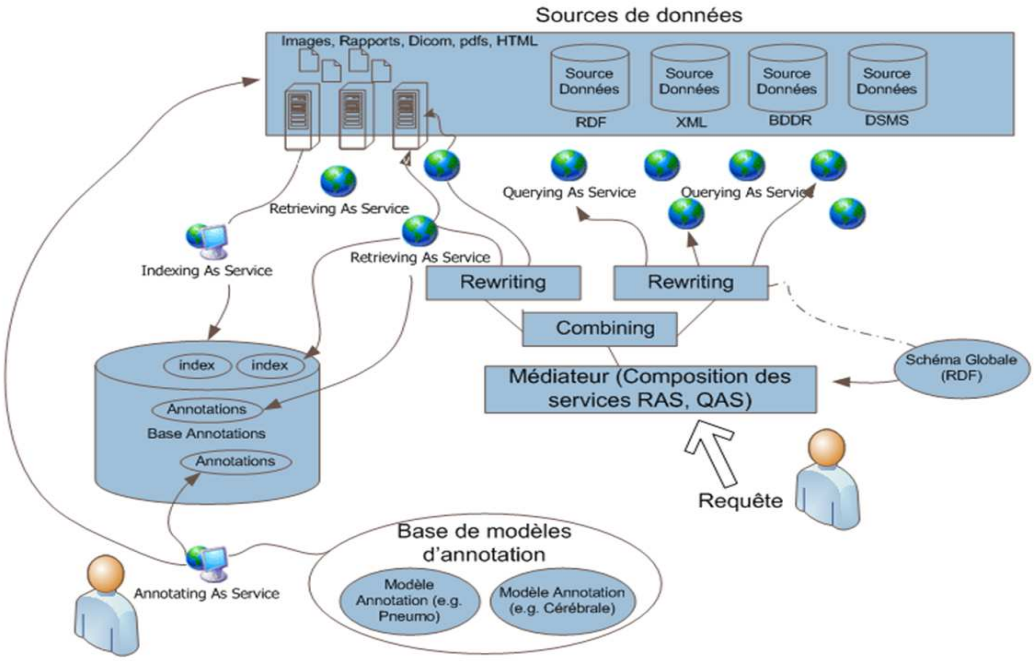

FIG. 1: Contexte du travail

Contributions

La contribution de ce travail est double :

- Dans un premier temps, nous commençons par la spécification des services web d'indexation pouvant être combinés avec d'autres services web d'interrogation. Cette spécification concerne les sources de données structurées comme les BDD ainsi que la combinaison des différents services web pour la réécriture de la requête dans l'architecture proposée de système de médiation. La modélisation de ces services web d'indexation est faite en RDF (Ressource description framework) qui est un modèle de description des données.
- Nous donnerons dans un deuxième temps un exemple d'utilisation des services

[2]http ://sharepoint.microsoft.com/en-us/product/capabilities/search/Pages/Fast-Search.aspx

web d'indexation et nous présenterons ensuite les techniques utilisées pour l'indexation (qui se fera d'une manière automatique ou manuelle selon le type de la source) et la réalisation des index exploitables par nos services web.

La suite de ce mémoire est structurée en trois parties, s'organisant comme suit :

- Dans la première partie, nous présenterons un état de l'art, qui a été divisé en deux sous parties, la première portera sur les aspects du web sémantique, et la seconde sur l'indexation sémantique.
- La deuxième partie a été divisé aussi en deux sous parties, une pour la conception des services web et l'autre pour la construction des index. Cette deuxième partie sera consacrée dans sa globalité, à la description de la conception détaillée de notre système. Cette conception décrite à l'aide de schémas, permet de montrer les différentes étapes à suivre et les fonctionnalités du système proposé.
- La troisième partie permet de concrétiser notre conception par la présentation des étapes de conception d'un prototype.

- Nous finirons, comme pour tout travail de recherche, par une conclusion qui récapitule la problématique que nous avons traitée et les résultats obtenus, ainsi que quelques améliorations prévues dans le futur.

Premi re partie

Notions de base et tat de l'art

1

Quelques aspects sur le web sémantique

1.1 Introduction

Le web sémantique est un concept développé par le W3C, il a pour but d'ajouter du sens au web et de le rendre plus intelligent. Dans ce chapitre, nous allons présenter quelques aspects du web sémantique qui nous serons nécessaires pour la suite. Nous commençons par définir ce que c'est que le web sémantique.

1.2 Web sémantique

Selon Tim Berners-Lee [5] : *le web sémantique n'est pas un web distinct mais bien un prolongement du web que l'on connaît, dans lequel, on attribue à l'information une signification clairement définie, ce qui permet aux ordinateurs et aux humains de travailler en plus étroite collaboration.*

Et dans une autre définition : *c'est un immense espace d'échanges de ressources entre machines permettant à des utilisateurs d'accéder à de grands volumes d'informations et à des services variés* [5].

La structure du web actuel est essentiellement syntaxique, son contenu est lisible par des humains et par des machines, mais il n'est compréhensible que pour les humains. L'idée est de changer la structure du web actuel, que l'on appelle web "présentable" ou "syntaxique", vers une autre structure, que l'on appelle web "intelligent" ou "compréhensible" par les machines, (voir la Figure 1.1). C'est de là qu'est née l'initiative du web sémantique : *un web qui parle aux machines* [4]. Pour ce faire, il est nécessaire de standardiser des langages et des outils adaptables à un maximum d'applications tout en conservant des propriétés permettant leur emploi dans les conditions d'échelle et de performance requises pour le web [6].

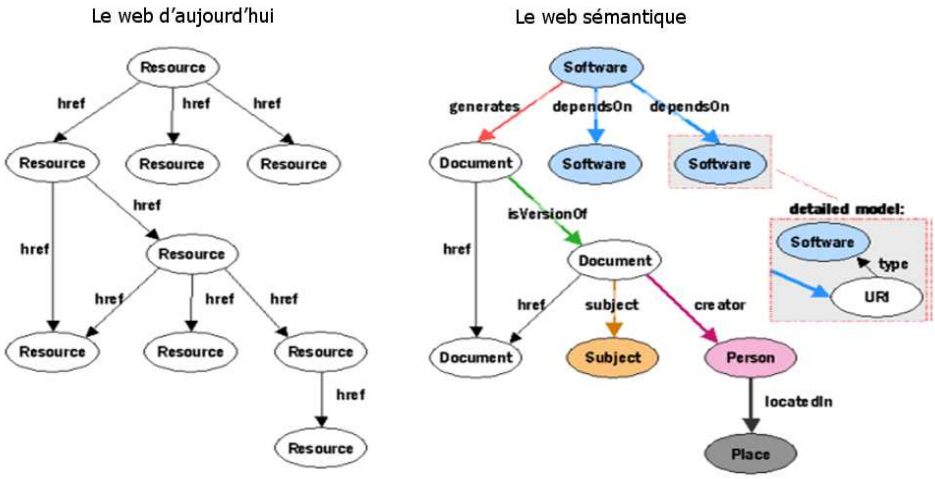

Fig. 1.1: Web d'aujourd'hui et le web sémantique

1.3 Services web

L'intérêt des services web est de favoriser une architecture orientée services, intégrant des systèmes hétérogènes complexes, fortement distribués et permettant la coopération et de nouvelles formes de collaboration entre les applications distantes. Un de ses intérêts est donc de faciliter l'interconnexion entre ces différentes applications, indépendamment des plateformes et des langages de programmation utilisés. Les services web semblent être la solution de l'avenir pour implémenter les systèmes distribués, aujourd'hui, ces services sont distribués à large échelle sur Internet [4].

En général, un service web se concrétise par un agent, réalisé selon une technologie informatique donnée. Un demandeur (utilisateur) utilise ce service à l'aide d'un agent de requête, il rentre alors des Inputs et attend des Outputs (saisir des entrées précises et avoir des sorties correspondantes). Le fournisseur et le demandeur partagent une même sémantique du service web, tandis que l'agent et l'agent de requête partagent une même description du service pour coordonner les messages qu'ils échangent (voir la Figure 1.2) [7] .

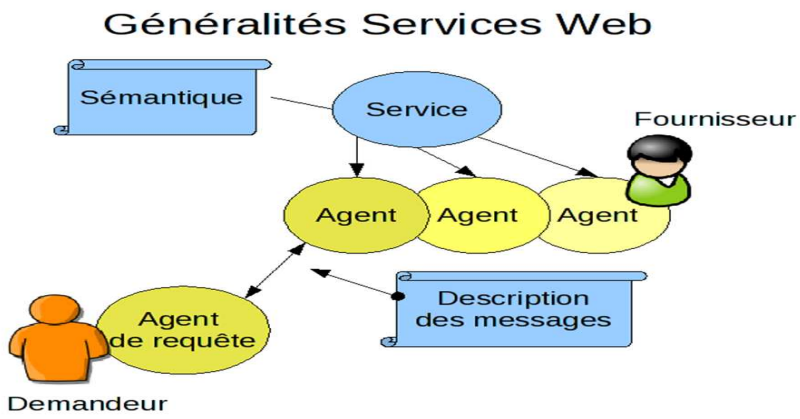

Fig. 1.2: Description du service web

1.4 Systèmes de médiation

Un système de médiation est un système intermédiaire, c'est une interface entre l'utilisateur et les services du web d'un domaine donné. Il doit donner l'impression à l'utilisateur qu'il n'utilise qu'un unique système alors que la satisfaction de sa demande peut exiger de composer plusieurs services.

Parmi les différentes grandes catégories d'applications de ces systèmes de médiation, on peut citer les applications de recherche d'information, celles d'aide à la décision en ligne et celles, de manière plus générale, de gestion de connaissances au sens large [1].

A titre d'exemple, on peut donner l'illustration du premier type d'applications. Supposons qu'un utilisateur pose la requête suivante : quels sont les maladies cancéreuses traitées à l'hôpital de Tlemcen ? lesquelles sont curables ?
Supposons l'existence de deux sources d'information. La première, Internet Medical Data Base, utilise un SGBD relationnel et contient une liste de maladies, précisant pour chacune le type, la partie du corps atteinte par cette maladie et les symptômes. La seconde source d'information, peut utiliser des fichiers XML contenant, par maladie, les différents cas traités et, pour chaque cas, le nom du patient, son état, et l'adresse de l'hôpital.
La réponse à la requête devra être construite en interrogeant chacune d'elles et en combinant les résultats de l'interrogation de façon à offrir à l'utilisateur une réponse globale.

1.5 Ontologies

Considérées comme des éléments fondamentaux du web sémantique, les ontologies y sont utilisées pour déterminer les index conceptuels décrivant les ressources sur le web. Les ontologies peuvent être définies comme des spécifications d'un vocabulaire de représentation pour un domaine partagé du discours qui peut inclure des définitions de classes, des relations, de fonctions et d'autres objets [8]. Les ontologies sont

utilisées en général pour permettre aux machines de raisonner et d'interpréter des informations, ainsi que d'améliorer la pertinence des recherches. Les ontologies permettent aux humains et aux machines de partager les connaissances du domaine et de coopérer ensemble.

Le principe consiste à définir une interprétation commune d'une partie du monde réel, et modéliser les concepts et les relations entre concepts par des classes et des relations entre classes, exemple dans la Figure 1.3.

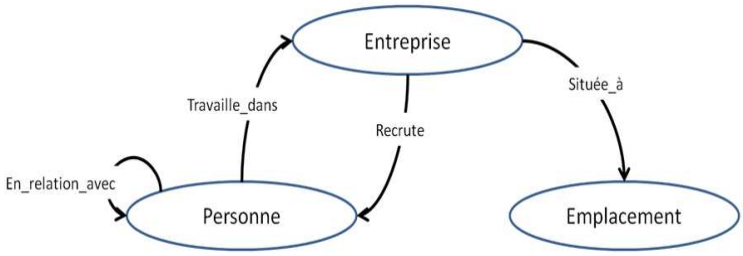

FIG. 1.3: Exemple d'ontologie

Les deux rôles des ontologies sont donc [9] :

– Définir une sémantique formelle pour l'information permettant son exploitation par un ordinateur (et donc des inférences) ;

– Définir une sémantique d'un domaine du monde réel, fondée sur un consensus et permettant de lier le contenu exploitable par la machine avec sa signification pour les humains.

Il est important de noter que le terme sémantique est employé dans un sens différent de celui utilisé dans le langage naturel. C'est-à-dire, le mot sémantique signifie ici "interprétable par les machines". Les machines devront être capables d'utiliser des ressources provenant de diverses sources [4].

1.6 RDF/RDFS/SPARQL

1. RDF

   RDF (Resource Description Framework) est un modèle conceptuel, associé à une

syntaxe, un cadre pour la description de ressources dont le but est de permettre à une communauté d'utilisateurs de partager les mêmes méta données pour des ressources partagées. Il a été conçu initialement par le W3C pour permettre de structurer l'information accessible sur le web et de l'indexer efficacement [10].

C'est une façon de représenter le monde sous la forme de déclarations simples, cette représentation est composée de 3 éléments :

- Un Sujet (ressource),
- Un Prédicat (propriété),
- Un Objet (valeur).

On s'en sert généralement pour la description de métadonnées externes à la ressource décrite. Puisqu'il ne s'agit pas d'un vocabulaire de métadonnées, on s'en sert alors avec des vocabulaires issus d'autres normes [11].

2. RDFS

RDFS (RDF Schema) permet de définir des vocabulaires RDF, principalement :

- Des classes,
- Des relations de sous-classe,
- Le typage des prédicats : domaine, co-domaine,
- Une relation de sous-propriété.

RDFS offre les moyens de définir un modèle (ou bien encore un schéma) de méta données qui permet de :

- donner du sens aux propriétés associées à une ressource ;
- formuler des contraintes sur les valeurs associées à une propriété afin de lui assurer aussi une signification.

Prenons l'exemple de [10], si l'on a une propriété qui représente un auteur, on peut exiger que les valeurs de cette propriété soient une référence à une personne (et non pas une voiture). On peut aussi vouloir restreindre quelles sont les propriétés s'appliquant à une ressource. Cela n'a probablement aucun sens d'autoriser une propriété "date de naissance" à être appliquée à un morceau de musique [10].

Les objets de ce langage (des graphes étiquetés) sont munis d'une sémantique formelle en théorie des modèles, ce qui permet de définir une relation de subsomption

entre les documents RDFS [12].

3. SPARQL

   SPARQL est un standard du W3C, un langage d'interrogation ontologique et un langage de requêtes adapté à la structure spécifique des graphes RDF. Il représente une amélioration de RDQL [13] (qui est le langage le plus utilisé pour interroger les ontologies implantées par RDF et il utilise une syntaxe similaire au SQL). SPARQL ajoute de nouvelles fonctionnalités pour construire les graphes de sortie.

1.7 Conclusion

Ces dernières années, le Web a rapidement évolué, car les données échangées sont devenues très hétérogènes. Les architectures orientées services, et en particulier les services Web, permettent l'accessibilité, la découverte et l'utilisation universelle de n'importe quelle application logicielle sur le Web en utilisant des normes ouvertes.
Le web sémantique propose des solutions formalisées pour améliorer la recherche sémantique des ressources. De nouvelles spécifications sont apparues pour améliorer cette recherche. L'un des objectifs du W3C est de permettre la recherche de ressources à la fois par des humains et par des machines. Le web sémantique s'est finalement orienté vers une solution d'indexation basée sur les ontologies [4].
Dans ce qui suit, nous parlerons de l'indexation d'une manière générale et de son rôle dans la recherche d'informations, on parlera également de ses techniques et de quelques travaux précédents sur l'indexation.

2

État de l'art sur l'indexation sémantique

2.1 Introduction

La recherche d'information (RI) suscite depuis fort longtemps l'attention de la communauté scientifique. La mise en œuvre de solutions capables d'améliorer la performance a toujours été primordiale. Des systèmes de recherche d'information (SRI) ont été conçus, leur objectif est de fournir aux utilisateurs les documents pertinents par rapport aux besoins qu'ils expriment. Les SRI utilisent des listes inversées qui rassemblent différents termes choisis pour représenter les contenus des documents et les liens vers ces documents. En complément, à chaque couple (terme, document) est associé un poids qui représente l'importance du terme dans un document. Cette conception est ainsi obtenue par un processus nommé indexation qui selon l'AFNOR est [14] : *l'opération qui consiste à décrire et à caractériser un document à l'aide de représentations des*

concepts contenus dans ce document, c'est-à-dire transcrire en langage documentaire les concepts après les avoir extraits du document par une analyse. [...] La finalité de l'indexation est de permettre une recherche efficace des informations présentes dans un fonds de documents et d'indiquer, sous une forme concise, la teneur d'un document. Lorsqu'une requête est soumise au système, les termes qu'elle contient sont mis en correspondance avec les termes d'indexation extraits des documents pour en déduire les documents à restituer à l'utilisateur. La phase d'indexation est donc une phase primordiale dans le processus de recherche. Dans la littérature, diverses méthodes et stratégies ont été proposées pour l'indexation [15].

2.2 Techniques d'indexation

L'indexation peut être faite d'une manière manuelle, automatique, de façon assistée (semi automatique) ou bien par concepts (indexation conceptuelle) [13].

L'indexation manuelle

Ce type d'indexation repose habituellement sur un jugement de signification plus ou moins intuitif, toujours lié à l'indexeur. Le travail à réaliser pour la mise au point d'une indexation est assez important : connaissance du contenu de l'information, choix des concepts à représenter et traduction de ces concepts en descripteurs. De plus, les mêmes notions peuvent être exprimées de manières très diverses. En raison de ces divers problèmes, des méthodes d'indexation automatique sont donc apparues [16].

L'indexation automatique

Cette indexation utilise des méthodes logicielles pour extraire et établir une liste ordonnée (un index), de tous les mots et leurs occurrences apparaissant dans les documents, et qui correspondent le mieux au contenu informationnel d'un document [16].

En construisant un index avec les mots extraits des textes intégraux, l'indexation automatique élargit considérablement le champ de la recherche et autorise des requêtes

plus souples. Ce type d'indexation suppose un certain nombre d'opérations [13] :

- Le filtrage des mots (analyse morphologique).
- La lemmatisation (Stemming) (analyse lexicale).
- La modélisation vectorielle.
- La pondération.

L'indexation automatique a de nombreux avantages, elle permet par exemple [13] :

- De limiter les choix parfois subjectifs de l'indexeur.
- D'alléger le travail requis par une indexation manuelle.
- D'éviter les incohérences des interprétations différentes entre plusieurs indexeurs.

L'indexation semi-automatique

Les systèmes actuels tentent de remplacer l'homme pour une importante part de son expertise et de lui épargner de nombreuses tâches ; mais, ils ne le remplacent pas complètement. L'indexation automatique suppose une intervention totale du système, ce qui est loin d'être le cas, car l'intervention humaine est toujours nécessaire, d'où l'indexation semi-automatique [17].

L'indexation conceptuelle

L'indexation conceptuelle consiste en toute explicitation symbolique de connaissances contenues dans les documents ou bien à leur propos, en permettant la recherche et la manipulation. Ceci dépasse la simple explicitation des concepts contenus dans les documents, mais recouvre également tout ajout de connaissances pouvant servir d'une manière ou d'une autre, par exemple, la position géographique d'une caméra au moment où un plan audiovisuel a été tourné, ou l'âge du réalisateur [17] [13].

L'annotation

Les notions d'annotation et d'indexation semblent équivalentes, néanmoins nous relèverons les différences suivantes :

- Indexer, c'est avant tout décrire un document pour le retrouver ;

– Annoter, c'est décrire l'interprétation du document par un lecteur, en vue de n'importe quelle tâche d'exploitation future de ce document.

– On indexe pour rechercher plus tard, on annote pour donner des traces de son interprétation, pour documenter la tâche que l'on est en train d'accomplir. Ces traces pourront alors être destinées à soi-même, ou partagées [17].

Dans le cas général, annoter un document, c'est : attacher à l'une de ses parties une description qui correspond à l'usage que l'on souhaitera en faire plus tard [17].

2.3 Indexation sémantique et ontologies

L'indexation sémantique via des ontologies s'appuie sur les technologies du web sémantique. Dans ce type d'approches, la connaissance du domaine (terminologique en particulier) est représentée sous forme d'ontologies, c'est-à-dire en particulier de concepts, d'instances de ces concepts et de relations [15].

2.4 Quelques travaux connexes à l'indexation sémantique

Plusieurs travaux ont contribué à l'avancement de la recherche dans le domaine de l'indexation et de la recherche d'informations. Nous pouvons citer :

– Baziz et al. [18] proposent une indexation utilisant le " noyau sémantique " d'un document. Il s'agit d'un ensemble de concepts pondérés suivant leur représentativité dans les documents et liés entre eux par des mesures de similarité. Cette structure dépend de la mesure de similarité considérée. Baziz et al. ont remarqué que la pondération des concepts est un point crucial, autant que le choix de ces concepts, pour les performances du système. L'idée de noyau sémantique permet de rendre graphiquement de façon claire à l'utilisateur les concepts dans un document et leurs liens.

– Seco et al. [19] pensent qu'une ontologie seule suffit à trouver le contenu informationnel des nœuds. Leurs thèse est qu'il est possible de retirer de la structure

de cette ontologie un sens au nombre d'hyponymes qu'a un concept : plus un concept a de descendants, plus il est spécialisé par d'autres concepts, moins il est lui-même caractéristique. Pareillement, les feuilles de la taxonomie ont une valeur informationnelle maximale.

- Song et al. [20] proposent un modèle de RI basé sur des ontologies de domaine, définies avec OWL. Les différentes ontologies de domaines sont intégrées pour former une ontologie unique. Les termes définis dans l'ontologie sont alors utilisés d'une part comme métadonnée pour annoter les contenus du web et d'autre part comme termes d'indexation de la collection.
- Gilles Hubert et al. [15] proposent un modèle de données dans le cadre d'une indexation à base d'une ontologie de référence. Cette structure de données permet en outre une mise à jour dynamique et en temps réel des résultats de l'indexation lors de la mise à jour de la collection de documents. Cette structure assure ainsi la cohérence permanente entre l'index, le corpus et l'ontologie de référence. L'avantage principal de ce modèle est qu'il n'est plus nécessaire de reconstruire l'index car il est à jour à tout moment. Ainsi, cette structure permet de mettre en place une indexation sémantique dynamique.

2.5 Conclusion

Cette partie traite l'indexation, ses techniques et quelques travaux précédents. Nous avons commencé par donner les différentes manière de mettre en place une indexation, qui peut se faire automatiquement, manuellement, d'une manière assistée ou par concepts. Nous avons également introduit l'indexation sémantique peut se base sur les ontologies et fait un tour d'horizon des travaux les plus significatifs et relatifs à l'indexation sémantique par ontologies. Le prochain chapitre sera consacré à la description de la démarche d'indexation de différentes sources d'information, ainsi qu'à la description des services web d'indexation qui auront pour but d'exploiter les résultats du processus d'indexation réalisé préalablement automatiquement ou manuellement, selon le type de la source.

Deuxième partie

Approche d'indexation sémantique à base de services web

3

Conception des services web d'indexation

3.1 Introduction

Dans les chapitres précédents nous avons présenté les notions de base pouvant nous servir à mettre en œuvre une indexation sémantique à base de services Web. Dans ce chapitre, il s'agit en partie de mobiliser ces notions afin de concevoir des services web permettant une indexation sémantique de sources d'informations hétérogènes et distribuées de nature médicale. Ces services doivent, en outre, être capables de s'insérer dans le cadre d'un système de médiation (notamment lors de la réécriture de requêtes impliquant ces services d'indexation). De ce fait, la suite de ce chapitre sera consacrée à la présentation de notre proposition dans un cadre d'usage global dont l'objectif final est la médiation entre systèmes hétérogènes. Pour ce faire, nous allons commencer par décrire l'architecture globale du système de médiation. Cette architecture sera le

support du processus d'indexation et d'exploitation des résultats de notre approche. Nous allons décrire par la suite les principales étapes concernées par l'indexation à l'aide de diagrammes. Nous détaillerons également les caractéristiques des services web et les algorithmes nécessaires pour la réalisation de l'indexation.

3.2 Contexte général : Médiation à base de services Web

Dans la Figure 3.1, nous présentons l'architecture globale du système, afin de montrer la partie concernée par ce travail dans son contexte général, c'est à dire : l'architecture d'un système de médiation et d'interrogation des sources de données hétérogènes médicales (rapports médicaux, imagerie médicale annotée, ...) à base de services web.

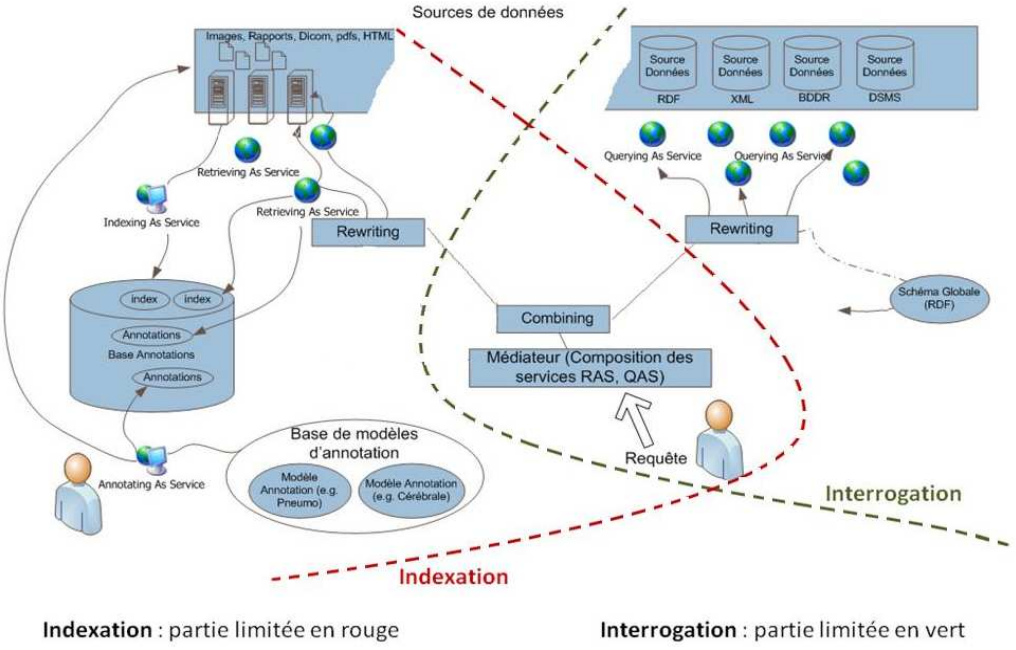

Fig. 3.1: Architecture du système global

3.2.1 Description de l'architecture globale

Le système global est divisé en deux parties, une partie pour l'interrogation et l'autre pour l'indexation. La partie interrogation concerne les sources de données structurées qui peuvent être interrogées par des langages classiques : SQL, SPARQL, ..., alors que la partie indexation concerne les sources de données semi ou non structurées qui peuvent être indexées et seront interrogées en utilisant des méthodes de recherche qui se basent sur des index.

La partie qui nous concerne dans ce travail est celle de l'indexation. Nous remarquons que dans notre partie indexation (délimitée en rouge) dans la Figure 3.1, l'utilisateur a comme vue l'interface du service web, qui se base sur un système intermédiaire représentant le système médiateur, c'est dans cette interface que l'utilisateur peut envoyer sa requête. Nous remarquons aussi qu'il peut y avoir une combinaison de services web.

3.2.2 Approche proposée

Bien que notre cadre global soit concerné par deux phases (l'indexation et l'interrogation moyennant des services web), notre étude se focalisera essentiellement sur le premier aspect. En effet, la phase d'indexation, étant en amont de l'interrogation, est au cœ œur de notre approche et en constitue l'élément premier pour la combinaison de différents services web pour l'évaluation d'une requête.

Notre objectif est de proposer des techniques d'indexation sémantique et de spécifier sous forme de services Web une interface permettant l'exploitation des index sémantiques proposés.

Le schéma de la Figure 3.2 représente l'approche proposée.

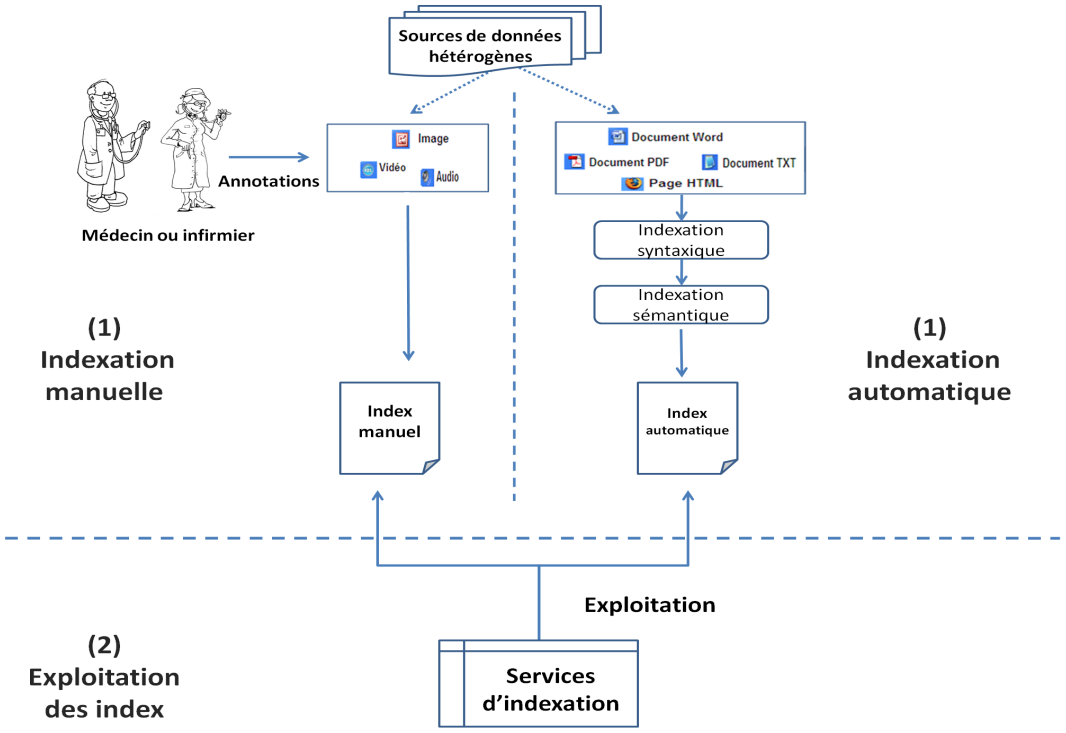

Fig. 3.2: Approche proposée

Sur le schéma de la figure précédente 3.2, nous remarquons que la phase d'indexation se réalise en deux traitements, selon le type de la source de données (sources de données hétérogènes) :

- Premier traitement : indexation faite automatiquement et concerne les sources textuelles (documents).
- Deuxième traitement : indexation faite d'une manière manuelle (des annotations faites par des humains) et concerne les sources non textuelles.

Les services web d'indexation considérés de ce travail concernent différents index issus de l'indexation faite préalablement. Nous distinguons notamment deux type index :

- index manuel = index issu des annotations de documents non textuels.
- index automatique = index issu d'une indexation automatique syntaxique et sémantique des documents textuels.

Ces services web d'indexation peuvent être intégrés et combinés avec d'autres services web d'interrogation (concernant les sources de données structurées) de l'architecture

globale, afin de trouver les sources de données pertinentes qui répondent à une requête donnée. Les réponses partielles obtenues par ces services sont combinées pour délivrer une réponse globale. Il est à noter que ces aspects concernant la réécriture de requêtes afin de d'obtenir une combinaison de résultats satisfaisant au plus une requête ne sont pas traités dans ce travail de recherche. Il sont traité dans le cadre d'un sujet de master recherche commencé parallèlement a notre travail.

3.3 Ontologie de médiation

Comme nous l'avons déjà mentionné, notre indexation s'inscrit dans le cadre d'une approche d'intégration de données par médiation [21]. Dans une telle approche, il est courant de définir, conceptuellement et de manière centralisée, un schéma global ou une ontologie regroupant l'ensemble des prédicats modélisant le domaine d'application du système médiateur. Dans notre cas qui est le domaine médical et afin de soutenir l'intégration des données des différentes sources, l'utilisateur posera ses requêtes dans les termes du vocabulaire structuré du domaine médical fourni par l'ontologie représentant l'ensemble des termes modélisés et utilisés par les différentes sources intégrées.

Le rôle de cette ontologie est d'établir la connexion entre les différentes sources accessibles en se fondant sur la définition de vues abstraites décrivant de façon homogène et uniforme le contenu des sources d'informations en termes des concepts de l'ontologie. Les sources d'informations pertinentes, pour l'évaluation d'une requête, sont calculées par réécriture de la requête en termes de ces vues (partie interrogation). Parmi ces vues, les services web d'indexation que nous allons proposer peuvent être utilisés.

Un exemple d'ontologie médicale de médiation

Afin de montrer un scénario global dans lequel nous allons exemplifier notre approche d'indexation à base de services web, nous présenteront dans un premier temps le schéma global décrivant l'ontologie de notre système (Figure 3.3).

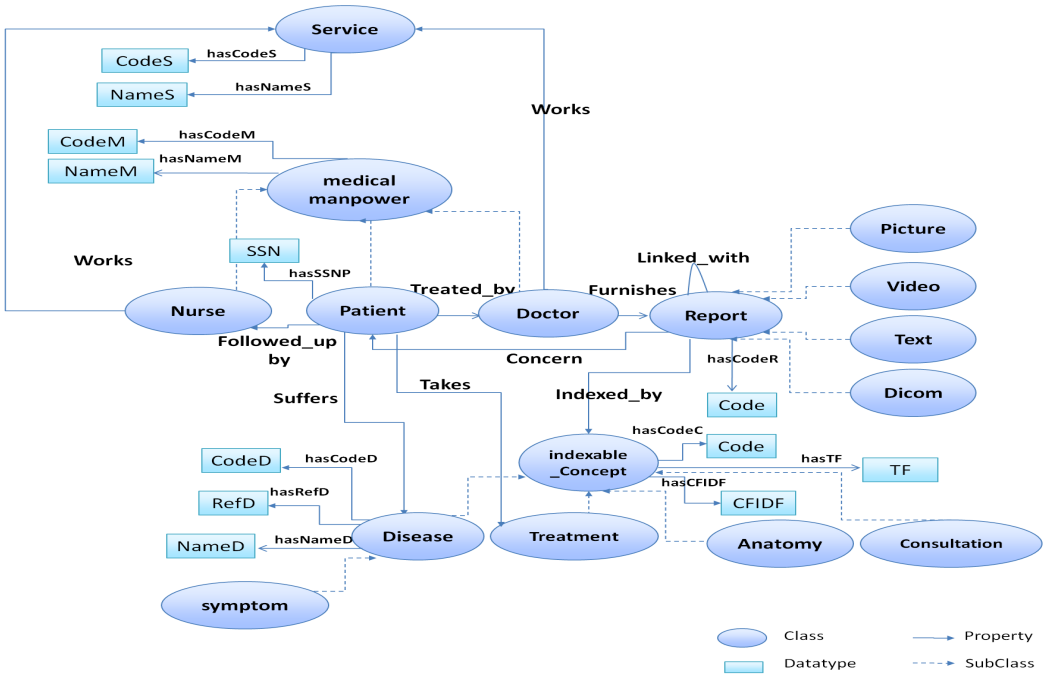

Fig. 3.3: Ontologie de médiation proposée

L'usage d'une ontologie lors de la phase d'indexation permet de rendre un certain nombre de services dont le plus important est la levée des ambiguïtés des sens des termes utilisés pour l'indexation. L'usage d'une ontologie permet aussi une meilleure représentation des connaissances contenues dans les documents. En termes d'indexation sémantique, les concepts de l'ontologie sont associés à chaque document selon les sémantiques qui y sont véhiculées. Ainsi, en plus de lier les documents à des termes pondérés comme dans les approches classiques [22], ces documents sont liés à des termes inter-connectés faisant partie d'une ontologie où les relations disposent d'une sémantique claire et non ambiguë (synonymie, équivalence, relation hiérarchiques, etc.). Dans cet exemple (Figure 3.3), notre ontologie du domaine d'application a été définie comme un ensemble de classes, et chaque classe dispose de :

- *propriétés*, e.g. la classe `Service` a un code de service (propriété : `hasCodeS`).
- *sous-classes*, e.g. le lien (`rdfs:subClass`) entre la classe `Doctor` et la classe `Medical_Manpower` signifiant que la classe `Doctor` est sous-classe de la classe `Medical_Manpower`.

Une classe peut être également liée à une ou plusieurs classes, e.g. `Report` est fourni

par un médecin `Doctor` et concerne un `patient`.

Afin de présenter les services Web pour l'indexation, nous allons présenter dans un premier temps les services web permettant l'interrogation des données dans le système de médiation. Ceci est nécessaire pour garder une cohérence notamment lorsque la réponse à une requête doit être fondée sur des services d'interrogation. Ces services d'interrogation doivent prendre en compte les caractéristiques de l'indexation pour permettre une future réécriture des requêtes et une combinaison des résultats.
Les différents services d'interrogation et d'indexation, seront décris par des vues RDF à partir de l'ontologie de médiation.

3.4 Services web d'interrogation

L'interrogation effective des sources se fait via un médiateur, qui traduit ou réécrit les requêtes en termes de vues. Comme nous l'avons mentionné précédemment dans ce mémoire, la partie interrogation des sources hétérogènes qui se base sur la réécriture des requêtes et de l'intégration des différents services web n'est pas notre objectif immédiat (c'est la seconde phase), néanmoins, il est nécessaire pour notre propos d'expliciter quelques exemples de services web d'interrogation afin de montrer comment ces derniers vont être combinés à nos services web d'indexation.
Le tableau suivant (Tableau 3.1) est un récapitulatif des différents services web d'interrogation utilisant la même ontologie proposée précédemment (l'ontologie de l'architecture globale).

Service	Fonctionnalités	Contraintes
S_1($a, ?b)	Donne les médecins(b) traitant le patient(a)	$a < p1000$
S'$_1$($a, ?b)	Donne les médecins(b) traitant le patient(a)	$a >= p1000$
S_2($a, ?b)	Donne les infirmiers (b) qui soignent le patient(a)	
S_3($a, ?b)	Donne les médecins(b) travaillant dans un service(a)	
S_4($a, ?b, ?c, ?d, ?e)	Donne les rapports : image(b), vidéo(c), texte(d) et dicom(e) d'un patient(a)	
S_5($a, ?b)	Donne les maladies (b) traitées d"un patient (a)	
S_6($a, ?b)	Donne les infirmiers (c) travaillant dans un service(a)	
S_7($a, ?b)	Donne les patients (b) atteints d'une maladie (a)	
S_8($a, ?b)	Donne les rapports (b) fournis par un médecin (a)	

Tab. 3.1: Services web d'interrogation

Syntaxe utilisée

Le langage utilisé pour décrire les services web est RDF avec la syntaxe de représentation N3[1].

Comment lire le service

Le symbole '$' représente les entrées (Inputs) et le symbole ' ?' représente les sorties (Outputs).

Exemple d'utilisation

Le service web d'interrogation est une requête SPARQL conjonctive "contient l'opérateur and représenté par .". Prenons le service S_1 comme exemple. La définition de ce service en RDF avec la syntaxe N3 est la suivante :

```
S1($a,?b) :-
  (?M1  rdf:type   O:Doctor) .
```

[1]http ://www.w3.org/DesignIssues/Notation3

```
(?M1  O:CodeP  ?b) .
(?P1  rdf:type    O:Patient) .
(?P1  O:SSN_P   ?a) .
(?P1  O:Treated_by ?M1)
```

Explications

$a (inputs) de type `Patient` et ?b (output) de type `Doctor` dans notre ontologie.

- 'M1' est une variable de type 'Doctor' qui aura un 'CodeP' comme code personnel (du personnel médical) correspondant à chaque valeur de sortie qui est 'b'.
- 'P1' est une variable de type 'Patient' qui a comme code 'SSN_P' ayant comme valeur 'a'.
- Le patient 'P1' est traité par 'M1' (Treated_by).

On aura alors comme résultat : les médecins (b) traitant le patient(a).

Afin de schématiser graphiquement le service 'S1' représenté en RDF, on a la figure suivante (Figure 3.4) :

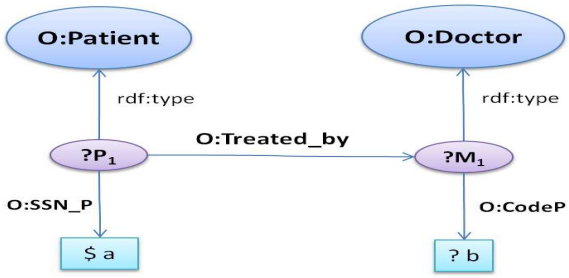

Fig. 3.4: Représentation RDF graphique du service S1 d'interrogation

Description du sch$ma

Dans la représentation graphique de RDF, les nœuds représentent les variables, et les arcs sont des propriétés.

Ce service permet de trouver les Médecins correspondant à la variable $M1$ ayant un $CodeP$ correspondant à la variable b en cherchant les patients correspondant à la variable $P1$ ayant un SSN_P correspondant à la variable a tels que les patients sont traités par ces médecins (propriété $Treated\_by$).

3.5 Services web d'indexation

Pour notre système, nous avons proposé deux services web d'indexation décrits dans le tableau suivant (Tableau 3.2). Les services web d'indexation proposés exploitent les résultats de l'indexation automatique et manuelle.

Service	Fonctionnalités	Contraintes
S1i($a,$seuil, ?b)	Donne tous les concepts indexables (b) cités dans le rapport (a) ordonnés selon n	$n >$ $seuil
S2i($a,$m, ?b)	Donne les m premiers rapports (b) concernant un concept (a) ordonné par t, Rapport =texte et/ou vidéo et/ou dicom et/ou image	

Tab. 3.2: Services web d'indexation

Syntaxe utilisée

La syntaxe des services est celle de RDF.

Explication des contraintes

1. Le service $S1i$ donne tous les concepts indexables (qui ont été indexés) du rapport (a). Ces concepts (les sorties) seront ordonnés d'une manière décroissante selon un certain n (n est le $CFIDF$: propriété du concept qui sera définie plus tard) qui est une valeur entière calculée pour chaque concept, supérieur à un seuil donné en entrée également.
2. Le service $S2i$ donne les m premiers rapports (b) concernant un concept (a). Les rapports seront ordonnés aussi selon un t (t est le tf (term frequency) : la fréquence d'apparition d'un concept dans un rapport, c'est une propriété du rapport qui sera détaillée par la suite). Les m premiers de ces rapports seront pris comme résultat

Exemple d'utilisation

Le service web d'indexation = requête SPARQL conjonctive augmentée de modificateurs de résultats (LIMIT, OFFSET, ORDER BY)

La définition du service S1i en RDF est la suivante :

```
S1i($a,$seuil, ?b) :-
    (?C1   rdf:type O:indexable\_Concept)  .
    (?C1   O:CodeC ?b) .
    (?C1   O:hasCFIDF ?CFIDF) .
    (?R1   rdf:type   O:Report) .
    (?R1   O:CodeR  ?a) .
    (?R1   O:Indexed_by  ?C1)
    FILTER (?CFIDF > $seuil)
    ORDER BY  ?CFIDF
```

Explication

a (inputs) de type Rapport dans l'ontologie (Report) et b(outputs) de type Concept indexable, dans l'ontologie (indexable_Concept).

- '$C1$' est une variable de type `indexable_Concept` qui aura un `Codec` comme code correspondant à chaque valeur de sortie qui est $?b$. Chaque `C1` a une valeur `CFIDF`.
- Les `CFIDF` sont filtrés et seulement les concepts ayant le `CFIDF > seuil` seront pris dans les résultats.
- `R1` est une variable de type `Rapport` qui a comme code `CodeR` ayant comme valeur a (l'entrée).
- Le rapport `R1` est indexé par `C1`.
- Les résultats seront ordonnés par `CFIDF`.

On aura alors comme résultat : les concepts (b) cités dans le rapport (a).

Afin de représenter graphiquement le service 'S1i' représenté en RDF, on a la figure suivante (Figure 3.5) :

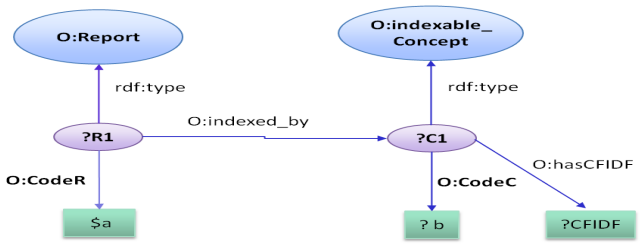

Fig. 3.5: Représentation RDF graphique du service S1i d'indexation

Description du schéma

Ce service permet de trouver les concepts correpondant à la variable $C1$ ayant un $CodeC$. Les concepts trouvés doivent correpondre à la variable b du rapport correpondant à la variable $R1$ ayant un $CodeR$, qui a a comme variable. Les concepts indexent le rapport et sont ordonnés par la variable $CFIDF$.

Pour le second service, la définition RDF du service S2i est la suivante :

```
S2i($a,$m,?b):-
(?R1   rdf:type    O:Report) .
(?R1   O:CodeR  ?b) .
(?C1   rdf:type    O:indexable_Concept) .
(?C1   O:codeC   ?a) .
(?R1   O:indexed_by   ?C1) .
(O:indexable_Concept  O:hasTF ?TF) .
ORDER BY ?TF
LIMIT m
```

Explication : a (inputs) de type Concept indexable dans l'ontologie (indexable_Concept) et b (outputs) de type Rapport dans l'ontologie (Report).

- '$R1$' est une variable de type 'Rapport' qui aura un '$CodeR$' comme code correspondant à chaque valeur de sortie qui est 'b'.
- '$C1$' est une variable de type 'Concept indexable' qui a comme code '$codeC$' ayant comme valeur 'a'.

– Le rapport '$R1$' est indexé par '$C1$'.

– Les rapports (les sorties) seront ordonnés par 'tf'.

– Les m premiers rapports seront pris uniquement.

On aura alors comme résultat : les rapports (b) concernant un concept(a).

La représentation graphique RDF correspondant à ce service 'S2i' est la suivante (Figure 3.6) :

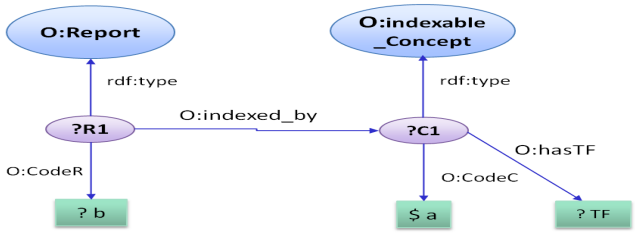

Fig. 3.6: Représentation RDF graphique du service S2i d'indexation

Description du schéma

Ce service permet de trouver les rapports correpondant à la variable $R1$ ayant un $CodeR$ correpondant à la variable b en cherchant les concepts correpondant à la variable $C1$ ayant un $CodeC$ correpondant à la variable a tels que les rapports sont liés à/indexés par ces concepts par une propriété indexé_par (ObjectProperty) et sont ordonnés par la variable TF.

3.6 Exemple d'intégration des services web

Les services web d'indexation que nous avons proposé ne sont pas capables de résoudre tous les problèmes, la composition de services web permet de répondre aux besoins complexes des utilisateurs, par la combinaison de plusieurs services web. Dans l'exemple suivant, nous illustrons l'utilisation et l'intégration des différents services d'interrogation et d'indexation.

* Soit la requête de l'utilisateur suivante :

```
Donner les rapports contenant le concept 'y1' fournis par un médecin
travaillant dans un service 'ser0'./ y1 est une maladie par exemple.
```

Requête en RDF

Dans ce qui suit nous présentons la requête en RDF :

```
Q(x1,y1,x2,y2):-
?R rdf:type  O:report
?R O:Indexed_by ?D
?R  O:CodeR   ?x1
?D rdf:type O:Disease
?D  O:hasNameD  ?y1
?R O:furnishes ?M
?M  rdf:type  O:Doctor
?M O:hasCodeM ?x2
?M O:hasNameM ?y2
?M O:dWorks ?S
?S  rdf:type O:Service
?S O:hasCodeS 'ser0'
?S O:hasNameS ?y3
```

Représentation RDF graphique de la requête

La représentaion graphique de la requête en RDF est la suivante (voir Figure 3.7) :

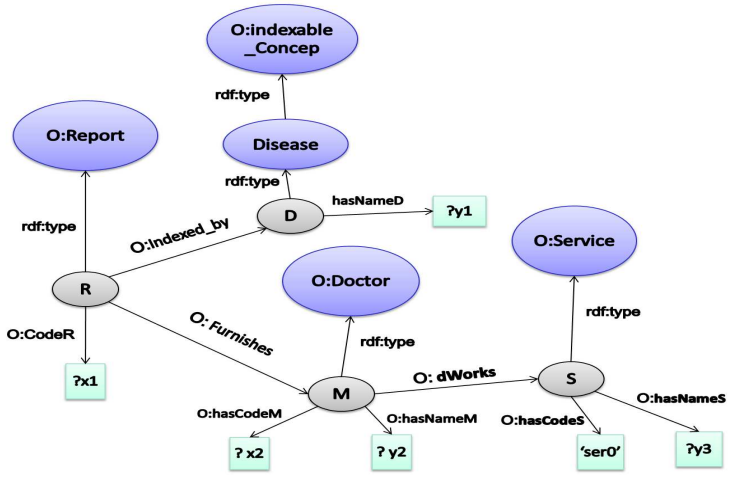

Fig. 3.7: Représentation RDF graphique de la requête

Solution de la requête

La solution de cette requête est la suivante (Figure 3.8) :

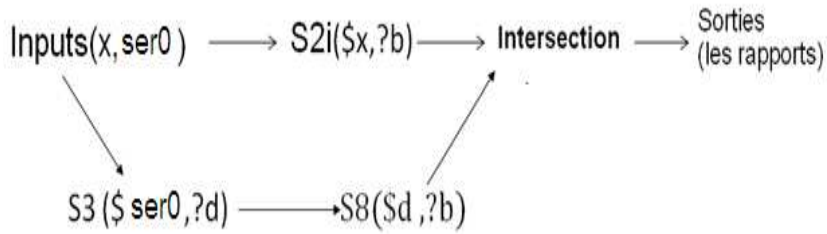

Fig. 3.8: Solution de la requête

Pour répondre à la requête, des services d'interrogation et d'indexation ont été intégrés.

Nous avons en entrée la maladie x et le service $ser0$:

1. Le service web d'interrogation S3 est lancé avec $ser0$ en entrée afin de trouver les médecins qui travaillent dans le service $ser0$.
2. Le service d'interrogation S8 est lancé après, avec comme entrée les médecins obtenus de (1) pour avoir les rapports fournis par ces médecins.
3. En parallèle à (1) et (2), le service web d'indexation S2i est lancé avec x comme entrée qui est une maladie (c'est un concept car c'est une sous classe de 'indexable_Concept') afin de trouver les rapports contenant le concept x.
4. Une intersection est faite entre le résultat de (2) et le résultat de (3) pour avoir les rapports parlant de x et fournis par les médecins travaillant dans $ser0$.

Dans le chapitre qui suit, nous présentons les étapes de notre approche d'indexation et de construction des index.

4

Construction des index

4.1 Approche de la construction d'index

Les services web d'indexation présentés précédemment doivent exploiter et rechercher dans des index physiques qui ont été créés suite à une indexation. Dans ce que suit nous présentons l'approche de l'indexation suivie dans ce travail, c'est à dire les étapes de la construction des index.

L'indexation est une forme de reformulation. Indexer, c'est reformuler le contenu d'un document dans une forme plus adaptée à son contexte d'exploitation dans une application donnée, on indexe donc, en vue d'une application. On ne parle plus seulement d'indexation mais également d'enrichissement, d'annotation, de marquage et de métadonnées [23].

Nous avons au départ un ensemble de ressources (fichiers) hétérogènes. Selon le type

de la ressource, deux traitements sont possibles (Figure 4.1) :

1. Indexation automatique : liée aux documents textuels, ex : fichiers textes.
2. Indexation manuelle : ça concerne les annotations faites par des humains (e.g. médecins, etc.) sur les ressources non textuelles, e.g. une vidéo, une image.

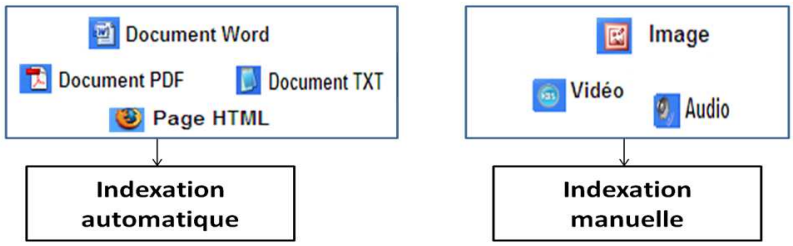

Fig. 4.1: Types d'indexation

4.1.1 Indexation automatique

Dans ce type d'indexation, nous nous inspirons des travaux de Baziz et al. [24][25][26]. La construction des index utilisés par les services web proposés dans l'indexation automatique, se fait en deux phases principales : l'indexation syntaxique et l'indexation sémantique, comme il est montré dans la Figure 4.2.

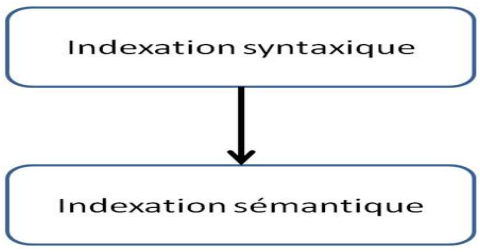

Fig. 4.2: Phases de l'indexation automatique

a- Indexation syntaxique

L'objectif de l'indexation syntaxique est d'extraire les termes des documents et de les stocker avec leur nombre d'occurrences dans un index physique (voir la Figure 4.3).

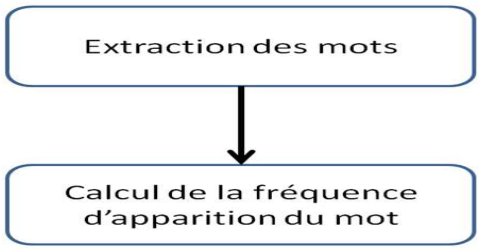

Fig. 4.3: Phases de l'indexation syntaxique

- Extraction des mots

Dans cette première phase de construction des index, il s'agit d'extraire les termes significatifs du document, c'est-à-dire de voir pour chaque terme rencontré, s'il ne fait pas partie d'une liste de mots vides (non utiles), exemple : les articles, les pronoms, ... etc, nommés aussi les " stop-words ". Si le terme est un stop-word il sera alors ignoré et ne sera pas pris pour indexer le document, Sinon il sera pris dans l'index.

- Calcul de la fréquence d'apparition du mot

Pour les mots significatifs, la fréquence d'occurrences doit être calculée. A chaque fois qu'un mot est rencontré dans le document, on incrémente sa fréquence (nombre d'apparition). Cette valeur obtenue est nommée *tf* (Term Frequency).

Le résultat de cette phase est le fichier " Index_Syn " : Index Syntaxique.
Il est constitué comme suit :
Pour chaque document, on trouve tous les mots qui appartiennent à ce document avec leurs fréquences d'occurrence.
N° document => {(terme$_1$, i fois),..., ($terme_n$, j fois)}, i,j : nombre d'occurrence du mot.
Comme exemple du résultat de cette étape, (voir la Figure 4.4)

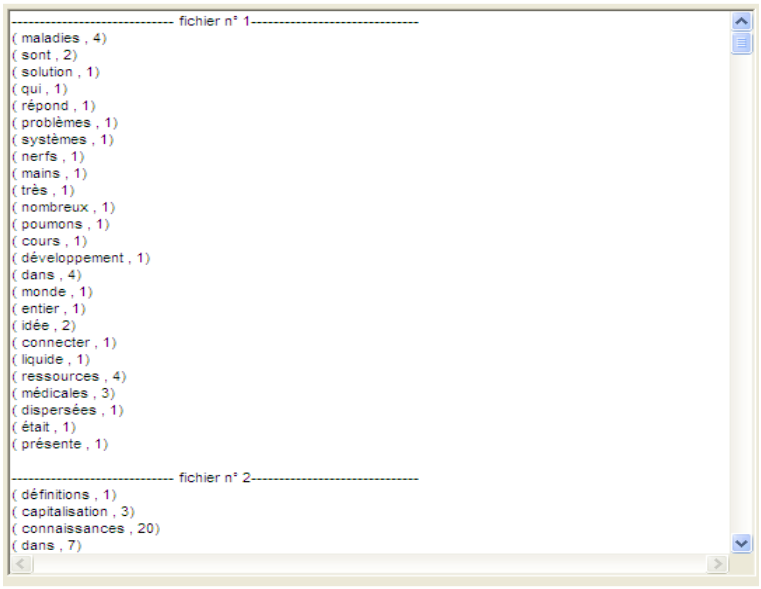

Fig. 4.4: Exemple d'index syntaxique

La phase de l'indexation syntaxique (élimination des stop-words et calcul du tf) se résume par l'algorithme suivant (Algorithm 1) :

```
    Require: docs : liste_Document
    Output: Index_Syn : index
1.1 i,j,k : entier ; /* i : Indice lignes ; j : Indice termes des ligne ;
    k : Indice fichiers ;                                              */;
1.2 mot_vide : booleen ; ligne : Ligne;
    /* Type Ligne (contient une liste de termes et une longueur)       */;
1.3 For k = 0 ; i < docs.nombre ; i++ do
        /* Faire ce traitement pour tous les fichiers                  */;
1.4     ouvrir_fichier_enCours();
1.5     For i = 0 ; i < fichier_enCours().longueur ; i++ do
            /* Faire ce traitement pour toutes les lignes              */;
1.6         ligne <- fichier_enCours().ligne(i);
            /* Récupérer la première ligne du fichier                 */;
1.7         For j = 0 ; i < ligne(i).nbr_termes ; i++ do
1.8             mot_vide <- Comparer(ligne[i].terme[j],Stop_List);
1.9             If mot_vide = False then
                /* Si le terme n'est pas un mot vide                  */;
1.10            If Index_Syn.Contient(ligne[i].terme[j]) = False then
                    /* Le terme n'existe pas dans l'Index_Syn         */;
1.11                Index_Syn.ajouter(ligne[i].terme[j],0);
                    /* tf=0 si terme n'existe pas dans index          */;
1.12            ELSE
1.13                Index_Syn.incrementer_tf(ligne[i].terme[j]);
                    /* Terme existe : incrémenter le tf               */;
1.14                END If;
1.15            END For;
1.16        END;
1.17    If fin_fichier() = true then
1.18        index_syn.New_ligne() ; /* Passer à la ligne pr doc suivant */
1.19    ;
1.20 END;
```

Algorithm 1: Indexation Syntaxique

L'algorithme précédent (Algorithm 1) se déroule ainsi :

- On fait le traitement suivant pour tous les fichiers :
- Ouvrir le fichier en cours, et pour toutes les lignes du fichier, faire le traitement suivant :
- Pour la ligne en cours, on compare tous ses termes avec la stop liste, pour déterminer si le terme est un mot vide.
- Si le terme n'est pas un mot vide et qu'il ne figure pas déjà dans l'index_Syn, on l'ajoute et son *tf* sera égal à 0. Sinon, si le terme figure dans l'index_Syn alors on incrémente son *tf* (sa fréquence d'apparition).
- Sinon si le terme est un mot vide, il sera alors ignoré.
- A la fin du fichier en cours, on fait un saut de ligne dans l'index_Syn, afin de stocker, s'il y en a, les termes du fichier suivant.

b- Indexation sémantique

Cette phase accepte en entrée les informations issues de l'indexation syntaxique. Elle est constituée des étapes suivantes (voir la Figure 4.5) :

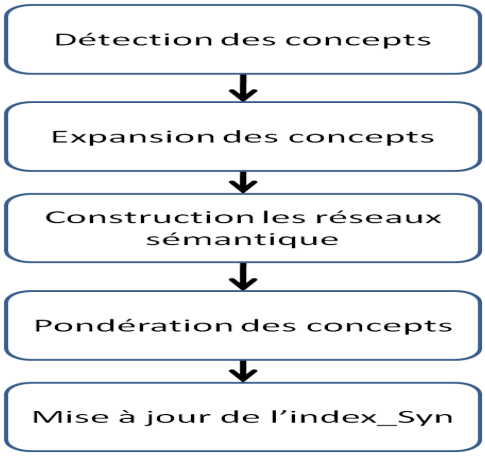

Fig. 4.5: Phases de l'indexation sémantique

Dans l'Index_Syn (issu de l'indexation syntaxique) et pour chaque document, on fait :

– **Détection des concepts**

Un terme est dit concept s'il appartient à au moins une entrée de l'ontologie.

La détection des concepts se fait en prenant les termes un à un et en les projetant sur l'ontologie pour détecter ce qui est concept de ce qui ne l'est pas.

Exemple : "Diabète" est un concept car il correspond à une entrée de l'ontologie qui est de type maladie alors que le terme "agir" par exemple ne sera pas pris comme concept.

A la différence de l'approche de Baziz et al. [26], le cas de détection des concepts formés par mots adjacents exemple : " infections pulmonaires ", n'est pas traité. On aura alors le terme " infections " qui sera pris comme un premier concept ainsi que " pulmonaire " qui sera pris comme un second concept.

Les concepts multi mots formés par l'approche de Baziz [26] sont rarement ambigus. Mais pour des raisons d'optimisation de la performance, d'amélioration de vitesse d'exécution et afin de diminuer la complexité de l'algorithme, nous avons opté pour cette démarche qui ne traite que les mots simples.

Dans cette étape, on fait la projection du document sur l'ontologie.

– **Expansion des concepts en utilisant l'ontologie**

Pour chaque concept détecté, un traitement spécifique est fait, il consiste à :

* Détecter les liens entre les différents concepts et de les relier ensemble en se basant sur l'ontologie.

* Etendre les concepts par leurs synonymes, dérivés et concepts de la même famille.

Une liste de synonymes et de sens sera attribuée à chaque concept, des liens et des relations entre concepts seront créés, ce qui forme une sorte de réseau pour chaque concept.

Dans cette étape on fait la projection de l'ontologie sur le document.

– **Construction du réseau sémantique**

Dans les deux étapes précédentes, nous avons utilisé notre ontologie pour représenter le contenu des documents sous forme de :

- Concepts.

- Relations entre concepts.

Nous avons eu comme résultat et pour chaque concept un nombre de termes qui lui sont associés. On appellera cette association le réseau sémantique du terme.

Nous définissons un réseau sémantique comme suit :

Définition

L'ensemble de termes R_k= { t_1,...,t_p } forme un réseau avec un terme t du document doc, c'est-à-dire :

- R_k est formé par l'union des éléments n_i de type terme qui sont en relation avec t, où relation ={ synonymie, dérivé, même_famille, ...}.

- ET $\forall\ t(n_i) \in R_k$, $t(n_i) \in doc$.

– **Pondération des concepts**

Il existe plusieurs approches pour pondérer les termes significatifs d'un document ou d'une requête. Nombre d'entre elles se basent sur les facteurs *tf* et *idf* qui permettent de considérer les pondérations locales et globales d'un terme.

Dans ce mémoire, on distingue la fréquence d'apparition d'un terme dans un document (term frequency, *tf*) qui a été déjà calculée dans l'indexation syntaxique et la fréquence d'apparition de ce même terme dans toute la collection considérée (inverse document frequency, *idf*).

La mesure *tf*idf* permet d'approximer la représentativité d'un terme dans un document, surtout dans les corpus de documents de tailles homogènes [27].

Dans notre approche, nous allons utiliser une variante de cette mesure qui le *Cf*idf*.

* Calcul du Cf

Le Cf est alors non pas la fréquence du terme mais celle du concept. Pour être calculé, on doit ajouter au *tf* du terme initial tous les *tf* des termes qui lui ont été associés dans le même document lors des phases précédentes, i.e les *tf* des termes du réseau sémantique.

Cf = *tf(t)* + $\sum$ *tf*(réseau). tel que t est le terme en cours et *tf*(réseau) sont tous les termes du réseau.

* Calcul du *idf*

Le *idf* est le nombre d'occurrences d'un terme dans tous les documents.

idf sera calculé comme suit : $idf(t) = \sum_{j=1}^{nbr_Doc} tf(t)$.

– **Mise à jour de l'index_Syn**

Pour faire la mise à jour de l'index_syn on ne prend en compte que les concepts ayant une entrée dans l'ontologie (c'est à dire les concepts ayant été détectés). Les termes qui appartiennent à leur réseau seront pris comme étant des concepts aussi avec la même pondération (qui est la somme) (enrichir l'index_Syn avec les termes du réseau créés ex : "synonymes").

On aura en résultat, un index sémantique (Index_Sem) qui est constitué comme suit :

N°document -> {{(concept_1$_k$, i), (concept_1$_{k+1}$,i)}, ... , {(concept_n_k,j), (concept_n_{k+1}, j)}}.

i,j : pondération du concept selon *Cf.idf* déjà calculé, k :1... m / m :entier.

Le résultat de cette étape est l'index sémantique (Index_Sem), il sera exploité par le service web S1 d'indexation.

La phase de l'indexation sémantique se résume par l'algorithme suivant (Algorithm 2) :

```
    Require: Index_Syn : index
    Output: Index_Sem : index sémantique
2.1 k : entier ; /* k : indice ;                                    */;
2.2 est_Concept : booleen ;
2.3 For k = 0 ; k < Index_Syn.nombre_Termes ; k++ do
        /* Faire ce traitement pour tous les termes de Index_Syn   */;
2.4     est_concept <– projection_sur_ontologie();
2.5     If est_concept = True then
            /* Le terme correspond à une entrée de l'ontologie     */;
2.6         expansion_avec_ontologie();
            /* Détecter les liens entre concepts et les étendre    */;
2.7         construction_réseau_sém();
            /* Le concept et les termes qui lui sont associés      */;
2.8         pondération_Concept();
            /* Le concept sera pondéré selon Cf.idf                */;
            /* Ces différentes étapes sont exécutées seulement si le
            terme est un concept                                    */;
2.9     Else /* Le terme sera alors ignoré                         */;
2.10    END If ;
2.11    MAJ_index_Syn();
        /* MAJ de l'index_Syn -> Seulement les concepts sont pris  */;
        /* On aura en sortie Index_Sem                              */
2.12 ;
2.13 END
```

Algorithm 2: Indexation Sémantique

L'algorithme précédent (Algorithm 2) se déroule ainsi :

- On fait le traitement suivant pour tous les termes de l'index_Syn (index syntaxique) :
- Projection du terme en cours sur l'ontologie.
- Si le terme est concept, faire le traitement suivant :
- Expansion du terme avec l'ontologie, ce que générera un réseau sémantique, on fait ensuite la pondération des concepts du réseau sémantique.
- Sinon si le terme n'est pas concept (ne correspond à aucune entrée de l'ontologie), il sera alors ignoré.
- La mise à jour de l'index_Syn est faite ensuite, en ignorant les termes non concepts, et en ajoutant les nouveau terme du réseau sémantique.

Création d'un index inversé

Cette étape n'est pas nécessaire pour l'indexation sémantique, mais elle est utile pour faire la création de l'index qui sera utilisé par le service web S2 d'indexation.
La création de l'index inversé consiste à : faire correspondre à tous les concepts, les documents qui les contiennent avec leurs pondérations, pour cela on doit regrouper tous les documents dans lesquels un concept apparait.

`Concept 1 −− > <document contenant le concept et sa pondération>`

`Concept 2 −− > <document contenant le concept et sa pondération >`

...

`Concept n −− > <document contenant le concept et sa pondération >`

Pour créer cette liste :

A partir de l'Index_Sem et pour tout concept de chaque document on fait un parcours de la liste qu'on veut créer et qui correspond aux concepts avec leurs documents, si le concept figure déjà on ajoute le document en cours aux autres documents contenant ce concept, sinon on ajoute le concept à la fin de la liste, ce processus est répété jusqu'à ce que tous les concepts de tous les documents soient traités.
Le résultat de cette étape (Index_Sem_Inv), l'Index Sémantique Inversé sera exploité par le service web S2 d'indexation.

L'algorithme suivant (Algorithm 3), permet de faire la création de l'index inversé :

```
Require: Index_Sem : index sémantique
Output: Index_Sem_Inv : index sémantique inversé
3.1 k, p : entier ; /* k : indice ;                                    */;
3.2 For k = 0 ; k < Index_Sem.nombreConcepts ; k++ do
        /* Faire ce traitement pour tous les concepts de tous les
        documents à partir de Index_Sem                               */;
3.3     p <– pondération ;
        /* Pondération du concept en cours dans le document           */;
3.4     If concept.enCours.existDans_Index_Sem_Inv() = True then
            /* si le concept existe déjà dans Index_Sem_Inv           */;
3.5         Index_Sem_Inv[concept.enCours].Ajout[num_Doc, p];
            /* ajouter le document en cours aux autres documents
            contenant ce concept dans Index_Sem_Inv avec pondération
            */;
3.6     Else Index_Sem_Inv[concept.enCours].Ajout[concept, num_Doc, p];
        /* ajouter à l'index : concept, document et pondération       */;
3.7     END If;
3.8     END
```

Algorithm 3: Création de l'index inversé

L'algorithme précédent (Algorithm 3) se déroule ainsi :

- On fait le traitement suivant pour tous les documents, à partir de l'Index_Sem (Index sémantique) :
- Faire ce traitement pour tous les concepts :
- Si un concept apparait déjà dans l'index inversé, alors ajouter le document en cours avec la pondération du concept à l'index inversé (Index_Sem_Inv).
- Sinon si un concept n'apparait pas déjà dans l'index inversé, alors ajouter le concept, le document en cours et la pondération du concept à l'index inversé (Index_Sem_Inv).

4.1.2 Indexation manuelle

Comme nous avons dit précédemment, ce type d'indexation est fait manuellement, c'est-à-dire que les documents seront annotés manuellement par des médecins (des utilisateurs) sur les ressources non textuelles, ex : une vidéo, une image (Figure 4.6).

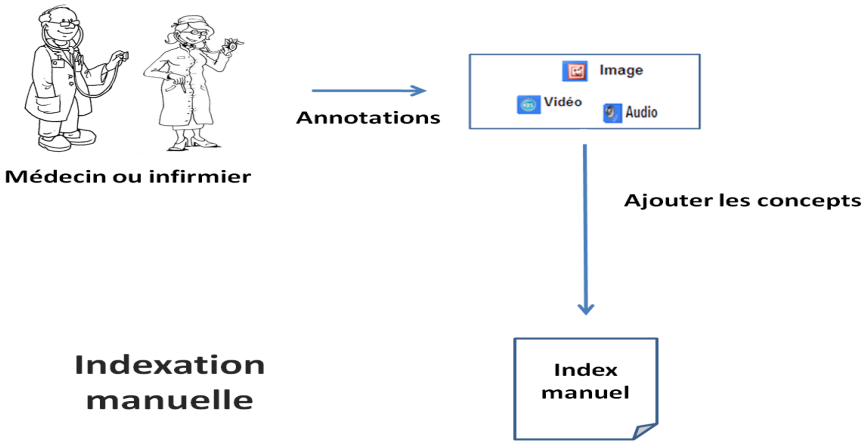

Fig. 4.6: Indexation manuelle

Création de l'index Sémantique manuel

Après l'annotation manuelle d'un fichier non textuel, faite par un spécialiste (médecin par exemple) en se basant sur les concepts d'une ontologie, ces derniers (les concepts) seront stockés dans un index et serviront pour indexer le fichier. On considère que pour l'indexation manuelle, un fichier est pertinent pour un concept, s'il est annoté par ce concept et qu'on n'a pas la notion de tf et $Cf.IDF$. Le fichier sera alors retourné en résultat s'il contient le concept recherché.

On aura alors pour un fichier f, un ensemble de couple (concept, instance) :

`N°fichier -> {{concept_1, instance }, ..., { concept_`n`, instance}}.`

Le résultat de cette étape est un index manuel (Index_Sem_Man).

Remarque :

Plusieurs travaux de recherche ont été réalisés dans le cadre de l'indexation manuelle basée sur les ontologies, on peut citer :

- Un système d'indexation sémantique des images cérébrales qui a été réalisé par Gaouar et al. [28].
- La conception et l'utilisation d'ontologies pour l'indexation de documents audio-visuels par A. Isaac [6].

Ces travaux peuvent être exploités, pour une éventuelle amélioration de notre travail.

Cr§ation de l'index S§mantique manuel invers§

Pour chaque instance de concept, on recherche les documents qui ont été annotés par cette même instance, et on les regroupe, ce qui permet d'avoir :

```
Instance 1 -> <documents contenant l'instance>
...
Instance n -> <documents contenant l'instance>
```

Le résultat de cette étape est un index manuel inversé (Index_Sem_Man_inv).

4.2 Recherche dans l'index

Quatre index seront utilisés par les services web d'indexation :
Pour l'indexation automatique, le service web S1 utilisera (Index_Sem) et le service web S2 utilisera (Index_Sem_Inv).
Pour l'indexation manuelle, le service web S1 utilisera (Index_Sem_Man) et le service web S2 utilisera (Index_Sem_Man_inv).

Pour les deux services web, la recherche est effectuée dans les index manuels et dans les index automatiques. La recherche se fait en évaluant la requête avec le respect d'un seuil et la prise en considération des pondérations dans le cas de l'indexation automatique.

1. Index_Sem (Service S1) : permet de faire entrer un code d'un document (un fichier), et la liste des concepts que contient ce document sera retournée et ordonnée par la pondération pour l'indexation automatique après avoir filtrer les concepts ayant une pondération inférieure à un certain seuil. Les fichiers indexés manuellement sont retournés automatiquement (Index_Sem_Man).

2. Index_Sem_inv (Service S2) : permet de faire entrer un concept, et la liste de documents qui concerne le concept sera retournée, ils seront ordonnés par ordre décroissant du *tf* (fréquence d'apparition du concept dans le document).
   On considère pour l'index manuel inversé (Index_Sem_Man_inv) qu'un document est automatiquement retourné et qu'il est pertinent pour le concept, car il a été annoté manuellement par un expert.

4.3 Conclusion

Cette dernière partie nous a permis de détailler et de mettre le point sur les fonctionnalités de l'approche proposée. Nous avons commencé par la présentation de l'architecture du système global de médiation, afin de situer notre partie. Nous avons donné ensuite un exemple d'ontologie de médiation, à partir duquel les services web d'indexation que nous avons proposé ont été décrits, sous formes de vues RDF pour permettre leur intégration avec les services web d'interrogation dans le système global. Nous avons également décrit les différentes étapes de la construction des index exploitables par nos services web d'indexation à l'aide de diagrammes. Des algorithmes sont présentés aussi pour décrire les différentes étapes de l'approche d'indexation proposée. Le chapitre suivant permettra de concrétiser notre conception par la présentation des étapes d'implémentation d'un prototype.

Troisième partie

Prototype

5

Conception d'un prototype

5.1 Introduction

Afin de concrétiser et de valider notre approche d'indexation et d'utilisation des deux services web proposés, nous avons conçu un prototype reflétant le fonctionnement de notre proposition, nous avons réalisé une première version de ce prototype dont les résultats font l'objet du présent chapitre. Nous commencerons d'abord par décrire l'environnement dans lequel nous avons réalisé le prototype puis nous discuterons les résultats que nous avons obtenu.

5.2 Environnement de développement

Nous avons développé notre application sur une machine Intel Core2duo, avec une vitesse de 1.6 GHZ, dotée d'une capacité mémoire de 2GB de RAM sous Windows XP. L'application est développée en utilisant le langage de programmation Java.

5.3 Pourquoi java ?

Java est un langage de programmation orienté objet et un environnement d'exécution récent, développé par Sun Microsystems en 1991.
Java possède de nombreuses caractéristiques qui font de lui un langage de choix, il permet de développer des applications client-serveur. Coté client, les applets sont à l'origine de la notoriété du langage. C'est surtout coté serveur que java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (Java Server Pages) qui peuvent se substituer à PHP, ASP et ASP.NET [29].

5.4 Description du fonctionnement de notre prototype

Dans cette partie, nous allons présenter les interfaces de notre application à travers un exemple qui a été réalisé.

5.4.1 Accès à l'application

La Figure 5.1 est la première interface de notre prototype qui apparaît à l'utilisateur, elle représente l'interface du lancement du prototype.

Fig. 5.1: Première interface de notre prototype

Cliquer sur " Continuer " permet le lancement du prototype, une autre interface apparait, elle représente la fenêtre principale de notre prototype (voir Figure 5.2).

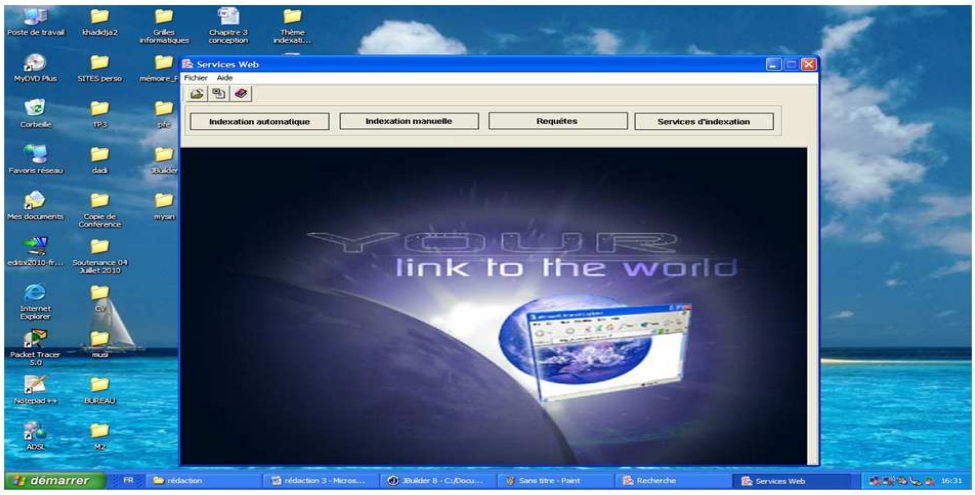

Fig. 5.2: Fenêtre principale de notre prototype

La barre du menu contient deux menus Fichier et Aide. Chaque menu est composé d'un ensemble d'Items. Des raccourcis vers les menus les plus importants, existent sous forme de boutons au menu principal (voir Figure 5.3).

Fig. 5.3: Boutons du Menu principal

A partir du menu principal (Figure 5.3), l'utilisateur a le choix entre quatre possibilités :

1- Lancer une indexation automatique

Ça consiste à faire une indexation sémantique automatiquement, à un corpus de fichiers (un répertoire).

2- Lancer une indexation manuelle

L'indexation manuelle, permet d'annoter manuellement les fichiers non textuels, tels que les images.

3- Écrire une requête

Cela permet de rendre une réponse à une requête plus ou moins complexe, en faisant une combinaison des réponses partielles des termes composant la requête.

4- Utiliser les services web d'indexation

L'utilisation des services web permet d'exploiter les index créés par les phases d'indexation précédentes.

5.4.2 Utilisation du prototype

Le prototype a pour but d'étudier la validité de l'approche proposée. Nous allons présenté dans cette partie quelques captures d'écran des utilisations de notre prototype.

1. **L'indexation automatique**

Si l'utilisateur clique sur le bouton " Indexation automatique " du menu principal (Figure 5.3), une interface apparait, offrant la possibilité de lancer l'indexation automatique d'un ensemble de fichiers (voir Figure 5.4).

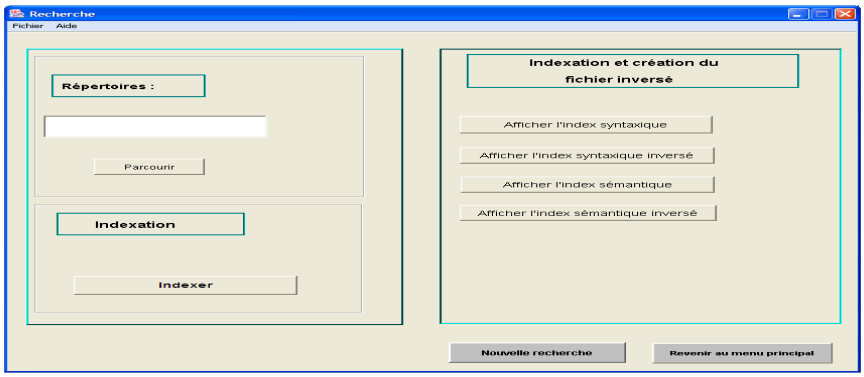

Fig. 5.4: Interface d'indexation automatique

L'utilisation de cette interface est comme suit (Figure 5.5) :

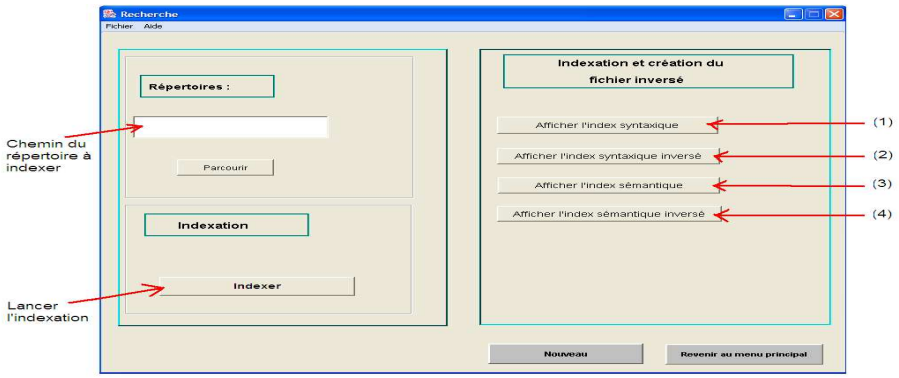

Fig. 5.5: Utilisation de l'interface d'indexation automatique

Dans la partie gauche de l'interface (Figure 5.5), l'utilisateur peut sélectionner le chemin du répertoire à indexer, en cliquant sur le bouton " Parcourir ", il peut ensuite lancer l'indexation en cliquant sur " Indexer ".

Dans la partie droite de l'interface (Figure 5.5), l'utilisateur peut visualiser les différents index créés lors de la phase d'indexation. Il peut afficher :

(a) L'index syntaxique : créé lors de la phase de l'indexation syntaxique (Index_Syn).

(b) L'index syntaxique inversé : c'est le résultat inversé de l'index créé par la phase d'indexation syntaxique.

(c) L'index sémantique : créé lors de la phase d'indexation sémantique à partir de l'index syntaxique (Index_Sem).

(d) L'index sémantique inversé : créé de la phase de construction de l'index sémantique inversé (Index_Sem_Inv).

2. **L'indexation manuelle**

Si l'utilisateur clique sur le bouton " Indexation manuelle " du menu principal (Figure 5.3), une interface apparait, offrant la possibilité de faire manuellement l'indexation des fichiers non textuels (voir Figure 5.6).

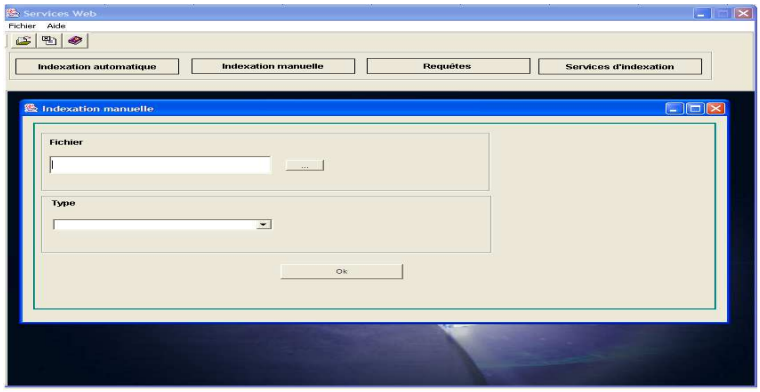

Fig. 5.6: Interface d'indexation manuelle

L'utilisation de l'interface est comme suit (voir Figure 5.7) :

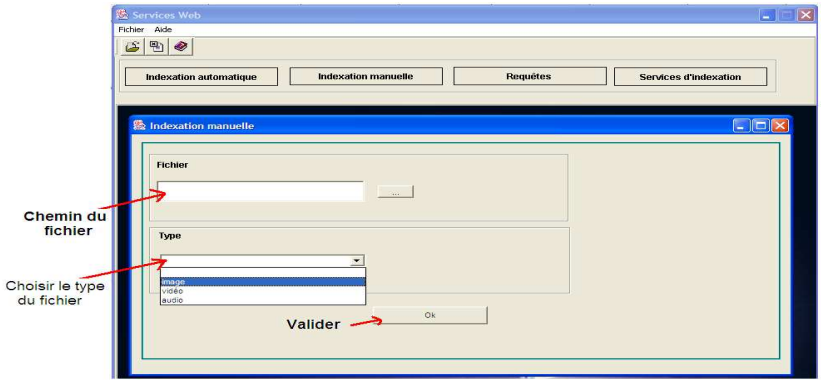

Fig. 5.7: Utilisation de l'interface d'indexation manuelle

L'utilisateur doit indiquer le chemin du fichier qu'il veut indexer manuellement, et il choisit son type : (image, vidéo, audio), chaque type a son traitement spécifique.

3. **Les requêtes**

Si l'utilisateur clique sur le bouton " Requêtes " du menu principal (Figure 5.3), une interface apparait, elle offre la possibilité de faire la saisie d'une requête (voir Figure 5.8).

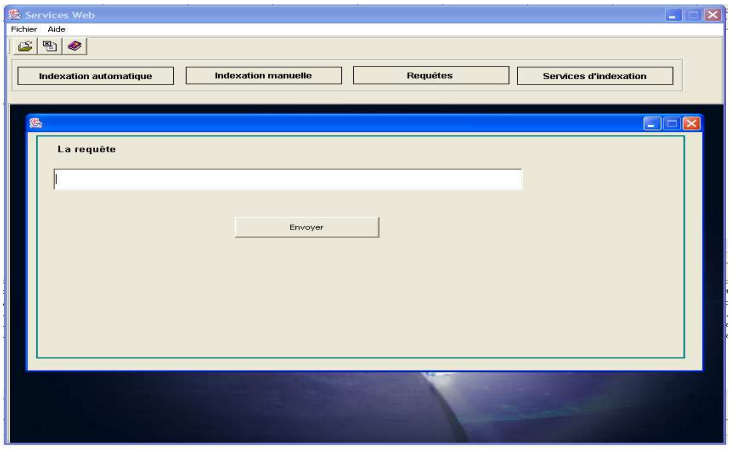

Fig. 5.8: Interface des requêtes

Pour cela, une zone est réservée pour faire la saisie de la requête.

Cliquer sur " Envoyer " permet de valider la saisie de la requête (Figure 5.9).

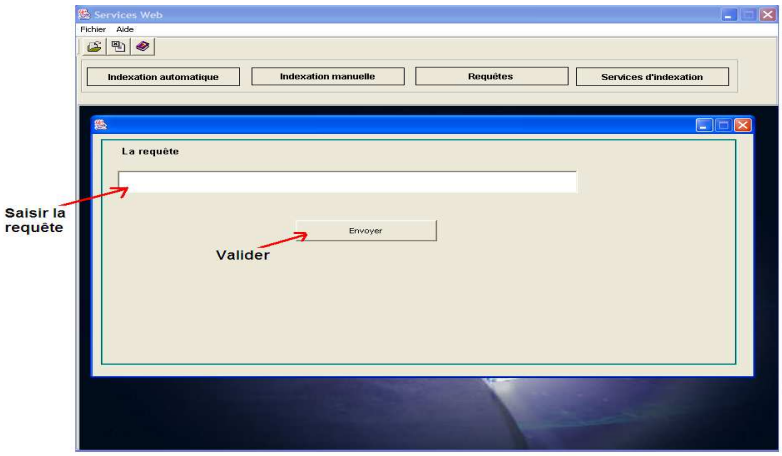

Fig. 5.9: Utilisation de l'interface des requêtes

4. **Les services web**

Si l'utilisateur clique sur le bouton " Services d'indexation " du menu principal (Figure 5.3), une interface apparait, elle offre la possibilité de tester les deux services web d'indexation que nous avons proposé (un services pour avoir comme résultat les fichiers à partir des concepts, et un service pour avoir les concepts à partir d'un fichier) (voir Figure 5.10).

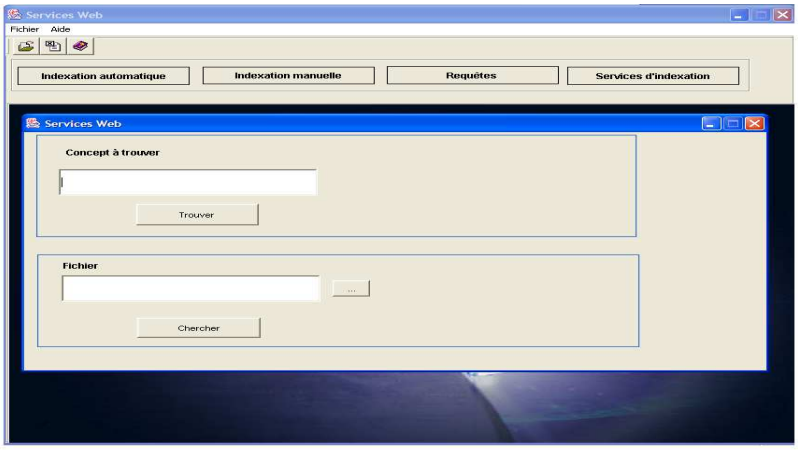

Fig. 5.10: Services Web d'indexation

L'explication de l'utilisation des deux services web est représentée par la Figure 5.11 :

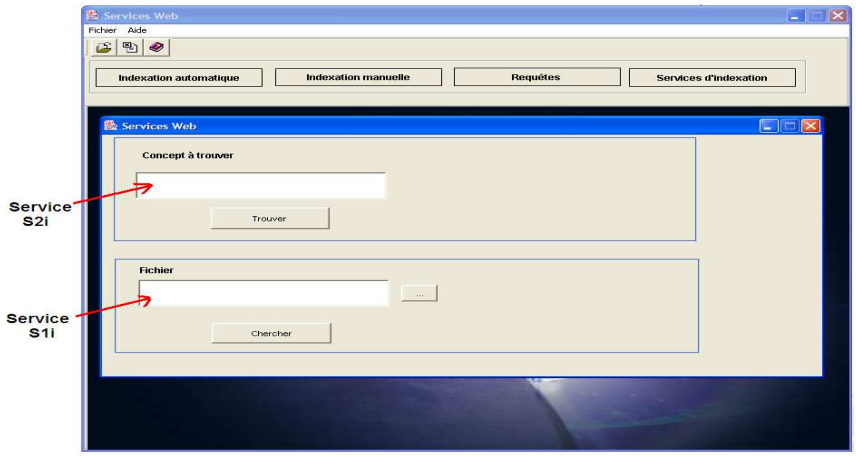

Fig. 5.11: Utilisation de l'interface des services Web d'indexation

Dans la Figure 5.11 :

1. On peut remarque que l'utilisateur peut saisir le concept qu'il souhaite rechercher dans la zone réservée au 'Concept'. En cliquent sur " Trouver ", une recherche est lancée afin de chercher le concept et retourner tous les fichiers contenant ce concept. La recherche est faite dans l'index sémantique inversé (Index_Sem_Inv) et dans l'index sémantique manuel inversé (Index_Sem_Man_Inv) , car ce sont les index qui contiennent les concepts avec leurs fichiers correspondants.
   Ce traitement représente le fonctionnement du service web S2i d'indexation (comme il a été définit dans la partie conception).
2. L'utilisateur peut aussi indiquer le chemin d'un fichier dans la zone 'Fichier'. En cliquant sur " Chercher ", une recherche est lancée afin de trouver tous les concept du fichier sélectionné. La recherche est faite dans l'index sémantique (Index_Sem) et dans l'index sémantique Manuel (Index_Sem_Man), car ce sont ces index qui contiennent tous les fichiers avec les concepts qui leurs indexent.
   Ce traitement représente le fonctionnement du service web S1i d'indexation (comme il a été défini dans la partie conception).

5.5 Conclusion

Dans ce chapitre, nous avons présenté notre prototype, i.e une application qui démontre la faisabilité de l'approche d'indexation et l'utilisation des deux services web proposés. Nous avons commencé par décrire le langage de programmation utilisé pour son développement ainsi que quelques interfaces du prototype. Nous avons également montré un exemple d'utilisation du prototype avec ses différents étapes. Nous notons que dans un travail ultérieur, cette application pourra être intégrée dans un JSP (Java Server Pages) ou applet pour qu'elle puisse être destinée au web, et donc traiter les données distribuées.

Conclusion générale et perspectives

Dans ce mémoire, nous avons présenté une approche d'indexation sémantique des sources de données hétérogènes et distribuées qui se base sur des services web. Ces services web exploitent les index générés de l'indexation.
Le but des services web d'indexation proposés est non seulement d'exploiter les index, et alors de rendre possible la recherche dans des sources de données (semi ou non structurées), mais aussi d'être intégrés et réutilisés dans un cadre global d'intégration, et par d'autres services web d'interrogation qui concernent les sources de données structurés. Le but de cette intégration est de gérer l'hétérogénéité, de considérer toutes les sources d'information, et les rendre en réponse aux utilisateurs. Cela est possible grâce à une réécriture des requêtes qui se fera selon des vues spécifiques aux différents services web définis en RDF dans le système d'intégration global.

Notre approche n'est donc qu'une première phase dans le cadre général d'un travail de recherche concernant la composition des services web pour l'interrogation des sources de données hétérogènes et distribuées.

L'approche d'indexation proposée dans ce travail, peut être considérée comme une indexation des sources semi ou non structurées dans un cadre applicatif qui est le domaine médical, l'approche s'appuie sur deux types d'indexation :

- Une indexation manuelle pour les sources non textuelles tel que les vidéos, les images,... etc. Ce type d'indexation s'effectue manuellement par des experts.

- Une indexation automatique pour les documents textuels. Elle se fait automatiquement en deux phases, une phase d'indexation syntaxique et une autre phase d'indexation sémantique.

Nous avons pu proposer deux services web d'indexation qui exploitent les index issus de notre approche d'indexation, et représentent des vues intégrables dans le système global.

- Le premier service web S1i : permet d'avoir tous les concepts qui indexent un fichier.
- Le deuxième service web S2i : permet d'avoir les fichiers portant sur un concept. Les concepts sont identifiés en se basant sur un exemple d'ontologie de médiation du domaine médical.

Pour une continuation de notre travail, plusieurs perspectives peuvent être envisagées :

- Nos travaux futurs consisteront à proposer et implémenter une approche d'indexation manuelle pour tous les fichiers non textuels, ou intégrer d'autres travaux d'indexation manuelle existants tels que le travail de Gaouar et Benyeless [28] pour l'indexation des images par exemple.
- Tester les performances de notre approche d'indexation en la comparant à d'autres approches selon des métriques qu'on définira.
- Étendre notre approche par la possibilité de faire le suivi de la dynamique (ajout, suppression, modification) des fichiers et de mettre à jour les index selon le cas.
- Compléter toutes les fonctionnalités du prototype et l'améliorer et l'intégrer dans le système global (avec la partie interrogation).
- Modifier l'application (le prototype) pour qu'elle soit destinée au web.
- Il est à signaler qu'une éventuelle amélioration du service web S2 d'indexation en introduisant la possibilité de faire plusieurs recherches simultanées, et combiner les résultats (par des opérateurs logiques par exemple) pourra résoudre le problème des requêtes non simples comportant plusieurs concepts (des phrases).

Bibliographie

[1] P. Laublet, C. Reynaud, and J. Charlet. *Sur quelques aspects du web sémantique.* In *IPPS/SPDP '99/JSSPP '99 : Proceedings of the Job Scheduling Strategies for Parallel Processing*, pp. 162–178 (London, UK, 1999).

[2] O. L. Tim Berners-Lee, James Hendler. *The semantic web.* Scientific American **22**, 3 (2001).

[3] M. Barhamgi. *Composing DaaS Web Services Application to eHealth.* Ph.D. thesis, Université Claude Bernard Lyon I (2010).

[4] A. Boukhadra. *La composition dynamique des services Web sémantiques a base d'alignement des ontologies owl-s.* Mémoire de magister, Ecole national Supérieur d'Informatique, Alger (2011).

[5] B.-L. Tim, H. James, and L. Ora. *The semantic web.* Scientific American (May 2001).

[6] A. Isaac. *Conception et utilisation d'ontologies pour l'indexation de documents audiovisuels.* thèse de doctorat, Université Paris IV – Sorbonne, France (Décembre 2005).

[7] W3C. URL `http ://www.w3.org/TR/ws-arch/`.

[8] V. Mezaris, L. Kompatsiaris, M.ichael, and G. Strintzis. *Region based image retrieval using an object ontology and relevance feedback*. Eurasip Journal on Applied Signal Processing (2004).

[9] J. Charlet, P. Laublet, and C. Reynaud. *Web sémantique quels apports pour la médecine*. 1ere journée WSM, Rennes (2003).

[10] xmlfr.org. *Rdf/rdfs*. URL `http ://xmlfr.org/documentations/tutoriels/`.

[11] W3C. *Rdf*. URL `http ://www.w3.org/RDF/`.

[12] J. Baget. *Homomorphismes d'hypergraphes pour la subsomption en rdf*. In *Actes 3e journées nationales sur modèles de raisonnement (JNMR), Paris (FR)*, pp. 1–24 (2003).

[13] H. henni M'hamed. *Approche ontologique pour la modélisation sémantique, l'indexation et l'interrogation des documents Coraniques*. Mémoire de magister, Ecole Supérieur d'Informatique, Oued-Smar, Alger.

[14] *Information et documentation, principes généraux pour l'indexation des documents*. Rapport technique, AFNOR (1993).

[15] G. Hubert, J. Mothe, B. Ralalason, and B. Ramamonjisoa. *Modèle d'indexation dynamique à base d'ontologies*. Conférence en Recherche d'Information et Applications (2009).

[16] M. E. H. Widad. *Indexation humaine et indexation automatisée : la place du terme et des environnements*. ufr idist/cersates, lille 3, France (2004).

[17] Y. Prié. *Sur la piste de l'indexation conceptuelle de documents, une approche par l'annotation lisi*. UFR Informatique, Université Claude Bernard Lyon 1 (2000).

[18] M. Baziz, M. Boughanem, N. Aussenac-Gilles, and C. Chrisment. *Semantic cores for representing documents in ir*. In *SAC*, pp. 1011–1017 (2005).

[19] N. Seco, T. Veale, , and J. Hayes. *An intrinsic information content metric for semantic similarity in wordnet.* In *In Proceedings of ECAI'2004, the 16th European Conference on Artificial Intelligence* (2004).

[20] J. F. Song, Z. W. Ming, X. W. Dong, L. G. Hui, and X. Z. Ning. *Ontology-based information retrieval model for the semantic web.* In *International Conference on e-Technology, e-Commerce and e-Service, EEE '05*, pp. 152 – 155 (2005).

[21] W. Gio. *Mediators in the architecture of future information systems.* In *Computer*, pp. 38–49 (1992).

[22] N. Faraj, R. Godin, R. Missaoui, S. David, and P. Plante. *Analyse d'une méthode d'indexation automatique basée sur une analyse syntaxique de texte.* In *Canadian Journal of Information and Library Science*, pp. 1–21 (Canada, 1996).

[23] B. Menon. *L'indexation à l'heure du numérique.* Journée d'étude ADBS, dans Documentaliste – Sciences de l'information (2004).

[24] M. Baziz, M. Boughanem, and N. Aussenac-Gilles. *Evaluating a conceptual indexing method by utilizing wordnet.* In *CLEF*, pp. 238–246 (2005).

[25] M. Baziz, M. Boughanem, and N. Aussenac-Gilles. *A conceptual indexing approach for the trec robust task.* In *TREC* (2005).

[26] M. Baziz, M. Boughanem, and S. Traboulsi. *A concept-based approach for indexing documents in ir.* In *INFORSID*, pp. 489–504 (2005).

[27] A. Ventresque. *Espaces vectoriels sémantiques : enrichissement et interprétation de requêtes dans un système d'information distribué et hétérogène.* thèse de doctorat, Université de Nantes, France (2008).

[28] I. Gaouar and A. Benyelles. *Un système d'indexation sémantique des images cérébrales.* Mémoire d'ingénieur d'état en informatique, Université de Tlemcen, Algérie (2010).

[29] Wikipedia. *java.* URL `http ://fr.wikipedia.org/wiki/Java`(*langage*).