Numéro d'identification :

UNIVERSITÉ DE BOURGOGNE
— U.F.R. DES SCIENCES ET TECHNIQUES —

THÈSE

présentée à l'Université de Bourgogne
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : **Informatique**
Formation Doctorale : **Informatique**
École Doctorale : **Environnements Santé STIC - E2S**

Enrichissement de requêtes et visualisation sémantique dans une coopération de systèmes d'information : méthodes et outils d'aide à la recherche d'information

par

Guillermo Valente GÓMEZ CARPIO

Soutenue le 14 Décembre 2010 devant le Jury composé de :

Danielle BOULANGER, Professeur, Université Jean Moulin - Lyon III, ... Rapporteur
Marie-Christine FAUVET, Professeur, Université Joseph Fourier, ... Rapporteur
Fabrice JOUANOT, Maître de conférences, Université Joseph Fourier ,..... Examinateur
Jean Marcel PALLO, Professeur, Université de Bourgogne, ... Examinateur
Nadine CULLOT, Professeur, Université de Bourgogne, ... Directeur de thèse
Lylia ABROUK, Maître de conférences, Université de Bourgogne, ... Co-encadrant

Numéro d'identification :

UNIVERSITÉ DE BOURGOGNE
— U.F.R. DES SCIENCES ET TECHNIQUES —

THÈSE

présentée à l'Université de Bourgogne
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : **Informatique**
Formation Doctorale : **Informatique**
École Doctorale : **Environnements Santé STIC - E2S**

Enrichissement de requêtes et visualisation sémantique dans une coopération de systèmes d'information : méthodes et outils d'aide à la recherche d'information

par

Guillermo Valente GÓMEZ CARPIO

Soutenue le 14 Décembre 2010 devant le Jury composé de :

Danielle BOULANGER, Professeur, Université Jean Moulin - Lyon III, ... Rapporteur
Marie-Christine FAUVET, Professeur, Université Joseph Fourier, ... Rapporteur
Fabrice JOUANOT, Maître de conférences, Université Joseph Fourier ,..... Examinateur
Jean Marcel PALLO, Professeur, Université de Bourgogne, ... Examinateur
Nadine CULLOT, Professeur, Université de Bourgogne, ... Directeur de thèse
Lylia ABROUK, Maître de conférences, Université de Bourgogne, ... Co-encadrant

Remerciements

La réalisation de cette thèse de doctorat fut une occasion merveilleuse de rencontrer et d'échanger avec de nombreuses personnes. Ce travail n'aurait pas pu être réalisé sans l'intervention d'un certain nombre de personnes qui m'ont apporté une aide précieuse.

Je tiens à exprimer tout d'abord mes remerciements aux membres du jury, qui ont accepté d'évaluer mon travail de thèse.

Je remercie tout particulièrement Mme Nadine CULLOT qui m'a appris la démarche de recherche scientifique, pour avoir fait preuve de patience et pour avoir accepté de diriger cette thèse. Je tiens également à la remercier pour la confiance et la sympathie qu'elle m'a témoignées au cours de ces années de thèse.

J'adresse mes plus sincères remerciements à Mme Lylia ABROUK pour son encadrement efficace et son soutien qui s'est avéré déterminant pour mener ce travail à terme.

À toutes les deux, je souhaiterais ici leur témoigner ma sincère reconnaissance pour tous les conseils et les remarques objectives apportées au cours de ces années.

Je remercie également Mme Danielle BOULANGER et Mme Maire-Christine FAUVET d'avoir accepté d'être les rapporteurs de ce manuscrit et pour l'intérêt qu'elles ont porté à ce travail.

J'aimerais également remercier Monsieur Jean-Marcel PALLO, Professeur à l'Université de Bourgogne, qui m'a fait l'honneur d'exercer les fonctions de Président du jury.

Je remercie aussi Monsieur Fabrice JOUANOT, pour avoir accepté d'être examinateur.

Je tiens enfin à remercier les amis, doctorants ou non qui m'ont aidé au cours de ces années de thèse.

Je dédie cette thèse
À Francisca MARQUEZ ADAME, mon épouse.
À Victoria et Kira Ségolène, nos filles.
À vous tous

Sommaire

Table des figures

Liste des tableaux

Introduction et Problématique

1

Introduction

Jusqu'à aujourd'hui, les caractéristiques des systèmes de recherche d'information ont considérablement évolué grâce à l'informatique et aux réseaux de communication, ainsi qu'à la croissance considérable du volume de données disponibles généré par ces derniers. Réunir suffisamment de données était le principal problème de la recherche d'information, mais avec la masse de données, l'objectif principal est devenu de localiser et d'interpréter les informations afin d'en extraire des connaissances. Pour cela, des approches et outils sont nécessaires afin de faciliter l'accès à l'information.

Dans un système de recherche d'information, nous retrouvons principalement deux éléments : les documents qui peuvent être : un texte, un morceau de texte, une page Web ou bien une image et une requête qui exprime le besoin d'information de l'utilisateur. Si un système de recherche d'information est un système qui permet de retrouver une information pertinente par rapport à une requête dans une grande collection de documents, alors, un document pertinent est un document qui doit contenir l'information que l'utilisateur recherche. Les résultats de la recherche dépendent de l'utilisateur et la requête soumise et reflètent la qualité d'un système de recherche d'information.

L'objectif du Web sémantique est de rendre le contenu accessible et interprétable par des agents logiciels. Ce nouveau Web ouvre la voie au développement de nouveaux outils en ajoutant une couche supplémentaire de connaissances afin d'assister l'utilisateur dans sa recherche. Le Web sémantique repose sur l'association de métadonnées explicites aux informations et leur mise en relation avec des ontologies. Les métadonnées explicites permettent d'ajouter des informations additionnelles afin de décrire les pages Web ou des ressources et les ontologies pour décrire un domaine avec ses concepts et les relations qui existent entre ses concepts. Par conséquent, la description sémantique des ressources est un concept fondamental du Web sémantique. Pour toutes ces raisons, nous nous intéressons dans notre travail au processus de recherche d'information et l'exploitation des données, basée sur l'utilisation de la sémantique des informations et les ontologies.

1.1 Contexte de la thèse

Notre travail s'inscrit dans le cadre d'un système de coopération basé sur des ontologies appelé OWSCIS (Ontology and Web Service based Cooperation of Information Sources) qui permet la coopération de sources de données hétérogènes. L'architecture de OWSCIS s'appuie sur des ontologies à deux niveaux : (i) au niveau des sources d'information avec plusieurs sources de données et (ii) au niveau de la coopération. Pour le premier niveau, des ontologies locales sont utilisées pour décrire la sémantique des informations, et une ontologie de référence sert à décrire la sémantique du domaine. Les ontologies locales sont mises en relation avec l'ontologie de référence et les sources d'information sont explicitées sémantiquement à l'aide des ontologies locales.

L'architecture d'OWSCIS est composée de plusieurs services, parmi lesquels un service d'interrogation et un service de visualisation. Le service d'interrogation permet à l'utilisateur de soumettre une requête sur l'ontologie de référence, nous nous intéressons dans notre travail au processus de recherche dans ce service pour l'amélioration de la pertinence des résultats. Le service de visualisation fournit à l'utilisateur une représentation "pertinente" des résultats de sa requête, l'utilisateur peut soumettre sa requête à travers ce service.

1.2 Problématique

Une coopération permet de mettre en relation des systèmes d'information variés. Faire coopérer des systèmes d'information nécessite de les relier sémantiquement. Le Web sémantique vise à fournir un moyen de partage des données issues de sources hétérogènes, l'idée principale est d'exploiter la description sémantique des ressources. Ceci, nécessite que ces ressources soient décrites par des annotations qui peuvent être exploitées par les utilisateurs ou les agents logiciels. La rapidité de l'évolution de la masse d'informations nécessite un besoin d'organisation des données, les ontologies sont apparues pour la représentation des données échangées afin de faciliter la communication entre les différents acteurs du domaine. Elles sont utilisées pour la représentation des données et pour la recherche d'information.

La recherche d'information basée sur la sémantique représente une méthode pertinente, néanmoins le processus de recherche d'information soulève quelques problèmes :

1. Le grand volume de données et l'accès aux informations pertinentes : avec la grande masse d'informations disponibles, la recherche d'information est devenue une tâche difficile qui demande un effort intellectuel considérable. L'utilisateur a besoin d'être assisté et aidé pendant son processus de recherche afin de retrouver et d'exploiter les réponses pertinentes répondant à son besoin.
2. La représentation des données : l'absence ou le manque de structure dans la représentation des résultats pour un utilisateur rend la navigation

difficile et l'exploitation du résultat quasiment impossible. La quantité spectaculaire de ressources sur le Web a généré un besoin de structuration et d'organisation afin de rendre les informations facilement exploitables.

Afin de reprendre les différents problèmes posés, la problématique de notre thèse peut se résumer à fournir des méthodes et des outils pour répondre à la question suivante :

Comment répondre aux besoins des utilisateurs pour la recherche d'information dans une grande masse de données, en les assistant et leur offrant des outils pour obtenir des résultats pertinents, d'une façon conviviale et dynamique ?

1.3 Contributions

Ce travail s'inscrit dans le cadre de la recherche d'information en utilisant les technologies du Web sémantique. Nous présentons dans ce qui suit les principales contributions de cette thèse. D'une manière générale, notre travail vise à améliorer l'architecture de coopération OWSCIS en proposant une architecture complète intégrant une interrogation et une visualisation basée sur la sémantique des informations.

Architecture de coopération basée sur les ontologies Au niveau global de la coopération, une ontologie de référence décrit la sémantique du domaine de coopération, cette ontologie est utilisée pour les services d'interrogation et de visualisation. Le service d'interrogation est responsable de la décomposition et recomposition de la requête pour son traitement. Il est composé du module d'enrichissement de requête. Le service de visualisation proposé pour améliorer l'architecture, se compose de trois modules de visualisation et un environnement de paramétrage.

Enrichissement de requête Notre contribution consiste à proposer une méthode d'enrichissement de requêtes appelé QUEXME (QUEry EXpansion MEthod) au niveau du module d'enrichissement de la requête du service d'interrogation de OWSCIS. QUEXME est une méthode basée sur l'analyse du comportement des utilisateurs. Il utilise un algorithme basé dans la notion d'importance d'un concept par rapport aux autres concepts et d'un ensemble de concepts par rapport à la requête.

Service de visualisation Notre contribution consiste à proposer une architecture du service de visualisation SEVI (SErvice de VIsualisation) intégrée dans OWSCIS. SEVI est un service composé de trois modules de visualisation : requête, enrichissement et résultats, ainsi qu'un environnement de paramétrage en 2 et 3 dimensions. Ce service vise à assister l'utilisateur dans son processus de recherche d'information afin de répondre à ses besoins

1.4 Approche

Dans ce travail de thèse, nous avons traité le problème de la recherche d'information en proposant une approche d'enrichissement de requête, complétée par un service de visualisation pour l'exploitation des données. Nous proposons une architecture pour les services d'interrogation et visualisation, pour cela, nous définissons une méthode d'enrichissement des requêtes et de visualisation de l'ontologie, de la requête et des résultats.

Un aperçu général de notre approche, illustrant les différentes phases, est présenté sur la figure 1.1. Les phases sont détaillées dans les chapitres suivants.

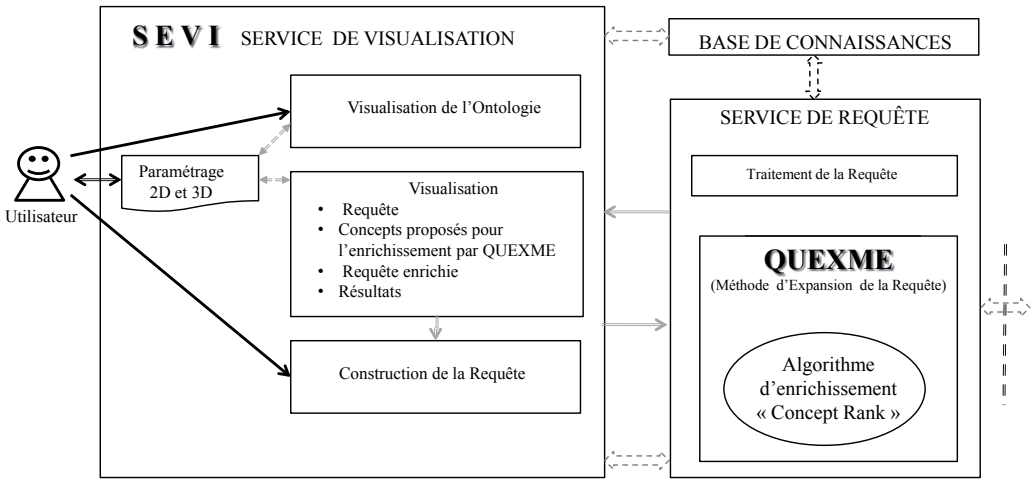

FIGURE 1.1 – Notre approche

1.5 Plan de la thèse

La suite de cette thèse est composée de trois parties.

1.5.1 Première partie

Dans la première partie, nous présentons l'état de l'art de notre travail composé de trois chapitres. Le chapitre 2 présente les concepts du Web sémantique et les ontologies. Les différents travaux dans le domaine de l'enrichissement des requêtes sont présentés dans le chapitre 3. Le chapitre 4 est consacré à une introduction générale au concept de visualisation. Nous présentons ensuite les travaux sur la visualisation d'ontologies.

1.5.2 Deuxième partie

Dans la deuxième partie, nous présentons notre proposition d'enrichissement de requêtes et de visualisation. Le chapitre 5 présente l'architecture OWSCIS et une vue générale de notre approche et des solutions proposées. Notre approche d'enrichissement de requêtes est détaillée dans le chapitre 6. Dans le chapitre suivant, nous présentons l'architecture et la méthode de visualisation du service de visualisation.

1.5.3 Troisième partie

La dernière partie de cette thèse est consacrée à l'expérimentation de notre approche ainsi qu'à l'évaluation des résultats. Le dernier chapitre présente l'expérimentation effectuée pour évaluer l'approche proposée. L'ontologie du SEMIDE (Système Euro-Méditerranéen d'Information sur les savoir-faire dans le Domaine de l'Eau) est utilisée pour valider la méthode d'enrichissement proposée.

Première partie

Etat de l'art

2

Le Web sémantique et les ontologies

Ce chapitre présente un aperçu des concepts généraux de notre travail. La section 2.1 présente le Web sémantique et ses différentes applications, la section 2.2 présente différentes définitions du terme ontologie, leurs classifications ainsi que leurs utilisations, enfin nous concluons dans la section 2.3.

Sommaire

2.1 Le Web sémantique

Le Web sémantique (*Semantic Web*) se présente comme une extension du Web classique. C'est un moyen d'extraction et d'exploitation de l'intelligence collective du Web, en rendant le contenu plus facile à échanger et à partager entre les différents utilisateurs. Ce nouveau Web vise à faciliter l'exploitation des ressources par les machines en accédant aux ressources grâce à leur représentation sémantique. Selon Goble et al. [BC07b], le Web sémantique vise à favoriser la découverte, l'échange d'information et l'intégration de l'information par l'étiquetage (*tagging*) ou le balisage (*marking up*), et à développer des technologies et infrastructures pour annoter le contenu du Web avec des informations additionnelles appelées "métadonnées" pouvant être traitées par des agents logiciels.

2.1.1 Définitions

Le Web classique se base essentiellement sur la structure du document et les liens entre les pages Web [BC07a], ce qui rend l'exploitation de l'information quasiment impossible par les machines. A la différence de cela, le Web sémantique offre une représentation sémantique du contenu, ce qui permet aux agents logiciels l'extraction de l'information. En 2001, Berners-Lee appelé "le père du Web" a introduit la notion du Web sémantique. Il le décrit [BLHL01] comme une extension du Web tel qu'il existait et permet de définir précisément la sémantique du contenu et ainsi de favoriser un meilleur traitement des requêtes des utilisateurs par des logiciels.

Nagarajan [Nag06] présente le Web sémantique comme *"...une vision avec l'idée de disposer de données sur le Web définies et liées de telle manière qu'il puisse être utilisé par les machines non seulement à des fins d'affichage, mais pour l'automatisation, l'intégration et la réutilisation des données entre diverses applications".*

Le Web sémantique comprend un certain nombre de technologies organisées en couches interdépendantes. Dans ces différentes couches, nous pouvons citer les métadonnées, les ontologies, ainsi que la logique et l'inférence [GG07]. La figure 2.1 illustre ces différentes couches.

Les différentes couches sont les suivantes :

1. **URI (*Uniform Resource Identifier*)** permet d'identifier les ressources sur les réseaux.
2. **XML (*Extensible Markup Language*)**, le langage extensible de balisage est une simplification du langage normalisé de balisage généralisé SGML (*Standard Generalized Markup Language*). Ses deux principes essentiels : 1) la structure d'un document XML est définissable et valable par un schéma et 2) un document XML est entièrement transformable dans un autre document XML. Enfin, son objectif initial est l'interopérabilité, de faciliter l'échange automatisé de contenus entre systèmes d'in-

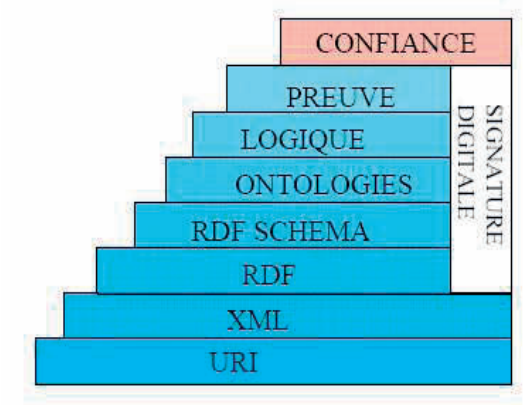

FIGURE 2.1 – Les couches du Web sémantique

formation hétérogènes et particulièrement adapté pour l'envoi de documents sur le Web.

3. **RDF (*Resource Description Framework*)** est un modèle de données doté d'une syntaxe à balises qui permet de représenter des objets ou des ressources ainsi que des relations entre ces objets sous forme de triplets. Il est destiné à décrire de façon formelle les ressources Web et leurs métadonnées et de permettre également le traitement automatique des descriptions. Un document structuré en RDF est un ensemble de triplets. Chaque triplet RDF est une association de (sujet, prédicat, objet) où le sujet représente la ressource à décrire, le prédicat représente un type de propriété applicable à cette ressource et l'objet représente une donnée ou une autre ressource.
4. **RDF-Schéma ou RDFS** est un vocabulaire qui permet de décrire les classes et propriétés pour des ressources RDF. Ce langage extensible de représentation des connaissances appartient à la famille des langages du Web sémantique et fournit des éléments de base pour la définition d'ontologies ou vocabulaires destinés à structurer des ressources RDF.
5. **Ontologies et Logique.** Ces couches concernent la définition de langages ontologiques comme OWL (*Ontology Web Language*) qui s'appuie sur la logique de description [SS04]. Le langage OWL est préconisé comme standard par le W3C consortium pour modéliser des ontologies. OWL permet de définir des terminologies constituées de concepts et de propriétés entre ces concepts. Il permet de spécifier des hiérarchies de concepts et de rôles. Les concepts peuvent être décrits par des axiomes. OWL permet de modéliser les informations d'un domaine afin de faciliter l'échange d'informations par des processus automatisés.
6. **Preuve.** Les langages logiques permettent la mise en œuvre d'outils de raisonnement. Des outils de raisonnement sont disponibles pour des lan-

gages comme OWL et permettent par exemple de tester la cohérence des informations, de les classifier, etc. Les modèles à base de règles comportent également des moteurs d'inférence qui permettent d'inférer des informations à partir des informations décrites. Des efforts sont en cours, pour uniformiser ces différentes approches. La spécification du langage SWRL (Semantic Web Rule Language) s'inscrit dans ces travaux émergents.

7. **Confiance.** La couche confiance est située au haut de la pyramide. Elle concerne l'utilisation de signatures numériques et d'autres types de connaissances afin de garantir la fiabilité et l'origine des informations, par des recommandations d'agents de confiance, de la notation, des organismes de certification et des organismes de consommateurs.

2.1.2 Les métadonnées

Le Web sémantique est un moyen d'échange et de partage des ressources grâce à la représentation sémantique du contenu. Ces ressources sont décrites par des informations structurées additionnelles "les métadonnées". Les métadonnées sont des "données sur les données" [CS06]. L'utilisation des métadonnées permet de décrire le contenu de la ressource afin de rendre son contenu exploitable. Les métadonnées peuvent exister sur différents niveaux distincts. Les informations décrivant la syntaxe, la structure et le contexte sémantique. La figure 2.2 illustre la représentation faite par Sheth[She03], des différents types de métadonnées :

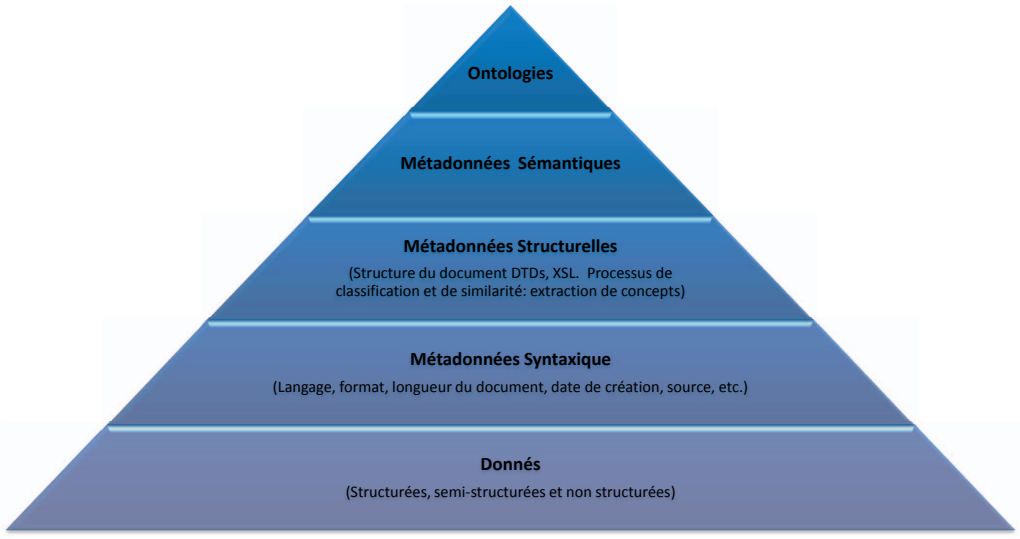

FIGURE 2.2 – Types de métadonnées et des annotations sémantiques selon Sheth

- **Métadonnées syntaxique.** Les métadonnées syntaxiques sont la forme la plus simple de métadonnées et décrivent l'information non contextuelle sur le contenu en fournissant des informations très générales, en associant des étiquettes. Certaines métadonnées syntaxiques peuvent être la taille du document, l'emplacement ou la date de création.
- **Métadonnées structurelles.** Les métadonnées structurelles fournissent des informations concernant la structure du contenu et décrivent comment les éléments sont assemblés ou arrangés.
- **Métadonnées sémantiques.** Les métadonnées sémantiques ajoutent les relations, les règles et les contraintes aux métadonnées syntaxiques et structurelles. Elles fournissent un contexte pour l'interprétation des informations basées généralement sur un modèle de métadonnées d'un domaine spécifique ou une ontologie. Ces métadonnées permettent aux applications de comprendre la signification réelle des données.

2.2 Les ontologies

Face à l'émergence des sources d'informations disponibles de plus en plus nombreuses et complexes, il est nécessaire de permettre une description de ces informations non seulement en termes de structure (aspect syntaxique) mais également en termes de signification (aspect sémantique). La description de métadonnées sur les sources d'informations prenant en compte la description structurelle mais aussi sémantique des informations est un problème important. Cette sémantique peut être exprimée à l'aide d'ontologies.

Une ontologie permet la modélisation d'un domaine de connaissances et peut être vue comme un modèle conceptuel d'un domaine particulier, qui décrit les concepts de ce domaine et les relations entre ces concepts. Elle est généralement considérée comme une base de connaissances et est au centre des développements émergents du Web sémantique. Les ontologies permettent la modélisation d'informations agréées par une communauté de personnes et accessibles par une machine pour développer des services automatisés et par conséquent, jouent un rôle de référence pour décrire la sémantique des informations à partager.

L'utilisation d'ontologies permet la représentation formelle des connaissances à l'aide de modèles basés sur des logiques (logiques de représentation, logique de description) et le raisonnement sur ces connaissances à l'aide d'outils d'inférences (vérification de la cohérence des informations, classification des informations, etc.). Les connaissances modélisées dans une ontologie peuvent être partagées et/ou réutilisées dans différents environnements (applications) afférant à un même domaine d'intérêt, selon Dobson et al [DS06].

2.2.1 Définitions

Il existe dans la littérature plusieurs définitions d'une ontologie. Le terme ontologie a commencé à se répandre au début des années 90, une première définition est donnée par Neches et ses collègues [NFF+91] :

> "An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary." [1]

La définition la plus citée est celle proposée par Thomas Gruber [Gru93] :

> "An ontology is an explicit specification of a conceptualization." [2].

Quelques années plus tard, Chandrasekaran [CJ96] a défini une ontologie comme une théorie du contenu sur les sortes d'objets, les propriétés de ces objets et leurs relations possibles dans un domaine spécifié de connaissances. De cette façon, l'ontologie fournit les termes potentiels pour décrire les connaissances sur ce domaine. D'une manière plus pragmatique Hafner [HF96] rejoint cette définition reprise par Noy [NH97]. En raison des différents sens du terme "ontologie", la plupart des chercheurs en intelligence artificielle conviennent que les fondations ontologiques d'un modèle de connaissances sont l'ensemble des catégories de haut niveau et des relations utilisées pour construire les entités du modèle. Chandrasekaran [CJB99] définit les éléments qui constituent une ontologie dans un monde constitué d'objets, dont les propriétés ou attributs peuvent prendre des valeurs. Les objets peuvent être associés par des relations. Les propriétés et les relations peuvent changer au cours du temps et ces changements mettent en jeu des événements et des processus, éventuellement associés par la relation de causalité. Dans le Web sémantique, une ontologie est vue comme un ensemble de connaissances, y compris le vocabulaire et les relations sémantiques, avec des règles simples d'inférence et de logiques relatives à des sujets particuliers [GBJJ05].

Dans le cadre de notre travail, nous retiendrons la définition générale suivante : *Une ontologie est une représentation des informations agréée par une communauté de personnes et accessible par des hommes et agents logiciels pour faciliter l'échange et le partage de ces informations.*

Plusieurs classifications des ontologies ont été proposées dans la littérature. Nous présentons dans la suite quelques unes de ces classifications que nous jugeons représentatives.

1. Une ontologie définit les termes et les relations comportant le vocabulaire d'un thème aussi bien que les règles afin de combiner les termes et les relations pour définir les extensions pour le vocabulaire.
2. Une ontologie est une spécification formelle et explicite d'une conceptualisation.

2.2.2 Classification des ontologies

Plusieurs classifications ont été définies et se basent sur différents critères. Nous retiendrons les classifications proposées par G. Van Heijst 97 [VHSW97], N.Guarino 97 [Gua97] et O. Lassila et D. McGuinness [LM01].

Van Heijst a proposé [VHSW97] deux types de classification des ontologies selon différents critères. La première classification se base sur les types et la richesse des structures utilisées dans l'ontologie. Selon ces critères, il distingue trois catégories d'ontologies :

- `Les ontologies terminologiques` qui sont utilisées pour spécifier les termes du vocabulaire d'un domaine de connaissances.
- `Les ontologies d'information` qui spécifient la structure/le schéma d'une base de données pour permettre le stockage d'informations.
- `Les ontologies qui modélisent de la connaissance` qui proposent des structures internes plus riches et qui sont davantage définies en fonction de leurs utilisations comme par exemple le partage d'informations.

Il propose également une classification des ontologies qui s'appuie sur la prise en compte des "objectifs" de la modélisation. Il retient quatre catégories d'ontologies selon ce critère :

- `Les ontologies d'applications` qui spécifient les informations nécessaires à une ou plusieurs applications particulières.
- `Les ontologies de domaine` qui expriment la conceptualisation des connaissances d'un domaine particulier.
- `Les ontologies génériques` qui modélisent des connaissances tranverses à différents domaines. Typiquement les ontologies génériques définissent des concepts comme les notions d'état, d'événement, d'action, etc.
- `Les ontologies de représentation` qui visent à expliciter les conceptualisations sous-jacentes aux formalismes de représentation des connaissances. Elles représentent les entités du monde réel sans a priori, de façon "neutre". Ces concepts des ontologies de représentation peuvent être utilisés dans les ontologies génériques ou les ontologies de domaine. Cette classification est illustrée par la figure 2.3.

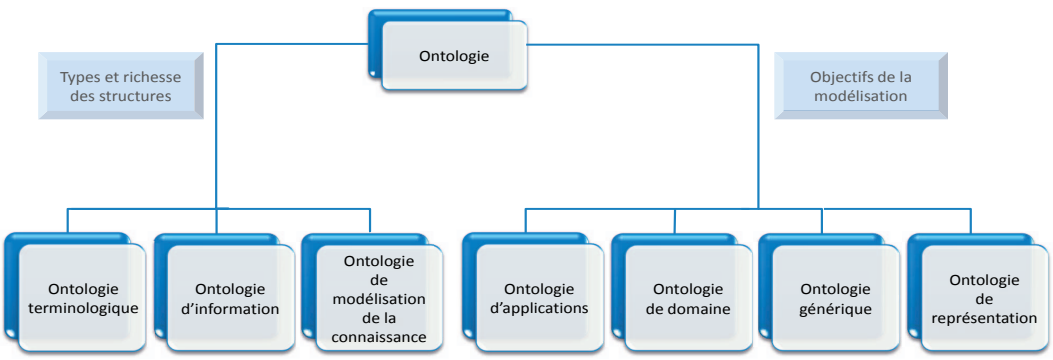

FIGURE 2.3 – Classification des ontologies selon Van Heijst [VHSW97]

Une autre classification des ontologies a été proposée par Guarino [Gua97] en quatre catégories (figure 2.4). Cette classification se base sur le degré de généralité ou du niveau de dépendance d'une tâche. Il considère :

- `Les ontologies de haut niveau` qui décrivent des concepts très généraux comme par exemple, l'heure, lieu, événement, action, etc.
- `Les ontologies de domaine` qui décrivent le vocabulaire par rapport à un domaine générique, i.e. la médecine, l'architecture, etc.
- `Les ontologies des tâches ou d'activités` qui décrivent une tâche ou une activité spécifique, i.e. les ventes, le diagnostic, etc.
- et finalement, `les ontologies d'applications` où les concepts dépendent à la fois d'un domaine et d'une activité en particulier.

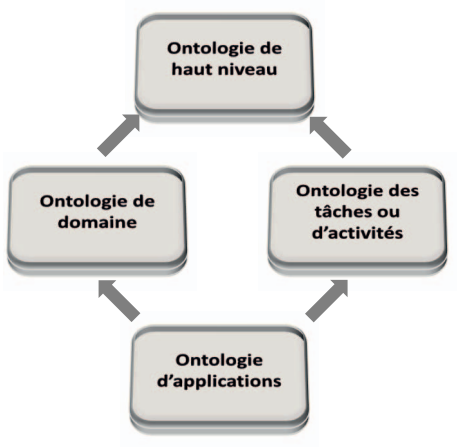

FIGURE 2.4 – Classification des ontologies selon Guarino

Lassila and MacGuinnes [LM01] ont également proposé une classification "continue" des ontologies en se basant sur les informations qu'elles doivent exprimer et la richesse de leurs structures internes. Cette classification est représentée sur la figure 2.5.

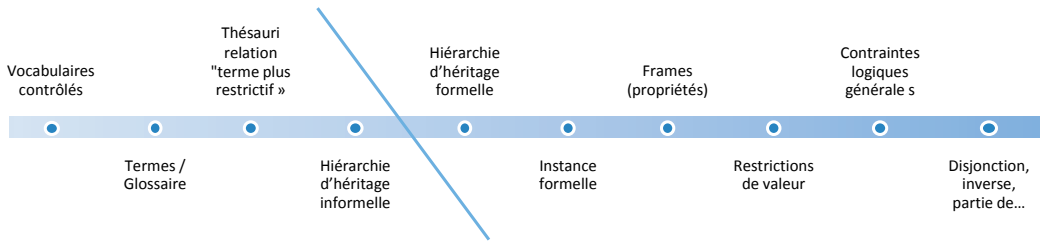

FIGURE 2.5 – Classification des ontologies selon Lassila et McGuinness

Nous pouvons distinguer deux parties dans la proposition : d'une part les ontologies "légères" basées sur la définition de vocabulaires et de relations sémantiques simples avec des vocabulaires contrôlés, des glossaires, des thésau-

rus, des ontologies avec des hiérarchies d'héritage informelles et d'autre part les ontologies qui ont des structures internes plus riches avec des hiérarchies d'héritage formelles, des instances formelles, la représentation de propriétés, de restrictions de valeurs, de contraintes logiques générales et l'utilisation d'opérateurs logiques comme la disjonction et la notion de relations inverses.

2.2.3 Ontologies et systèmes d'information

Les systèmes d'information (SI) sont essentiellement des outils permettant de capturer et représenter les connaissances sur certains domaines. Les professionnels et les chercheurs SI ont traditionnellement cherché à identifier les problèmes de modélisation et de représentation des connaissances au sein des SI. En 1998, Sheth [She98] met l'accent sur les nouveaux besoins des SI. La nouvelle génération des SI devra être capable de résoudre l'interopérabilité sémantique de SI, dans laquelle un fait peut être plus qu'une description, afin de faire bon usage de l'information disponible dans internet et l'informatique distribuée.

Les ontologies servent selon Pisanelli [PGS02] à :

1. créer une compréhension partagée pour unifier différents points de vue,
2. faciliter la communication entre les personnes impliquées dans la construction du SI,
3. permettre la réutilisation des connaissances du domaine, faciliter la récupération, l'intégration et l'interopérabilité entre les sources hétérogènes de la connaissance,
4. fournir une base de représentation des connaissances du domaine,
5. aider à identifier les catégories sémantiques de domaine. Ainsi, l'utilisation d'ontologies dans le développement des SI peut établir des correspondances et des relations entre les entités d'information des différents domaines.

Les ontologies sont de plus en plus un outil efficace dans la recherche et le développement de la discipline des SI et démontrent que leur utilisation contribue à l'amélioration de la qualité du produit final. Frank [Fra97] discute de l'intérêt d'utiliser une ontologie pour les besoins des systèmes d'information géographique pour modéliser les relations spatiales. Guarino [Gua98] s'intéresse au rôle central que peuvent jouer les ontologies dans les systèmes d'information futurs. Pisanelli [PGS02] démontre l'efficacité de l'approche ontologique dans des domaines spécifiques comme la pêche, le domaine clinique et le domaine génétique. Viinikkala [Vii04] décrit et illustre l'utilisation des ontologies dans la discipline des systèmes d'information.

2.3 Conclusion

Dans ce chapitre, nous avons présenté les principaux concepts de notre domaine de recherche. Nous avons présenté le Web sémantique et ses différents éléments. Les ontologies sont apparues comme la principale représentation de la sémantique d'un domaine donné, elles servent à la représentation des données afin de faciliter la communication entre les différents acteurs d'un domaine. Elles sont utilisées dans le processus de recherche d'information à deux niveaux : la représentation des données et la recherche d'information. Dans le chapitre suivant, nous présentons un état de l'art ainsi qu'un bilan sur les travaux qui traitent de l'enrichissement de requêtes dans la recherche d'information.

3

Enrichissement de requêtes

Dans ce chapitre, nous présentons des approches d'enrichissement de requêtes en décrivant des méthodologies et outils existants. Nous abordons essentiellement les approches que nous avons étudiées, et qui traitent complètement ou même partiellement du problème d'enrichissement de requêtes. Nous avons identifié deux types d'approches : (i) les approches basées sur les requêtes des utilisateurs et (2) les approches basées sur le contenu des ressources. La suite de ce chapitre est organisée comme suit : dans la section 3.1 sont présentées quelques définitions, la section 3.2 présente les différentes approches étudiées et enfin une analyse de l'état de l'art est donnée dans la section 3.3

Sommaire

3.1 Processus d'enrichissement de Requêtes

La recherche d'information se compose en général de deux phases, (i) la phase d'indexation qui consiste à représenter au mieux le contenu des ressources et (ii) la phase de recherche qui consiste à répondre à la recherche de l'utilisateur en donnant des réponses en fonction de sa requête. La figure 3.1 schématise ce processus. Les documents sont annotés puis les éléments de la requête sont comparés avec les annotations pour sélectionner les documents réponses.

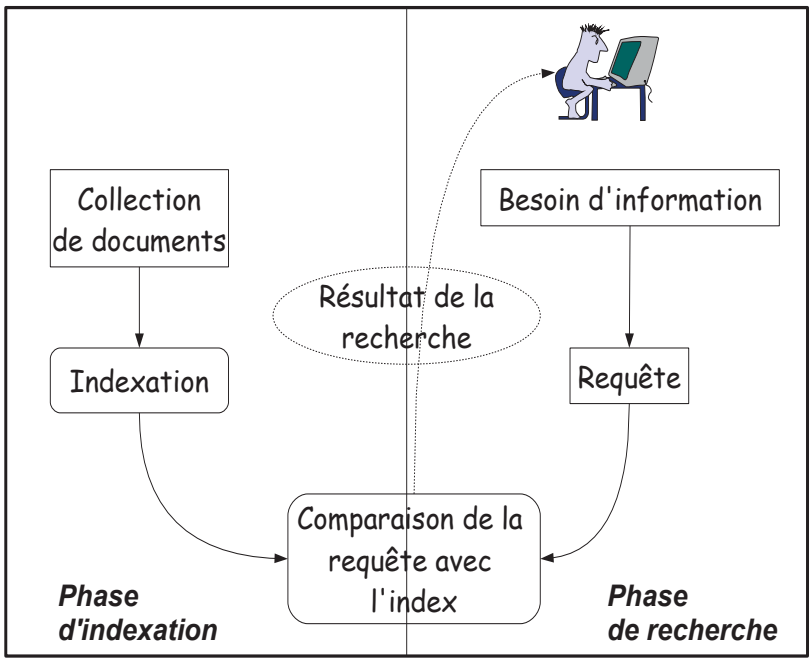

FIGURE 3.1 – Les étapes dans le processus de recherche d'information

Afin de répondre au mieux aux besoins de l'utilisateur, sa requête peut être enrichie afin de rendre les résultats les plus pertinents possibles. Les méthodes d'enrichissement de requêtes sont étudiées dans le domaine de l'informatique, en particulier dans le domaine de traitement du langage naturel et de la recherche d'information (RI). Plusieurs travaux se sont intéressés à l'enrichissement de requêtes en utilisant différentes approches, en se basant par exemple sur la similarité entre termes (synonymie, relation généralisation/spécialisation, etc.). L'enrichissement de requêtes peut se présenter sous deux formes : la reformulation de requête qui consiste à modifier ou compléter une requête avant d'effectuer la recherche correspondante et la proposition de nouveaux termes en pondérant les résultats selon leur pertinence. Les nouveaux termes peuvent être ajoutés automatiquement ou agréés par l'utilisateur pour construire une nouvelle requête enrichie.

Dans les moteurs de recherche, le processus d'enrichissement se compose de deux principales étapes :

- l'évaluation de la requête saisie par l'utilisateur,
- et la proposition de nouveaux termes pour la requête.

Avant de présenter les différentes approches retenues, nous donnons quelques définitions de l'enrichissement de requêtes.

3.1.1 Définitions

Plusieurs définitions d'enrichissement de requêtes *Query Expansion (QE)* ont été données dans la littérature, nous en présentons dans cette section quelques unes.

Salton et McGill [SM86] définissent l'enrichissement de requêtes dans les systèmes de recherche d'information (SRI) comme un processus qui vise à rendre les résultats plus clairs et précis en permettant à l'utilisateur de modifier sa requête pour améliorer la pertinence de ses résultats.

Selon Abberley et al [AKRR99], l'enrichissement de requêtes dans leur système permet de reformuler les requêtes et améliorer le processus de recherche d'information.

Efthimiadis [Eft96] qui a également proposé une classification des méthodes d'enrichissement de requêtes, donne la définition suivante : *" l'enrichissement de requêtes ou l'enrichissement de termes est un processus qui vise à compléter la requête en proposant des termes supplémentaires, et est considéré comme une amélioration de la recherche d'information "*. Il donne également les définitions suivantes :

- L'enrichissement de requêtes est une approche qui peut être appliquée quelle que soit la recherche ou les méthodes utilisées.
- La requête initiale telle qu'elle est saisie par l'utilisateur peut être une représentation inadéquate ou incomplète des besoins de l'utilisateur, soit de lui-même ou de la représentation des idées dans les documents, base de données, etc.
- L'enrichissement de requêtes peut avoir lieu à la formulation de requête initiale, à la phase de reformulation de requêtes, ou bien les deux.
- Le concept plus général de modification de requête peut impliquer la suppression des termes de la requête. Dans notre travail, nous nous intéressons à l'ajout de nouveaux termes.

À l'ère du Web sémantique, Vechtomova et Wang [VW06] présentent l'enrichissement de termes des requêtes comme une amélioration de la formulation de la requête initiale dans la recherche de documents. Ces termes sont généralement choisis parmi les documents entiers, des paragraphes ou des parties du document où apparaissent les termes de la requête. Ils ajoutent que la relation sémantique entre les termes diminue en fonction de la distance qui les sépare dans le texte.

3.1.2 Classification des méthodes d'enrichissement de requêtes

Quand on parle de requêtes, selon Efthimis [Eft96] la simplicité de la recherche peut être réduite à deux étapes :

1. **Formulation de la requête initiale.** L'utilisateur construit la première stratégie de recherche et la soumet au système.
2. **Reformulation de la requête.** Après avoir eu quelques résultats de sa recherche, l'utilisateur améliore les résultats en modifiant sa recherche (i) manuellement, (ii) semi-automatiquement, ou (iii) automatiquement.

L'enrichissement de la requête est un cas particulier de la reformulation de requêtes, et est basé sur des méthodes proposées dans la classification donnée par [Eft96]. La figure 3.2 illustre cette classication et est réalisée en fonction de trois différents critères :

1. L'interaction de l'utilisateur dans le processus pour améliorer la sélection des termes supplémentaires.
2. Les types de ressources utilisées pour trouver les termes d'une requête supplémentaire.
3. Les méthodes et algorithmes utilisés pour sélectionner les termes à ajouter.

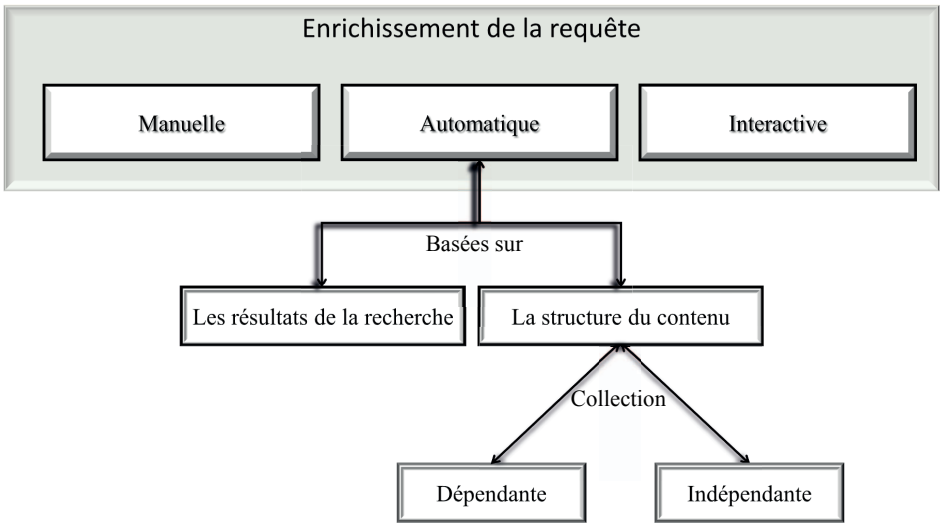

FIGURE 3.2 – Classification selon Efthimiadis [Eft96]

Nous pouvons classer les différentes méthodes d'enrichissement comme suit : **manuelle**, **automatique** (médiation de l'utilisateur) ou **interactive** (assistée par l'utilisateur).

- *MQE, Manual Query Expansion.* L'enrichissement manuel de requêtes est basé sur les modèles de recherche booléenne et les modes d'interaction entre l'utilisateur et le système de recherche.
- *AQE, Automatic query Expansion.* L'enrichissement automatique de requêtes consiste à la proposition automatique de nouveaux termes par le système. Cette étape est transparente pour l'utilisateur dans le processus de recherche d'information.
- *IQE, Interactive Query Expansion.* L'enrichissement de requêtes interactif consiste à l'interaction entre le système et l'utilisateur pour le choix des termes de la requête enrichie. D'une part, le système propose les termes et les présente à l'utilisateur et d'autre part, les utilisateurs sélectionnent des termes en fonction de leurs préférences.

Ces différentes méthodes sont basées sur :

1. Les résultats de la recherche (*Based on Search Results, BSR*) : en utilisant les documents issus de la requête de l'utilisateur, en prenant les plus pertinents afin d'extraire de nouveaux termes qui peuvent servir à enrichir la requête initiale.

2. La structure du contenu (*Based on Knowledge Structures, BKS*) : qui est indépendante du processus de recherche. Elle peut avoir deux méthodes différentes :
   - collection dépendante (*collection dependent, CD*), être basée sur un corpus et,
   - collection indépendante (*collection independent, CI*) être indépendante du corpus.

Le choix de nouveaux concepts dans la requête initiale est important pour la pertinence des résultats. Ce choix est motivé par l'interaction entre l'utilisateur et le système.

3.2 Approches pour l'enrichissement de requêtes

Au fil des ans, les changements technologiques dans les systèmes d'information et le Web ont évolué et avec eux les types d'enrichissement de requêtes utilisés. Il existe de nombreux systèmes d'information qui travaillent avec les différentes méthodes d'enrichissement, mais nous aimerions mettre l'accent uniquement sur des méthodes interactives d'enrichissement des requêtes qui sont actuellement les plus courantes. Selon ce critère retenu, nous classons certains ouvrages et identifions leurs méthodes d'enrichissement des requêtes et les ressources qu'ils utilisent pour calculer les termes supplémentaires. Nous classons ces travaux dans deux catégories : les approches basées sur le résultats de la recherche de l'utilisateur et les approches basées sur la structure des ressources.

3.2.1 Approches basées sur les résultats de recherche

Les systèmes présentés sont basés sur différentes méthodes d'enrichissement de requêtes, mais utilisent tous les résultats (commentaires ou feed-back, profil de l'utilisateur ou comportement de l'utilisateur) en tant que ressources pour le calcul de nouveaux termes pour enrichir les requêtes des utilisateurs. Nous avons identifié deux types d'approches (i) les approches basées sur les documents réponses, (ii) les approches basées sur une interaction avec l'utilisateur en étudiant son profil ou son comportement.

3.2.1.1 Approches basées sur les documents réponses

L'enrichissement de requêtes est réalisé avant la phase de recherche, les méthodes statistiques sont apparues pour l'enrichissement de requêtes, elles sont basées sur le calcul de co-occurence de termes. Cela consiste à créer une relation entre des termes qui sont cités dans les mêmes documents.

[RBEB02] présentent un système de réecriture de requêtes en utilisant une méthode de classification non supervisée. Un système nommé SIAC est utilisé pour la recherche documentaire, il permet de classer les documents retournés à l'aide d'un arbre de décision. Les phrases issues des documents retournés sont regroupées dans les feuilles de l'arbre en fonction de la requête, une méthode de classification non supervisée est utilisée afin de classer les phrases et les documents selon qu'elles contiennent ou pas des termes. L'enrichissement de requêtes se fait à partir des expressions booléennes représentant les *meilleures* feuilles (feuilles possédant un grand nombre de documents).

Le système proposé par Cui et al [CWNM02] est basé sur des analyses statistiques de corpus de documents. La méthode d'enrichissement de requêtes utilise les résultats de recherche comme ressources pour le calcul de nouveaux termes pour enrichir les requêtes des utilisateurs. Le système extrait des corrélations entre les termes de la requête et les termes du document en analysant les logs. L'objectif est d'augmenter la fréquence des termes apparaissant conjointement sur un même document et de sélectionner les termes avec le plus grand coefficient. L'hypothèse de cette approche est que les termes de la requête de l'utilisateur et les termes des documents réponses que l'utilisateur sélectionne (cliques) sont corrélés. En analysant ces liens les auteurs ont défini une mesure statistique pour le calcul de la corrélation entre les deux espaces.

Qiu et al [QF93] proposent une approche statistique basée sur un thésaurus de similitude, il est représenté par une matrice de similarité entre les termes. Ce thésaurus représente l'indexation des termes dans le corpus de documents. L'enrichissement de la requête se fait après sélection de termes pondérés. Ils utilisent le modèle Generalised Vector Space Model (GVSM), une extension du modèle vectoriel classique. Ce modèle a été développé afin de répondre aux critiques selon lesquelles les unités lexicales ne sont pas de bonnes bases pour l'espace vectoriel classique puisqu'ils ne sont pas indépendants [CYF+97]. Le modèle GVSM inclut tous les termes de l'enrichissement et utilise la requête

enrichie afin de classer les documents réponses.

Ils utilisent également le modèle Latent Semantic Indexing (LSI) afin de trouver les termes qui décrivent les documents et les requêtes. Cet algorithme est utilisé par les moteurs de recherche afin d'évaluer le contenu d'un site en fonction des mots clés utilisés. Ils obtiennent de bons résultats en choisissant les concepts qui sont sémantiquement liés à l'ensemble des termes de la requête plutôt qu'à chacun de ses termes.

Guelfi et Pruski [GP06] ont développé un cadre formel basé sur la logique et la théorie des graphes pour l'ajout de termes dans les requêtes, ces termes sont extraits du thésaurus Wordnet. L'approche consiste à enrichir une requête en se basant sur l'ensemble des résultats retournés par le moteur de recherche Google. Les documents sont organisés dans un graphe dont les sommets représentent les concepts des pages Web et les arêtes les liens sémantiques entre eux. Cette approche a été étendue [GPR07] pour la rendre compatible avec les ontologies OWL en utilisant les relations.

3.2.1.2 Approches basées sur l'interaction avec l'utilisateur

Dans le cadre de Gossple [BGLK09], pour enrichir la requête d'un utilisateur, les auteurs utilisent la probabilité de passer d'un *tag* à un autre comme indicateur de leur similarité. Ce système est basé sur l'utilisateur afin d'améliorer la recherche d'information sur le Web, il utilise les réseaux sociaux afin d'enrichir une requête. Le but de ce système est de découvrir les utilisateurs partageant les mêmes intérêts en construisant un réseau social avec des distances sémantiques entre les utilisateurs. La requête d'un utilisateur u est enrichie avec des *tags* considérés "similaires" d'autres utilisateurs proches de l'utilisateur u. Un réseau personnel pondéré d'utilisateurs partageant les mêmes intérêts (items) est utilisé pour construire la matrice *TagMap* pour chaque utilisateur u. La matrice représente la relation d'une paire de tags utilisés dans le processus d'enrichissement. Les tags des utilisateurs proches sont ajoutés dans la *TagMap*. L'algorithme *TagRank* basé sur l'algorithme *PageRank* [BP98] exploite les *TagMap* afin de déterminer les meilleurs tags candidats pour l'enrichissement de la requête. Il considère les tags proches des termes de la requête. La probabilité de passer d'un tag à un autre dépend du degré de similarité entre les tags.

Le système AIRA (*Adaptive Information Research Assistant*) [BBB04] construit un profil utilisateur afin d'enrichir la requête. Lors de la recherche, le système identifie un contexte à partir de la requête et du profil de l'utilisateur. L'enrichissement de la requête passe par deux phases : (i) apprentissage du vocabulaire et (2) apprentissage de requêtes. Le profil de l'utilisateur est mis à jour en fonction de sa recherche. A l'issue des deux phases les documents les plus pertinents sont retournés à l'utilisateur. Le fonctionnement du système repose sur l'hypothèse que l'utilisateur maintient à jour un ensemble de documents ou de références qui représentent son profil et intérêts qui sont représentatives de ses centres d'intérêts. C'est à partir de cet ensemble que

le système extrait et structure les ressources représentatives de l'utilisateur. L'apprentissage du vocabulaire vise à identifier les meilleurs termes pour la recherche de documents et l'apprentissage de la requête recherche la combinaison optimale des termes sélectionnés dans la première phase. Lors de cette phase les requêtes sont soumises aux moteurs de recherche et les documents réponses sont évalués et les requêtes pondérées en fonction de l'évaluation. Pour chaque requête sélectionnée de nouveaux termes sont ajoutés.

Dans le but d'offrir un accès personnalisé aux données semi-structurées afin d'augmenter la pertinence des données, Zayani [ZPCS06] présente l'architecture d'un système d'adaptation avec des fonctionnalités permettant l'enrichissement d'une requête à partir du profil utilisateur et de le mettre à jour en fonction des requêtes de l'utilisateur. Les auteurs proposent un modèle utilisateur avec des caractéristiques permanentes et évolutives pour stocker les interactions. Les informations permettent l'évaluation de la requête afin de l'enrichir et rendre le résultat plus pertinent. Les requêtes sont enrichies avec des termes représentant les intérêts de l'utilisateur. Le résultat d'une même requête peut différer d'un utilisateur à un autre.

3.2.2 Approches basées sur la similarité entre concepts

D'autres approches sont basées sur l'utilisation de ressources "externes". Ces ressources peuvent être des bases de connaissances telles que les thésaurus ou ontologies. Les systèmes de recherche d'information ont évolué avec l'apparition du Web sémantique et visent à exploiter les relations sémantiques entre les termes d'une requête et/ou des documents afin d'enrichir le contexte de la requête de l'utilisateur. Nous avons retenu dans notre étude quelques approches que nous avons estimées pertinentes. Nous différencions dans la présentation des travaux (i) d'une part la similarité entre les termes de la requête et les documents et (ii) d'autre part les travaux basés sur la similarité sémantique entre les termes d'une ressource externe (ontologie, dictionnaire, thésaurus...). Si les termes rajoutés proviennent des documents de la collection, on parle de retour de pertinence *(relevance feedback)*. Par contre, s'ils proviennent d'une ressource externe, on parle de reformulation de requête.

3.2.2.1 Approches basées sur la similarité requêtes/documents

Le système (*Query Similarities and relevant Documents, QSD*) [HKJD02] stocke pour chaque requête la liste des documents jugés les plus pertinents. Pour chaque nouvelle requête, le système effectue un calcul de similarité avec les requêtes précédentes et propose les résultats pertinents des requêtes similaires. L'enrichissement de la requête se fait par les étapes suivantes :

- calculer la similarité entre la nouvelle requête et les précédentes requêtes (cosinus des vecteurs de requêtes),
- sélectionner les requêtes similaires (supérieur à un seuil défini),
- pour cet ensemble de requêtes sélectionner les documents pertinents,

– calcul du vecteur document des documents sélectionnés,
– enrichir la requête avec le vecteur document pondéré.

L'analyse formelle des concepts est une formalisation mathématique de l'analyse des données qui utilise la structure des treillis pour la représentation d'une relation binaire entre deux ensembles. L'utilisation de cette structure a débuté dans les années 80 [GW97]. Dans le domaine de la biologie, Messai [MDNST06] présente une méthode basée sur l'analyse formelle des concepts et l'utilisation d'ontologies afin d'enrichir des requêtes. Il construit un treillis de concepts pour représenter la requête. Il classe les métadonnées dans un treillis de concepts, chaque concept représente une paire (ressource, métadonnées) où les métadonnées sont des ensembles de termes. Le concept construit à partir d'une requête est ajouté dans le treillis afin de déterminer les documents les plus pertinents répondant à la requête de l'utilisateur. Un document est pertinent pour une requête s'il partage au moins une de ses métadonnées avec la requête, le nombre de métadonnées communes reflète le degré de pertinence du document. Dans la phase de "raffinement" de la requête, les métadonnées ajoutées à la requête sont soit plus spécifiques soit plus générales que celles de la requête initiale (raffinement par généralisation ou par spécification). Ce raffinement est basé sur des ontologies au moment de l'indexation des ressources afin d'améliorer le processus de recherche d'information.

Une approche qui vise à améliorer la requête est présentée dans [LJ10] en ajoutant des mots vides. Partis de l'observation que les requêtes ne contiennent généralement pas de ponctuation ni mots vides entre deux termes d'une requête, ils modifient la requête en ajoutant les mots vides et ceci en se basant sur un corpus de documents afin d'améliorer la recherche d'information des moteurs de recherche multilingues.

3.2.2.2 Approches basées sur liens sémantiques

Depuis quelques années, plusieurs travaux se sont intéressés à l'utilisation des liens sémantiques entre les termes afin d'enrichir les requêtes. Ces travaux utilisent un vocabulaire structuré allant d'une simple taxonomie à l'utilisation d'ontologies complexes.

D'après [Voo94] les méthodes statistiques ont moins de succès que les méthodes utilisant les relations sémantiques et linguistiques. Il propose une méthode d'enrichissement de requêtes basée sur les relations sémantiques entre les termes (synsets) dans Wordnet [MBF+90]. Gong et al [GCU05] utilisent les relations hyperonymie, hyponymie et les relations de synonymie de Wordnet pour l'enrichissement de la requête. Wong [WH10] propose une nouvelle mesure pour calculer la similarité entre termes afin d'enrichir les requêtes. Une mesure de similarité est calculée en interprétant le graphe de termes comme un réseau électrique.

[VC04] décrit comment les liens sémantiques entre les noms et verbes peuvent améliorer les résultats des systèmes de recherche d'information, la requête étendue se compose donc des termes de la requête originale et des

verbes qualia. Un lien qualia relie un nom et un verbe reliés sémantiquement (par exemple couper est la fonction du terme couteau). Navgli [NV03] propose une approche qui consiste à extraire des concepts et relations d'une ontologie et les ajouter dans la requête initiale en se basant sur les liens entre les termes. Une ontologie personnalisée [XFPF10] est utilisée pour une recherche d'information sémantique en offrant à l'utilisateur une interrogation par concepts avec l'ensemble des termes décrivant le concept.

Afin d'améliorer la recherche dans le Web sémantique et permettre à l'utilisateur de créer des requêtes simples, le système SPARK [ZWX+07] traduit des mots clés d'une requête en requêtes *sparql*. La structure de SPARK se compose de deux modules :

a) traitement de l'ontologie, qui indexe automatiquement les ressources d'une ontologie choisie comme la base de connaissances et b) de la construction de la requête formelle, il prend des mots-clés en entrée, et retourne une liste classée des requêtes en langage *sparql* en sortie. La traduction se fait en trois étapes : le mise en correspondance des termes, la construction du graphe de requêtes et le classement des requêtes.

1. La mise en correspondance des termes de la requête avec les ressources de la base de connaissances consiste à trouver les ressources des ontologies correspondant aux termes (classes, instances..). Deux méthodes sont utilisées : la *méthode morphologique* en comparant deux phrases et la *méthode sémantique* en utilisant une ressource externe par exemple *Wordnet*.
2. Pour la construction de graphes, les ressources sont divisées dans différents ensembles de requêtes en utilisant l'algorithme *Minimum spanning Tree*.
3. Le classement de requêtes consiste à calculer la probabilité pour construire une requête F à partir d'un terme K dans la base de connaissances D.

3.3 Bilan

Dans ce chapitre, nous avons présenté un état de l'art concernant l'enrichissement des requêtes. Nous avons identifié deux types d'approches :

- Les méthodes basées sur les résultats de la recherche.
- Les méthodes basées sur les ressources externes, telles que des bases de connaissances, thésaurus ou ontologies.

En ce qui concerne le premier type d'approche, nous avons différencié celles utilisant les documents réponses pour extraire les termes servant à l'enrichissement de la requête de celles utilisant l'interaction avec l'utilisateur. Cette interaction peut être le choix de l'utilisateur des documents retournés ou plus généralement la construction du profil utilisateur en fonction de ses actions. Le deuxième type d'approche utilise les liens sémantiques entre les termes du document et des requêtes.

L'utilisation d'une ontologie ou autre ressource utilisant un vocabulaire contrôlé, ainsi que les relations entre les termes pour l'enrichissement de la requête permet de prendre en compte des termes qui ne sont pas utilisés dans la requête mais pouvant retourner des résultats proches de la requête initiale. Cependant, ce type d'approche peut amener à une recherche trop générale ou trop spécifique pour l'utilisateur si par exemple un utilisateur cherche des informations sur un sport, ceci peut l'amener à une recherche sur tous les types de sports. L'utilisation d'une ontologie permet de trouver des termes proches sémantiquement mais ne prend pas en compte les termes cherchés en même temps.

L'utilisation du profil de l'utilisateur permet de cibler l'enrichissement et les résultats en fonction de la recherche de l'utilisateur. Cependant ce type d'approche nécessite une phase d'apprentissage assez importante étant donnée que l'étude se fait sur un utilisateur. La recherche d'un utilisateur peut se propager sur d'autres utilisateurs si ces derniers soumettent des requêtes similaires, mais cette propagation doit prendre en compte l'ensemble des termes d'une requête.

Pour toutes ces raisons, nous présentons dans le chapitre 6 notre approche d'enrichissement de requêtes basée sur le comportement des utilisateurs en prenant en compte tous les termes de la requête initiale. Nous introduisons plusieurs notions sur lesquelles nous nous basons comme "séquence de requêtes" ou "graphe de concepts". Notre approche est principalement basée sur l'idée de "popularité" d'un concept par rapport aux autres concepts. La recherche de l'utilisateur est également basée sur une ontologie de référence.

4

Visualisation sémantique

CEs dernières années, plusieurs techniques et outils de visualisation ont été développés. Le développement du Web sémantique et des ontologies permettent d'envisager des techniques de visualisation plus pertinentes des informations, basées sur cette connaissance de la sémantique des informations. L'objectif de ce chapitre est de proposer une classification et une analyse des techniques de visualisation existantes selon certains critères sélectionnés afin de proposer une méthodologie de visualisation des informations en prenant en compte leur sémantique.

Sommaire

4.1 Vers une visualisation sémantique

Selon le dictionnaire de la langue française, visualiser signifie « rendre visible de manière claire » et dans le domaine de l'informatique, cela signifie « afficher ou faire paraître des éléments sur un écran ». Michael Friendly [Fri09] donne la définition suivante *"La visualisation de l'information est l'étude interdisciplinaire de la représentation visuelle des collections de grande envergure de l'information non-numérique, tels que les fichiers et les lignes de code dans les systèmes informatiques, des bibliothèques et des bases de données bibliographiques, des réseaux de relations sur Internet, etc."*.

Le domaine de la visualisation de l'information a émergé de la recherche de en interaction homme-machine (IHM), informatique, infographie ou visualisation graphique, la conception visuelle et la psychologie [SB03]. Ce domaine est considéré comme un élément essentiel dans la recherche scientifique, les bibliothèques numériques, l'exploration de données, l'analyse des données financières et les études de marché. James et Cook [SB03] présentent la visualisation de l'information comme des représentations visuelles permettant aux utilisateurs de voir, d'explorer et de comprendre de grandes quantités d'informations à la fois.

Depuis l'apparition du World Wide Web en 1990, le Web est considéré comme un système hypertexte contenant des documents liés entre eux et accessibles à travers Internet. Le Web est aujourd'hui un moyen extraordinairement flexible et économique pour la communication et l'accès aux informations et services.

La quantité spectaculaire de ressources sur le Web a généré un besoin de structuration et d'organisation afin de rendre les informations facilement exploitables. Le Web sémantique est apparu comme un ensemble de technologies visant à rendre le contenu des ressources accessible et réutilisable par les utilisateurs et les agents logiciels, ces technologies visent à remédier à l'absence d'une organisation claire. Le Web sémantique qui se présente comme une extension du Web classique dépasse les limites de ce dernier en introduisant des descriptions explicites de la sémantique des informations, la structure interne et la structure globale, les contenus et les services disponibles. Il vise à classer, structurer et annoter les ressources avec une sémantique explicite traitable par des machines [Ado01].

La recherche d'information dans le Web classique se base essentiellement sur la structure des documents. Dans le Web sémantique les machines accèdent aux ressources grâce à leur représentation sémantique. Ces ressources sont décrites par des informations additionnelles structurées. Pour représenter ces informations, ou pour les relier à des connaissances partagées, le Web sémantique a introduit la notion d'ontologie. Les ontologies servent à la représentation d'informations afin de faciliter l'échange et leur exploitation de données par les machines.

Dans ce contexte, la visualisation de l'information permet d'exploiter les résultats fournis par la recherche sémantique en rendant l'information inconnue

plus facile à repérer et à exploiter, en mettant en évidence les relations dans les informations. La visualisation de l'information permet d'analyser de grandes quantités de ressources en donnant une vue d'ensemble et des vues détaillées des informations. Les résultats des recherches peuvent être manipulés par le biais d'une interface graphique.

La suite de ce chapitre est organisée de la manière suivante : la section 4.2 présente les critères retenus dans le cadre d'un système de visualisation pour le Web sémantique. Différentes techniques de visualisation basées sur ces critères sont présentées dans la section 4.3 et quelques systèmes qui les utilisent. La section 4.4 présente des outils dédiés à la visualisation d'ontologies et une analyse de ces outils est présentée dans la section 4.5. Enfin, nous concluons dans la section 4.6.

4.2 Critères de classification

Cette section présente tout d'abord les critères de sélection retenus pour la description et la classification des systèmes puis un bilan pour permettre de déterminer les caractéristiques requises pour un système de visualisation sémantique. Les critères de classification retenus s'appuient sur les critères proposés par Kimani et al. [KCC02] : la dynamique de construction, le type d'adaptativité et le mode d'interaction des systèmes enrichis, et des critères complémentaires que nous avons jugé également pertinents. L'ensemble de ces critères est le suivant :

- *La dynamique de construction de la visualisation.* La visualisation peut être construite de deux façons : statique ou dynamique. La construction statique est basée sur une pré-analyse de l'ensemble des structures des informations à visualiser. Le *rendu visuel* est à la charge de l'administrateur qui construit une représentation qui doit être adaptée à l'ensemble de la visualisation. La construction dynamique est faite par le système en utilisant des paramètres de configuration et d'autres informations disponibles comme des règles.
- *Le type d'adaptativité.* Ce critère concerne la capacité d'un système à apporter des modifications dans l'environnement de l'utilisateur. L'adaptativité inclut la spécification de l'environnement de façon plus ou moins automatisée et la transformation de cet environnement en se basant une analyse de l'évolution des environnements de l'ensemble des utilisateurs. La spécification peut être *(i)* manuelle, l'administrateur applique manuellement les modifications à apporter, *(ii)* semi-automatique, le système est capable d'évoluer par une analyse des changements réalisés dans le temps à l'environnement mais l'administrateur doit valider ces modifications ou *(iii)* automatique, aucune validation n'est nécessaire.
- *La personnalisation.* Elle permet aux utilisateurs de préciser ou de modifier certains aspects de l'environnement. La personnalisation peut être basique et intelligente. Dans la personnalisation basique, l'utilisateur spé-

cifie explicitement le paramétrage des affichages ou des vues. Dans la personnalisation intelligente, l'utilisateur contribue partiellement à paramétrer les affichages, mais le système est capable de faire des raisonnements pour sélectionner le paramétrage le plus approprié.
- *Le mode d'interaction.* Il concerne la façon dont l'utilisateur interagit avec le système. Trois types d'interaction peuvent être cités : *(i)* la manipulation directe où les utilisateurs effectuent des modifications sur la représentation visuelle en agissant sur les objets affichés à l'écran (pointer, déplacer, etc.), *(ii)* les menus qui permettent à l'utilisateur de faire apparaître les options disponibles et *(iii)* les formulaires pour aider le remplissage des paramètres pour la visualisation.

Afin de compléter ces caractéristiques, nous retenons quatre critères complémentaires : la dimension, la représentation de la visualisation, les techniques conceptuelles et les mécanismes utilisés pour le développement des systèmes.
- *La dimension.* Il existe deux façons de visualiser les résultats, dans le plan en 2D, en utilisant des images des représentations de types d'arbres, graphes ou tout autre stratégie d'organisation ou bien en 3D si cela semble pertinent pour l'information à représenter avec également différentes stratégies de présentation.
- *La représentation graphique.* Elle concerne tous les paramètres pris en compte pour la présentation de l'information comme les couleurs, la forme, la taille, etc.
- *Les techniques conceptuelles.* Elles concernent les techniques sous-jacentes de représentation de la sémantique des informations qui peuvent être reflétées de façon plus ou moins directe dans la visualisation.
- *Les mécanismes du système.* Ils incluent les langages utilisés pour l'implémentation du système mais également l'utilisation d'algorithmes spécifiques.

4.3 Les techniques de visualisation

Les techniques de visualisation sémantique reposent sur plusieurs points comme : *(i)* le codage visuel à proprement parler qui est la façon dont les données sont affichées à l'écran *(ii)*, l'usage de métaphores qui peuvent être utilisées afin de faire la correspondance entre l'espace de l'information et le "monde réel" de l'utilisateur *(iii)* et les techniques conceptuelles qui correspondent aux mécanismes utilisés pour représenter et extraire la sémantique des données.

LeGrand [Gra01] propose trois types de techniques de visualisation : (i) arbres et graphes, (ii) les cartes et (iii) les mondes virtuels.
- Les arbres et graphes : les arbres présentent l'avantage d'une interprétation grâce aux relations hiérarchiques pouvant représenter de nombreux éléments, ils ont l'inconvénient de ne pas gérer un grand volume de données. Les Graphes tout comme les arbres s'adaptent à la représentation

de la structure globale de l'information, ils font apparaître clairement la structure des données et les relations, mais ils deviennent complexes quand le volume des données est important et la visualisation devient difficile.

- Les cartes : les cartes conceptuelles *"Concept Map"* sont une représentation graphique dans laquelle les concepts sont liés entre eux par des liens pour former un réseau/carte. Les cartes thématiques *"Topic Maps"* qui réfèrent à un standard ISO, permettent d'offrir un niveau sémantique pour organiser et classer l'information. Un *topic map* ressemble fortement à une ontologie, mais il se base sur un élément central appelé topic. Le but du *topic map* est de définir de manière complète un *topic* en lui donnant des rôles et des associations avec d'autres *topics*.
- Les mondes virtuels : les mondes virtuels sont créés artificiellement par un programme informatique. Il est possible de se déplacer et d'interagir avec ce monde virtuel. Sa représentation peut être en deux ou en trois dimensions qui est préférable, il peut simuler le monde réel ou non, il peut également interagir avec des agents informatiques. L'utilisation de mondes virtuels comme moyen de représentation de l'information reste fortement liée aux types d'informations à visualiser et plus adaptée à des domaines spécifiques. Dans le cadre général de la recherche d'information sur le Web ou dans des ensembles structurés comme des thesaurus ou des ontologies, ce type de représentation n'est en général pas pertinente.

Nous avons retenu quelques systèmes de visualisation jugés pertinents pour notre étude. La description de ces outils se fait en se basant sur les critères de classification présentés plus haut.

4.3.1 Les arbres et graphes

Star Tree Viewer . Cet outil est proposé dans l'environnement de développement d'applications Inxight Star Tree Studio et modélise l'information sous la forme d'arbres hyperboliques. Il aide à créer une interface de navigation visuelle dans les sites internet. La technique des arbres hyperboliques (hyperbolic trees) repose sur une croissance exponentielle du nombre d'informations affiché en fonction de l'éloignement du centre de l'arbre. Cette technique a été utilisée dans différents systèmes, comme le système proposé par N. Dushay [Nao04]. Les nœuds qui représentent les pages Web sont organisés dans une structure hiérarchique. Les couleurs des liens changent en fonction des relations qui existent entre les pages. Le mode d'interaction est une manipulation directe (ouvrir les pages en cliquant sur les nœuds, changer la racine de l'arbre, etc.). La personnalisation est effectuée manuellement par l'administrateur. A.Kobsa [Kob04] propose une comparaison d'outils de visualisation dont la visualisation Tree Viewer avec une expérimentation par des utilisateurs notamment pour analyser la satisfaction des utilisateurs.

Natto View Cet outil [SiOM99] utilise un ensemble de techniques pour visualiser et interagir de façon dynamique sur les espaces d'informations de structure de graphes comme le Web, l'information est visualisée en 3D. Le rendu visuel joue un rôle important dans le système, les nœuds représentent les pages et les arcs les liens entre les pages. Le mode d'interaction est la manipulation directe. Les nœuds peuvent être organisés selon les besoins de l'utilisateur.

WebBook et WebForager WebBook [CRY96] utilise la métaphore du livre afin de représenter un ensemble de pages Web. Une visualisation interactive 3D est utilisée afin de représenter la relation existante entre les pages du livre. Chaque page du WebBook représente une page Web. L'organisation de la visualisation se fait en utilisant des couleurs sur les liens afin que l'utilisateur différencie les différents types de liens (internes, externes.). Web Forager est utilisé pour représenter l'espace de travail d'un utilisateur où l'utilisateur peut interagir avec les livres, ainsi qu'avec d'autres objets de l'environnement (bibliothèque.)

4.3.2 Les cartes

WebOFDAV Ce système proposé par [HEC98] utilise une approche de navigation basée sur des sous-espaces disponibles nécessaires à l'utilisateur. Ce système est basé sur OFDAV (Online Force-Directed Animated Visualization), il génère et calcule une séquence incrémentale de *frames* ou des cartes (*maps*) correspondant à l'espace d'informations parcouru par l'utilisateur. Il génère des sous-graphes et le voisinage des informations parcourues afin d'aider l'utilisateur dans sa navigation. Plusieurs paramètres pour l'organisation de la visualisation sont utilisés comme colorier les nœuds en fonction de la navigation. Cet outil n'est pas adaptatif et utilise la manipulation directe et les menus comme mode d'interaction.

Navigational View Builder L'outil est basé sur l'utilisation de différentes techniques conceptuelles comme le *clustering* ou la hiérarchisation qui aident à rendre plus compréhensible et efficace l'espace d'information [MF95]. Le concepteur des vues peut spécifier des modes de représentation des informations (couleurs, formes, etc.). Par exemple, les nœuds peuvent représenter des fichiers Web (documents, media). Le type du nœud peut représenter un type de fichier, la taille d'un nœud peut représenter la taille du fichier, les couleurs peuvent être liées à des thématiques, etc. La construction de la visualisation est statique, la manipulation directe (action sur les objets, menus) est utilisée comme mode d'interaction.

Wave Cet outil [KN95] organise les objets du Web dans les classes conceptuelles qui indiquent leurs similarités et leurs différences pour leur compréhension par l'utilisateur. Il utilise des agents autonomes pour analyser et classer les

informations. La construction de la visualisation est dynamique et en 3D. La personnalisation est semi-automatique. Les paramètres de visualisation varient en fonction des objets Web représentés.

NIRVE Sebrechts et al [SCL+99] proposent une évaluation de l'outil de visualisation NIRVE. Cet outil permet la visualisation de résultats de recherche. Elle peut être en mode textuel, en 2D et 3D. Plusieurs paramètres sont utilisés pour l'organisation de la visualisation (boite, globe, couleurs). Les liens entre les descripteurs et les concepts sont affichés avec une légende interactive, la personnalisation est manuelle. La représentation a été testée avec différentes métaphores [CLS00].

ET-MAP Cet outil [CSO96] utilise comme approche les réseaux de neurones afin d'analyser et classer automatiquement le contenu des documents (pages Web). La représentation est en 2D avec une hiérarchie de plusieurs niveaux. L'organisation de la visualisation utilise les cartes avec différents niveaux, ainsi que d'autres paramètres (couleurs, forme). La construction de la visualisation est dynamique avec une personnalisation automatique.

UNIVIT L'outil UNIVIT (Universal Interative Visualization Tool) proposé par Legrand et Soto [LeG02], [LS01], [LS00], permet de manipuler des documents XML grâce à l'utilisation du DOM (Document Object Model), en fournissant une visualisation en 2D ou 3D de tout fichier XML. L'information est organisée de façon hiérarchique avec des vues détaillées en présentant chaque partie de l'arborescence. La visualisation est construite dynamiquement afin de faciliter l'exploitation de l'information. La navigation est réalisée à l'aide de la souris et de menus. UNIVIT permet la visualisation de cartes thématiques et l'organisation des données.

L'objectif principal dans les systèmes de visualisation est d'améliorer l'interaction avec l'utilisateur en l'aidant dans sa navigation. L'analyse des systèmes présentés montre l'environnement suivant :

- Dynamique de construction de la visualisation. Tous les systèmes utilisent une construction dynamique de la visualisation à l'exception de StarViewer qui utilise une méthode de construction statique où la construction de la visualisation est basée sur une pré-analyse donnée pour l'administrateur pour le rendu visuel.
- Adaptativité. La plupart des systèmes ne sont pas adaptatifs à l'exception de WAVE qui utilise une adaptativité de personnalisation semi-automatique.
- Personnalisation. La personnalisation se fait de manière basique par la moitié des outils. Les outils qui utilisent une personnalisation intelligente tiennent compte des besoins des utilisateurs pour permettre un paramétrage de l'environnement automatique.

- Mode d'interaction. En ce qui concerne le mode d'interaction, tous les systèmes utilisent *(i)* la manipulation directe, en particulier UNIVIT et Wave qui permettent d'effectuer plusieurs opérations sur les objets (zoom, rotation, etc), *(ii)* les menus à différents niveaux, *(iii)* les formulaires sont utilisés par ET-MAP.
- Dimension. Tous les outils proposent des visualisations en 2D. La visualisation en 3D est également proposée dans beaucoup d'outils mais la personnalisation de la représentation graphique n'est pas toujours possible.
- La représentation graphique. La façon dont les éléments sont affichés (tailles, couleurs, ...) est plus ou moins importante selon les systèmes mais elle peut être une réelle façon d'apporter de l'information à l'utilisateur s'il peut personnaliser de façon importante cette représentation comme c'est le cas dans des systèmes comme *Navigational View Builder*.
- Les techniques conceptuelles. La majorité des systèmes s'appuient sur des représentations hiérachiques pour représenter l'information ou sur des structures de graphe. On peut noter cependant les approches proposées dans les systèmes comme *Web Book et Web Forager* qui utilise la notion de métaphore d'un livre pour représenter les informations.

L'analyse de ces systèmes permet d'identifier des caractéristiques souhaitables pour un système de visualisation et montre également que la prise en compte du type des informations à visualiser est importante pour choisir ces caractéristiques. L'annexe D présente deux tableaux de comparaison récapitulatifs des systèmes présentés selon les critères d'analyse retenus. Dans le cadre de notre travail, il est nécessaire de visualiser une ontologie et les requêtes d'un utilisateur. Nous allons donc, dans un premier temps nous intéresser plus spécifiquement à des outils de visualisation d'ontologies, puis nous présenterons les critères que nous avons retenus et notre outil au chapitre 7.

4.4 Visualisation d'ontologies

Les ontologies permettent de représenter toutes les connaissances de domaines précis. Mais cette masse d'informations est difficilement visualisable dans sa globalité, et beaucoup de travaux tentent de résoudre ce problème. Le but de cette section est de présenter différentes techniques et outils de visualisation d'ontologies existants.

Protégé Protégé[3] [NFM00] est l'outil le plus utilisé dans le domaine des ontologies. Il est intuitif et permet de créer ou modifier une ontologie de façon dynamique et conviviale. Protégé représente une ontologie de manière parcellaire et de façon textuelle. L'ontologie n'est pas affichée dans sa globalité, elle peut être affichée sous la forme de cinq parties :

3. http ://protege.stanford.edu/

- *Metadata* : afin de donner une brève description de l'ontologie en général, c'est à dire comment elle est codée et les différents espaces de noms associés.
- *OWL classes* : pour représenter la hiérarchie des classes et voir en détail la classe sélectionnée.
- *Properties* : de la même manière que *OWL classes*, il affiche toutes les propriétés associées aux classes avec une description détaillée.
- *Individuals* : affiche la liste des instances d'une classe.
- *Forms* : sert à gérer l'espace d'affichage des instances.

Plusieurs *plugins* d'affichage ont été proposés pour permettre la représentation graphique des informations de l'ontologie dans l'environnement Protégé. Nous présentons brièvement ici les *plugins* Jimbalaya, GViz et OntoViz.

Jambalaya Jambalaya [4] [SMS+01] est un *plugin* de Protégé offrant plusieurs affichages alternatifs à ceux de Protégé. Ces affichages sont fortement basés sur la théorie des graphes et la théorie des *treemaps*. Un *treemap* est une visualisation d'une hiérarchie sous forme de rectangles imbriqués. Il permet d'avoir accès en un coup d'œil à toutes les feuilles d'un arbre et de remonter aisément la hiérarchie. Jambalaya propose quatre types de vues :

- *Class and individual tree* : cette vue sert à représenter l'ontologie sous la forme d'un arbre, une option permet de le disposer de manière radiale.
- *Nested treemap* : cette vue permet d'avoir accès rapidement à l'arbre des classes et des instances. Cette vue ne permet pas d'afficher les relations.
- *Domain/Range* : cette vue affiche quant à elle toutes les relations à l'aide d'une flèche allant du domaine à la classe cible. Cela permet de voir l'organisation relationnelle de l'ontologie.
- *Nested view* : cette vue est un mélange des deux précédentes. Elle affiche les classes et les instances sous forme de *treemap* et les relie avec les relations comme dans la vue domain/range.

La figure 4.1 illustre l'affichage avec la vue *Class and individual tree*.

TGViz (Touchgraph Visualization Tab) [H.A03] est un *plugin* de Protégé qui permet de visualiser en deux dimensions la représentation graphique des instances et les liens ou propriétés qui existent entre elles. Il permet à l'utilisateur de modifier la profondeur du graphe, changer les couleurs, effectuer des rotations et zoomer. La figure 4.2 montre un exemple de visualisation avec TGViz.

OntoViz [Sin03] est un *plugin* permettant à Protégé de visualiser en deux dimensions des ontologies. OntoViz se compose de trois parties : (i) une partie pour visualiser l'ontologie, (ii) une autre représentant sous forme d'arbre les classes de l'ontologie et la dernière (iii) pour indiquer quelles sont les classes et les instances qui sont présentées dans la partie de droite. Cette dernière partie

4. http ://Webhome.cs.uvic.ca/ chisel/projects/jambalaya/jambalaya.html

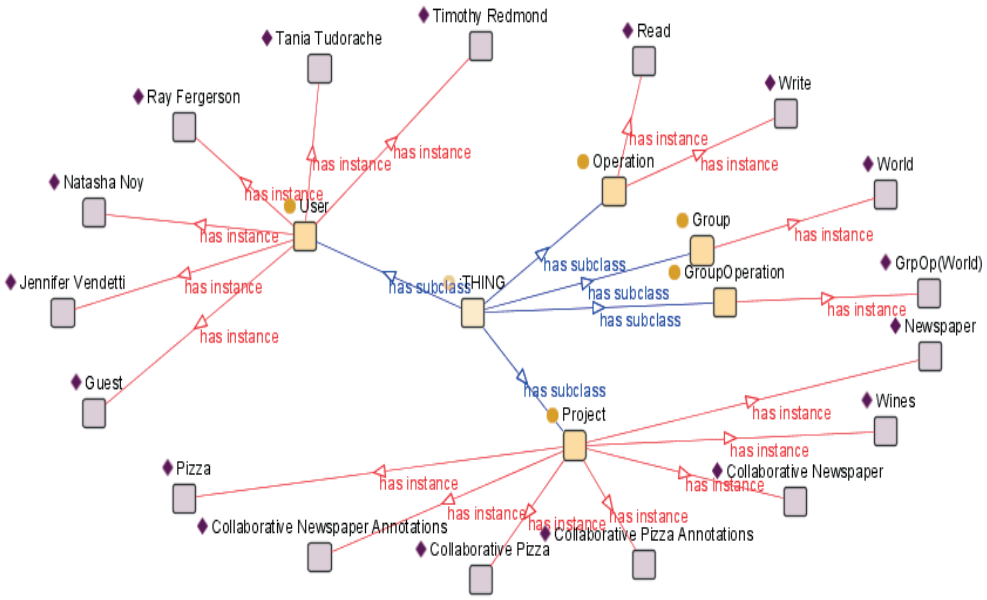

FIGURE 4.1 – La vue des classes et individus de jambalaya

comporte des cases à cocher pour indiquer les paramètres de présentation. L'utilisateur peut sélectionner les classes et instances à afficher, et également zoomer sur des parties du graphe. La figure 4.3 illustre la visualisation sous Protégé avec le plugin OntoViz.

Il existe cependant d'autres environnements de création et d'édition d'ontologies.

Kaon Le système KAON [5] fournit un environnement complet pour le développement d'applications basées sur les ontologies. L'affichage dans Kaon est basé sur le concept de graphe que l'on peut déplier. Le graphe obtenu se réorganise dynamiquement pour essayer d'être le plus lisible possible. Chaque type d'élément possède sa propre couleur et, comme dans Protégé, quand on sélectionne un élément, ses informations détaillées sont affichées dans une fenêtre à part pour pouvoir regarder en détail ses différentes propriétés. La figure 4.4 montre un exemple de visualisation avec KAON.

5. http ://kaon.semanticWeb.org/

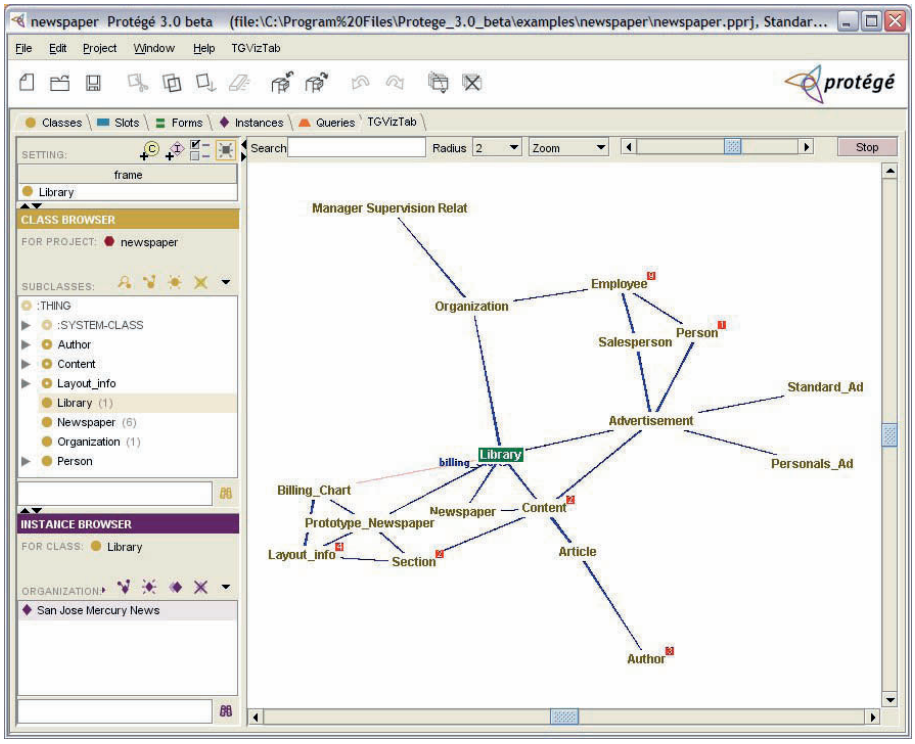

FIGURE 4.2 – Visualisation avec TGViz

SWOOP [6] se rapproche dans sa présentation de Protégé, en étant moins complet. Il utilise un affichage basé sur trois éléments : un arbre des classes, un arbre des propriétés et une liste de tous les éléments de l'ontologie. Les données précises sur l'élément sélectionné sont données dans une *frame* séparée. Il propose une fonctionnalité permettant d'afficher l'ontologie sous forme de *crop circle*. Il représente chaque classe sous forme de disque et le disque d'une sous classe est donc contenu dans le disque de la superclasse, ce qui est similaire à la notion de *treemap*.

OntoSphere [7] [BBP05] propose plusieurs vues permettant d'afficher une partie de l'ontologie en trois dimensions. La première vue permet d'afficher les relations de l'ontologie en répartissant les concepts sur une sphère et en symbolisant les relations à l'aide de vecteurs. La deuxième vue permet de modéliser la hiérarchie des classes sous la forme d'un cône. La dernière vue permet de présenter dans le détail un concept en affichant tous les liens qu'il a avec un autre élément de l'ontologie.

6. http ://www.mindswap.org/2004/SWOOP/
7. http ://ontosphere3d.sourceforge.net/

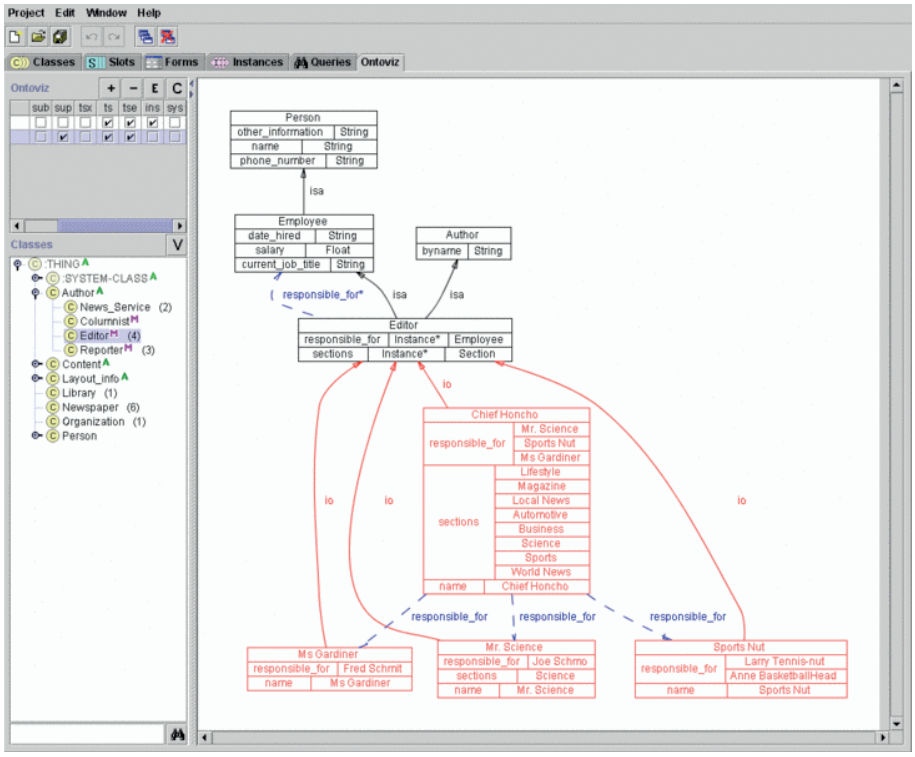

FIGURE 4.3 – Visualisation avec OntoViz

4.5 Analyse

Protégé permet d'afficher l'ontologie en plusieurs parties. Séparer les différents types d'informations permet de mieux structurer une ontologie volumineuse mais rend impossible la visualisation d'informations éloignées dans l'ontologie. La taille des ontologies type dans *Kaon* dépasse largement la capacité d'affichage, ce type de visualiseur est très performant pour des ontologies très ciblées et donc peu volumineuses. La logique de changer la couleur suivant le type d'élément est très intéressante, par contre si les noms ne sont pas clairs, il peut paraître difficile d'associer les couleurs aux éléments. L'affichage de *jambalaya* est très complémentaire de protégé et permet de donner à ce programme une interface de visualisation intuitive. La représentation dans *swoop* est très intéressante car elle permet d'avoir une vue d'ensemble de l'arbre des classes et de connaitre rapidement les instances d'une classe. Par contre, l'affichage définit beaucoup d'espace vide et la vue paraît trop grande par rapport à son contenu. La représentation des relations est également difficile à visualiser. En ce qui concerne *OntoViz* et *TGViz* seules les relations d'héritage sont visibles. La visualisation devient illisible avec une ontologie volumineuse (nombre de nœuds important) [KTV+08].

Tous ces outils permettent une visualisation en deux dimensions, ce type d'affichage ne permet pas de faire apparaître de manière claire l'ontologie dans

néanmoins la version 2 du projet se base sur OWL-DL ce qui palliera ce défaut.

L'affichage dans Kaon est basé sur le concept de graphe que l'on peut déplier.

Le graphe obtenu se réorganise dynamiquement pour essayer d'être le plus lisible possible néanmoins la taille des ontologies types dépasse largement la capacité d'affichage comme le montre l'exemple (Figure 2.3) qui n'est qu'à moitié deployé.

Chaque type d'éléments possède sa propre couleur et, comme dans protégé, quand on sélectionne un élément, ses informations détaillées sont affichées dans une frame à part pour pouvoir regarder en détail ses différentes propriétés.

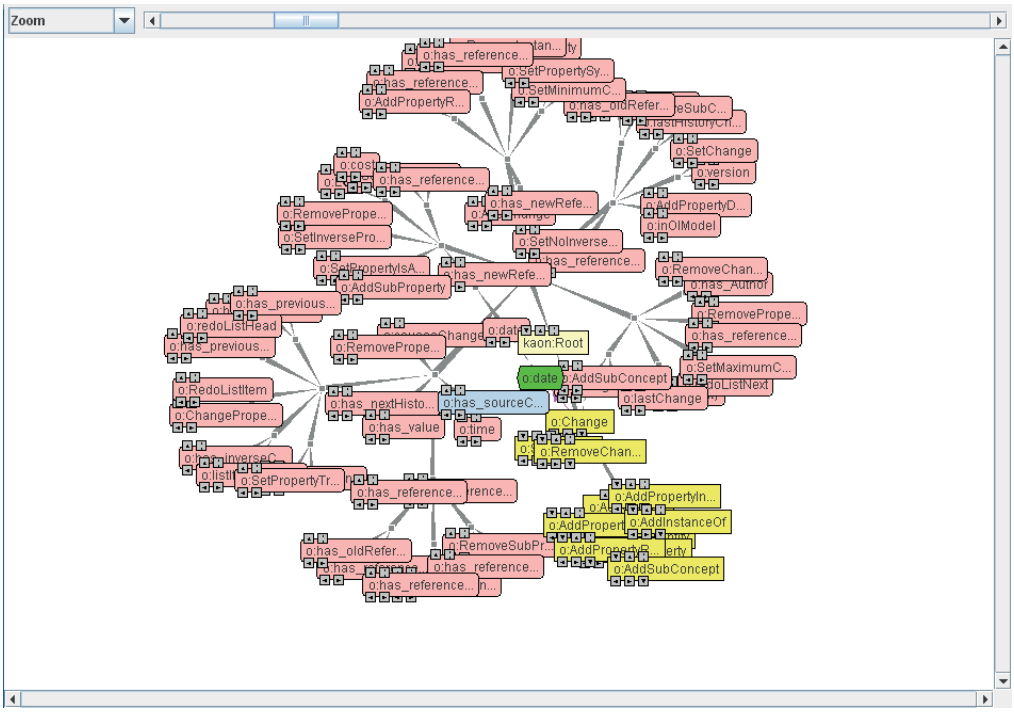

FIG. 2.3 – graphe obtenu avec kaon

FIGURE 4.4 – Visualisation avec KAON

Analyse

Ce type de visualiseur est très performant pour des ontologies très ciblées et donc peu volumineuses. Mais sa lisibilité est inversement proportionnelle à sa taille.

La logique de changer la couleur suivant le type d'élément est très intéressante malheureusement si les noms de ses éléments ne sont pas clairs, il peut paraître difficile d'associer les couleurs aux éléments.

4.6 Conclusion

sa globalité. *Protégé* le fait mais en la divisant en sous parties. En ce qui concerne les outils de visualisation en trois dimensions, dans *ontosphere*, plus le nombre de relations affichées est important, plus il est difficile de discerner les concepts cachés et de différencier les relations. Dans la vue détaillée, la visualisation est focalisée sur un seul élément, et ne permet pas de voir la place du concept dans l'ontologie.

[7]Kaon 2 : `http://kaon2.semanticweb.org`

La visualisation dans le Web sémantique est un domaine peu étudié jusqu'à présent : les travaux concernant la visualisation se limitent généralement à la visualisation d'ontologies et ne traitent pas de la recherche d'information utilisant ces technologies du Web sémantique. Dans notre travail, nous proposons une méthodologie de visualisation de résultats dans le processus de recherche d'information en exploitant les relations sémantiques entre les ressources et ceci en nous basant sur une ontologie de référence. Ce travail s'inscrit dans le cadre du développement plus complet d'une architecture de coopération de sources d'information hétérogènes basée sur l'utilisation d'ontologies pour exprimer la sémantique des informations partagées. Cette architecture appelée OWSCIS (Ontology and Web Service based Cooperation of Information Sources) est présentée dans le chapitre suivant.

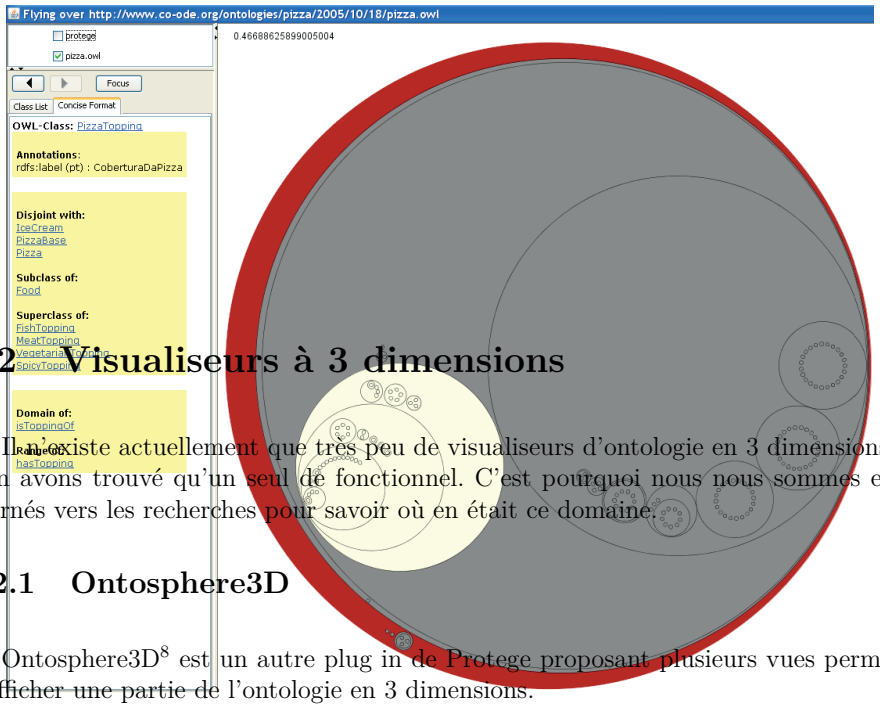

FIG. 2.8 – Crop circles de Swoop

FIGURE 4.5 – Visualisation avec SWOOP

Analyse

Cette représentation est très intéressante car elle permet d'avoir une vue d'ensemble de l'arbre des classes et de connaitre rapidement les instances d'une classe.

Un au c cette représent ésente et nous n e pour savoir qu

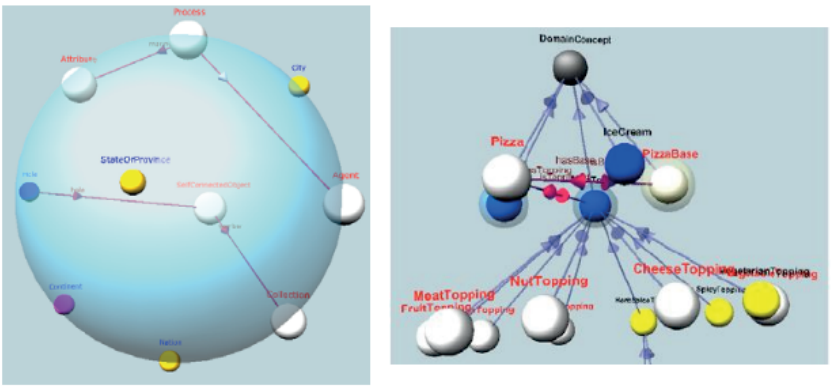

FIG. 2.9 – vue relationnelle et vue hiérarchique d'ontosphere3D

FIGURE 4.6 – Visualisation avec OntoSphere

La dernière vue permet d'obtenir une version détaillée d'un concept en affichant tous les liens qu'il a avec un autre élément de l'ontologie. Cette vue est vraiment focalisée sur un seul élément et ne permet pas de se rendre compte de la place qu'occupe cet élément par rapport au reste de l'ontologie (Figure 2.10).

2.2 Visualiseurs à 3 dimensions

Il n'existe actuellement que très peu de visualiseurs d'ontologie en 3 dimensions nous n'en avons trouvé qu'un seul de fonctionnel. C'est pourquoi nous nous sommes ensuite tournés vers les recherches pour savoir où en était ce domaine.

2.2.1 Ontosphere3D

Ontosphere3D[8] est un autre plug in de Protege proposant plusieurs vues permettant d'afficher une partie de l'ontologie en 3 dimensions.

La première vue permet d'afficher les relations de l'ontologie en répartissant les concepts sur une sphère et en symbolisant les relations à l'aide de vecteurs (Figure 2.9), lorsque le nombre de relations affichées est important, plus il est difficile de discerner les concepts cachés et de différencier les relations.

La seconde vue permet de modéliser la hiérarchie des classes sous la forme d'un cône. Mais cela devient vite confus c'est pourquoi une profondeur est fixée et si l'on descend cela replie le haut de la hiérarchie.

Malheureusement, l'affichage proposé par Swoop définit beaucoup d'espace vide et la vue paraît trop grande par rapport à son contenu.

Ontosphère est à notre avis assez proche de Jambalaya sans réussir à l'égaler.

[8] Ontosphere3D : http://ontosphere3d.sourceforge.net/index.html

Deuxième partie

Approche retenue

5

Interrogation et visualisation sémantique dans une coopération de Systèmes d'Information

Dans ce chapitre, nous présentons le contexte et les objectifs de notre travail et nous donnons une vue d'ensemble des solutions proposées qui portent sur (i) la définition d'une architecture de coopération de systèmes d'information basée sur l'utilisation d'ontologies ii) la spécification d'une méthode d'enrichissement de requêtes et (iii) la visualisation des informations.

Sommaire

5.1 Contexte et objectifs

Notre travail s'inscrit dans la continuité des recherches menées dans le domaine de la coopération de systèmes d'information et des systèmes de recherche d'information au sens large. Les coopérations de systèmes d'information ont évolué dans le temps vers une meilleure prise en compte de la sémantique des informations partagées. Les architectures ont progressé depuis des architectures de type fédérations (faiblement ou fortement couplées) avec des intégrations plus ou moins fortes des schémas des bases de données vers des architectures plus ouvertes de type médiateurs - wrappers ; les wrappers étant chargés des traductions au niveau des sources locales et les médiateurs étant chargés de gérer les échanges et les requêtes au niveau de la coopération. Ces architectures permettent de prendre en compte différentes sources d'information comme des bases de données mais également des sources de type semi-structurées comme les documents XML. Elles peuvent également permettre une meilleure prise en charge de la sémantique des informations notamment en se référant à une représentation partagée des connaissances à l'aide d'ontologies. Ces préoccupations de meilleure prise en compte de la sémantique des informations, rejoignent les travaux développés dans le cadre plus large de la recherche d'information et notamment les travaux émergents dans le domaine du Web sémantique. Dans ce domaine, plusieurs travaux se sont intéressés à la recherche sémantique pour améliorer la pertinence des réponses aux recherches, mais ceci a également augmenté la complexité du processus de recherche d'information. Les moteurs de recherche permettent de trouver des ressources, en général associées à des mots clés, en se basant sur des annotations associées aux ressources. L'indexation de documents consiste à repérer dans le contenu d'un document certains mots ou expressions significatives dans un contexte donné. Il est également important dans le processus de recherche d'information, que l'interrogation se fasse de manière automatique sans manipulation de la part de l'utilisateur et qu'il reçoive une réponse sans connaître les étapes intermédiaires. La recherche d'information est une tâche complexe dont l'élément de base qui est la formulation d'une requête, est à la charge de l'utilisateur qui peut être plus ou moins qualifié. Cette formulation peut aller de l'expression de simples mots-clés à des phrases en langage naturel, ou des requêtes plus structurées notamment dans le cadre de la coopération de systèmes d'information. Quel que soit le mode d'interrogation, un challenge important est la mise en place d'outils d'aide pour les utilisateurs. Cette aide peut varier de la simple auto-complétion des mots ou des expressions à des méthodes plus riches d'analyses des comportements des utilisateurs ou toute autre technique.

Notre travail vise à apporter des solutions pour enrichir le processus d'interrogation d'une coopération de systèmes d'information basée sur l'utilisation d'ontologies pour représenter la sémantique des informations partagées et à répondre à la problématique de ce domaine particulièrement sur trois points : (i) la définition d'une architecture de coopération de systèmes d'information basée sur les ontologies qui intègre un service d'interrogation avec (ii) une

méthode d'enrichissement des requêtes pour guider l'utilisateur dans ses recherches et (iii) un service de visualisation des informations qui peuvent être les connaissances partagées, les requêtes et les résultats.

Plusieurs problèmes se posent qui concernent d'une part la mise en place d'une architecture de coopération qui permette la prise en compte de la sémantique des informations partagées et d'autre part la nécessité d'aider et guider les utilisateurs pour l'interrogation de la coopération, avec des outils d'aide et de visualisation conviviaux. L'architecture retenue, appelée OWSCIS (Ontology and Web Service based Cooperation of Information Sources), s'appuie sur une représentation de la sémantique des informations de la coopération à deux niveaux. Au niveau local, les informations sont mises en correspondance avec une ontologie locale qui est une modélisation partielle des informations partagées de la coopération en lien avec les informations des sources de données locales. Au niveau global, une ontologie de domaine recouvre l'ensemble des informations partagées par la coopération et les ontologies locales sont mises en correspondance avec l'ontologie de domaine appelée ontologie de référence. La section 5.2 donne un aperçu général de l'architecture OWSCIS et de son mode de fonctionnement pour l'interrogation. L'architecture retenue peut être vue comme une architecture de type médiateurs - wrappers avec des outils de traduction et de mise en correspondance au niveau des sources de données locales (wrappers) et des outils et services au niveau de la coopération (médiateurs). L'interrogation de la coopération se fait au niveau de l'ontologie de référence. Les différents mappings mis en place entre les sources de données et les ontologies locales et entre les ontologies locales et l'ontologie de référence permettent d'interroger la coopération, de manière transparente pour l'utilisateur, sur les différentes sources d'informations grâce au service d'interrogation qui se charge des étapes de décomposition et recomposition des requêtes.

Dans ce contexte, un problème important concerne l'interaction des utilisateurs avec le système coopératif et donc avec les informations contenues dans l'ontologie de référence pour spécifier sa requête, car l'utilisateur ne connaît pas a priori les termes de l'ontologie de référence et il ne sait pas comment exprimer sa requête ni quelle est la meilleure façon d'obtenir des résultats pertinents. Il est donc nécessaire de mettre en place des méthodes et des outils d'aide à l'utilisateur.

La figure 5.1 illustre l'interaction entre un utilisateur et le système coopératif. L'interrogation passe par l'interrogation de l'ontologie de référence qui est incluse dans un ensemble appelé "Base de connaissances" et qui comporte, en plus de l'ontologie de référence, un annuaire de mapping qui lie les informations de l'ontologie de référence aux informations des sources de données locales grâce aux ontologies locales. Cette base de connaissance comporte également des outils utilisés lors de l'ajout d'une nouvelle source d'information dans la coopération.

La visualisation des informations manipulées par l'utilisateur est également un élément essentiel pour l'interrogation de la coopération. Ces informations

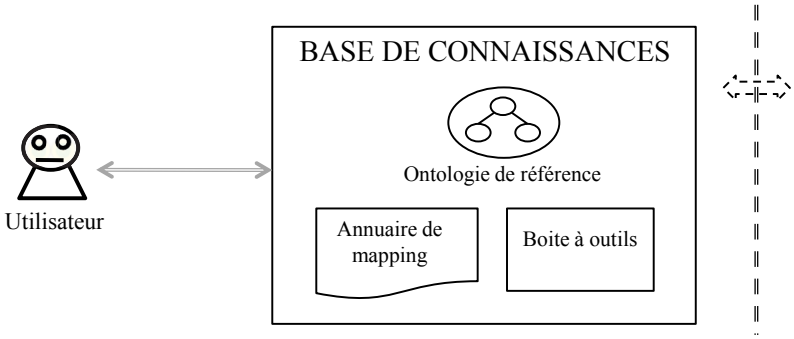

FIGURE 5.1 – Interaction utilisateur-système

incluent l'ontologie de référence sur laquelle doit être émise la requête, la visualisation de la requête et de ses résultats. Il est également nécessaire de définir des outils d'aide à la construction d'une requête posée sur l'ontologie de référence et d'aide pour l'enrichissement de la requête pour guider l'utilisateur dans une séquence de recherche.

Pour atteindre ces objectifs, notre travail est centré sur la proposition de méthodes et la mise en place d'outils permettant d'aider et de guider les utilisateurs dans le processus de recherche d'information et en les assistant dans la construction de requêtes. Il se compose essentiellement de trois parties qui sont :

- la prise en compte dans l'architecture d'un service de visualisation et d'un service de requête plus complet permettant de guider l'utilisateur dans ses recherches,
- une méthode d'enrichissement des requêtes, intégrée dans le service de requête,
- la visualisation de l'ontologie de référence, de la requête posée par l'utilisateur ou enrichie et des résultats.

La figure 5.2 donne un extrait de l'architecture d'OWSCIS pour les services concernés par notre travail. Ces services interagissent avec la base de connaissances et essentiellement à travers l'ontologie de référence.

L'objectif de notre approche est d'enrichir la requête d'un utilisateur en lui fournissant les résultats de sa requête, mais également une liste de termes associés aux termes de sa requête initiale. Les termes complémentaires proposés sont ceux jugés comme pertinents ou proches des termes initiaux de la requête. Ces termes peuvent être utilisés par l'utilisateur pour affiner sa requête ou pour construire une nouvelle requête. Le processus d'interrogation comporte trois étapes principales.

1. Première étape : l'utilisateur exprime une requête qui est posée sur l'ontologie de référence en se référant à ses concepts et ses propriétés. L'onto-

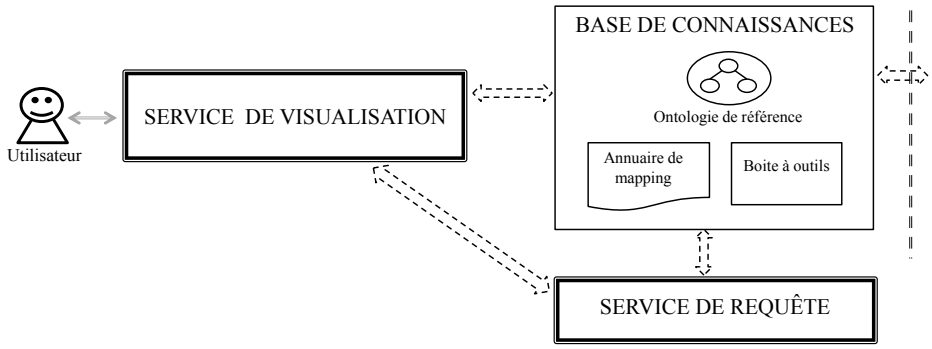

FIGURE 5.2 – Services de requête et de visualisation

logie de référence ne comporte pas d'instances. Les informations sources sont gérées dans les sources de données locales et accessibles par le biais des ontologies locales liées à l'ontologie de référence. Le service de requête est chargé de l'exécution de la requête et fournit un document résultat exprimé dans les termes de la requête, et sous un format XML. La construction de la requête est assistée par un outil spécifique.

2. Deuxième étape : parallèlement à la résolution de la requête initiale par le service de requête, un enrichissement de cette requête est réalisé. Cette phase d'enrichissement consiste tout d'abord en une analyse de la requête initiale pour en extraire ses concepts et éventuellement les compléter par des concepts voisins (synonymes ou liés) dans l'ontologie de référence. Puis, la phase d'enrichissement qui consiste à rechercher dans l'ontologie de référence des concepts jugés "proches" pour enrichir la requête initiale est effectuée. La recherche des concepts complémentaires se base sur la notion de popularité d'un concept. Un aperçu de la méthode d'enrichissement est donné dans la section 5.3 et le détail complet est décrit dans le chapitre 6. Une première phase d'enrichissement de cette requête est réalisée par le service d'interrogation qui extrait les concepts

3. Troisième étape : elle concerne la visualisation proprement dite des informations (requête, requête enrichie et résultats) pour l'utilisateur.

Nous présentons dans la suite de ce chapitre l'architecture de coopération OWSCIS et nous donnons un aperçu de la méthode d'enrichissement de requêtes retenue et les principes de visualisation adoptés dans le processus de recherche d'information. Le chapitre 6 détaille la méthode d'enrichissement et le chapitre 7 est dédié aux aspects visualisation.

5.2 L'architecture générale : OWSCIS

L'architecture OWSCIS (Ontology and Web Service based Cooperation of Information Sources) permet la coopération de sources de données hétérogènes. Les sources d'information locales possèdent des données sous forme semi-structurées comme des documents XML ou dans des bases de données relationnelles par exemple. La sémantique des informations locales est exprimée à l'aide d'ontologies locales qui peuvent être soit préexistantes aux données, soit créées lors de l'adhésion de la source locale à la coopération. Ces ontologies locales sont mises en correspondance avec une ontologie de référence de domaine qui représente la sémantique de l'ensemble des données de la coopération. Un utilisateur peut poser une requête sur la coopération sur l'ontologie de référence et accéder de façon transparente à l'ensemble des données de la coopération. Des outils pour la mise en correspondance des sources de données locales avec les ontologies locales, et des ontologies locales avec l'ontologie de référence ont été développés. L'architecture s'appuie sur la définition de services pour les requêtes, la visualisation et la mise en correspondance des ontologies locales avec l'ontologie de référence. La figure 5.3 décrit succinctement cette architecture.

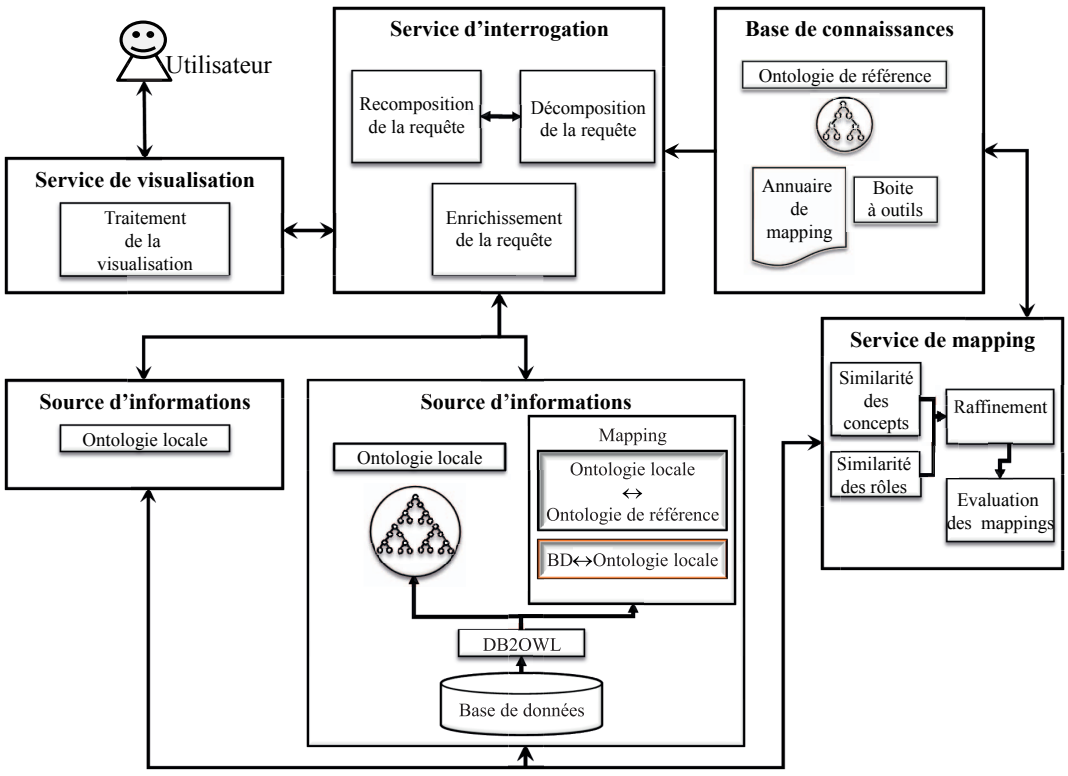

FIGURE 5.3 – Architecture générale d'OWSCIS

Cette architecture est composée de cinq composants principaux :

- Un service d'interrogation qui permet à l'utilisateur de soumettre une requête. Cette requête est posée sur l'ontologie de référence. Elle est décrite en langage SPARQL. Elle est décomposée en sous-requêtes qui sont envoyées aux fournisseurs de données pertinents. Puis les résultats retournés par les fournisseurs sont recomposés et transmis avec la requête au module de visualisation. Un module spécifique permet l'enrichissement de la requête avant son exécution pour permettre une visualisation plus complète des résultats. Un outil d'aide à la construction de la requête permet d'assister l'utilisateur pour décrire sa requête de façon graphique et interactive.
- Les fournisseurs de données sont les sources d'informations locales. La sémantique de ces informations est exprimée à l'aide des ontologies locales qui sont elles mêmes reliées à l'ontologie de référence. Ils assurent la résolution des sous-requêtes qui leur sont transmises en utilisant les *mappings* entre les données source et les ontologies locales. Les résultats sont donnés sous forme de fichiers SPARQL. Un outil DB2OWL [CGY07] a été développé pour la génération semi-automatique d'une ontologie locale à partir d'une base de données relationnelle.
- La base de connaissances comporte l'ontologie de référence, une bibliothèque des méthodes de calculs utilisées pour le rapprochement des ontologies locales et de l'ontologie de référence et un catalogue de fournisseurs de données avec des informations sur le contenu de ces sources de données.
- Le service de *mapping* permet la mise en correspondance l'ontologie locale et l'ontologie de référence. Ce service s'appuie sur les méthodes de comparaison sémantique décrites dans la base de connaissances.
- Le service de visualisation qui propose deux modules principaux qui permettent :

  1. la visualisation de l'ontologie de référence permettant à l'utilisateur de naviguer dans l'ontologie de façon conviviale et pour l'aider à exprimer sa requête.
  2. la visualisation des requêtes, des enrichissements et des résultats.

Plusieurs travaux ont déjà été réalisés dans le cadre de la mise en place de l'architecture OWSCIS. Ils concernent :

- d'une part les travaux développés dans le cadre de la thèse de Thibault Poulain [Pou09] essentiellement centrés sur la définition d'une méthode semi-automatique de mise en correspondance des ontologies locales et de l'ontologie de référence. Cette méthode s'appuie sur la notion de calcul d'une distance sémantique entre les termes des ontologies. Plusieurs mesures peuvent être utilisées et pondérées. Ces méthodes sont intégrées dans la boîte à outils de la base de connaissance d'OWSCIS.
- d'autre part les travaux développés dans le cadre de la thèse de Raji Ghawi [GHA10] qui s'est intéressé à l'ensemble des méthodes et outils

permettant la mise en correspondance des sources de données locales et des ontologies locales à la fois pour la représentation des connaissances et également pour l'interrogation de la coopération.

Notre travail vient compléter ces travaux au niveau de l'interaction de l'utilisateur avec la coopération OWSCIS au niveau du processus d'interrogation. Le service de visualisation adopté dans l'architecture OWSCIS permet à l'utilisateur de visualiser l'ontologie de référence, la requête, l'enrichissement de la requête et des résultats sous différentes formes, en utilisant différentes techniques de visualisation.

5.3 L'enrichissement de requêtes : QUEXME

D'une manière générale, l'enrichissement de requêtes vise à aider un utilisateur à soumettre une requête à un moteur de recherche que ce soit sur le Web ou dans un autre système. L'idée principale est d'enrichir la requête initiale de l'utilisateur afin de rendre le résultat le plus pertinent possible. Le processus d'enrichissement est réalisé en deux phases : (i) la formulation de la requête par l'utilisateur, et (ii) l'enrichissement de la requête. Ce processus peut être visible ou pas pour l'utilisateur. La requête peut être enrichie de façon automatique avant d'être résolue ou bien des éléments d'enrichissement peuvent être proposés à l'utilisateur pour améliorer sa recherche.

L'approche développée dans la méthode QUEXME (QUery EXpansion METhod) s'appuie sur une analyse du comportement usuel des utilisateurs pour proposer à un utilisateur qui soumet une requête, des termes additionnels à sa requête initiale. Le processus d'enrichissement de requêtes est illustré par la figure 5.4.

Le processus d'expansion des requêtes vise à proposer à l'utilisateur des termes pertinents pour enrichir sa requête et le guider dans sa recherche d'information. Les différentes étapes sont les suivantes :

1. L'utilisateur soumet sa requête qui est exprimée dans les termes de l'ontologie de référence. La requête émise sur la coopération est spécifiée en langage SPARQL. La construction de la requête est guidée de façon graphique et interactive. Le processus d'enrichissement est basé sur l'analyse des concepts décrits dans la requête. Ces concepts sont extraits de la requête initiale. Pour le processus d'enrichissement, une requête peut donc être vue comme un ensemble de concepts.
2. L'algorithme d'enrichissement est appliqué, des concepts sont ajoutés à la requête initiale. Les concepts ajoutés sont également des concepts issus de l'ontologie de référence.
3. La liste des concepts de la requête initiale et les concepts additionnels sont proposés à l'utilisateur avec les résultats de la requête initiale. Il peut affiner sa requête en utilisant ces nouveaux termes ou en sélectionnant certains d'entre eux.

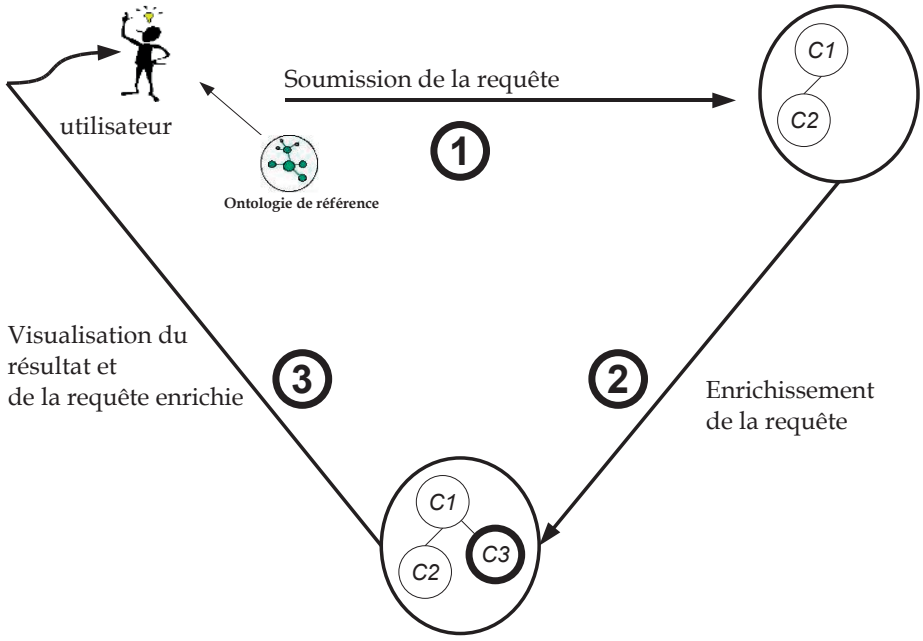

FIGURE 5.4 – Processus d'enrichissement

Le processus d'enrichissement de requêtes peut être décomposé en deux phases : une phase d'apprentissage qui permet l'initialisation du processus d'enrichissement qui est faite à partir d'un jeu de requêtes (plusieurs séquences de requêtes des utilisateurs modèles) et une phase d'enrichissement qui comporte les mêmes étapes que la phase d'apprentissage pour le traitement d'une requête mais en plus, l'enrichissement proprement dit. La figure 5.5 illustre ces deux phases.

Phase d'apprentissage. C'est la première phase du processus de recherche. Cette étape est basée sur l'analyse d'un ensemble de séquences de requêtes correspondant à des sessions de recherche - modèles - d'utilisateurs. L'enchaînement des requêtes est utilisé afin de construire un graphe de concepts. Ce graphe lie les concepts utilisés dans des requêtes successives et est utilisé dans la phase d'enrichissement.

Phase d'enrichissement. C'est la phase d'enrichissement à proprement parler. Il s'agit de sélectionner de nouveaux concepts considérés comme pertinents pour enrichir la requête de l'utilisateur. La notion de pertinence ou de voisinage sémantique s'appuie sur la notion de popularité d'un concept et celle d'importance d'un concept par rapport à un ensemble de concepts. A la fin de cette phase, les nouveaux concepts sont proposés à l'utilisateur qui peut les

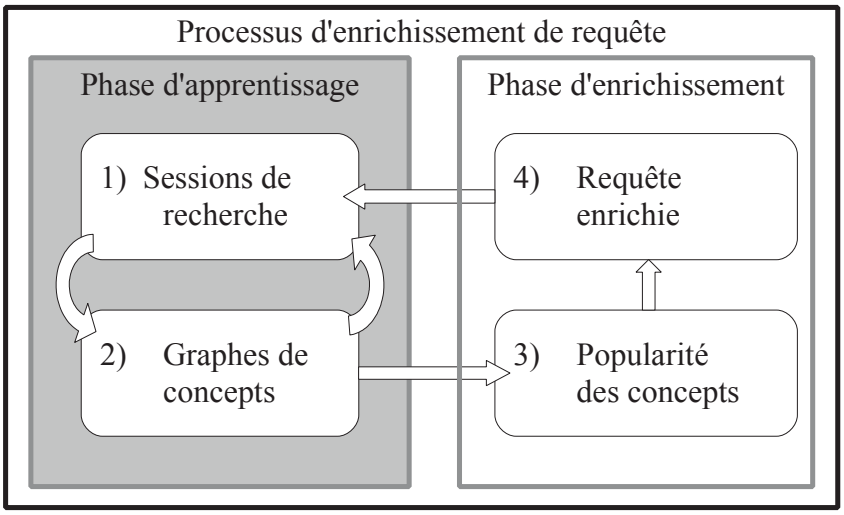

FIGURE 5.5 – Les deux phases du processus d'enrichissement des requêtes

utiliser pour reformuler sa requête ou poursuivre sa séquence de recherche. Le chapitre 6 détaille la méthode d'enrichissement.

5.4 La visualisation des informations : SEVI

Nous donnons dans cette section un aperçu général du service de visualisation SEVI (SErvice of VIsualization) et des différents modules qui le composent. Le processus de visualisation des informations doit permettre de répondre aux deux questions : "Quoi visualiser ?" et "Comment ?". Les informations à visualiser sont liées au processus d'interrogation mis en œuvre dans OWSCIS et concernent l'ontologie de référence pour la construction de la requête et les informations liées à la requête elle-même. La requête étant émise sur l'ontologie de référence, il est possible de proposer une visualisation de la requête sous forme d'un extrait de cette ontologie après analyse de ses concepts et de ses propriétés. Les informations à visualiser concernent également les résultats de la requête initiale qui pourra avoir été complétée pour améliorer la visualisation de ces résultats et la requête enrichie avec la liste des concepts additionnels proposés avec l'extrait de l'ontologie enrichie correspondant.

Les techniques de représentation visuelle choisies répondent à la question "Comment ?". Le service de visualisation permet l'affichage des informations sous différentes formes en 2D et selon les informations en 3D. L'environnement est paramétrable par l'utilisateur. Il offre des outils conviviaux pour afficher les informations en se basant sur l'ontologie de référence sur laquelle sont exprimées les séquences de recherche dans la coopération.

La visualisation des informations vise à fournir à l'utilisateur un environnement dynamique pour l'interrogation en représentant les concepts et les pro-

priétés d'une requête et son voisinage sémantique lié aux concepts additionnels proposés ainsi que les résultats des requêtes sous différentes formes. L'environnement comporte également des outils d'aide notamment pour la construction des requêtes. L'utilisateur peut paramétrer son environnement de visualisation. La figure 5.6 illustre l'architecture générale du service de visualisation.

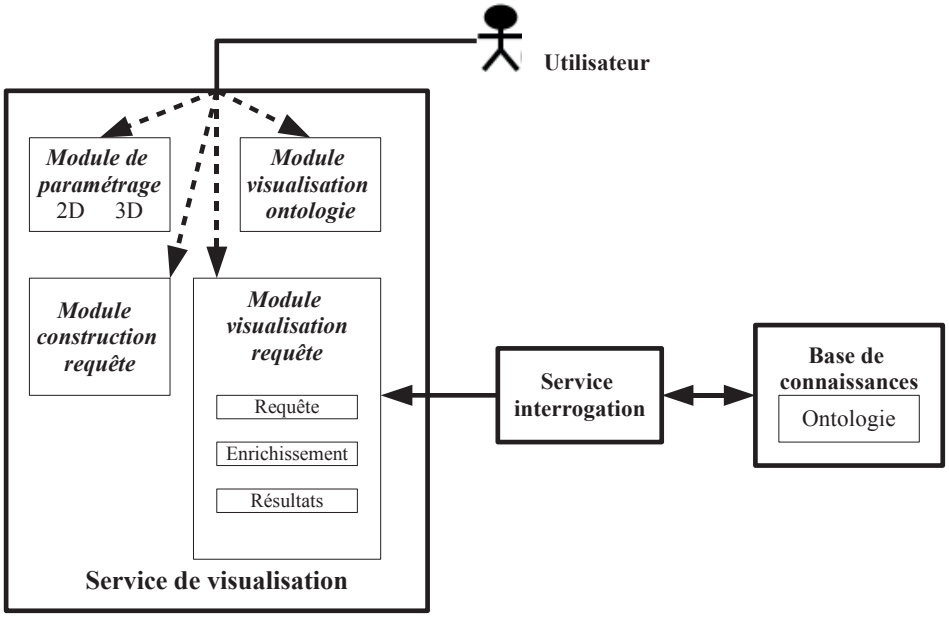

FIGURE 5.6 – Architecture générale du service de visualisation

Le module de visualisation de l'ontologie permet à l'utilisateur de visualiser l'ontologie de référence. Cette ontologie est incluse dans la base de connaissance de l'architecture OWSCIS. Elle peut être affichée sous différentes formes (2D et 3D) et l'utilisateur peut paramétrer son environnement d'affichage et sélectionner les éléments qu'il souhaite visualiser.

Le module de paramétrage permet la personnalisation de l'environnement pour les différentes visualisations proposées. Ces paramètres concernent l'organisation de la visualisation et le rendu graphique (choix des couleurs pour les concepts par exemple pour différencier différents thèmes, choix des couleurs et des formes pour les arcs pour noter les différentes relations, etc.).

Le module de visualisation de requête est composé de trois sous modules : le sous-module de visualisation d'une requête avec ses concepts et ses

relations, le sous-module enrichissement pour prendre en compte et visualiser les concepts additionnels proposés et le sous-module de visualisation des résultats de la requête.

Le module de construction de requête permet d'aider et d'assister l'utilisateur dans le processus de recherche pour la construction des requêtes.

Le chapitre 7 décrit l'ensemble l'environnement de visualisation et les outils proposés pour l'aide à l'interrogation dans l'architecture de coopération OWSCIS.

5.5 Conclusion

Nous avons présenté dans ce chapitre le contexte et les objectifs du travail proposé qui s'intègre dans le cadre général de la définition d'une architecture de coopération basée sur l'utilisation d'ontologies, appelé OWSCIS (Ontology and Web Service based Cooperation of Information Sources).

Notre contribution concerne l'interaction de l'utilisateur avec la coopération pour son interrogation. Pour prendre en compte cette interaction et proposer aux utilisateurs un environnement dynamique et convivial, deux services ont été définis qui visent à permettre l'interrogation de la coopération de façon assistée avec des outils d'aide pour la construction des requêtes et pour leur enrichissement et à offrir des outils pour la visualisation des informations de façon personnalisée dans un environnement riche en s'appuyant sur l'ontologie de référence.

- Le service de visualisation a la charge de la présentation (organisation et rendu graphique) des informations manipulées lors de l'interrogation de la coopération i.e. la requête et son voisinage sémantique qui inclut les concepts additionnels proposés par l'enrichissement, et les résultats des requêtes. Lors d'une session de recherche d'information, l'utilisateur peut émettre une requête initiale puis relancer d'autres requêtes en s'aidant des concepts additionnels présentés. Les informations sont présentées de façon dynamique et paramétrable pour faciliter l'interaction de l'utilisateur sur l'environnement pour l'aider dans ses recherches.
- Le service d'interrogation intègre une méthode d'enrichissement de requêtes pour guider et aider l'utilisateur dans la construction de ses requêtes. L'enrichissement est basé sur une analyse du comportement usuel des utilisateurs lors d'une session de recherche. De nouveaux termes issus de l'ontologie de référence, sont proposés dans la phase d'interrogation pour enrichir la requête initiale. Notre approche est basée sur la notion de **popularité** d'un concept par rapport à un autre concept et par rapport à un ensemble de concepts (requête).

La méthode d'enrichissement QUEXME (QUery EXpansion MEthod) et l'environnement de visualisation sont présentés respectivement aux chapitres 6 et 7.

6

L'enrichissement de requêtes : QUEXME

Dans ce chapitre, nous présentons la méthode d'enrichissement de requêtes QUEXME (QUery EXpansion MEthod) que nous proposons. Cette méthode se base sur la notion de popularité d'un concept pour rechercher des concepts proches de la requête de l'utilisateur pour son enrichissement.

Sommaire

6.1 Introduction

Dans les systèmes de recherche d'information et notamment sur le Web, les moteurs de recherche s'appuient sur l'indexation des documents à l'aide de mots clés pour permettre leur recherche. La recherche d'information est un processus qui demande un choix précis de mots et que l'utilisateur sache exactement ce qu'il cherche. L'enrichissement de requête consiste à ajouter ou proposer des termes à la requête initiale de l'utilisateur dans le but d'améliorer les résultats de la recherche en les rendant plus pertinents, ces termes peuvent être des synonymes ou des termes associés. Plusieurs travaux se sont intéressés à l'enrichissement de requêtes, les méthodes peuvent être manuelles, automatiques ou assistées basées sur des thésaurus ou des réseaux sémantiques (ex : Wordnet [8]) en prenant en compte le besoin de l'utilisateur.

La méthode proposée QUEXME (QUery EXpansion MEthod) [GCAC09a], [GCAC09b] est basée sur la notion de popularité d'un concept et l'analyse du comportement usuel des utilisateurs, en utilisant l'ontologie de référence de l'architecture de coopération OWSCIS. Dans ce contexte, la recherche d'information se concrétise par la formulation d'une requête émise sur la coopération. La requête est exprimée dans les termes de l'ontologie de référence et les termes additionnels proposés jugés proches sémantiquement et pertinents sont également issus de l'ontologie de référence. L'usage de l'ontologie de référence comme base pour l'interrogation de la coopération permet de mettre en œuvre des méthodes de recherche et d'enrichissement qui s'appuient sur un ensemble d'informations bien définies avec les concepts et les propriétés de l'ontologie. Dans le cadre plus général des moteurs de recherche sur le Web, la recherche de termes synonymes ou liés peut passer par l'utilisation de bases de connaissances externes qui peuvent également être des ontologies.

6.2 La méthode d'enrichissement QUEXME

La méthode QUEXME comporte deux phases qui sont (i) la phase d'apprentissage qui correspond à une phase d'initialisation du processus et (ii) la phase d'enrichissement proprement dite qui correspond au processus complet de recherche de concepts additionnels pour une requête. La méthode est basée sur l'hypothèse qu'un utilisateur qui recherche des informations va effectuer une séquence de requêtes pour obtenir les informations qu'il souhaite. Le processus est dynamique et consiste à proposer à partir d'une requête initiale d'un utilisateur, des concepts proches et jugés pertinents pour enrichir sa requête en complément aux résultats de la requête émise. L'utilisateur peut alors reformuler ou compléter sa requête pour affiner sa recherche. Cette étape se répète pour une séquence de recherche. La figure 6.1 donne un aperçu général de l'approche mise en œuvre dans la méthode QUEXME. La phase d'apprentissage consiste

8. http ://wordnet.princeton.edu/

à analyser un ensemble de séquences de requêtes "modèles" qui servent à initialiser la construction d'un graphe de concepts et les valeurs d'une matrice qui sont utilisées pour le calcul de la pertinence d'un concept relativement à une requête. La phase d'enrichissement comporte également une étape de création d'un graphe de concepts et la mise à jour des valeurs de la matrice des concepts, mais elle comporte une deuxième étape qui concerne la recherche des termes additionnels pour la requête. La recherche des concepts additionnels nécessite plusieurs calculs et notamment (i) le calcul du ConceptRank d'un concept que l'on peut apparenter à la notion de popularité d'un concept, (ii) le calcul de l'importance d'un concept qui exprime sa "pertinence" pour la requête posée et (iii) le calcul des concepts candidats.

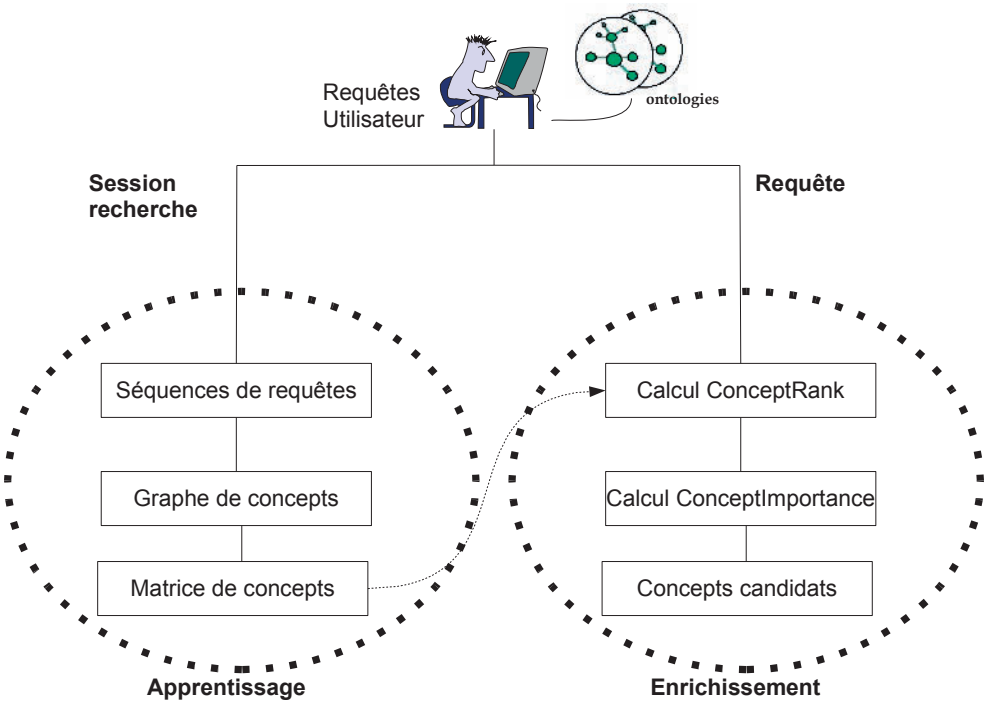

FIGURE 6.1 – Approche générale QUEXME

Les sections 6.2.1 et 6.2.2 donnent le détail des phases d'apprentissage et d'enrichissement mises en œuvre dans la méthode QUEXME. La description de la méthode est illustrée en s'appuyant sur une ontologie qui décrit des conférences académiques. Cette ontologie modélise les entités relatives à une conférence académique, ses acteurs et ses relations. Elle comporte des concepts comme les concepts *person*, *event*, *organization*, *workshop*, *location*, etc. Un extrait de cette ontologie est décrit dans l'annexe A.

6.2.1 Phase d'apprentissage

La méthode se base sur la notion de popularité d'un concept par rapport à une requête i.e. par rapport à l'ensemble des concepts qui la compose. Intuitivement, l'idée de popularité repose sur l'analyse des comportements des utilisateurs lors de sessions de recherche, pour détecter des concepts qui sont usuellement choisis lors de la recherche d'un ou des concepts spécifiques. Pour effectuer ces calculs de popularité, la méthode utilise une matrice qui permet de connaître pour chaque concept de l'ontologie, quels sont les concepts "liés" en se basant sur l'analyse des séquences de recherche des utilisateurs. On s'intéresse aux concepts cherchés "après" un concept, ce qui correspond dans la méthode QUEXME, à la construction d'un graphe de concepts à partir d'une séquence de recherche et à la mise à jour d'une matrice appelée matrice de concepts. Ce processus nécessite une première phase pour initialiser la matrice de concepts, cette phase correspond à la phase d'apprentissage. La phase d'apprentissage consiste à analyser un ensemble de séquences de requêtes, uniquement pour construire les graphes de concepts associés et mettre à jour au fur et à mesure la matrice des concepts. On peut considérer ces requêtes comme des requêtes modèles de la phase d'apprentissage. Dans le cadre de la coopération OWSCIS, l'ontologie qui est considérée est l'ontologie de référence.

Graphe de concepts et construction de la matrice

Un graphe de concepts est construit à partir d'une séquence de requêtes d'un utilisateur lors d'une session de recherche. Les nœuds représentent les concepts et les arcs expriment les relations entre les concepts. Un arc d'un nœud N_1 vers un nœud N_2 correspond à la recherche dans une requête du concept N_2 après la recherche d'une requête avec le concept N_1 par un utilisateur. Ce graphe est orienté et pondéré.

Dans la méthode QUEXME, une requête est un ensemble de concepts. Dans le cadre de la coopération en OWSCIS, ces concepts sont extraits de la requête SPARQL, émise sur la coopération.

Dans un premier temps, nous définissons les notions de séquence et sous-séquence de requêtes.

Définition 6.1 *Soit C l'ensemble de concepts du domaine, Q est une requête qui est un sous ensemble de C.*

- Une séquence de requêtes est un n-uplet $S(Q_1, Q_2, ..., Q_n)$ tel que Q_i est une requête. Une sous-séquence de la séquence $S(Q_1, Q_2, ..., Q_n)$ est un couple $Si(Q_i, Q_{i+1})$ tel que $i = 1...n-1$.
- La cardinalité $|Q|$ de la requête Q, représente le nombre de concepts appartenant à la requête.

Définition 6.2 *Un graphe de concepts $G_i(O_i, V_i)$ est un graphe orienté et pondéré pour une sous-séquence $S_i(Q_i, Q_{i+1})$ de la séquence $S(Q_1, ..., Q_n)$ tel que :*

- V_i est l'ensemble des arcs appartenant à $(Q_i \cap Q_{i+1}) \times (Q_{i+1} - Q_i)$.
- O_i est l'ensemble des concepts appartenant à $Q_i \cup Q_{i+1}$,
- Le poids $w_i(v_i)$ associé à l'arc $v_i \in V_i$ est défini par :

$$w_i(v_i) = \frac{1}{|Q_i \cap Q_{i+1}|} \tag{6.1}$$

Ce poids est basé sur le nombre de concepts communs que l'utilisateur soumet dans les deux requêtes d'une sous-séquence $S_i(Q_i, Q_{i+1})$.

exemple Soient les deux requêtes consécutives (sous-séquence) suivantes :
- Q_1 : person, organization, presenter
- Q_2 : person, presenter, workshop, socialevent

Nous déterminons l'ensemble de concepts du graphe et nous calculons les arcs et leurs poids. Les arcs sont construits à partir des concepts de l'intersection des requêtes Q_1 et Q_2, vers les concepts de la différence entre les deux ensembles des concepts des requêtes. Leur poids est calculé à partir de la cardinalité de l'intersection des requêtes Q_i et Q_{i+1}.
- $V_i = (\{\text{person, organization, presenter}\} \cap \{\text{person, presenter,workshop,socialevent}\}) \times (\{\text{person, presenter,workshop,socialevent}\} - \{\text{person, organization, presenter}\})$.
  $V_i = (\{\text{person, presenter}\} \times (\{\text{workshop, socialevent}\})$.
- $w_i(v_i) = \frac{1}{|\{person,presenter\}|} = \frac{1}{2}$

Nous obtenons le graphe illustré par la figure 6.2.

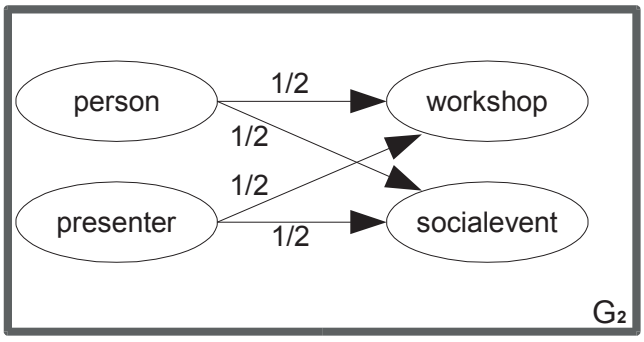

FIGURE 6.2 – Graphe de la sous-séquence

L'algorithme 1, décrit la construction d'un graphe pour une sous-séquence de requête.

Définition 6.3 *Un graphe G d'une séquence $S(Q_1, .., Q_n)$ est l'union des graphes G_i des sous-séquences $S_i(Q_i, Q_{i+1})$.*

$$G = \bigcup_{i=1}^{n-1} G_i \tag{6.2}$$

Algorithme 1 : Construction d'un graphe de sous-séquence

Données :
$S_i \longleftarrow (Q_i, Q_{i+1})$ une sous-séquence de requêtes
$C \longleftarrow \{c_1, c_2, ..., c_n\}$ ensemble des concepts $c_i \in (Q_x \cap Q_y)$
$D \longleftarrow \{d_1, d_2, ..., d_n\}$ ensemble des concepts $d_i \in (Q_{i+1} - Q_i)$
G_{S_i} graphe de la sous-séquence de requêtes (Q_i, Q_{i+1}) contenant les concepts de $C \cup D$
début
  répéter
    sélectionner c_i de C
    supprimer c_i de C
    // Construction de l'arc pondéré
    répéter
      sélectionner d_i de D
      supprimer d_i de D
      $v_i = \frac{1}{|(Q_i \cap Q_{i+1}|}$
      mettre un arc entre c_i et d_i avec un poids v_i
    jusqu'à $(D = \emptyset)$;
  jusqu'à $(C = \emptyset)$;
  Retourner G_{S_i}
fin

Le graphe de concepts d'une séquence de requêtes est représenté par une matrice. Elle correspond à une matrice carrée contenant pour chaque couple de concepts (C_1, C_2), la somme des poids des arcs reliant les deux concepts dans le graphe de concepts.

Nous définissons la notion de matrice d'une sous-séquence, et la notion de matrice pour toutes les séquences.

Définition 6.4 *Un graphe G_i d'une sous-séquence $S_i(Q_i, Q_{i+1})$ peut être représenté par la matrice MC_i tel que :*

$$MC_i(c_i, c_j) = \begin{cases} w_i(c_i, c_j) s'il\ existe\ un\ arc\ v_i : c_i \rightarrow c_j \\ 0 \qquad \text{sinon} \end{cases} \tag{6.3}$$

Définition 6.5 *Une matrice MC d'un graphe G pour une séquence $S(Q_1, Q_2 ...., Q_n)$ est la somme des matrices MC_i des sous-séquences $S_i(Q_i, Q_{i+1})$.*

$$MC = \sum_{i=1}^{n-1} MC_i \tag{6.4}$$

La table 6.1 donne un extrait des sous-séquences utilisées dans la première phase d'apprentissage de notre exemple. Elle décrit pour chaque sous-séquence, les deux requêtes considérées (Requête 1 et Requête 2), les arcs qui

TABLE 6.1 – Arcs et poids du graphe

Requête 1	**Requête 2**	**Arcs v_{12}**	**Poids w_{12}**
presentation , event	paper, presentation	presentation →paper	1
conference, socialevent, event	location, conference, socialevent	conference → location,socialevent →location	1/2
committee, organization, panel	person, committee, organization	committee → person,organization → person	1/2
event, attendee, presenter, keynotespeaker	person, presenter, attendee, keynotespeaker	presenter → person, attendee → person, keynotespeaker → person	1/3
workshop, socialevent, conference, presentation	organization, workshop, socialevent, conference, presentation,	workshop → organization, socialevent → organization, conference→organization, presentation→organization	1/4
chair, group	chair, organization, panel, committee	chair → organization, chair → panel, chair → committee	1

sont construits pour le graphe de concepts associé (arcs) et leur pondération (poids). Les arcs pondérés sont utilisés afin de mettre à jour la matrice des concepts en l'incrémentant. Cette matrice servira à déterminer les concepts candidats pendant la phase l'enrichissement.

Les figures 6.3 et 6.4 illustrent les graphes de concepts construits pour les deux premières sous-séquences de la table 6.1.

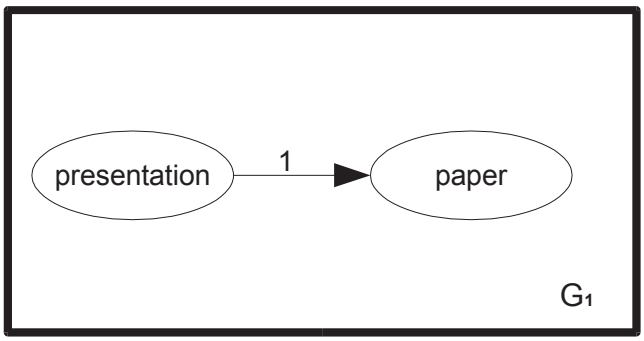

FIGURE 6.3 – Graphe de la 1ère sous-séquence de la phase d'apprentissage

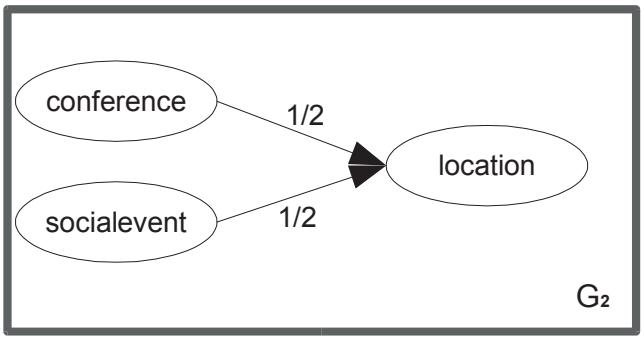

FIGURE 6.4 – Graphe de la 2ème sous-séquence de la phase d'apprentissage

TABLE 6.2 – Extrait de la matrice de concepts

Concepts	**paper**	**location**	**committee**	**organization**
workshop	0	0	0	0.25
socialevent	0	0.5	0	0.25
conference	0	0.5	0	0.25
presentation	1	0	0	0.25
chair	0	0	1	1

A partir des graphes de concepts construits pour les sous-séquences de requêtes, la matrice des concepts est calculée. La table 6.2 illustre la matrice obtenue à partir des sous-séquences de recherche présentées dans la table 6.1. Au fur et à mesure de la phase d'apprentissage, les valeurs de cette matrice sont incrémentées en utilisant les valeurs des arcs des graphes construits. Seuls les concepts des arcs apparaissant dans les graphes sont présentés dans la table 6.2. La matrice complète est une matrice carrée sur l'ensemble des concepts de l'ontologie. Cette matrice sert de base pour les calculs de popularité dans la phase d'enrichissement. Dans cet exemple, la phase d'apprentissage qui a été réalisée comportait 50 sous-séquences de requêtes.

6.2.2 Phase d'enrichissement

Lorsque la phase d'apprentissage a été effectuée, la phase d'enrichissement proprement dite, peut être mise en œuvre. Elle constitue le processus d'enrichissement proposé dans la méthode QUEXME. Cette phase comporte toutes les étapes décrites précédemment (construction des graphes et de la matrice), mais elle comporte également les phases de calcul et d'enrichissement de la requête par des termes additionnels. La figure 6.5 illustre un exemple de deux sessions de recherche (séquence de requêtes) faites par un utilisateur. La première séquence comporte 3 requêtes successives Q_1, Q_2 et Q_3 et la deuxième séquence comporte deux requêtes Q_4 et Q_5. Cet exemple est utilisé pour illustrer la phase d'enrichissement décrite dans cette section.

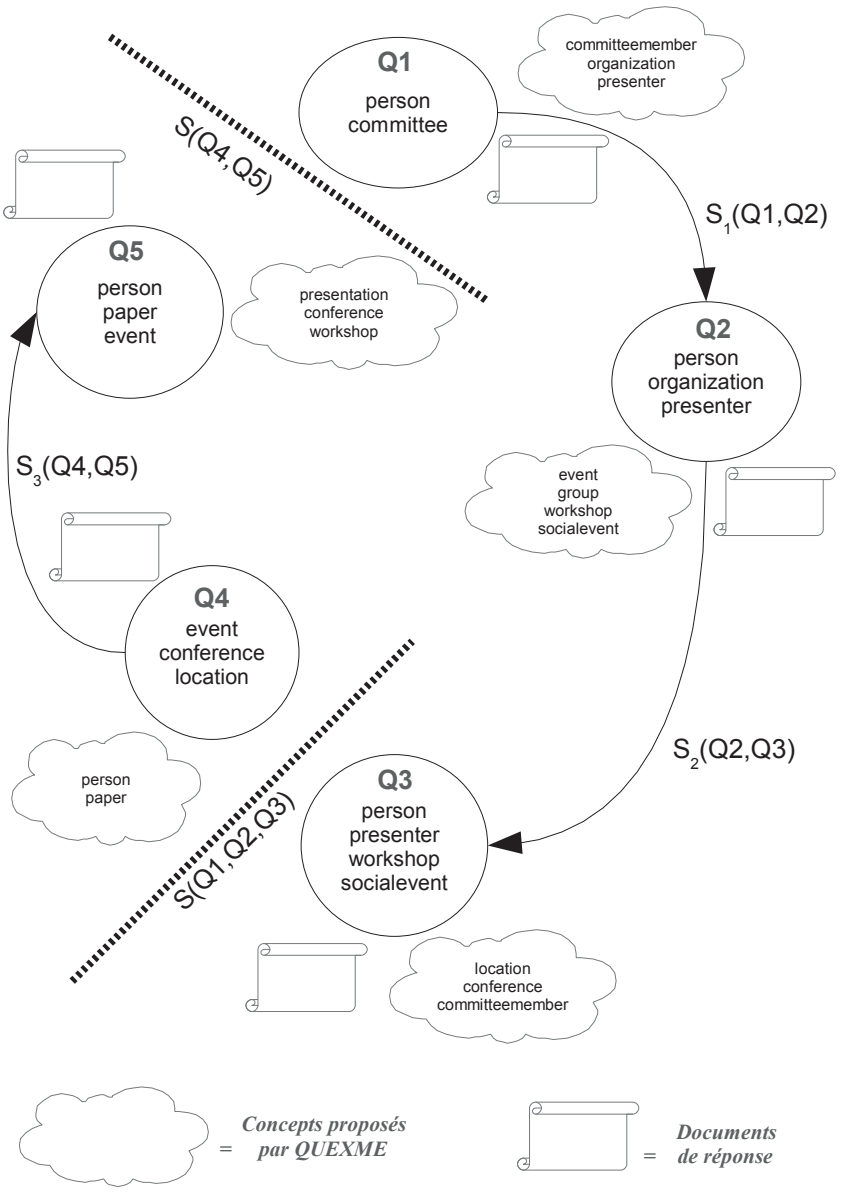

FIGURE 6.5 – Exemple de deux séquences de requêtes

Pour chaque sous-séquence d'une séquence de recherche, on construit de la même manière que dans la phase d'apprentissage, le graphe de concepts et on met à jour la matrice de concepts. Cette dernière est incrémentée avec les poids arcs du graphe de la nouvelle sous-séquence de requêtes. Le graphe illustré par la figure 6.6 représente la première sous-séquence de requête $S_1(Q_1, Q_2)$ de la figure 6.5.

La matrice des concepts est incrémentée à partir de valeurs des arcs pondérés des graphes construits. La table 6.3 montre les valeurs de la matrice des concepts après la phase d'apprentissage et les séquences de recherche $S(Q_1, Q_2, Q_3)$ et $S(Q_4, Q_5)$. Les valeurs de la matrice sont calculées à partir de

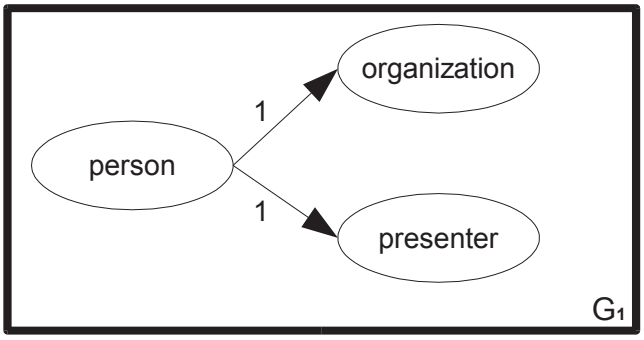

FIGURE 6.6 – Graphe de la sous-séquence S_1

ses valeurs courantes et les valeurs des poids des arcs des graphes de concepts. Dans la table 6.3 :

- w_G est le poids des arcs des graphes des sous-séquences construits à partir des requêtes de l'utilisateur,
- w_{M_i} est la valeur des poids de la matrice de concepts (phase d'apprentissage),
- $w_{M_{i+1}}$ est la valeur des poids de la matrice après la prise en compte des requêtes de l'utilisateur.

TABLE 6.3 – Valeurs de la matrice de concepts après les séquences de recherche $S(Q_1, Q_2, Q_3)$ et $S(Q_4, Q_5)$

.

Concepts $C_i \rightarrow C_j$	w_G	w_{M_i}	$w_{M_{i+1}}$
person→organization	1	2.5	3.5
person→presenter	1	1	2
person→workshop	0.5	0	0.5
person→socialevent	0.5	0.5	1
presenter→workshop	0.5	0	0.5
person→socialevent	0.5	0.5	1
event→person	1	0	1
event→paper	1	0.3	1.3

La phase d'enrichissement de la requête comporte en plus des étapes de calcul des graphes et de la matrice, les étapes de recherche de concepts candidats pour l'enrichissement de la requête. Ces étapes comportent le calcul d'une mesure de popularité d'un concept appelé *ConceptRank*, puis le calcul de l'importance d'un concept basée sur sa popularité et enfin la recherche de concepts candidats pour l'enrichissement et enfin la sélection de concepts à partir de seuils choisis.

Le calcul de la popularité d'un concept proposée en QUEXME est inspiré de l'idée implémentée dans l'algorithme du *PageRank* par le moteur de recherche

du Web Google. L'algorithme du *PageRank* s'appuie sur un graphe hypertexte qui représente les liens hypertextes entre des ressources. La popularité d'une page sur le Web va donc être liée au nombre de fois que cette page est accédée à partir d'autres pages. La notion de popularité proposée en QUEXME s'appuie sur un graphe de concepts avec des poids qui indiquent si un concept a été fréquemment recherché à partir d'un autre. Elle est liée au comportement usuel des utilisateurs lors de sessions de recherche. Nous rappelons brièvement la méthode de calcul du *PageRank* utilisée dans les moteurs de recherche comme Google, puis nous donnons le détail des calculs et mesures proposées dans QUEXME.

6.2.2.1 L'algorithme PageRank

Le PageRank est l'algorithme d'analyse des liens concourant au système de classement des pages Web utilisé par le moteur de recherche Google, pour déterminer l'ordre dans les résultats de recherche qu'il fournit [BP98]. Actuellement, d'autres critères sont également pris en compte dans le classement proposé par Google, le PageRank n'est qu'un critère parmi d'autres. Le principe du *PageRank* est donné par la définition suivante.

Définition 6.6 *Soit x une page Web. F_x l'ensemble de liens sortant de x, B_x l'ensemble des pages qui pointent sur x,*

- *$N_x = |Fx|$ le nombre de liens sortant de x,*
- *d est le facteur de normalisation. Le PageRank de la page x est défini par :*

$$PR(x) = (1-d) + d \sum_{y_i \in B_x} \left(\frac{PR(y_i)}{N(y_i)}\right) \tag{6.5}$$

Plus la somme des *PageRank* des pages qui pointent vers une page est grande plus le *PageRank* de la page est grand.
La méthode développée dans QUEXME consiste à considérer le graphe des concepts des séquences de requêtes, les arcs des graphes représentant les liens entre les concepts. Le *ConceptRank* peut s'apparenter à la notion de popularité d'un concept.

6.2.2.2 Calcul du ConceptRank d'un concept

Définition 6.7 *Soit c_i un concept de C, $B(c_i)$ l'ensemble de concepts tels que $MC(c_j, c_i) \neq 0$, d est le facteur de normalisation.*

Le *ConceptRank* du concept c_i, $CR(c_i)$ est défini par :

$$CR(c_i) = (1-d) + d \sum_{c_j \in B_{c_i}} \left(\frac{CR(c_j)}{N(c_j)}\right) \tag{6.6}$$

où $N(c_j)$ esr le nombre d'arcs du graphe G, partant de c_j vers un autre concept.

$$N(c_j) = \sum_{k=0}^{n} v_k \begin{cases} v_k = 1, & \text{if } MC(c_j, c_k) \neq 0 \\ v_k = 0, & \text{else} \end{cases} \quad (6.7)$$

Le calcul du *ConceptRank* (CR) nécessite l'itération du processus de calcul. Les valeurs des CR sont initialisées à 1.

Exemple Calcul du *ConceptRank* du concept *event* . A la première itération, la valeur des CR est égale à 1 puis le calcul est itéré plusieurs fois. Le calcul de la 1ère itération est la suivante. $CR(Event) = (1-0.85)+0.85((CR_{person}/N_{person})+(CR_{paper}/N_{paper})+(CR_{conference}/N_{conference})+(CR_{presentation}/N_{presentation})+(CR_{committeemember}/N_{committeemember})+(CR_{committee}/N_{committee})+(CR_{organization}/N_{organization})+(CR_{keynotetalk}/N_{keyntetalk})+(CR_{presenter}/N_{presenter})+(CR_{keynotespeaker}/N_{keynotespeaker}))$
$CR(Event) = (1-0.85)+0.85((1/12)+(1/2)+(1/5)+(1/5)+(1/5)+(1/5)+(1/4)+(1/3)+(1/5)+(1/5))$
$CR(Event) = (0.15)+0.85(2.363)$
$CR(Event) = 2.15855$
La valeur de $CR(Event)$ après 50 itérations est 4.656402.

Nous complétons la notion de popularité par celle d'importance d'un concept par rapport à un autre et par rapport à une requête.

6.2.2.3 Calcul de l'importance d'un concept

La notion d'importance d'un concept par rapport à un autre concept ou plus précisément l'importance d'un concept par rapport à une requête représente la mesure utilisée pour choisir les concepts candidats dans le processus d'enrichissement.

Définition 6.8 *Soit MC la matrice représentant le graphe G, et Q la requête. $CI(c_i, c_j)$ représente le ConceptImportance du concept c_i par rapport au concept c_j et est défini par :*

$$CI(c_i, c_j) = MC(c_i, c_j) \times CR(c_i) \quad (6.8)$$

où $MC(c_i, c_j)$ est la valeur de la matrice MC pour le couple (C_i, C_j) et $CR(c_i)$ est le ConceptRank de c_i.

Par exemple, le calcul de l'importance du concept *event* par rapport au concept *paper* est :
$CI(event, paper) = MC_{event,paper} x CR(event)$
$CI(event, paper) = 0.3 \times 4.656402$
$CI(event, paper) = 1.3969206$

Définition 6.9 *$I(c_i, Q_i)$ représente l'importance du concept c_i dans la requête Q_i et est défini par :*

$$I(c_i, Q_i) = \sum_{c_j \in Q_i} CI(c_i, c_j) \tag{6.9}$$

où $CI(c_i, c_j)$ représente l'importance du concept c_i par rapport au concept c_j défini plus haut. La table 6.4 montre les résultats des calculs de l'importance des concepts considérés par rapport à la requête person, commitee.

TABLE 6.4 – Résultats des facteurs d'importance

(Concept,Requête)	$I(c,Q)$
(person ,{person,committee})	11.961165
(chair , {person,committee})	4.894458
(attendee , {person,committee})	2.398824
(committeemember , {person,committee})	7.528373
(organization , {person,committee})	9.595937
(presenter , {person,committee})	7.6645746
(keynoteSpeaker , {person,committee})	3.9235027

6.2.2.4 Facteur Minimum d'importance

Afin de choisir les concepts pour l'enrichissement de la requête, nous considérons leur importance par rapport à la requête. Nous définissons un seuil minimum d'importance *minimum factor importance*. La requête Q_i est enrichie par le concept c_i si son *ConceptImportance* $I(c_i, Q_i)$ est supérieur au seuil d'importance.

Définition 6.10 *Soit I_{Q_i} l'ensemble des valeurs ConceptImportance dans la requête Q_i, $I_{Q_i} = \{I(c_1, Q_i), ..., I(c_n, Q_i)$. Le facteur d'importance minimum $Min(FI/Q_i)$ pour une requête Q_i est défini par :*

$$Min(FI/Q_i) = \frac{Max(I_{Q_i}) + Min(I_{Q_i})}{2} \tag{6.10}$$

Ce facteur représente la moyenne de valeurs minimales et maximales d'importance. En appliquant ce facteur sur notre exemple, on obtient le résultat suivant :

$Min(FI/Q_i) = \frac{(11.961165+2.398824)}{2}$

$Min(FI/Q_i) = 7.179994$

Les concepts retenus sont : *person, organization, presenter, committeemember*

6.3 Conclusion

Ce chapitre a décrit notre système d'enrichissement de requêtes qui a pour objectif d'aider l'utilisateur dans son processus de recherche d'informations. L'approche proposée est basée sur la notion de popularité d'un concept en analysant le comportement usuel des utilisateurs afin de sélectionner de nouveaux concepts à ajouter dans la requête. Nous avons introduit les définitions des notions utilisées dans l'approche : séquences et sous-séquences de requêtes, graphe et matrice de concepts, popularité d'un concept (*ConceptRank*), importance d'un concept par rapport à un autre concept et par rapport à une requête. La méthode repose sur une phase d'apprentissage pour initialiser le graphe et la matrice de concepts utilisés dans la recherche des concepts additionnels pour enrichir une requête. La phase d'enrichissement calcule le *ConceptRank* des concepts pour calculer leur importance par rapport à la requête à enrichir. Puis l'application d'un seuil permet de sélectionner un ensemble de concepts dans l'ensemble des concepts candidats. La méthode a été illustrée par un exemple. Un exemple plus complet pour valider la méthode est proposé au chapitre 8.

7

SErvice de VIsualisation SEVI

Dans ce chapitre, nous présentons le service de visualisation *SEVI*, proposé dans l'environnement de coopération OWSCIS. Ce service propose des fonctionnalités pour (i) aider l'utilisateur dans la construction de sa requête, (ii) visualiser l'ontologie de référence, (iii) visualiser les requêtes, leur enrichissement et leurs résultats. Il s'appuie sur la connaissance de l'ontologie de référence pour offrir à l'utilisateur un environnement convivial et dynamique de visualisation lors de l'interrogation de la coopération. Il propose des environnements de visualisation 2D et 3D personnalisables par l'utilisateur.

Sommaire

7.1 Introduction

La visualisation des informations est un point essentiel pour faciliter le ”repérage” et de ce fait l'exploitation des informations. Dans la recherche d'information, elle permet de donner une vue globale en rendant l'information recherchée plus facile à repérer et de ce fait plus facile à utiliser. Notre approche s'inscrit dans le cadre général de l'architecture de coopération de systèmes d'information OWSCIS que nous avons proposé, basée sur l'utilisation d'une ontologie de référence pour l'interrogation transparente de la coopération. Le service de visualisation s'appuie sur l'utilisation de l'ontologie de référence, ses concepts et ses relations sémantiques pour offrir un ensemble de fonctionnalités pour (i) visualiser l'ontologie, (ii) construire une requête, (iii) visualiser la requête, son enrichissement et ses résultats dans un environnement dynamique et convivial. Il offre des outils pour l'ensemble du processus de recherche dans la coopération en permettant la visualisation de tous les éléments nécessaires (ontologie, requêtes, enrichissement et résultats).

Une analyse et classification des différents systèmes de visualisation (chapitre 4) nous a permis de dégager des critères pour la mise en œuvre de ce service de visualisation. Les travaux étudiés se limitent généralement à la visualisation d'ontologies et ne s'intéressent pas au processus de recherche d'information. Dans le service *SEVI* (SErvice de VIsualisation) que nous proposons, nous présentons une approche de visualisation qui prend en compte les différents éléments du processus de recherche de l'utilisateur (de l'interrogation à l'affichage des résultats), nous soulignons l'importance de la séparation entre le ”quoi visualiser” du ”comment visualiser”.

Les éléments à visualiser dans le processus de la requête de la coopération sont : (i) l'ontologie de référence, (ii) la requête de l'utilisateur, (iii) les concepts proposés pour l'enrichissement, (iv) la requête enrichie et (v) les résultats de la recherche. Les techniques de représentation visuelles choisies répondent à la question ”comment ?”.

La suite de ce chapitre est organisée comme suit : La section 7.2 décrit l'architecture générale du service de visualisation, la section 7.3 présente le module de visualisation de l'ontologie avec ses différentes fonctionnalités. Les étapes de visualisation du processus de construction d'une requête sont présentées dans la section 7.4. La section 7.5 présente le module de visualisation de requêtes avec ses trois fonctionnalités de visualisation : requête, enrichissement et résultats, dans les deux environnements 2D et 3D. Enfin, nous concluons dans la section 7.6

7.2 Service de visualisation : SEVI

Comme nous l'avons mentionné dans le chapitre 5 , *SEVI* est un service de l'architecture générale d'OWSCIS. La figure 7.1 détaille les différents modules de ce service, ainsi que son interaction avec les autres services d'OWSCIS. Les

différentes étapes de traitement d'une requête sont également présentées.

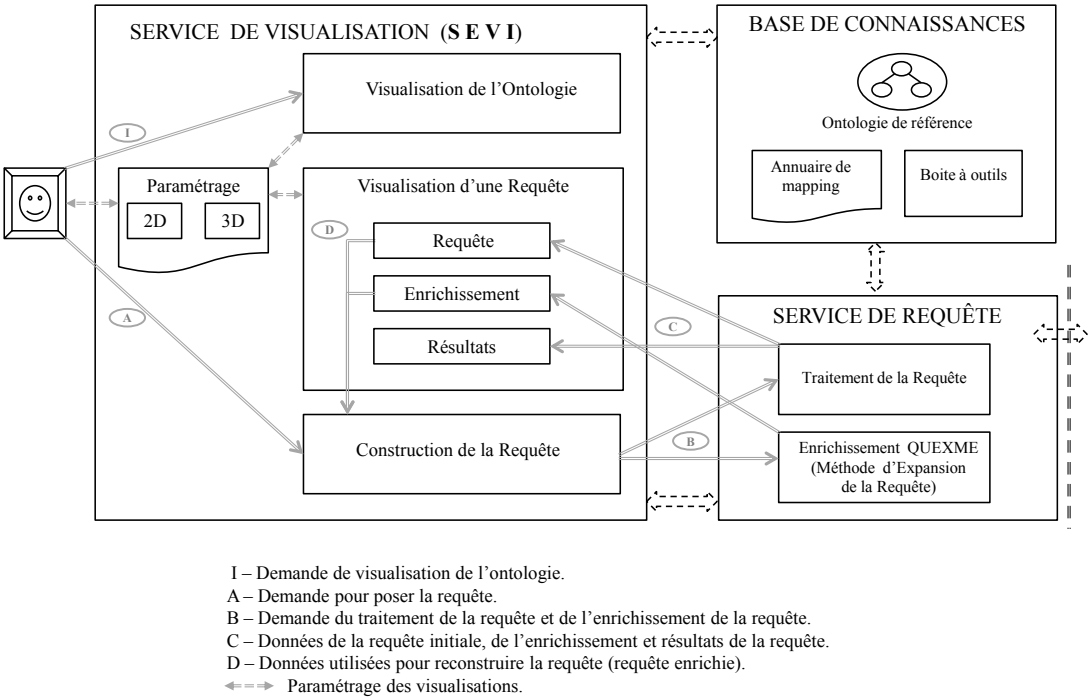

FIGURE 7.1 – Approche générale SEVI

Dans l'architecture générale, deux éléments de la coopération OWSCIS sont directement liés au service de visualisation :

- **La base de connaissances** qui contient l'ontologie de référence. L'ontologie est utilisée dans le processus de construction et d'enrichissement d'une requête et peut être affichée grâce au service de visualisation.
- **Le service de requête** qui a pour fonction le traitement de la requête par la coopération et son enrichissement en appliquant la méthode QUEXME présentée dans le chapitre 6. Ce service envoie au service de visualisation la requête, les concepts proposés pour l'enrichissement et les résultats afin d'avoir une représentation pertinente qui permette à l'utilisateur d'exploiter au mieux les résultats.

Le service de visualisation *SEVI* est composé de quatre modules de visualisation dont un environnement de paramétrage :

1. **Module visualisation ontologie.** Il permet à l'utilisateur de visualiser l'ontologie de référence utilisée par le système.
2. **Module visualisation requête.** Il est composé de trois sous-modules :
   - Le sous-module *Requête* qui permet de visualiser les concepts et les relations de la requête en se basant sur l'ontologie de référence.
   - Le sous-module *Enrichissement* qui permet de visualiser les concepts proposés par le service d'interrogation "QUEXME", afin de permettre

à l'utilisateur d'avoir une vue générale des concepts proposés dans l'ontologie.
- Le sous-module *Résultats* qui permet de visualiser les résultats de la requête.

3. **Module construction requête.** Il permet d'assister l'utilisateur dans le processus de recherche lors de la soumission de la requête. Il propose des outils pour construire dynamiquement une requête en s'appuyant sur l'ontologie de référence.
4. **Module paramétrage.** Il offre des outils à l'utilisateur pour le paramétrage et l'organisation de la visualisation.

Dans l'architecture générale, le service *SEVI* a pour rôle l'interaction avec l'utilisateur dans le processus de recherche. Les différents modules sont présentés dans la suite de ce chapitre.

7.3 Visualisation de l'ontologie

Ce module permet la visualisation de l'ontologie de référence utilisée par le système. C'est un module d'interaction directe avec l'utilisateur, il permet le paramétrage des sorties en 2D ou 3D selon le choix, il permet également de changer les paramètres en fonction de la recherche. Le fonctionnement du module de visualisation de l'ontologie est illustré par la figure 7.1. La fonctionnalité de visualisation de l'ontologie est identifiée sur la figure par la flèche annotée *[I - Demande de visualisation de l'ontologie]*.

Nous présentons dans la suite les deux environnements de visualisation (2D et 3D) proposés pour la visualisation de l'ontologie. L'ensemble des visualisations présentées dans ce chapitre est issu des prototypes (2D et 3D) de visualisation implémentés dans le projet OWSCIS. Ces prototypes ont été en partie élaborés avec la collaboration d'étudiants de Master STIC BD-IA (Bases de Données et Intelligence Artificielle) de l'Université de Bourgogne [LJ08] et d'un étudiant de l'Institut technologique de Nuevo León au Mexique [dJLL09] dans le cadre du projet tutoré de leur formation.

7.3.1 L'ontologie de référence

Afin d'illustrer la visualisation proposée, nous allons utiliser l'ontologie *Conférence* présentée au chapitre 6. Cette ontologie modélise les entités relatives à une conférence académique, ses acteurs et ses relations. Elle comporte des concepts comme les concepts *person*, *event*, *organization*, *workshop*, *location*, etc. Un extrait de cette ontologie est décrit dans l'annexe A. La figure 7.2 illustre une visualisation de cette ontologie avec les outils proposés (visualisation en 2D et une organisation des composants en *StaticLayout*).

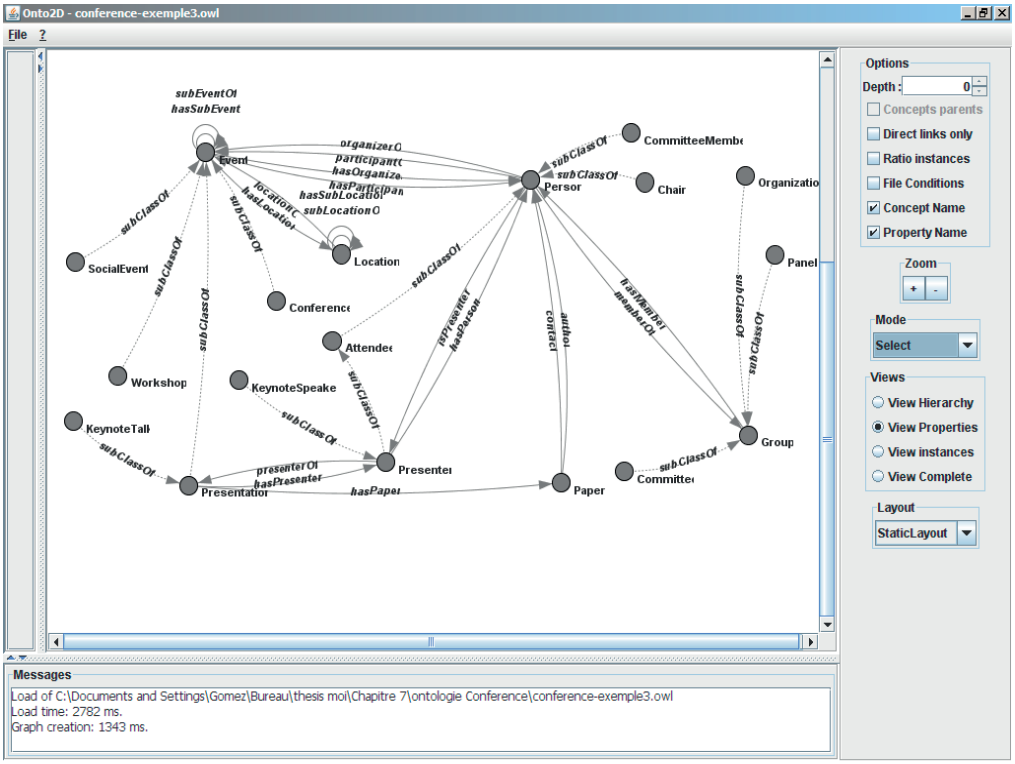

FIGURE 7.2 – Visualisation de l'ontologie

7.3.2 Environnement 2D

L'analyse et la classification des systèmes de visualisation étudiés, dans le chapitre 4, nous ont conduit à retenir un certain nombre de critères jugés pertinents dans le cadre des outils de visualisation que nous proposons. Nous donnons les caractéristiques de notre outil selon ces critères. Le module de visualisation 2D offre comme son nom l'indique différentes possibilités de visualisation en 2D. Notre outil de visualisation présente donc les caractéristiques suivantes :

- *La dynamique de construction de la visualisation.* Elle est de type *dynamique* car le système utilise l'ensemble des paramètres et des règles pour la construction de la visualisation.
- *Le type d'adaptativité.* Elle est *automatique* car elle est basée sur l'analyse de l'évolution de l'environnement.
- *La personnalisation.* Elle est également *automatique* car aucune validation (par un administrateur) n'est nécessaire pour la personnalisation de l'environnement et l'utilisateur peut voir directement les modifications effectuées.
- *Le mode d'interaction.* La manipulation directe, l'utilisateur peut agir directement sur les objets visualisés, comme afficher ou masquer des concepts de l'ontologie.
- La visualisation en 2D permet également la visualisation textuelle, comme

voir le nom des éléments à côté de chaque représentation.

Les choix retenus ont été faits pour apporter la plus grande convivialité et le plus grand dynamisme possible dans le paramétrage de la visualisation pour un utilisateur et de façon automatique par le système si nécessaire.

7.3.2.1 Les stratégies d'organisation de l'information

Ces stratégies permettent la visualisation des informations décrites dans l'ontologie sous différentes formes : soit sous forme d'arbres (structures hiérarchiques), soit sous forme de graphes. Ces stratégies d'organisation sont les suivantes :

- *FRLayout*, les nœuds sont placés en fonction d'un algorithme itératif qui suppose que les nœuds se repoussent lorsqu'ils sont proches, mais sont attirés par les nœuds reliés.
- *CircleLayout* réorganise les nœuds en cercle.
- *ISOMLayout* implémente un algorithme d'auto-organisation basé la méthode graphique de Meyer [Mey98].
- *KKLayout* implémente l'algorithme de Kamada-Kawai [KK88] pour les nœuds.
- *SpringLayout* est un gestionnaire de placement qui s'appuie sur les bords des composants graphiques afin de les placer dans la fenêtre, il définit des relations entre les bords des composants.
- *StaticLayout* est une disposition qui place les nœuds en fonction de leur GDS (Graphic Data System) .

La figure 7.3 illustre la visualisation réalisée avec notre environnement 2D avec la stratégie d'organisation de *CircleLayout*, où les éléments sont organisés sous forme d'un cercle. La figure 7.4 illustre la visualisation de l'ontologie avec la stratégie *ISOMLayout (Inverted Self-Organizing Map)*.

7.3.2.2 La représentation graphique

La représentation graphique ou le rendu visuel concerne la façon dont les éléments sont visualisés. Dans notre environnement (prototype actuel), le choix du rendu visuel pour les éléments de base (concepts, propriétés et instances) n'est pas directement paramétrable, seule l'organisation générale des éléments peut être personnalisée.

Nous utilisons pour les différents éléments de l'ontologie, la représentation commune aux principaux outils de visualisation d'ontologies :

- Les concepts sont représentés par des nœuds sous forme de petits cercles gris.
- Les propriétés sont représentées des flèches ou arcs orientés gris.
- Les instances sont représentées à l'aide de petits carrés gris.

Les noms de chacun des éléments apparaissent sous forme textuelle à côté de la représentation graphique. Tous ces éléments peuvent également être représentés sous forme de tableaux. Ces représentations graphiques sont illustrées

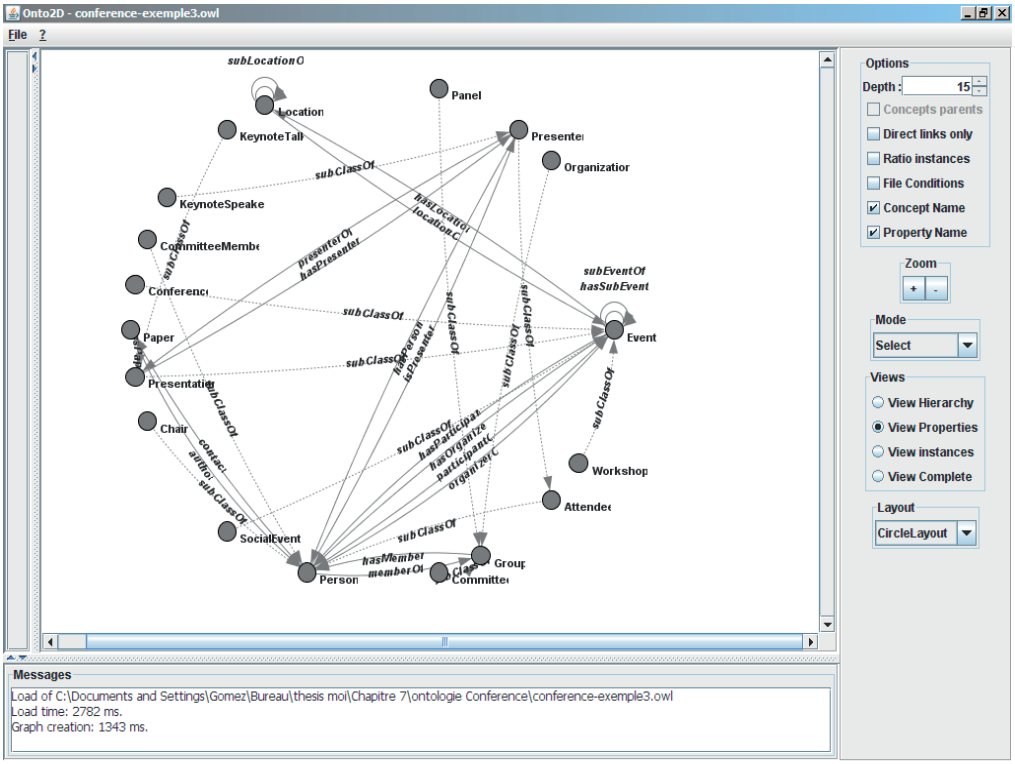

FIGURE 7.3 – Vue circleLayout

par la figure 7.3 avec une stratégie d'organisation de type *circleLayout* et la figure 7.4 avec une organisation de type *ISOMLayout*.

De plus, l'utilisateur a la possibilité de choisir les éléments qu'il souhaite visualiser (partie droite dans la fenêtre de visualisation). Ces options de paramétrage sont présentées dans la section suivante.

7.3.2.3 Le paramétrage de l'environnement de visualisation

La paramétrage permet à l'utilisateur de sélectionner et personnaliser les éléments à visualiser. Nous avons choisi de regrouper le paramétrage concernant l'affichage en quatre parties : options, zoom, mode d'utilisation et vues.

1. *Options* : elles permettent de gérer la visibilité ou le masquage de certains éléments, éventuellement selon leur niveau. L'utilisateur peut ainsi personnaliser plusieurs caractéristiques comme :
   - La profondeur du graphe : en sélectionnant le niveau qu'il souhaite. Certains éléments de la hiérarchie sont masqués pour permettre de mieux visualiser les informations d'un niveau donné.
   - La taille de la représentation des instances (*Ratio instances*) : la taille de la représentation des instances peut être adaptée en fonction du nombre d'instances dans un concept.
   - L'affichage des conditions : les restrictions définies sur un concept peuvent être visualisées.

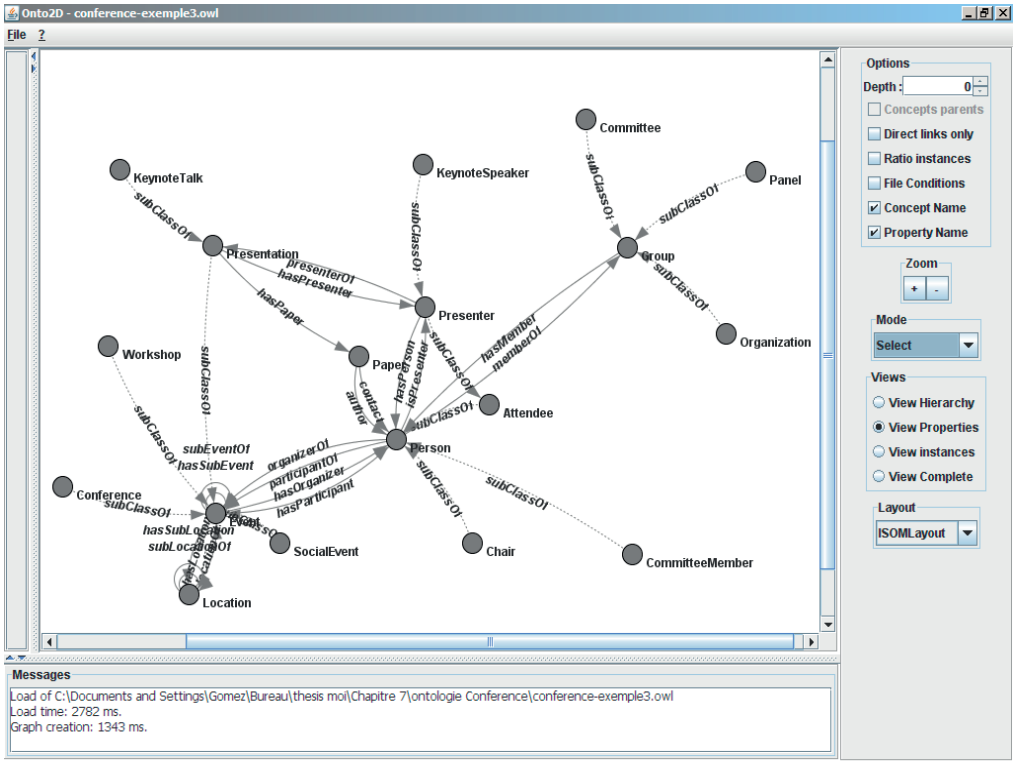

FIGURE 7.4 – Vue ISOMLayout

– L'affichage de libellé qui permet d'afficher ou de masquer le nom des concepts, des instances et des propriétés à côté de sa représentation.

2. *Zoom* : permet d'agrandir et réduire la taille des éléments.
3. *Mode d'utilisation* : l'utilisateur a la possibilité de choisir soit le mode *sélectionner*, il permet alors de sélectionner les éléments affichés, ce qui mettra en valeur l'élément sélectionné pour une meilleure exploitation des informations. Il peut également choisir le mode *Déplacer* qui lui permettra de réorganiser dynamiquement les éléments.
4. *Vues* : cette option permet de choisir les éléments à afficher dans la vue. Il est possible de sélectionner une vue hiérarchique des concepts, la visualisation ou non des propriétés et des instances ou de sélectionner la vue de l'ensemble de ces éléments (*Vue complète*).

Ces paramétrages sont disponibles dans la fenêtre de visualisation comme le montre la figure 7.4 par exemple.

7.3.3 Environnement 3D

La visualisation 3D permet une meilleure représentation des éléments dans l'espace. A l'heure actuelle, le prototype développé ne permet pas l'affichage d'une grande quantité d'informations en même temps. L'objectif retenu est davantage d'offrir une représentation plus conviviale des informations comme la

hiérarchie des concepts à partir d'un concept donné, ou toutes les propriétés et instances liées à un concept donné. Une manipulation directe (déplacements, rotations) est proposée sur les éléments visualisés en 3D. Les différents éléments nécessaires dans le processus de recherche d'information, de l'ontologie de référence aux résultats de la requête, peuvent être visualisés en trois dimensions. Les caractéristiques de l'outil 3D sont similaires à celles de l'environnement 2D, on peut cependant noter les ajouts suivants :

- Les éléments à afficher peuvent être sélectionnés, déplacés et tournés (rotation dans l'espace).
- La personnalisation du rendu visuel et du comportement des éléments est possible (et non fixé a priori comme en 2D), ceci en raison de l'objectif visé d'améliorer leur visualisation pour l'utilisateur.

7.3.3.1 Les stratégies d'organisation de l'information

La visualisation en 3D peut être réalisée soit sous forme hiérarchique pour représenter la hiérarchie de concepts à partir d'un concept sélectionné, soit sous forme de graphe pour représenter un concept sélectionné et l'ensemble de ses propriétés et instances.

1. *Arbre* : L'affichage en arbre permet la visualisation de la hiérarchie des concepts à partir d'un concept donné (sélectionné dans la partie textuelle de la fenêtre). Le concept sélectionné est centré au mieux dans la fenêtre de visualisation et sa sous-hiérarchie de concepts est affichée. La figure 7.5 illustre ce type d'affichage. Le concept *event* a été selectionné, l'ensemble de ses sous-concepts *conference, presentation, workshop, socialevent* est affiché. Ces informations sont aussi données en mode textuel à droite de la fenêtre avec en premier le nom du concept sélectionné et en dessous ses sous-concepts. Ils peuvent également être retrouvés également en mode textuel dans la partie gauche, dans la structure hiérarchique décrite.
2. *Graphe* : L'affichage en graphe permet l'affichage d'un concept sélectionné (dans la partie gauche de la fenêtre) avec l'ensemble de ses propriétés et instances. Il donne une vue détaillée d'un concept avec l'ensemble des informations qui lui sont relatives. La figure 7.6 illustre ce type d'affichage. Dans cet exemple, l'élément sélectionné est le concept de *presenter*, l'espace 3D affiche le concept et ses propriétés *(presenterOf, hasPerson)* et avec son sous-concept *keynotespeaker*.

7.3.3.2 La représentation graphique

Un rendu visuel (représentation graphique) est proposé par défaut. Il est également paramétrable par l'utilisateur.

La représentation graphique par défaut des différents éléments de l'ontologie (concepts, propriétés, instances, etc.) est la suivante :

- Les concepts sont représentés par des sphères bleues.

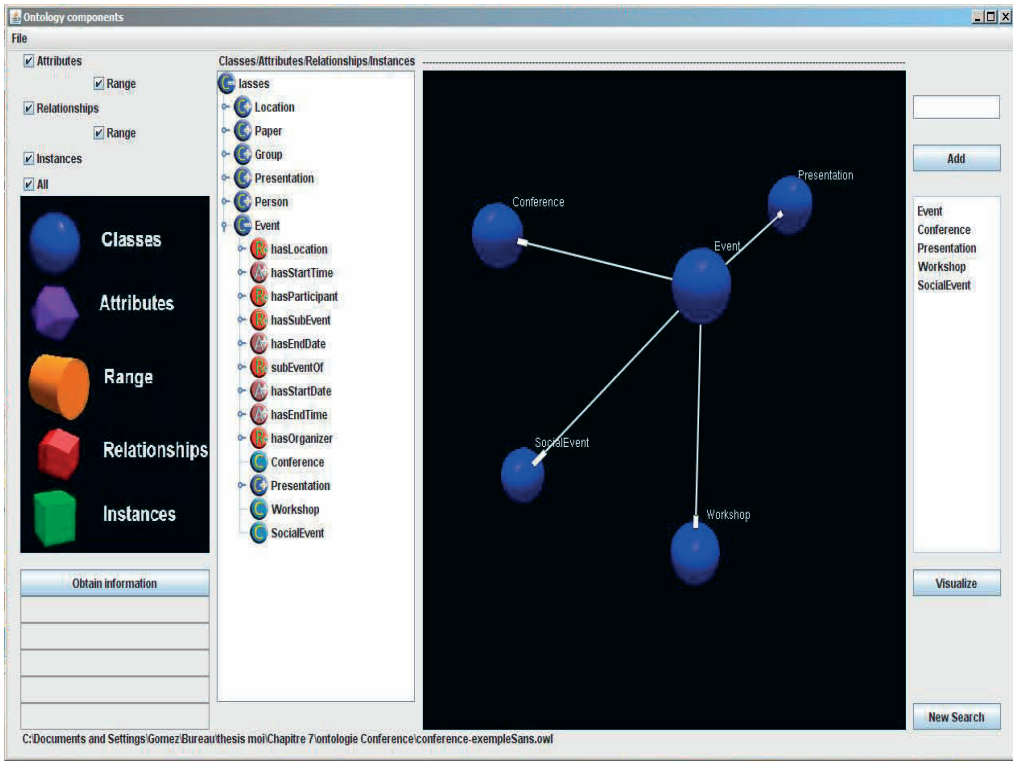

FIGURE 7.5 – Affichage d'arbre en 3D

- Les propriétés de type objet (relations) sont représentées par des décaèdres rouges.
- Les propriétés de type données (attributs) sont représentées par des dodécaèdres violets.
- Les domaines d'arrivée (range) des propriétés de type données sont des cylindres jaunes. Ils n'apparaissent que lorsque la propriété est explicitement sélectionnée.
- Les instances sont représentées par des cubes verts.

Les relations entre les éléments sont représentées par des lignes blanches avec l'extrémité plus épaisse indiquant le sens de la relation. Comme pour la visualisation en 2D, les noms des éléments sous forme de texte sont affichés.

Ces représentations graphiques sont illustrées par les figures 7.5 et 7.6. L'utilisateur a la possibilité de visualiser les options d'affichage. Dans la partie en haut à gauche, on peut voir les options de choix de paramétrage pour sélectionner et visualiser les différents éléments. La fenêtre de la partie centrale permet de visualiser l'arbre des éléments pour la sélection les éléments à visualiser.

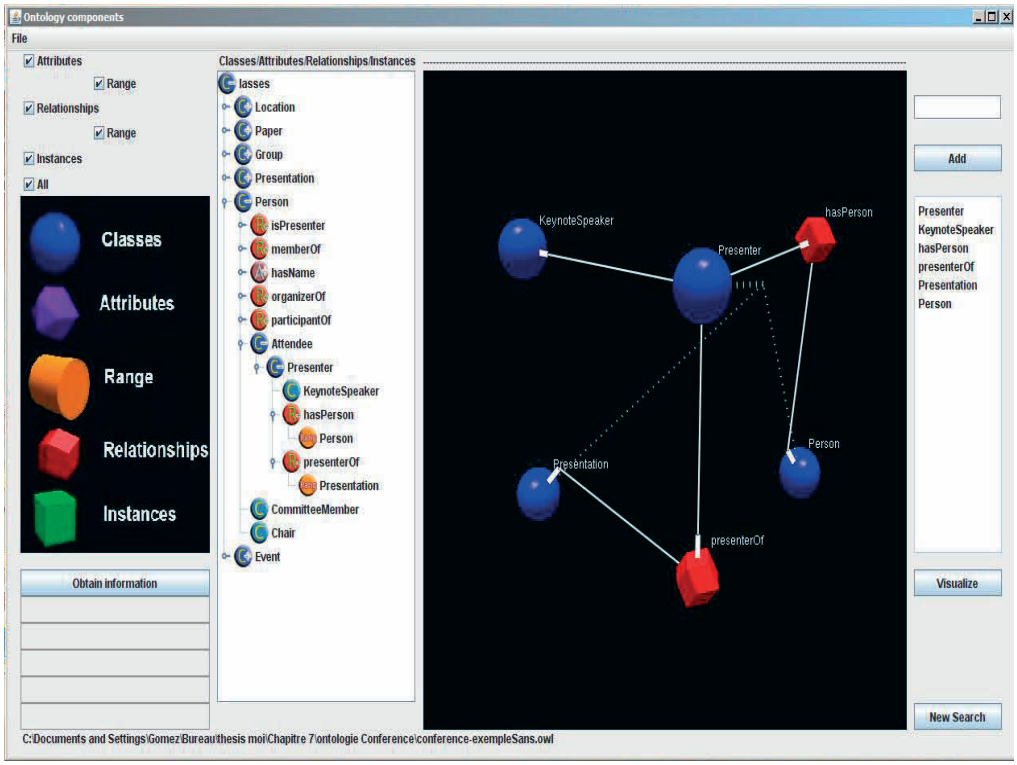

FIGURE 7.6 – Affichage de graphe en 3D

7.4 Construction d'une requête

Le module de construction de requêtes est un module d'aide pour l'utilisateur, pour l'assister dans la construction de sa requête. La requête, dans l'architecture de coopération OWCSIS, est posée sur l'ontologie de référence. La figure 7.1 montre le service de visualisation SEVI. La fonctionnalité de construction d'une requête est identifiée sur la figure par la flèche annotée *A - Demande pour poser la requête.*

Le module permet la construction dynamique de la requête dans un environnement graphique et sa génération automatique en langage SPARQL [SPA]. Dans l'environnement OWSCIS, le langage OWL [OWL] a été retenu pour la modélisation des ontologies locales et de référence et le langage SPARQL a été choisi pour exprimer les requêtes. Certaines contraintes liées à sa résolution dans la coopération sont imposées sur la forme des requêtes, dans le système OWSCIS et donc dans l'outil de visualisation SEVI.

Une requête en SPARQL est composée d'un ensemble de triplets (sujet, prédicat, objet). Dans SEVI, une contrainte complémentaire consiste à imposer que le prédicat soit une propriété de type objet ou données mais ne soit pas une variable.

La forme générale d'une requête est la suivante :

```
SELECT ?v1 ?v2 ?vi ?vn
```

```
WHERE {
(
...
sujetj (predicatj) (objetj).
...
}
FILTER ...
```

où :
- La partie *SELECT* comporte la liste des variables de la requête dont on souhaite obtenir les valeurs. Les variables sont préfixées par le caractère ? (par exemple ?*v*1)
- La partie *WHERE* spécifie la requête sous forme de triplets. Le sujet est une variable. Le prédicat est une propriété de type objet ou données. L'objet peut être soit une variable soit un littéral, soit une instance d'un concept. Les prédicats peuvent également être des prédicats prédéfinis du langage.
- La partie *FILTER* permet de fixer des conditions sur la requête.

7.4.1 Méthode de construction

La requête est construite de manière dynamique grâce au module de requête. L'objectif du module est de proposer à l'utilisateur un environnement assisté pour construire la requête en générant au fur et à mesure la requête en langage SPARQL. Cette construction est basée sur les étapes suivantes :

1. La description de liste des variables. L'utilisateur peut spécifier une variable et sélectionner son type parmi la liste des concepts de l'ontologie. La partie *SELECT* de la requête est automatiquement complétée et un triplet qui indique le type de variable est également ajouté. Par exemple, pour une variable ***x*** associée au concept ***person***, la variable ?***x*** est ajoutée dans la partie *SELECT* et le triplet ?***x a person*** est ajouté dans la partie *WHERE*.
2. La construction dynamique des triplets de la requête en répétant les étapes suivantes :
   A) choix d'une variable comme sujet,
   B) choix comme prédicat, d'une propriété (*object property* ou *datatype property*) ayant comme domaine de départ, le concept associé à la variable sélectionnée. Une présentation dynamique de ces propriétés est proposée par le module.
   C) choix, comme objet, (i) d'une variable typée si le prédicat est une propriété de type objet, ou (ii) d'une variable "libre" si la propriété est de type données ou (iii) d'un littéral.
3. De façon optionnelle, il est possible d'ajouter un filtre de la forme :

- condition basique : variable ($?x$), opérateur de comparaison ($<, >, =$, *etc.*), valeur ou variable.
- condition composée : condition basique, opérateur logique ($\&\&, |||, !$), condition basique.
- une expression régulière exprimée sous la forme *regex* de SPARQL.

La figure 7.7 résume ces étapes et les actions que l'utilisateur doit effectuer pour la construction d'une requête.

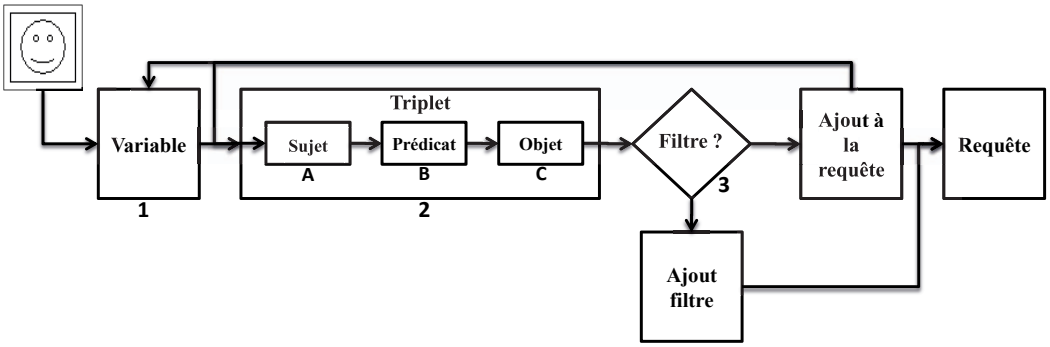

FIGURE 7.7 – Les étapes pour la construction de requête

Les tableaux 7.1, 7.2 et 7.3 détaillent respectivement les actions à réaliser par l'utilisateur pour (i) *typer* une variable, (ii) construire un triplet et (iii) construire un filtre conditionnel. Le typage d'une variable génère un triplet avec le prédicat prédéfini *a* de SPARQL.

TABLE 7.1 – Typage d'une variable

Sujet	**Prédicat**	**Objet**
Sélection d'une variable dans la liste des variables déjà construites	a	Séléction d'un concepts dans la liste des concepts

La figure 7.8 illustre de façon schématique les différents éléments de construction d'une requête. Les valeurs qui apparaissent sur la figure sont celles de la requête exemple présentée dans la section suivante. On peut noter les différentes parties de construction de la requête pour les variables, les triplets et les filtres.

TABLE 7.2 – Construction des triplets

Sujet	**Prédicat**	**Objet**
Sélection d'une variable	Sélection d'une propriété.	
	1) Si la propriété est de type objet	Sélection d'une variable typée du domaine d'arrivée de la propriété
	2) Si la propriété est de type données	Sélection d'une variable libre ou saisie d'une valeur de littéral.

TABLE 7.3 – Construction d'un filtre

Filtre		
L'utilisateur choisit la condition		
basique	par exemple : FILTER ($?z = 13$)	
composée	par exemple : FILTER (($?z >= 21$) && ($?z < 33$))	condition basique, opérateur (&&, |||,), condition basique.
regex	par exemple :FILTER regex (?y, "Wh")	regex : (expressions autorisées en SPARQL).

7.4.2 Exemple de construction d'une requête

Nous présentons dans cette section la méthode de construction d'une requête sur un exemple, nous utiliserons cette requête dans la suite de ce chapitre pour la visualiser, visualiser son enrichissement et ses résultats. La requête est exprimée sur l'ontologie *Conférence* déjà utilisée. La figure 7.9 présente l'extrait de l'ontologie utilisée pour la requête avec les concepts *event, presenter, presentation, etc.* et les propriétés *hasPresenter, isPresenterOf, hasName, hasTitle, etc.*.

L'utilisateur soumet la requête suivante : *Lister les noms des personnes qui sont des présentateurs et qui ont réalisé une présentation, avec le titre de la présentation. On ne retiendra que les personnes dont le nom commence par "GU".*

Pour construire cette requête l'utilisateur choisit une variable *x* dans la liste des variables et le concept ***presenter***. Le premier triplet généré par le système est :

```
?x a :presenter
```

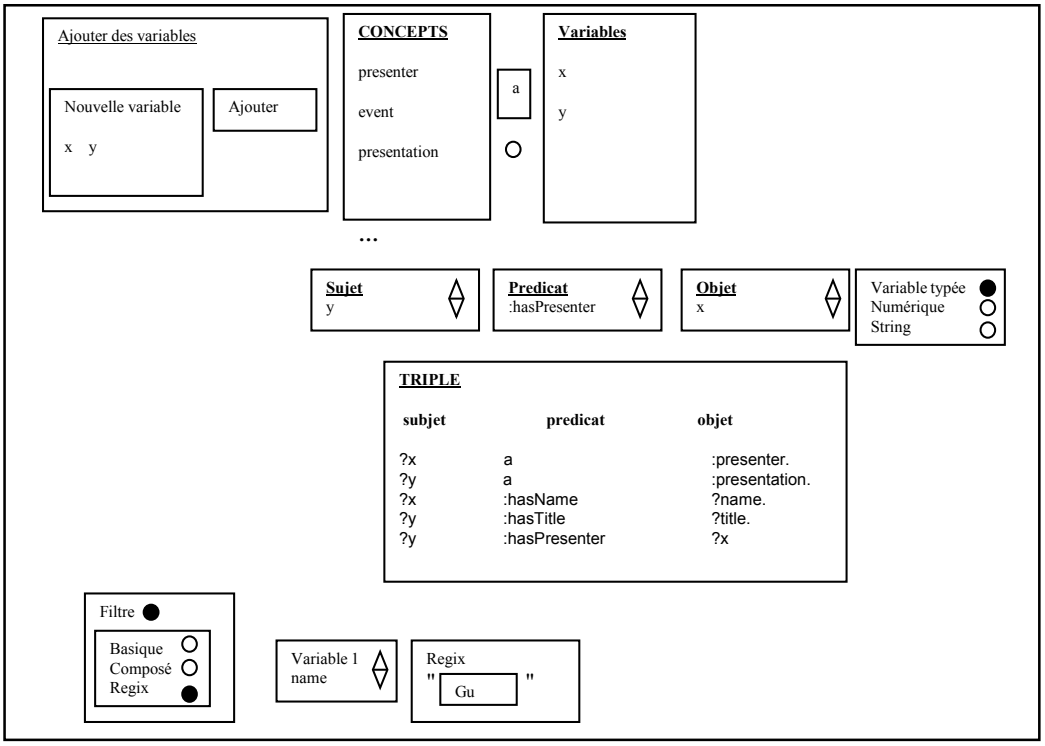

FIGURE 7.8 – Construction de la requête

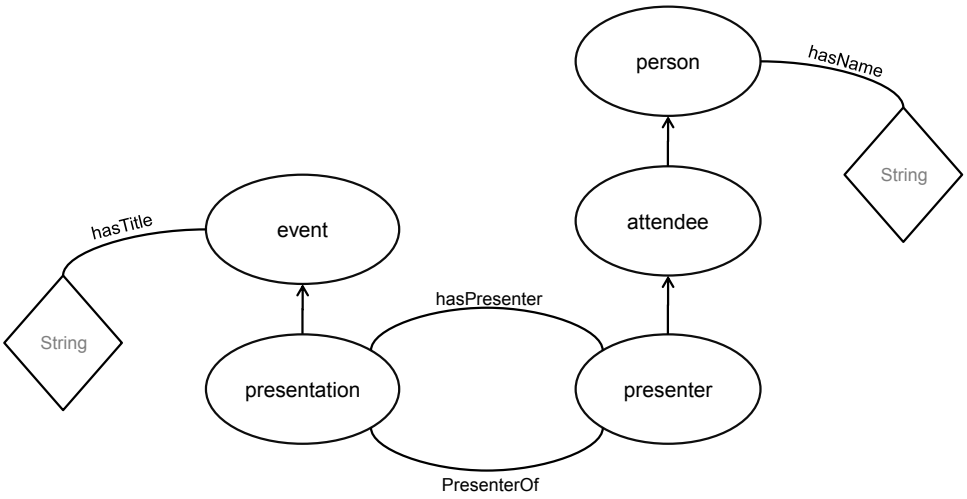

FIGURE 7.9 – Extrait de l'ontologie "*conference*"

De la même manière, l'utilisateur choisit une variable *y* et le concept ***presentation***, le triplet généré est :

```
?y a :presentation
```

Ensuite l'utilisateur construit les triplets qui permettent la sélection du nom de

présentateur et du titre de la présentation. Les propriétés *hasName* et *hasTitle* sont sélectionnées, et les variables utilisées sont des variables libres (non typées) ?*name* et ?*title*. Il construit également le triplet qui associe la présentation et le nom avec la propriété *hasPresenter*. Les sujets et objets de ce triplet sont des variables typées ?*x* et ?*y*. Les triplets suivants sont générés :

```
?x   :hasName   ?name.
?y :hasTitle  ?title.
?y  :hasPresenter  ?x
```

Une fois les triplets construits, l'utilisateur choisit le filtre (condition) en prenant les noms qui commencent par les lettres "GU", le filtre ajouté dans la requête est le suivant :

```
FILTER regex ( ?name, "^Gu")
```

La requête complète SPARQL générée est la suivante :

```
SELECT ?name ?title
WHERE
{
?x  a       :Presenter.
?y  a           :Presentation.
?x  :hasName    ?name.
?y  :hasTitle   ?title.
?y  :hasPresenter   ?x
FILTER regex (?name, "^Gu")
}
```

Lorsque la requête est construite, le module de construction de la requête l'envoie au service de requête pour son traitement et son enrichissement.

7.5 Visualisation d'une requête, enrichissement et résultats

Le module de visualisation d'une requête a pour objectif de proposer à l'utilisateur toutes les options de visualisation concernant la requête soumise. C'est un module directement lié au service de requête de l'architecture OWS-CIS. Son interaction avec l'utilisateur se fait à partir de l'environnement de paramétrage, ce qui permet des sorties en 2D et 3D. Ce module propose trois sous-modules de visualisation : requête, enrichissement et résultats.

1. *Requête* : Ce sous-module permet la visualisation de la requête à traiter. Il est également chargé d'extraire les concepts de la requête et d'assurer la transmission de ces informations vers le sous-module d'enrichissement.

2. *Enrichissement* : Ce sous-module récupère les concepts proposés pour l'enrichissement QUEXME du service de requête et offre des fonctionnalités pour visualiser cet enrichissement proposé. Il assure également la transmission des informations vers le module de construction de la requête pour permettre à l'utilisateur de construire une nouvelle requête *enrichie*.
3. *Résultats* : Ce sous-module assure la récupération et la visualisation des résultats pour la requête en cours de traitement. Ces résultats sont au format SPARQL-XML (SPARQL Query Results XML Format).

Le résultat d'une requête est transmis au module de visualisation sous forme d'un document XML avec la syntaxe (SPARQL-XML) sous la forme suivante :

Résultat SPARQL-XML

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
<head>
...
<variable name=.../>
</head>
<results>

        <result>
                <binding name= ...>
                        <bnode> ...</bnode>
                </binding>
                <binding name= ...>
                        <literal> ... </literal>
                </binding>
                ...
         </result>
         ...
 </results>
</sparql>
```

Le document XML comporte deux parties : la partie entête dans la balise *head* qui décrit la liste de variables de la requête et la partie résultat décrite dans la balise *results*. Chaque résultat est décrit dans une balise *result* avec la liste des variables et les valeurs associées. Les balises *bnode* sont utilisées pour des instances d'objet et les balises *literal* pour les valeurs de propriétés de type données. Le résultat sparql-xml de la requête de notre exemple est le suivant :

Résultat de la requête

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
<head>
<variable name="x"/>
<variable name="y"/>
<variable name="name"/>
<variable name="title"/>
</head>
<results>
        <result>
```

```
11                  <binding name="x">
12                          <bnode>x1</bnode>
13                  </binding>
14                  <binding name="y">
15                          <bnode>y1</bnode>
16                  </binding>
17                  <binding name="name">
18                          <literal>Guillermo</literal>
19                  </binding>
20                  <binding name="title">
21                          <literal>Ontology</literal>
22                  </binding>
23          </result>
24  </results>
25  </sparql>
```

La section suivante détaille la méthode mise en œuvre dans SEVI pour le traitement et la visualisation d'une requête dans l'environnement OWSCIS.

7.5.1 Méthode de visualisation

Le traitement d'une requête dans SEVI est illustrée par la figure 7.1. Le schéma présente toutes les étapes nécessaires pour la construction, le traitement, l'enrichissement d'une requête et les visualisations associées.

1. Construction de la requête. L'utilisateur demande au module de construction de la requête de l'assister pour soumettre sa requête (flèche A).
2. Envoi de la requête au service de requête (flèches B)
   (a) Traitement de la requête par le service de requête.
   (b) Enrichissement de la requête et retour des résultats de la requête enrichie.
3. Visualisation de la requête (flèches C)
   (a) Visualisation de la requête initiale.
   (b) Visualisation des résultats à partir de donnés envoyées par le service de requête.
   (c) Visualisation de l'enrichissement à partir du résultat de l'enrichissement de la requête par le service de requête.
4. L'utilisateur peut reconstruire la requête à partir du résultat de l'enrichissement, on reprend à partir de 1 pour construire la requête enrichie (flèche D).

Le processus de visualisation d'une requête dépend du besoin de l'utilisateur, il peut paramétrer l'environnement et changer de vue. Le processus se répète quand l'utilisateur soumet une autre requête ou pour un nouvel utilisateur.

7.5.2 La représentation graphique 2D

Nous avons présenté dans la section 7.3.2.2 les caractéristiques de la représentation graphique en deux dimensions pour la visualisation de l'ontologie. Cependant, cette représentation change selon la fonctionnalité du module.

1. **Requête** La représentation graphique est différente pour les éléments de la requête. Par défaut, les concepts de la requête apparaissent en bleu et sont situés au centre de la fenêtre de visualisation. Les propriétés de type objet de la requête sont également présentées dans la fenêtre de visualisation mais sans changement de couleur, elles sont identifiées par leur nom. Les propriétés de type de données de la requête sont visualisées en format texte dans la fenêtre à gauche. La figure 7.10 illustre la requête de notre exemple, les deux concepts *(presenter, presentation)* appartiennent à la requête et se trouvent en bleu au centre de la visualisation. La propriété *(hasPresenter)* de type objet est également visualisée et les propriétés de type données *(hasName et hasTitle)* peuvent être visualisées en texte sur la fenêtre à gauche. Dans cette fenêtre, d'autres informations sur les concepts comme leur super-concept sont également données en mode textuel sur le concept sélectionné.

2. **Enrichissement** La représentation graphique est également différente pour la fonctionnalité de visualisation de l'enrichissement. Les concepts proposés pour l'enrichissement par QUEXME sont placés au centre de la fenêtre de visualisation en jaune avec les autres concepts de la requête en bleu. La figure 7.11 illustre la proposition d'enrichissement de QUEXME pour la requête de l'exemple. Les concepts proposés sont *(event, person, attendee)* et les concepts de la requête sont *(presenter, presentation)*.

3. **Résultats** La représentation graphique est spécifique pour les instances qui appartiennent aux résultats de la requête. Les résultats sont sous forme de carrés colorés situés au centre de la fenêtre. Par défaut, toutes les instances sont en vert, mais si l'utilisateur en sélectionne une, elle devient rouge, ainsi que ses propriétés, afin que l'utilisateur puisse visualiser facilement les concepts et les propriétés liés. Le concept associé à l'instance est en vert foncé. La valeur de l'instance et ses propriétés sont également visualisées sous forme de texte dans une fenêtre à gauche. Les résultats de la requête de l'exemple sont les instance ($x1$) et ($y1$). ($x1$) a un nom (Guillermo_GOMEZ) et c'est la personne qui a une présentation ($y1$) dont le titre est (*Ontology*). La figure 7.12 illustre les résultats avec la sélection du concept ($y1$) qui est en rouge, cette instance a un titre (*ontology*) et un présentateur ($x1$). La figure 7.13 illustre le même résultat mais avec la sélection du concept ($x1$) qui est en rouge, cette instance est de type présentateur et a un nom *(Guillermo GOMEZ)*. Dans les deux cas, l'information textuelle est présentée dans la fenêtre à gauche des visualisations.

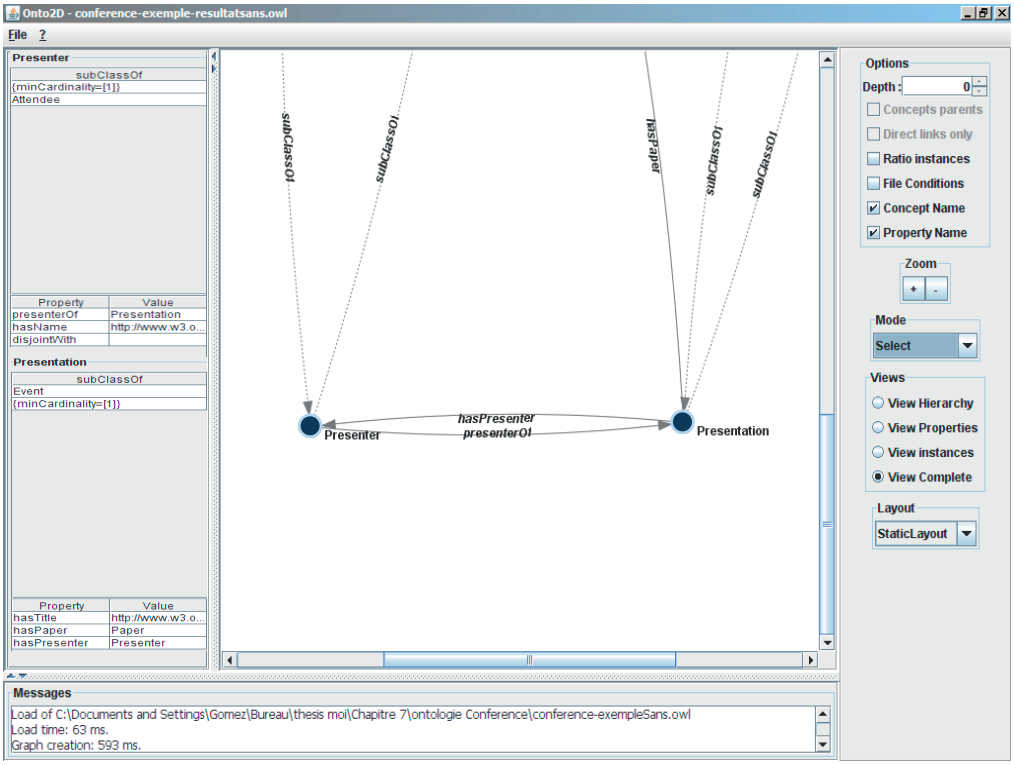

FIGURE 7.10 – Visualisation des éléments de la requête en 2D

7.5.3 La représentation graphique 3D

La visualisation d'une requête, de son enrichissement et de ses résultats est également proposée en 3D. L'objectif est de montrer de façon conviviale et dynamique ses différents éléments et d'offrir des outils pour paramétrer et manipuler la visualisation. Cette représentation en 3D est différente en fonction de la fonctionnalité du module mais utilise toujours les caractéristiques de la représentation graphique présentées dans la section 7.3.3.2.

1. **Requête**. Cette fonctionnalité permet l'affichage des éléments qui appartiennent à la requête et éventuellement des informations de son voisinage sémantique. Les éléments non directement inclus dans la requête sont représentés et liés par des lignes blanches en pointillés. La figure 7.15 illustre la visualisation de la requête de l'exemple. Elle visualise les deux concepts *(presenter, presentation)*, elle affiche également les propriétés *(hasName, hasTitle, hasPresenter)*. La visualisation présente également le concept *(person)* lié indirectement aux concepts *(presenter, presentation)*.
2. **Enrichissement** Un nouvel espace 3D de visualisation est généré avec les concepts proposés par QUEXME. Si les concepts sont liés dans l'ontologie, les relations sont représentées par des lignes blanches sinon par des lignes en pointillés blanches. La figure 7.15 illustre la proposition d'enrichissement de QUEXME pour la requête de l'exemple. La visuali-

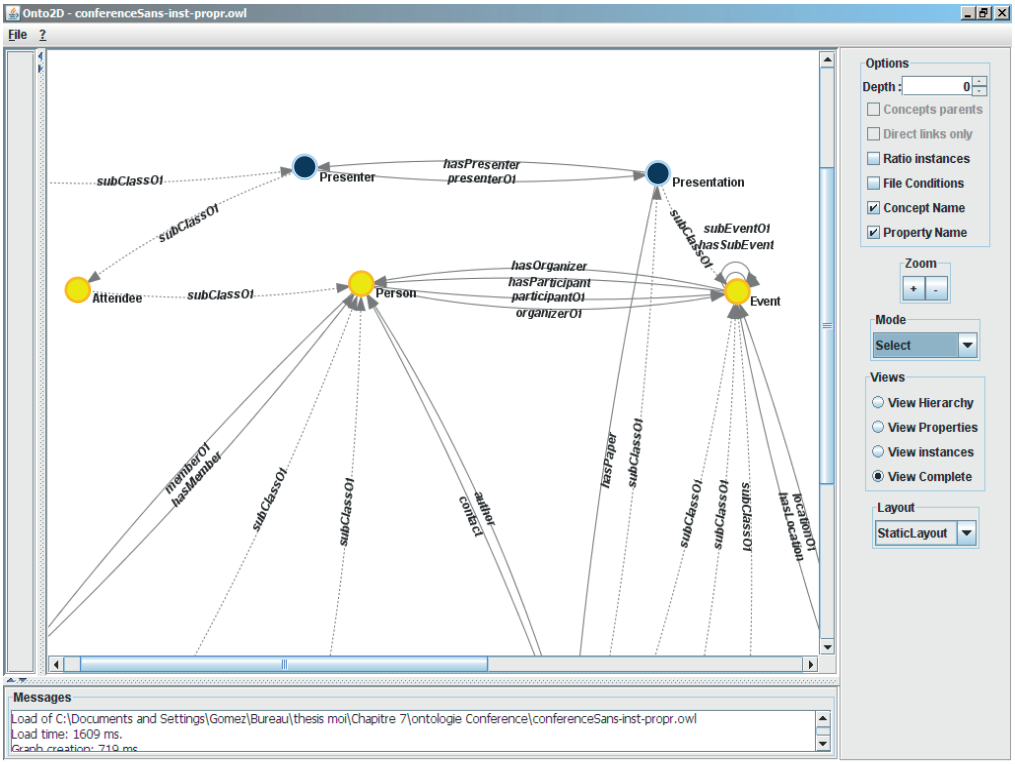

FIGURE 7.11 – Visualisation des concepts proposés pour l'enrichissement

sation permet de voir les trois concepts proposés *(event, person, attendee)* et les concepts de la requête *(presenter, presentation)*. Elle montre que les concepts pour l'enrichissement liés directement sont *(person,attendee, presenter)* et *(event, presentation)*.

3. **Résultats** Un nouvel espace 3D de visualisation est généré qui montre les instances des concepts de la requête. Cette représentation est complétée par des données textuelles qui sont affichées lors de la sélection d'une instance.La figure 7.16 illustre les résultats de la requête de notre exemple. La visualisation présente les instances liées à ses concepts, ($x1$) au concept (*presenter*) et ($y1$) au (*presentation*). Les lignes blanches pointillés entre les concepts indiquent l'existence des propriétés. Les propriétés de type de données (*hasTitle*) et (*hasName*), respectivement liés à ses concepts (*presentation*, *presentater*) sont également représentées. Une fenêtre textuelle peut donner les valeurs des instances de la façon suivante lorsqu'elles sont sélectionnées. Par exemple, pour l'instance ($y1$), nous avons :

Instance	Property	Object/Value
$y1$	hasPresenter	$x1$
$y1$	hasTitle	Ontology

Dans l'état actuel du prototype, les relations entre les instances ne sont pas visualisées, ce qui ne permet pas d'avoir une visualisation des résultats *complète*. Ces informations peuvent être visualisées en sélectionnant

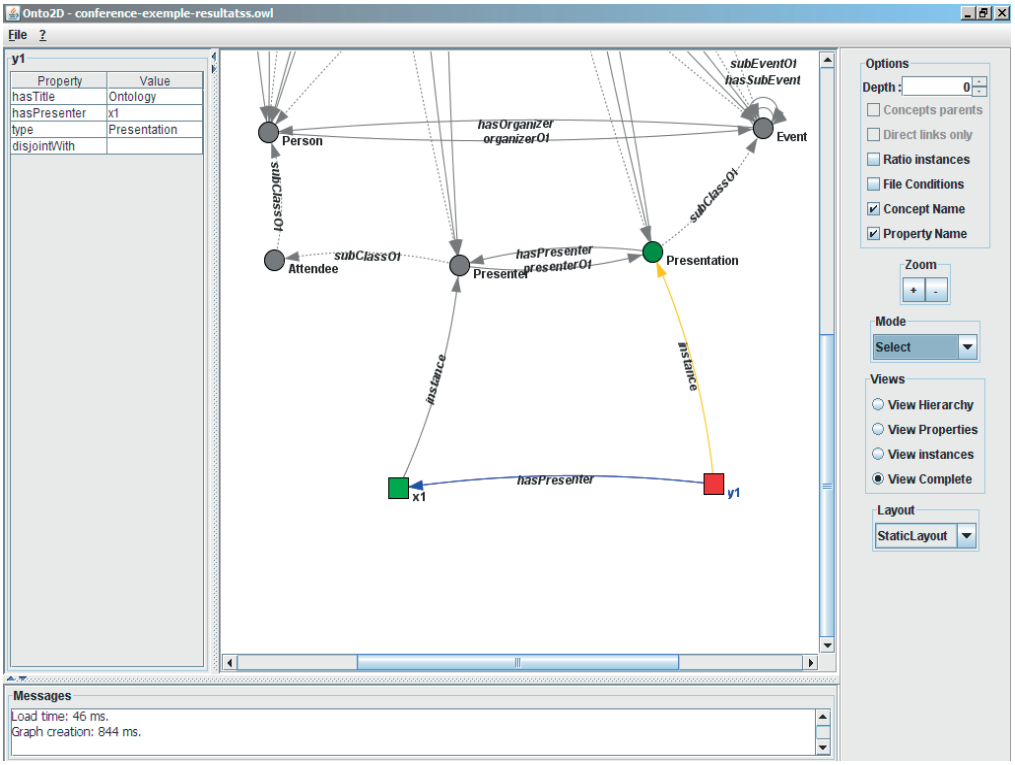

FIGURE 7.12 – Visualisation des résultats

l'instance pour connaître ses propriétés de façon textuelle.

Pour toutes les fonctionnalités, une visualisation de l'arbre 2D de l'ontologie est présentée au milieu de la fenêtre. Cette représentation hiérarchique peut être utilisée pour aider les utilisateurs à se situer dans l'ontologie, ou pour sélectionner d'autres éléments à afficher.

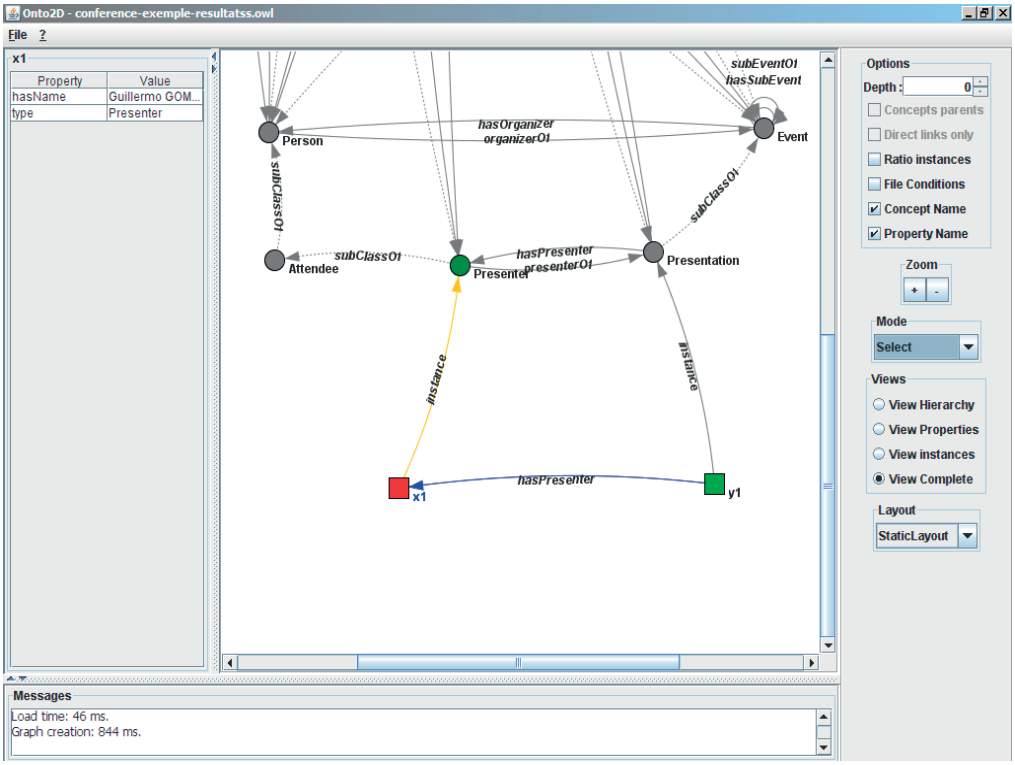

FIGURE 7.13 – Visualisation des résultats

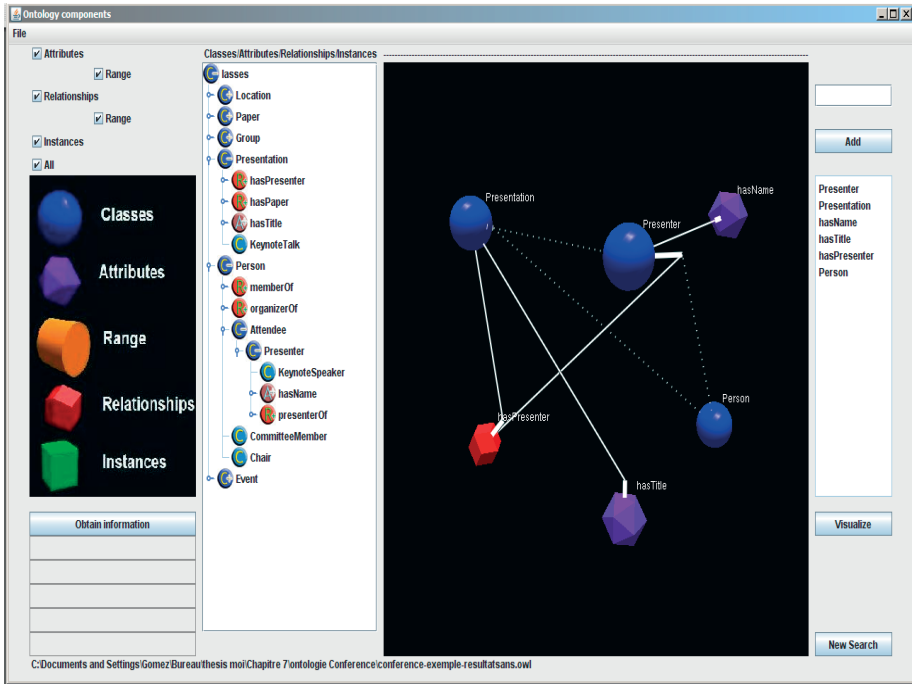

FIGURE 7.14 – Visualisation des éléments de la requête en 3D

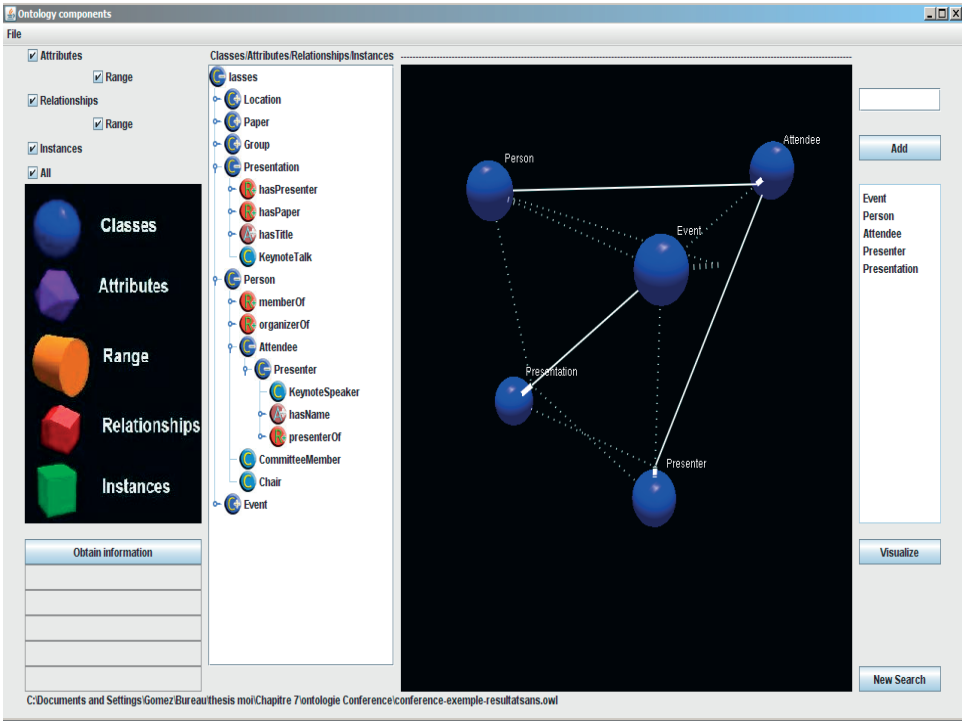

FIGURE 7.15 – Visualisation des concepts proposés pour l'enrichissement en 3D

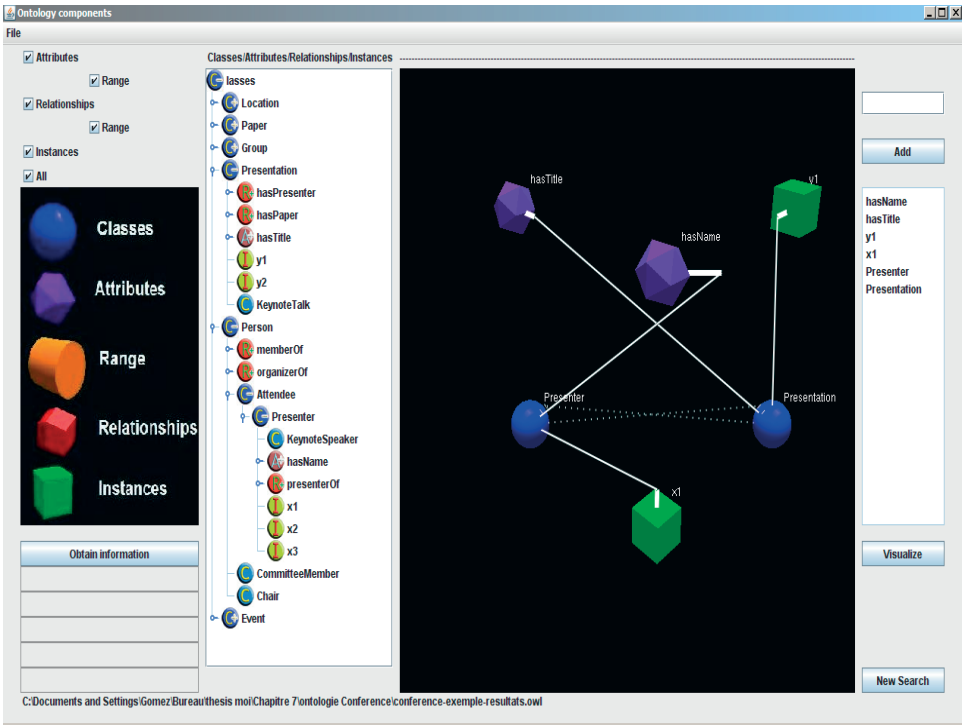

FIGURE 7.16 – Visualisation des résultats (instances) en 3D

7.6 Conclusion

Dans ce chapitre, nous avons présenté le service de visualisation SEVI. Ce service fait partie de l'architecture générale de OWSCIS et se compose de trois modules et un environnement de paramétrage : (i) visualisation de l'ontologie, (ii) construction de la requête et (iii) visualisation d'une requête. Le module de visualisation d'une requête offre trois fonctionnalités : requête, enrichissement et résultats.

Notre approche de visualisation vise à exploiter et afficher les relations sémantiques entre les concepts en se basant sur l'ontologie de référence. Elle a pour objectif d'assister l'utilisateur dans son processus de recherche d'information afin de répondre à ses besoins. Avec le service de visualisation l'architecture OWSCIS devient un système interactif en offrant un espace dynamique et convivial pour l'utilisateur en l'aidant à comprendre, localiser et visualiser les relations des éléments de sa requête construite dans le langage *sparql*. La visualisation est en 2D ou 3D. Enfin, le service de visualisation *SEVI* est indépendant de l'ontologie utilisée.

Troisième partie

Expérimentation et évaluation

8

Expérimentation

Ce chapitre présente la partie expérimentale de notre travail. Nous décrivons le protocole d'expérimentation appliqué et ses différentes étapes (apprentissage et enrichissement). Puis nous analysons les résultats obtenus en appliquant notre approche d'enrichissement de requêtes QUEXME.

Sommaire

8.1 Introduction

Nous décrivons dans ce chapitre l'expérimentation menée dans le cadre cette thèse [GCAC10]. L'objectif de ce chapitre est la présentation et l'évaluation de notre approche d'enrichissement de requêtes (chapitre 6). Dans le cadre de notre travail, nous avons utilisé, pour l'évaluation de notre approche, l'ontologie développée dans le projet SEMIDE (Système Euro-Méditerranéen d'Information sur les savoir-faire dans le Domaine de l'Eau)(SEMIDE/EMWIS [9]). Le SEMIDE est une initiative du partenariat Euro-Méditerranéen qui comporte 27 états membres de l'Union Européenne et 10 Pays Partenaires Méditerranéens. Il fournit un outil stratégique pour l'échange d'information et de savoir-faire dans le domaine de l'eau entre et à l'intérieur des pays du partenariat Euro-Méditerranéen. L'outil développé permet des recherches par thèmes, sous-thèmes etc., les thèmes pouvant être liés entre-eux. Ces informations sont modélisées au sein d'une ontologie "légère". Nous avons expérimenté notre approche en utilisant cette ontologie avec l'accord du Semide. Il est ainsi possible de construire des requêtes sur l'ontologie qui correspondent à des recherches dans la classification des thèmes de l'ontologie et d'appliquer la méthode d'enrichissement QUEXME pour enrichir les requêtes et proposer un ensemble de termes liés à ceux de la requête initiale émise par l'utilisateur.

La suite de ce chapitre est organisée comme suit : la section 8.2 présente la démarche adoptée pour l'expérimentation (protocole d'expérimentation).

Une description de l'ontologie du SEMIDE utilisée est présentée dans la section 8.3. La section suivante décrit les différentes phases de l'expérimentation et l'approche est évaluée dans la section 8.5. Enfin, nous concluons dans la section 8.6.

8.2 Protocole d'expérimentation

L'expérimentation a été menée en deux étapes qui correspondent aux deux phases nécessaires à l'utilisation de la méthode QUEXME.

- Etape d'apprentissage. La première étape concerne la constitution d'un corpus de requêtes modèles utilisé pour la phase d'apprentissage de la méthode QUEXME. Une base de requêtes modèles a été construite et les requêtes ont été soumises au système. Cette étape est destinée à l'initialisation des graphes et matrices de valeurs associées pour calculer la popularité d'un concept et donc ensuite sa pertinence pour enrichir une requête.
- Etape d'enrichissement. La deuxième étape concerne l'enrichissement. Un corpus plus important de requêtes ($\times 10$) utilisateur a été constitué et soumis au système pour être évalué. Dans cette étape, les graphes et matrices de valeurs sont également mis à jour, mais en plus, la phase d'enrichissement de la requête est appliquée. De façon plus précise, une

9. http ://www.semide.net/

session de recherche correspond à une séquence de requêtes (relatives à une même recherche). Lorsque 2 requêtes successives ne comportent aucun concept commun, le système considère qu'une nouvelle session de recherche est effectuée.

Après la mise en œuvre de ces 2 étapes, les résultats de l'expérimentation ont été évalués, ce qui correspond dans notre travail à la validation "manuelle" des enrichissements proposés par une personne (utilisateur du système ou expert de l'ontologie). La figure 8.1 résume les étapes de l'expérimentation proposée :

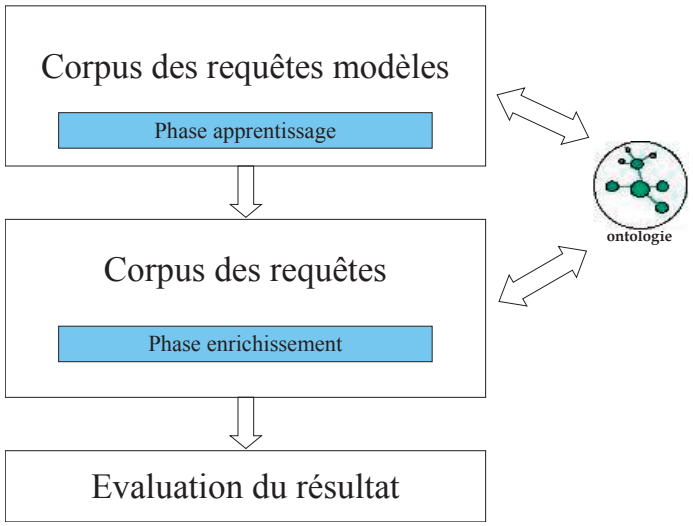

FIGURE 8.1 – Protocole d'expérimentation

8.3 L'ontologie du système SEMIDE

Nous avons évalué notre approche sur l'ontologie du système SEMIDE (Système Euro-Méditerranéen d'Information sur les savoir-faire dans le Domaine de l'Eau) comme nous l'avons expliqué en introduction de cette section. Cette ontologie a également été utilisée dans le cadre du travail proposé par L. Abrouk et E. Mino [AM04]. L'ontologie existante est basée sur les termes et les relations du thésaurus de l'office International de l'Eau (OIEau [10]).

Cette ontologie a été réalisée principalement afin de :

1. Partager le savoir commun entre les différents acteurs et partenaires dans le domaine de l'eau, soit de façon manuelle (navigation), soit de façon automatique (agents logiciels).
2. permettre l'intégration/l'importation d'ontologies d'autres domaines afin d'enrichir cette ontologie des savoir-faire dans le domaine de l'eau,

10. http ://www.oieau.fr/

3. faire en sorte de rendre le domaine explicite.

L'ontologie du SEMIDE est décrite par des concepts du domaine et des relations qui les relient. La figure 8.2 illustre un extrait de cette ontologie. Les noms des concepts ne sont pas significatifs mais chaque concept comporte un terme associé appelé "terme préféré". Le terme préféré représente la sémantique du concept. L'ontologie est essentiellement construite autour d'une hiérarchie de concepts (chaque concept a son terme préféré associé) et des relations sémantiques entre ces concepts. Les relations sémantiques sont par exemple de type "synonyme de" ou "relatif à". La figure 8.2 illustre la structure de l'ontologie (hiérarchie des concepts et termes préférés, les relations sémantiques ne sont pas représentées) :

- un nœud est un concept, représenté par un rectangle sur la figure (par exemple le concept C_1) ;
- les concepts sont organisés hiérarchiquement par des relations de spécialisation/généralisation, ici le concept C_2 est une spécialisation du concept C_1 ;
- chaque concept comporte un terme préféré, représenté dans un cercle lié au concept sur la figure.
- d'autres relations sémantiques peuvent lier des concepts comme des relations de synonymie.

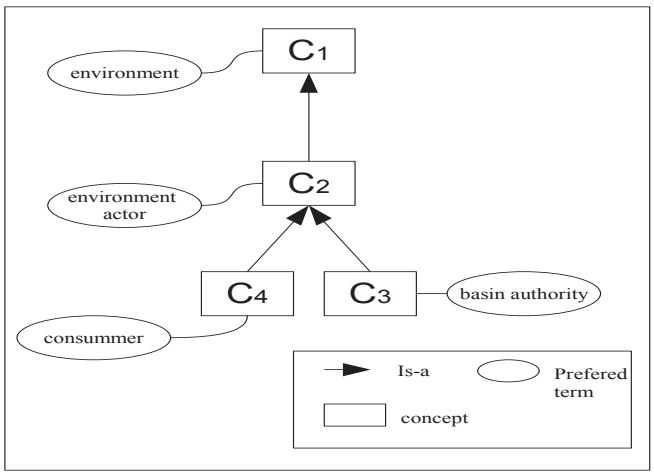

FIGURE 8.2 – Extrait de l'ontologie du Semide

Un extrait des informations modélisées dans l'ontologie est donné en annexe D. L'annexe présente de façon synthétique les thèmes (sujets) et sous-thèmes du domaine modélisés dans l'ontologie.

8.4 Etapes de l'expérimentation

Le processus d'expérimentation repose sur 2 étapes qui correspondent à la phase d'initialisation de la méthode QUEXME et à la phase d'utilisation, enrichissement des requêtes. Le principe appliqué est le suivant

1. Un corpus de 100 sous-séquences de requêtes modèles a été constitué à partir de l'ontologie. Une sous-séquence de requêtes correspond à la définition de 2 requêtes séquentielles dans une session de recherche.

2. La phase d'apprentissage analyse chaque sous-séquence de requêtes et met à jour les graphes et matrices de valeurs utilisées ensuite pour l'enrichissement.

3. Un corpus de 1000 requêtes utilisateur a ensuite été constitué et les requêtes ont été soumises au système pour tester l'étape d'enrichissement. Une session de recherche correspond à une séquence de requêtes. Une séquence comporte une requête initiale et une suite de sous-séquences de requêtes qui correspondent à l'ensemble des requêtes émises par un utilisateur pour une recherche donnée.

4. Pour chaque requête traitée lors de la phase d'enrichissement, les graphes et matrices de valeurs sont mis à jour et les calculs des termes pertinents pour l'enrichissement de la requête sont calculés et proposés.

8.4.1 Phase d'apprentissage

Au cours de cette première étape, nous avons construit une base de 100 sous-séquences de requêtes pour construire la matrice de concepts $\boldsymbol{MC}$. Le tableau 8.1 illustre quelques séquences de requêtes de cette phase. Il décrit les deux requêtes de la sous-séquence de requêtes (Requête1, Requête 2), les arcs qui sont définis pour cette sous-séquence (Arc v_{12} et leur poids associés (Poids)). Le graphe de concepts construit pour une séquence de requêtes est utilisé pour incrémenter la matrice de concepts.

Par exemple, pour la première sous-séquence présentée dans le tableau Q_1 *(environment, environment actor)* et Q_2*(environment actor, water actor)*, l'élément de la matrice $\boldsymbol{MC}$*(environment actor, water actor)* est incrémenté de 1 car un arc est défini entre les concepts *environment actor* et *water actor* avec un poids de 1 sur cet arc. Lorsque la matrice des concepts a été correctement initialisée, l'étape d'enrichissement peut être appliquée. La qualité des résultats de l'enrichissement est dépendante de la pertinence des sous-séquences de requêtes utilisées dans l'apprentissage car la matrice est à la base des calculs de l'importance d'un concept par rapport à une requête et donc à sa sélection ou non pour l'enrichissement.

TABLE 8.1 – Extrait de sous-séquences appliquées de la phase d'apprentissage

Requête 1	**Requête 2**	**Arc v_{12}**	**Poids**
environment, environment actor	environment actor, water actor	environment actor →water actor	1
administrative council, basin authority, services providing company	administrative council, basin authority, administrative organization	administrative council → administrative organization,basin authority →administrative organization	1/2
consumer	consumer, dry cleaning	consumer → dry cleaning	1
association, consumer, public opinion, environment	association, consumer, public opinion, services providing company	association → services providing company , consumer → services providing company, public opinion → services providing company	1/3

8.4.2 Phase d'enrichissement

Une fois la matrice des concepts initialisée, la phase d'enrichissement vise à proposer des termes pour enrichir les requêtes des utilisateurs. La phase d'enrichissement a été appliquée sur 1000 requêtes dans cette expérimentation. Nous rappelons ici brièvement les différentes étapes mises en œuvre dans la phase d'enrichissement qui comporte comme pour l'apprentissage une étape de construction des graphes et de mise à jour des matrices de valeurs mais également la recherche des concepts pertinents.

La figure 8.3 illustre des séquences de requêtes qui ont été réalisées et les termes proposés par le système.

Par exemple, pour la première requête de l'utilisateur Q_1 *(water actor, environment actor)*, l'enrichissement consiste à extraire les concepts de la requête : *water actor* et *environment actor* et à les analyser pour sélectionner les concepts pertinents pour son enrichissement.

Le tableau 8.2 montre la première séquence présentée sur la figure 8.3 avec : la requête émise, les concepts proposés pour l'enrichissement et les concepts retenus par l'utilisateur pour poursuivre sa recherche.

Pour calculer les concepts proposés pour l'enrichissement d'une requête, le système analyse chaque concept extrait de la requête et calcule l'importance des autres concepts de l'ontologie par rapport à la requête $I(c,Q)$. Ce calcul passe par le calcul des popularités des concepts $ConceptRank(CR)$ et l'importance d'un concept par rapport à un autre concept ($ConceptImportanceCI$). Il fournit une liste de concepts candidats pour l'enrichissement de la requête. La sélection se fait alors en se basant sur un facteur minimum d'importance

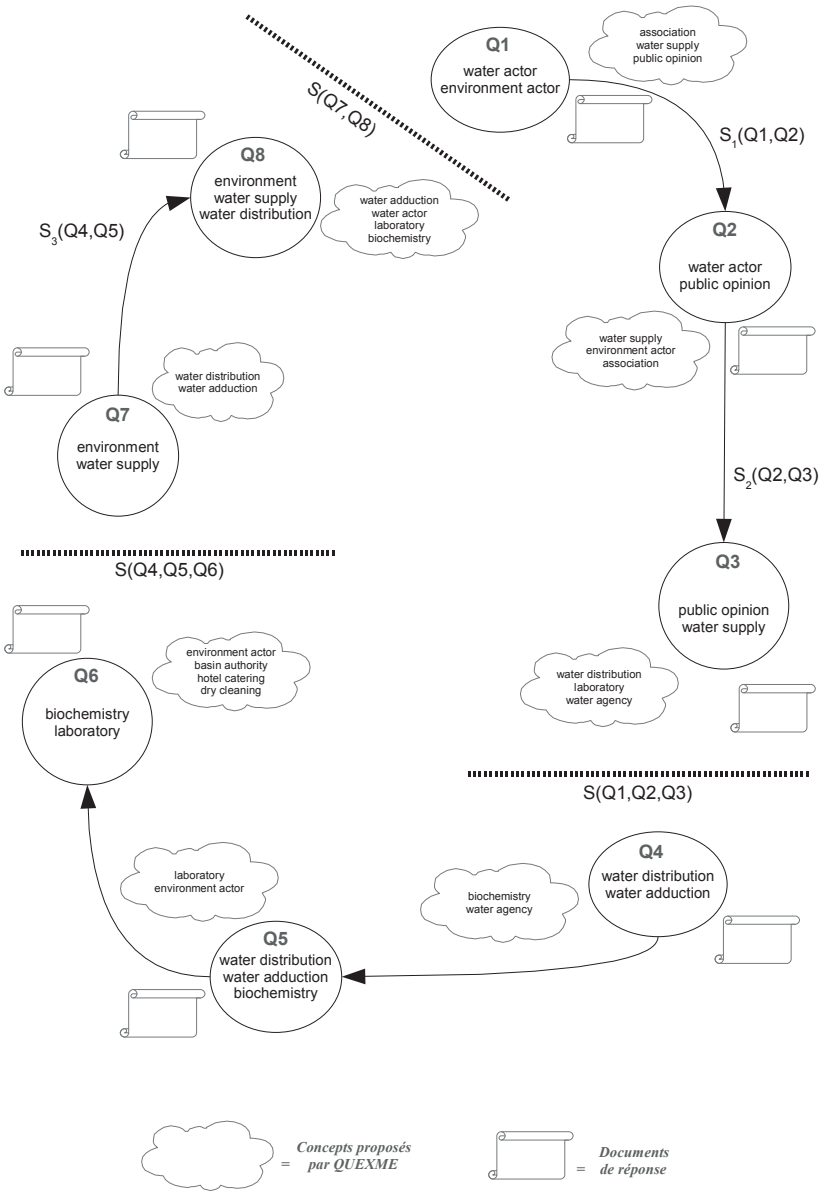

FIGURE 8.3 – Extrait de séquences et enrichissement proposé

$Min(FI/Qi)$ qui définit un intervalle de valeurs pour sélectionner les concepts à partir des importances des concepts par rapport à la requête. Les résultats de ces calculs sont illustrés par le tableau 8.3. Tous les concepts pour lesquels l'importance par rapport à la requête $I(c,Q)$ est supérieure au facteur minimum d'importance $Min(FI/Qi)$ sont sélectionnés et proposés pour enrichir la requête. Dans notre exemple, trois concepts *water supply, public opinion, association* sont retenus et proposés à l'utilisateur.

TABLE 8.2 – Enrichissement des requêtes de la séquence $S1$

Requête	**Concepts proposés par QUEXME**	**Concepts retenus de l'enrichissement**	**Concepts retenus de la requête initiale**
Q_1, water actor, environment actor	association, water actor, public opinion	public opinion	water actor
Q_2, water actor, public opinion	water supply, environment actor, association	water supply	public opinion
Q_3, public opinion, water supply	water distribution, laboratory, water agency	-	-

TABLE 8.3 – Concepts sélectionnés pour l'enrichissement

(Concept)	$I(c,Q_1)$	$I(c,Q_1) \geq Min(FI/Q_1)$
biochemistry	1.3342	
agricultural association	1.0000	
association	3.2141	x
consumer	0.0000	
services providing company	0.3333	
garage	0.0000	
public opinion	3.6684	x
community facility	0.8112	
water actor	2.4583	
water supply	6.1267	x

8.5 Evaluation

La phase d'expérimentation nous a permis de produire une ensemble de requêtes avec pour chaque requête un ensemble de concepts sélectionnés et proposés à l'utilisateur. Nous proposons dans cette section d'analyser et d'évaluer ces résultats selon deux points de vue : d'une part le comportement de l'utilisateur i.e. son comportement par rapport aux concepts proposés et leur utilisation dans sa session de recherche et d'autre part la validation des concepts proposés automatiquement par le système, par une personne (utilisateur ou expert).

8.5.1 Comportement des utilisateurs

Nous avons étudié le taux d'acception des concepts proposés aux utilisateurs, dans une séquence de recherche i.e. le pourcentage de concepts retenus entre une requête et la requête suivante. Nous avons sélectionné un échantillon de 100 requêtes pour cette étude.

Le tableau 8.4 montre sur quelques requêtes, les éléments considérés pour l'étude du comportement des utilisateurs. La première colonne décrit les concepts de la requête de l'utilisateur, la deuxième donne les concepts proposés par le système pour l'enrichissement de la requête. Les colonnes 3 et 4 donnent le nombre de concepts proposés par QUEXME et le nombre de concepts retenus par l'utilisateur. Les concepts non retenus par les utilisateurs sont en italique dans le tableau. La dernière colonne donne le taux d'acceptation des concepts.

TABLE 8.4 – Taux d'acceptation des concepts proposés

Concepts de la requête	**concepts proposés (QUEXME)**	**nombre concepts proposés**	**nombre concepts retenus**	% **acceptation**
agricultural association	international organization, national organization	2	2	100
aqueduct, environment	water actor, environment actor, *water distribution*	3	2	66
water agency	water supply, water distribution	2	2	100
environment, biochemistry	*environment actor*	1	0	0
services providing company, water agency	laboratory, *public opinion, administrative council*	3	1	33
hotel catering, garage	environment actor	1	1	100

La figure 8.4 représente sous forme d'un graphique le comportement des utilisateurs pour 24 requêtes de l'échantillon de 100 (prises au hasard).

Une analyse des résultats sur l'ensemble des échantillons montre que la satisfaction des utilisateurs est relativement similaire à celle des experts qui est décrite dans la section suivante. Le taux de satisfaction est supérieur à

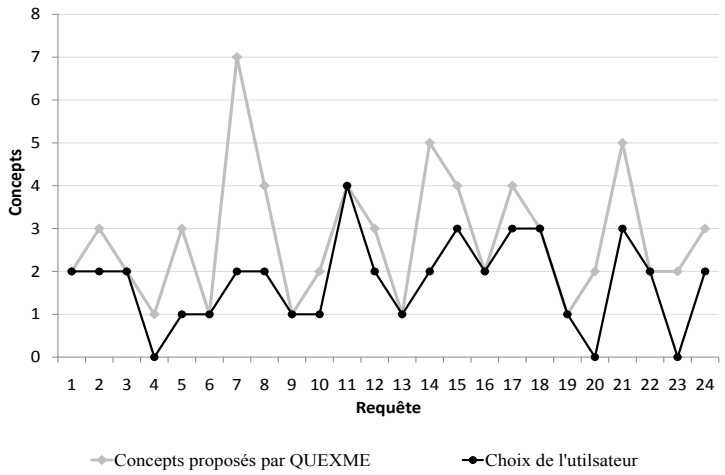

FIGURE 8.4 – Evaluation

80% pour environ 40% des requêtes. Le taux de satisfaction considéré est le rapport entre le nombre de concepts retenus sur le nombre de concepts proposés. Cependant une analyse plus détaillée des résultats montre que le taux de satisfaction n'est pas régulier pour une session de recherche, le taux est plus important au début de la session car la recherche est plus "large" et moindre en fin de session car la recherche "s'affine". On peut noter qu'un comportement "incohérent" d'un ensemble d'utilisateurs peut perturber la qualité du système car l'analyse est faite à partir des recherches réalisées par les utilisateurs. Cet effet peut être amoindri si la méthode est appliquée pour un ensemble important de requêtes (effet de lissage).

8.5.2 Validation

La phase de validation consiste en une analyse "manuelle" par une personne (utilisateur ou expert) des enrichissements proposés par le système. L'idée intuitive est de s'assurer, par une expertise manuelle, que les concepts proposés sont bien pertinents pour l'enrichissement de la requête posée. Nous avons également retenu un échantillon de 100 requêtes sur les 1000 réalisées dans l'expérimentation et nous avons calculé le pourcentage de requêtes dont le pourcentage de satisfaction (nombre de termes validés $\times 100/$ nombre de termes proposés) de l'expert est dans un intervalle donné : de 0 à 10%, de 10% à 50%, de 50% à 80% et de 80% à 100%. Le tableau 8.5 donne les résultats de ces calculs. Par exemple, 41% des requêtes analysées offre un taux de satisfaction de 80 à 100% i.e. que pour 41% des requêtes ont de 80 à 100% des termes proposés qui ont été validés par l'expert.

Une analyse plus détaillée de cette étude montre que si le nombre de concepts proposés est important, la pertinence de ces concepts diminue et que la validation du nombre de termes validés par l'expert diminue également. Le

TABLE 8.5 – Résultats de la validation par l'expert

Termes validés par l'expert (%)	**Nombre requêtes(%)**
80 à 100%	41.7
50 à 80%	29.1
10 à 50%	16.75
0 à 10%	12.5

choix des concepts sélectionnés en QUEXME est déterminé par un seuil fixé a priori mais qui pourrait être affiné par exemple pour ne jamais permettre la proposition de plus de quatre ou cinq concepts pour une requête. Cependant avec le paramétrage actuel de ce seuil qui est basé sur la moyenne des importances des concepts, les résultats sont globalement satisfaisants avec 40% des requêtes validées par l'expert pour 80 à 100% des concepts et 30% pour 50 à 80% des concepts. La non validation d'un concept signifie que sa pertinence pour la session de recherche n'est pas assez signifiante.

8.6 Conclusion

Dans ce chapitre, nous avons présenté l'expérimentation réalisée au cours de notre travail en décrivant les différentes étapes de notre approche. Nous avons présenté le protocole d'expérimentation, l'ontologie utilisée et les étapes de l'expérimentation. La dernière section a été consacrée à une analyse des résultats du point de vue d'un utilisateur lors d'une session de recherche et du point de vue d'un expert pour la pertinence des enrichissements proposés. La validation complète du prototype nécessiterait une expérimentation avec un passage à l'échelle sur des ontologies plus complexes, cependant l'ontologie du SEMIDE (Système Euro-Méditerranéen d'Information des savoir-faire dans le domaine de l'Eau) est un bon indicateur pour montrer la pertinence de la méthode QUEXME proposée.

Conclusion et perspectives

9

Conclusion

9.1 Synthèse

Les travaux présentés dans cette thèse traitent de la recherche d'information en utilisant les technologies du Web sémantique. Ils se situent dans le contexte d'un système de coopération basé sur des ontologies OWSCIS (Ontology and Web Service based Cooperation of Information Sources). Les principales contributions de cette thèse sont les architectures et méthodologies des services d'interrogation et de visualisation basées sur la sémantique des données.

Dans ce travail de thèse, nous abordons principalement le problème de la recherche d'information en essayant de répondre aux besoins des utilisateurs et fournir des résultats pertinents tout en aidant l'utilisateur dans son processus de recherche et l'exploitation d'un grand volume de données. Pour cela nous introduisons tout d'abord dans le chapitre 2 le concept du Web sémantique et ses différents éléments. Nous donnons également les différentes définitions des ontologies ainsi que leur utilisation et plus particulièrement dans la recherche d'information.

Afin de situer notre travail, nous présentons dans le chapitre 3 un état de l'art sur les travaux qui traitent l'enrichissement de requêtes, nous avons identifié deux types d'approches : (i) les approches basées sur les requêtes des utilisateurs et (ii) les approches basées sur le contenu des ressources. Avec la masse d'informations engendrée par les systèmes d'information, un besoin d'organisation est apparu pour l'exploitation des ressources par les utilisateurs. Pour répondre à ce besoin plusieurs outils de visualisation sont apparus pour une visualisation plus pertinente des informations. Nous présentons dans le chapitre 4 une analyse et classification des outils et techniques de visualisation sémantique et plus particulièrement la visualisation d'ontologies. Cet état de l'art nous conduit à la conclusion que les travaux concernant la visualisation se limitent généralement à la visualisation d'ontologies et ne traitent pas de la recherche d'information utilisant ces technologies du Web sémantique.

Nous présentons dans le chapitre 5 une vue générale des approches et so-

lutions proposées dans le cette thèse, ainsi que l'architecture générale OWSCIS. A partir de nos besoins et des limites des travaux présentés une nouvelle approche d'enrichissement de requête est présentée dans le chapitre 6. Cette approche permet de proposer de nouveaux termes à l'utilisateur en se basant sur la notion de popularité des concepts en utilisant les usages des utilisateurs afin de sélectionner de nouveaux termes à ajouter dans la requête. Notre approche est complétée dans le chapitre 7 par la proposition de méthodes pour la visualisation des informations dans le processus de recherche d'information. Ce service a pour fonction la présentation des résultats d'une requête émise par l'utilisateur sur l'ontologie de référence.

Les approches proposées dans cette thèse ont été implémentées, nous présentons dans le chapitre 8 la partie expérimentale de l'approche d'enrichissement en décrivant la construction du jeu de tests et l'évaluation de l'approche d'enrichissement de requêtes QUEXME sur l'ontologie du projet SEMIDE.

9.2 Résumé des principales contributions

Les travaux de cette thèse sont basés sur l'exploitation sémantique d'une ontologie de référence dans une architecture générale d'un système de coopération basé sur des ontologies. Ils concernent les aspects d'interrogation et de visualisation dans l'architecture.

9.2.1 Enrichissement de requêtes

Nous avons proposé dans cette thèse une nouvelle approche d'enrichissement de requêtes. Nous pouvons résumer les différentes contributions de ce premier axe de notre travail par :

- la contribution au niveau général de l'architecture par la création de l'architecture du module d'enrichissement dans le service d'interrogation dans OWSCIS,
- l'exploitation sémantique de l'ontologie de référence dans OWSCIS,
- une nouvelle approche d'enrichissement basée sur le concept d'importance des concepts, en exploitant les usages des utilisateurs,
- la possibilité de l'application de l'approche QUEXME dans un autre domaine en se basant sur une autre ontologie de référence. Notre expérimentation a été effectuée sur une ontologie du domaine de l'eau.

9.2.2 Visualisation

Le deuxième axe de travail qui concerne l'approche de visualisation, les différentes contributions sont les suivantes :

- la contribution au niveau général de l'architecture par la création de l'architecture du service de visualisation dans OWSCIS,

- l'exploitation sémantique de l'ontologie de référence d'OWSCIS en proposant une technique d'affichage de l'ontologie,
- un outil d'aide pour la recherche de l'utilisateur en proposant un module de construction de requêtes,
- un service de visualisation composé de trois modules et un environnement de paramétrage : (i) visualisation de l'ontologie, (ii) construction de la requête et (iii) visualisation d'une requête. Le module de visualisation d'une requête se compose de trois fonctionnalités : requête, enrichissement et résultats.
- l'exploitation de deux types d'affichage : en 2D et 3D,

D'une manière plus générale, ces contributions aident les utilisateurs à retrouver et exploiter rapidement une information parmi de grandes quantités de données en fournissant les résultats pertinents. Mettre en place l'architecture OWSCIS nous a conduit à une étude des systèmes d'information et des systèmes de coopération basés sur la sémantique. Les autres aspects de l'architecture d'OWSCIS ont été développés dans les travaux de thèse de Thibault POULAIN [Pou09] et Raji GHAWI [GHA10]. Thibault Poulain s'intéresse à la modélisation de la sémantique des informations en utilisant des ontologies, avec la génération d'une méthode semi-automatique de mise en correspondance de deux ontologies (locale et de référence) et une technique d'interrogation de la coopération basée sur l'utilisation des informations sémantiques décrites dans les ontologies et utilisant les différents mappings mis en place. Raji Ghawi s'intéresse à la mise en correspondance des sources de données locales avec les ontologies locales, il a développé plusieurs outils parmi lesquels : DB2OWL avec la création d'une ontologie locale à partir d'une base de données et le mapping d'une base de données à une ontologie locale, ou X2OWL qui génère une ontologie à partir de ressources de données XML.

9.3 Perspectives

Les perspectives de nos travaux sont nombreuses et portent principalement sur trois volets.

Utilisation des relations entre les concepts L'approche d'enrichissement se base sur les usages des utilisateurs en analysant les séquences de requêtes afin de déduire la "popularité" d'un concept. Il serait intéressant d'ajouter un autre facteur comme la distance sémantique entre les concepts candidats pour l'enrichissement.

Utilisation du profil utilisateur D'autres aspects concernant l'utilisateur pourraient être utilisés pour déterminer les concepts pour l'enrichissement comme le nombre de concepts utilisés dans la requête ou en utilisant des approches de regroupements d'utilisateurs afin de prendre en compte la similarité

entre les utilisateurs [AGACTG10] comme paramètre dans le calcul de la popularité d'un concept. Un concept pouvant être "plus populaire" s'il est utilisé par un utilisateur proche.

Outil de visualisation Plusieurs améliorations concernent le service de visualisation :

- l'amélioration de l'environnement 3D en utilisant d'autres outils ou langages de représentations des éléments 3D,
- introduire la notion de poids ou popularité de concepts pour l'affichage des éléments présentés à l'utilisateur,
- construire une nouvelle requête à partir de l'environnement 3D avec le choix simple du pointeur et un clic sur les concepts choisis,
- dans les environnements 2D et 3D, faire une vue "comparative" de la visualisation de la requête enrichie par rapport à la requête initiale en ajoutant des couleurs, tailles et formes différentes et en calculant la distance sémantique entre les concepts des deux requêtes.

Annexes

A

Extrait de l'ontologie conférence

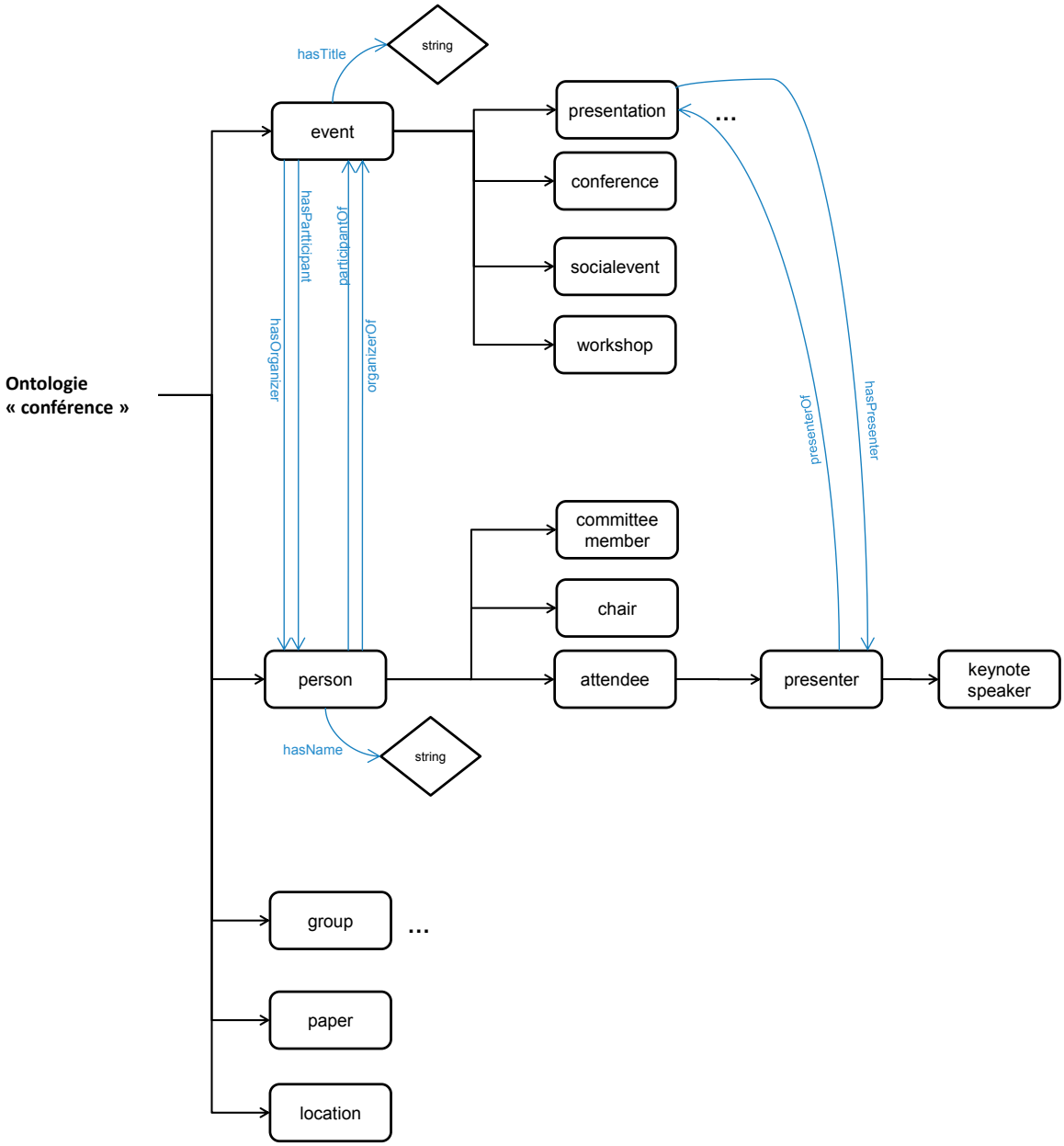

FIGURE A.1 – Extrait de l'ontologie conférence

B

Extrait de l'ontologie du SEMIDE

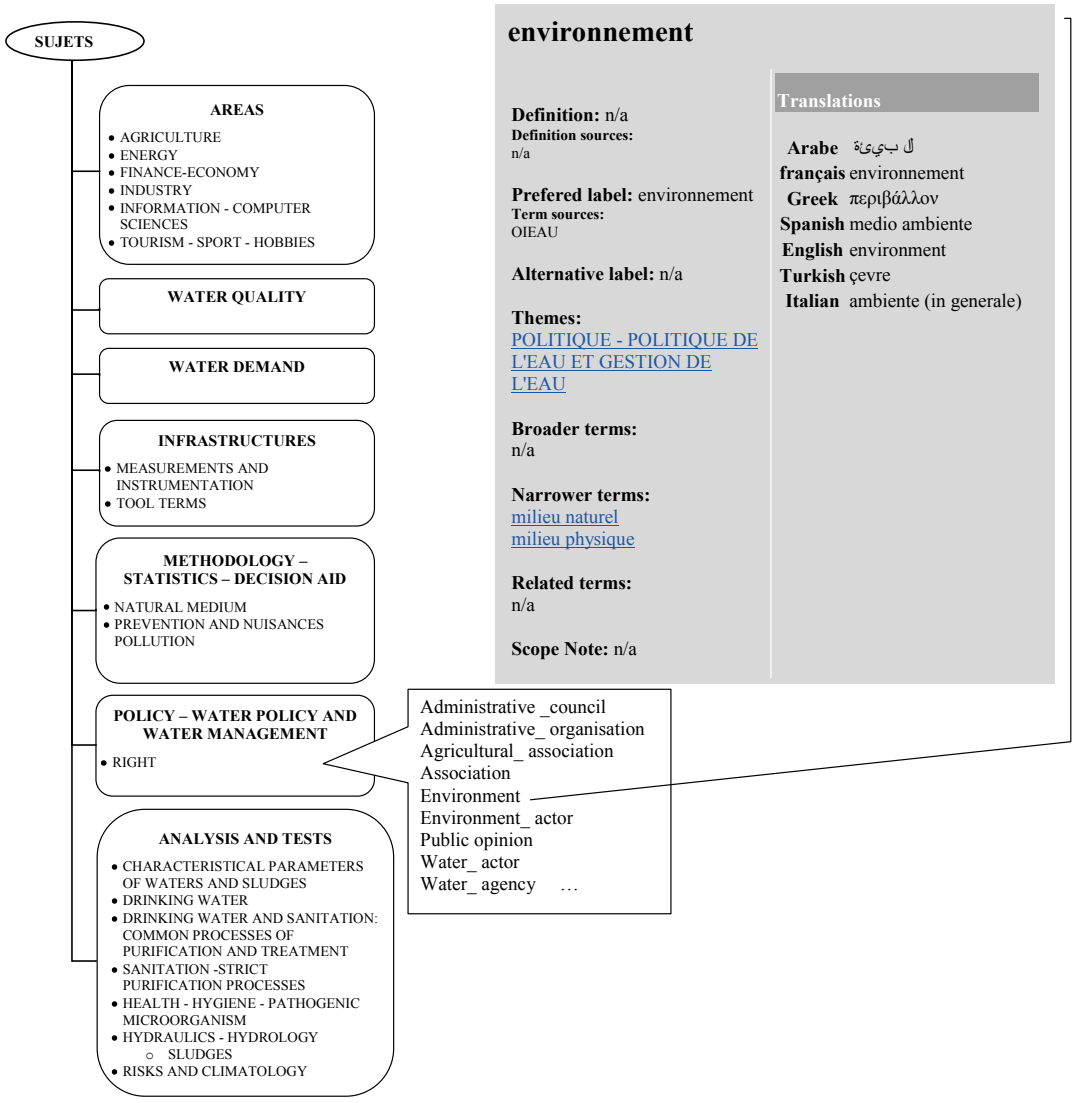

FIGURE B.1 – Extrait de l'ontologie du SEMIDE

C

Implémentation et extraits de scripts

C.1 Implémentation de l'outil de visualisation

Un outil de visualisation a été réalisé afin d'implémenter l'approche SEVI (chapitre 7) dans le cadre de l'architecture OWSCIS. Il se compose de deux parties : visualisation en 2D et une visualisation en 3D.

L'outil de visualisation a été implémenté en Java, il utilise la bibliothèque Jena [CDD+04]. Jena [11] est une bibliothèque de classes Java qui facilite le développement d'applications pour le web sémantique. Jena est utilisée au niveau de l'application pour parcourir toute une ontologie et charger au fur et à mesure les éléments dans un graphe. On utilise principalement les classes *FRLayout, CircleLayout, ISOMLayout, KKLayout et StaticLayout* afin de positionner les éléments [12].

La visualisation en 3D repose principalement sur *Java.awt.Component* afin de créer les différents composants de l'interface utilisateur et interagir avec ce dernier. La bibliothèque de programmation graphique JOGL [CW06] est utilisée [13].

11. http ://jena.sourceforge.net/

12. Le prototype de visualisation en 2D fait a été réalisé avec la participation des étudiants Simon Julienne et Bruno Lanaud du Master pro Bases de Données et Intelligence Artificielle (M2 BDIA) de l'Université de Bourgogne.

13. Le prototype de visualisation en 3D a été réalisé avec la participation de l'étudiant Miguel de Jesús López López comme pratique professionnelle de l'ingénierie de systèmes informatiques de l'Instituto Tecnológico de Nuevo León au Mexique.

C.2 extraits de scripts

Extrait du fichier de script visualisation

```
1  package ui;
2
3  import javax.media.opengl.GL;
4  import javax.media.opengl.GLAutoDrawable;
5  import javax.media.opengl.GLEventListener;
6  import javax.media.opengl.glu.GLU;
7  import javax.media.opengl.glu.GLUquadric;
8  import javax.swing.JOptionPane;
9
10 import java.text.NumberFormat;
11 import java.util.Enumeration;
12 import java.util.Vector;
13
14 import com.sun.opengl.util.GLUT;
15
16 import java.io.IOException;
17
18 import javax.swing.*;
19 import javax.swing.event.TreeExpansionEvent;
20 import javax.swing.event.TreeExpansionListener;
21 import javax.swing.event.TreeWillExpandListener;
22 import javax.swing.tree.DefaultMutableTreeNode;
23 import javax.swing.tree.DefaultTreeModel;
24 import javax.swing.tree.ExpandVetoException;
25 import javax.swing.tree.TreeNode;
26 import javax.swing.BorderFactory;
27 import javax.swing.Icon;
28 import javax.swing.ImageIcon;
29 import javax.swing.JButton;
30 import javax.swing.JCheckBox;
31 import javax.swing.JFileChooser;
32 import javax.swing.JFrame;
33 import javax.swing.JLabel;
34 import javax.swing.JMenuItem;
35 import javax.swing.JOptionPane;
36 import javax.swing.JPanel;
37 import javax.swing.JScrollPane;
38 import javax.swing.JTree;
39 import javax.swing.WindowConstants;
40 import javax.swing.tree.DefaultTreeCellRenderer;
41 import javax.swing.tree.TreeCellRenderer;
42 import java.util.Vector;
43
44 import java.awt.*;
45 import java.awt.event.*;
46 import java.io.File;
47 import java.io.FileReader;
48 import java.util.Enumeration;
49 import java.util.Iterator;
50 ////////////////////////////
51 import com.sun.opengl.util.BufferUtil;
52 import java.nio.IntBuffer;
53
54 import java.awt.geom.Rectangle2D;
```

```
55 import java.nio.ByteBuffer;
56 import java.nio.IntBuffer;
57 import java.util.ArrayList;
58 import java.util.Collections;
59 //*****************************
60 public class Renderer implements GLEventListener {
61
62         static Vector class = new Vector();
63         static Vector relations = new Vector();
64         static Vector instance = new Vector();
65         static Vector attributes= new Vector();
66         static Vector rank = new Vector();
67         static Vector concepts = new Vector();
68         //      static Vector coordinates = new Vector();
69
70         static boolean flag =false;
71
72         static float tx= 0.0f;
73         static float ty= 0.0f;
74         static float tz= 0.0f;
75
76         static JTree tree2 ;
77
78         static String nodoraiz="empty";
79         private float rotAngle = 0f;
80
81     private float xrot;                         // Rotates Cube On The X
        Axis
82     private boolean increaseX;
83     private boolean decreaseX;
84     private float yrot;                         // Rotates Cube On The Y
        Axis
85     private boolean increaseY;
86     private boolean decreaseY;
87     private float zoom = -15.0f;
88     private boolean zoomIn;
89     private boolean zoomOut;
90     private float up_down=0.0f;
91     private boolean up;
92     private boolean down;
93     private float left_right=0.0f;
94     private boolean left;
95     private boolean rigth;
96
97     private GLU glu = new GLU();
98
99     //LETRAS*************************************
100
101    private GLUT glut = new GLUT();
102    private NumberFormat format = NumberFormat.getNumberInstance();
103
104    ////////////SELECTION//////////////////////////////////
105    private int mouseX;
106    private int mouseY;
107    private int mouseZ;
108    private boolean isClicked = false;
109    ////////////////////////////////////////////////
110
111    public Renderer() {
```

```
112         format.setMinimumFractionDigits(2);
113         format.setMaximumFractionDigits(2);
114     }
115 
116     //************************************************
```

D

Classification des systèmes de visualisation

Système	Dynamique de construction	Adaptativité	Personnalisation	Mode d'interaction			
		Spécification de **l'environnement**		Manipulation directe	Menus	Formulaires	Manipulations additionnelles
Star Tree Viewer	S	Non	Ingénieur/Concepteur qui met en place le système	MD			▪ Double-clic sur les nœuds ▪ Glissement des nœuds pour les déplacer vers le centre de l'arbre
Navigational View Builder	D	Non	B	MD	-M		▪ Interaction minimale dans les menus
WebOFDAV	D	Non	B	MD	M		▪ Menu simple pour certaines des opérations
UNIVIT	D	Non	I	MD	-M		▪ Zoom ▪ Déplacement ▪ Rotation ▪ Filtrage
Natto View	D	Non	I Organisée selon les besoins des utilisateurs	MD	+ - M		▪ Certaines interactions proposées à travers des menus
WebBook y Web Forager	D	Non	B	MD	M		▪ Interaction basée sur les menus ▪ Web Book pour Web Forager
WAVE	D	SA (Présentation interactive et exploration des résultats)	B	MD			▪ Echelles conceptuelles pour expliquer la sémantique
NIRVE	D	Non	B Utilisateurs experts	MD	M		▪ Menu Windows ▪ Zoom de texte ▪ Interaction de visualisation entre 2D et 3D
ET-Map	D	Non	I	MD	+ - M	F	▪ Navigation interactive ▪ Affichage des résultats de mapping ▪ Style de navigation Pointer-et-cliquer ▪ Certaines interactions sont proposées à travers des menus ▪ Formulaires ▪ Cartes thématiques de catégories en multicouches ▪ Classification sémantique ▪ Concepts spatiaux ▪ Classification sémantique

S = Construction statique
D = Construction dynamique

NA = Pas d'adaptativité
SM = Spécification manuelle
SS = Spécification semi-automatique
SA = Spécification automatique

F = Formulaires

B = Basique
I = Intelligente

MD = Manipulation directe
- M = Utilisation minimale des menus
+ - M = Utilisation régulière des menus
M = Grande utilisation des menus

FIGURE D.1 – Tableau de comparaison récapitulatif des systèmes présentés selon les critères d'analyse retenus

Système	Dimension	Représentation graphique	Techniques conceptuelles	Mécanismes du système
Star Tree Viewer	2D	• N□□□□, □□□□□□□□□□□□□□□□□□□□et Couleurs des liens	• □□□□□□□□□□□□□□□□□□	• Java • Applet Java «fisheye »
Navigational View Builder	3D	• N□□□□□: Type, Taille, Forme , Couleur et Saturation des couleurs • Liens : Continu et Pointillée	• Classification (Clustering) • Structure hiérarchique	• Langage « C »
WebOFDAV	2D	• N□□□□□ • Couleurs	• Online Force-Directed Animated Visualization (OFDAV) • Cadre (Frames)	• OFDAV • Analyse des liens hypertexte • Basés sur les besoins de □□□□□□□□□□□□ • Sous-graphes
LIP6	2D 3D	• N□□□□□: Couleurs, Formes et Textures • Arcs : Ligne pleine et Ligne en pointillé • Cartes thématiques	• Cartes thématiques • □□□□□□□□□□□□□□□□□□□□s • Résultats en fichier XML • Arbres de cônes • Données organisées en hiérarchie • Données non organisées en hiérarchie □□□□□□-connex□□□□□	• XML (DOM) • Cartes thématiques (Topic Maps)
Natto View	3D	• N□□□□□ • Lignes	• Encodage visuel • WWW	• Techniques pour la visualisation et □interaction dynamique avec des espaces □□□□□□□□ □□□□□□ structurés en graphes • WWW
WebBook and Web Forager	2D 3D	• Livre interactive • Pages • Code de couleur	• Encodage visuel • Niveaux hiérarchiques • Arrangement hiérarchique	• Théorie □□□□□□ □□□□□□□□□ □□□□□□□□ □□□□□ • Pages Web
WAVE	3D	• Diagrammes • Objets 3D • N□□□□□: Couleur, Taille et Forme	• Encodage visuel • Arrangement hiérarchique	• Agents autonomes • Outils de reconnaissance de □□□□□□□□□ conceptuelle • Algorithme de positionnement □□□□□□ □□□
NIRVE	2D 3D	• Boîtes : Couleur, Forme et Taille • Globe ou Sphère : Icônes, Latitude, Arcs et Couleur • Rectangle : Formats Portrait ou Paysage et Code de couleur	• Classification (Clustering) • Encodage visuel	• Graphique Windows gérés par OpenGL • Menu Windows par Tcl/Tk • Xevents • Similarité des Concepts
ET-Map	2D	• Carte Niveau et Maillage • Maillage ou couches Ombre, Couleurs et Etiquettes	• Classification (Clustering) • Classification hiérarchique • Navigation	• Technique de IA Kohonen (carte auto-organisatrice) • Réseau neuronal • Outils de navigation • Navigation sur le Web • Pages Web

FIGURE D.2 – Tableau de comparaison récapitulatif des systèmes présentés selon les critères d'analyse retenus

Bibliographie

[Ado01] L. T. Adolfo. Ontologías en la web semántica. 2001.

[AGACTG10] Lylia Abrouk, David Gross-Amblard, Nadine Cullot, and Virginie Thion-Goasdoué. Tagging resources, tagging communities. In *ITNG*, pages 1253–1254, 2010.

[AKRR99] D. Abberley, D. Kirby, S. Renals, and T. Robinson. The THISL broadcast news retrieval system. In *Proc. ESCA Workshop on Accessing Information In Spoken Audio*, pages 19–24, Cambridge, 1999.

[AM04] Lylia Abrouk and Eric Mino. A framework to share water information. In *Proceedings of ICTTA04*, Damas,Syrie, 17-24 Avril 2004.

[BBB04] Jean-Christophe Bottraud, Gilles Bisson, and Marie-France Bruandet. Expansion de requêtes par apprentissage automatique dans un assistant pour la recherche d'information. In *Conference en Recherche Information et Applications, CORIA'04*, pages 89–108, 2004.

[BBP05] Alessio Bosca, Dario Bonino, and Paolo Pellegrino. P. : Ontosphere : more than a 3d ontology visualization tool. In *In : SWAP 2005, the 2nd Italian Semantic Web Workshop. CEUR Workshop Proceedings. (2005*, 2005.

[BC07a] Christopher J. O. Baher and Hei-Hoi Cheung, editors. *Semantic Web*. Springer, 2007.

[BC07b] Christopher Baker and Kei-Hoi Cheung, editors. *Knowledge discovery for biology with Taverna. Producing and consuming semantics in the Web of science. Semantic Web*, chapter 16, pages 355–395. Springer-Verlag New York Inc, 2007.

[BGLK09] Marin Bertier, Rachid Guerraoui, Vincent Leroy, and Anne-Marie Kermarrec. Toward personalized query expansion. In *SNS '09 : Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 7–12, New York, NY, USA, 2009. ACM.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5) :34–43, 2001.

[BP98] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30 :107–117, 1998.

[CDD+04] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena : implementing the semantic web recommendations. In *WWW Alt. '04 : Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83, New York, NY, USA, 2004. ACM.

[CGY07] N. Cullot, R. Ghawi, and K. Yétongnon. Db2owl : A tool for automatic database-to-ontology mapping. In Michelangelo Ceci, Donato Malerba, and Letizia Tanca, editors, *15th Italian Symposium on Advanced Database Systems (SEBD 2007)*, pages 491–494, 2007.

[CJ96] B. Chandrasekaran and J. R. Josephson. Representing Function as Effect. In *The 13th National Conference on Artificial Intelligence, AAAI-96. Workshop on Modeling and Reasoning about Function*, Portland, Oregon, August 1996.

[CJB99] B. Chandrasekaran, John R. Josephson, and V. Richard Benjamins. What are ontologies, and why do we need them ? *IEEE Intelligent Systems*, 14(1) :20–26, 1999.

[CLS00] J. Cugini, S. Laskowski, and M. Sebrechts. Design of 3d visualization of search results : Evolution and evaluation. In *IST/SPIE's 12th Annual International Symposium : Electronic Imaging 2000 : Visual Data Exploration and Analysis*, USA, 2000.

[CRY96] Stuart K. Card, G. George Robertson, and W. York. The webbook and the web forager : video use scenarios for a world-wide web information workspace. In *CHI '96 : Conference companion on Human factors in computing systems*, pages 416–417, New York, NY, USA, 1996. ACM Press.

[CS06] Jorge Cardoso and Amit Sheth. The semantic web and its applications. In *Semantic Web Services, Processes and Applications*, volume 3 of *Semantic Web and Beyond*, pages 3–33. Springer US, 2006.

[CSO96] Hsinchun Chen, Chris Schuffels, and Richard Orwig. Internet categorization and search : A self-organizing approach. *Journal of Visual Communication and Image Presentation*, 7(1) :88 – 102, March 1996.

[CW06] Jim X. Chen and Edward J. Wegman. *Foundations of 3D Graphics Programming : Using JOGL and Java3D*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[CWNM02] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *WWW'02 :*

Proceedings of the eleventh International Conference on World Wide Web, pages 325–332. ACM Press, 2002.

[CYF+97] Jaime G. Carbonell, Yiming Yang, Robert E. Frederking, Ralf D. Brown, Yibing Geng, and Danny Lee. Translingual information retrieval : A comparative evaluation. In *The 15th International joint Conference on Artificial Intelligence*, pages 708–714, 1997.

[dJLL09] Miguel de Jesus LOPEZ LOPEZ. Ingénierie de systèmes informatiques de l'Instituto Tecnologico de Nuevo Leon au Mexique, 2009. Projet de pratique professionnelle.

[DS06] Glen Dobson and Peter Sawyer. Revisiting Ontology-based requirements engineering in the age of the Semantic Web. in : Dependable Requirements Engineering of Computerised Systems at NPPs. Institute for Energy Technology (IFE), Halden, 2006.

[Eft96] Efthimis N. Efthimiadis. Query Expansion. *Annual Review of Information Science and Technology, ARIST.*, 31 :121–187, 1996.

[Fra97] Andrew Frank. Spatial ontology : A geographical information point of view. In Oliviero Stock, editor, *Spatial and Temporal Reasoning*, pages 135–153. Springer Netherlands, 1997.

[Fri09] Michael Friendly. Milestones in the history of thematic cartography, statistical graphics, and data visualization. 2009. http://datavis.ca/milestones/.

[GBJJ05] C. Ghaoui, V. Bannore, L.C. Jain, and M. Jain. *Knowledge-Based Virtual Education : User-Centred Paradigms*. Springer, 2005.

[GCAC09a] Guillermo Valente Gomez-Carpio, Lylia Abrouk, and Nadine Cullot. A Query Expansion Methodology in a Cooperation of Information Systems based on Ontologies. In *WEBIST 2009 - Proceedings of the Fifth International Conference on Web Information Systems and Technologies*, pages 256–262. INSTICC Press, March 2009.

[GCAC09b] Guillermo Valente Gomez-Carpio, Lylia Abrouk, and Nadine Cullot. QUEXME ; A Query Expansion Method Applied to Water Information System. In *SITIS 2009 - Proceedings of the Fifth International Conference on Signal Image Technology & Internet Based Systems*. ACM, December 2009.

[GCAC10] Guillermo Valente Gomez-Carpio, Lylia Abrouk, and Nadine Cullot. The Query Expansion Method "QUEXME" in an application environment. In *Journal of Multimedia Processing and Technologies.*, volume 1 of *Print ISSN : 0976-4127 Online ISSN : 0976-4135*. Digital Information Research Foundation, March 2010.

[GCU05] Zhiguo Gong, Chan Wa Cheang, and Leong Hou U. Web query expansion by WordNet. In Springer 2005. Lecture Notes in Computer Science 3588, editor, *16th International Conference on Database and Expert Systems Applications, DEXA'05*, pages 166–175, Copenhagen, DenFmark, August 22-26 2005.

[GG07] Dov Greenbaum and Mark Gerstein. Semantic web standards : Legal and social issues and implications. In Christopher J. O. Baker and Kei-Hoi Cheung, editors, *Semantic Web*, pages 413–433. Springer US, 2007.

[GHA10] Raji GHAWI. *Coopération des systèmes d'informations basée sur les ontologies*. PhD thesis, Université de Bourgogne, 2010.

[GP06] Nicolas Guelfi and Cédric Pruski. On the use of ontologies for an optimal representation and exploration of the web. *Journal of Digital Information Management*, 4 :159–168, 2006.

[GPR07] Nicolas Guelfi, Cédric Pruski, and Chantal Reynaud. Les ontologies pour la recherche ciblée d'information sur le web : une utilisation et extension d'owl pour l'expansion de requêtes. In *Actes d'IC*, pages 61–73, 2007.

[Gra01] Bénédicte Le Grand. *Extraction d'information et visualisation de systèmes complexes sémantiquement structurés*. PhD thesis, Université Pierre et Marie Curie, 2001.

[Gru93] Gruber T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) :199–220, 1993.

[Gua97] Nicola Guarino. Semantic Matching : Formal ontological distinctions for information organization, extraction, and integration. *Lecture Notes in Computer Sciencies, Technology LNAI*, 1299 :139–170, 1997.

[Gua98] Nicola Guarino. Formal ontology and information systems. pages 3–15. IOS Press, 1998.

[GW97] B. Ganter and R. Wille. Applied lattice theory : Formal concept analysis, 1997.

[H.A03] H.Alani. Tgviztab : An ontology visualisation extension for protégé. In *In Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering*, Sanibel Island, Florida, USA, 2003.

[HEC98] Mao Lin Huang, Peter Eades, and Robert F. Cohen. Webofdav-navigating and visualizing the web on-line with animated context swapping. In *WWW7 : Proceedings of the seventh international conference on World Wide Web 7*, pages 638–642, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[HF96] Carole D. Hafner and Natalya Fridman. Ontological foundations for biology knowledge models. In *4th International Conference on Intelligent Systems for Molecular Biology, ISMB*, pages 78–87. AAAI Press, 1996.

[HKJD02] Armin Hust, Stefan Klink, Markus Junker, and Andreas Dengel. Query expansion for web information retrieval. In *Informatik bewegt : Informatik 2002 - 32. Jahrestagung der Gesellschaft für Informatik e.v. (GI)*, pages 176–182. GI, 2002.

[KCC02] Stephen Kimani, Tiziana Catarci, and Isabel F. Cruz. Web rendering systems : Techniques, classification criteria and challenges. In *Visualizing the Semantic Web*, pages 63–89. 2002.

[KK88] Tomihisa Kamada and Satoru Kawai. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 41(1) :43–56, 1988.

[KN95] Robert Kent and Christian Neuss. Creating a web analysis and visualization environment. *Computer Networks and ISDN Systems*, 28, 1995.

[Kob04] A. Kobsa. User experiments with tree visualization systems. In *InfoVis 2004, IEEE Symposium on Information Visualization*, 2004.

[KTV+08] Akrivi Katifori, Elena Torou, Costas Vassilakis, Giorgos Lepouras, and Constantin Halatsis. Selected results of a comparative study of four ontology visualization methods for information retrieval tasks. In Oscar Pastor, André Flory, and Jean-Louis Cavarero, editors, *RCIS*, pages 133–140. IEEE, 2008.

[LeG02] B. LeGrand. Topic maps visualization. *XML Topic Maps - Creating and Using Topic Maps for the Web, Jack Park et Sam Hunting*, 2002.

[LJ08] Bruno Lanaud and Simon Julienne. Master pro Bases de Données et Intelligence Artificielle (M2 BDIA) de l'Université de Bourgogne, 2008. Visualisation d'ontologies.

[LJ10] Johannes Leveling and Gareth J.F. Jones. Query recovery of short user queries : on query expansion with stopwords. In *SIGIR '10 : Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 733–734, New York, NY, USA, 2010. ACM.

[LM01] Ora Lassila and Deborah Mcguinness. The role of frame-based representation on the Semantic Web. Technical Report 01, Knowledge Systems Laboratory (KSL), Stanford University, Stanford, California, 02 2001.

[LS00] B. LeGrand and M. Soto. Information management - topic maps visualization. In *XML Europe 2000*, 2000.

[LS01] B. LeGrand and M. Soto. Topic maps metrics. In *Knowledge Technologies Conference*, 2001.

[MBF+90] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to wordnet : An on-line lexical database. *International Journal of Lexicography*, 1990.

[MDNST06] Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smaïl-Tabbone. Treillis de concepts et ontologies pour interroger l'annuaire de sources de données biologiques BioRegistry. *Ingénierie des Systèmes d'Information : Systèmes d'information spécialisés, ISI'06*, 11(1) :39–60, 2006.

[Mey98] Bernd Meyer. Competitive learning of network diagram layout. In *Proc.VL'98-1998 IEEE Symposium on Visual Languages*, pages 56–63. IEEE CS Press, 1998.

[MF95] S. Mukherjea and James D. Foley. Visualizing the world-wide web with the navigational view builder. In *Proceedings of the Third International World-Wide Web conference on Technology, tools and applications*, pages 1075–1087, New York, NY, USA, 1995. Elsevier North-Holland, Inc.

[Nag06] Meenakshi Nagarajan. Semantic annotations in Web Services. In *Semantic Web Services, Processes and Applications*. Springer, 2006.

[Nao04] Dushay Naomi. Visualizing bibliographic metadata - a virtual book spine viewer. *Cornell University, National Science digital Library. D-Lib Magazine*, 10, 2004.

[NFF+91] Robert Neches, Richard Fikes, Tim Finin, Tom Gruber, Ramesh Patil, Ted Senator, and William R. Swartout. Enabling technology for knowledge sharing. *AI Magazine*, 12(3) :36–56, 1991.

[NFM00] Natalya Fridman Noy, Ray W. Fergerson, and Mark A. Musen. The knowledge model of protégé-2000 : Combining interoperability and flexibility. In *EKAW '00 : Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, pages 17–32, London, UK, 2000. Springer-Verlag.

[NH97] Natalya Fridman Noy and Carole D. Hafner. The state of the art in ontology design : A survey and comparative review. *AI Magazine*, 18(3) :53–74, 1997.

[NV03] R. Navigli and P. Velardi. An analysis of ontology-based query expansion strategies. In *14th European conference on machine learning (ECML 2003)*, 2003.

[OWL] OWL. Owl. http://www.w3.org/TR/owl-features/.

[PGS02] Domenico M. Pisanelli, Aldo Gangemi, and Geri Steve. Ontologies and Information Systems : The marriage of the century. In *Proceedings of LYEE Workshop*, 2002.

[Pou09] Thibault Poulain. *Une approche orientée sémantique pour l'interrogation d'une coopération de systèmes d'information basée sur les ontologies*. PhD thesis, Université de Bourgogne, 2009.

[QF93] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *16th Annual International Conference on Research and Development in Information Retrieval, SIGIR*, pages 160–169, New York, 1993.

[RBEB02] Christian Raymond, Patrice Bellot, and Marc El-Beze. Enrichissement de requêtes pour la recherche documentaire selon une classification non-supervisé. In *13Ëme Congrés Francophone AFRIF-AFIA de Reconnaissance des Formes et d'Intelligence Artificielle (RFIA'2002)*, pages 625 – 632, Angers, France, Janvier 2002.

[SB03] Ben Shneiderman and Benjamin B. Bederson. *The Craft of Information Visualization : Readings and Reflections*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[SCL+99] M.M. Sebrechts, J.V. Cugini, S.J. Laskowski, J. Vasilakis, and M.S. Miller. Visualization of search results : a comparative evaluation of text, 2d, and 3d interfaces. In *SIGIR '99 : Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 1999. ACM Press.

[She98] Amit P. Sheth. *Changing focus on interoperability in information systems : from system, syntax, structure to semantics*, chapter 0, pages 1–25. Interoperating Geographic Information System, 1998.

[She03] A. Sheth. Semantic Meta Data for enterprise information integration. *DM Review magazine*, July 2003.

[Sin03] Michael Sintek. OntoViz Tab : Visualizing Protege Ontologies, 2003.

[SiOM99] Hidekazu Shiozawa, Ken ichi Okada, and Yutaka Matsushita. 3d interactive visualization for inter-cell dependencies of spreadsheets. In *INFOVIS*, pages 79–, 1999.

[SM86] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[SMS+01] Margaret Anne Storey, Mark Musen, John Silva, Casey Best, Neil Ernst, Ray Fergerson, and Natasha Noy. Jambalaya : Interactive visualization to enhance ontology authoring and knowledge acquisition in protégé. In *in Protege. Workshop on Interactive Tools for Knowledge Capture (K-CAP-2001*, 2001.

[SPA] SPARQL. Sparql. http://www.w3.org/TR/rdf-sparql-query/.

[SS04] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.

[VC04] P. Sebillot V. Claveau. Extension de requêtes par lien sémantique nom-verbe acquis sur corpus. In *Traitement automatique des langues naturelles, TALN'04*, 2004.

[VHSW97] G. Van Heijst, A. Th. Schreiber, and B. J. Wielinga. Using explicit ontologies in KBS development. *International Journale Human-Computer studies*, 47(2-3) :183–292, 1997.

[Vii04] Mika Viinikkala. Ontology in Information Systems. 8109103 Ohjelmistotuotannon, 22 March 2004.

[Voo94] E. M. Voorhees. Query expansion using lexical-semantic relations. In *SIGIR '94 : Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[VW06] Olga Vectomova and Ying Wang. A study of the effect of term proximity on query expansion. *Journal of Information Science*, 32(4) :324–333, July 2006.

[WH10] Shuguang Wang and Milos Hauskrecht. Effective query expansion with the resistance distance based term similarity metric. In *SIGIR '10 : Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 715–716, New York, NY, USA, 2010. ACM.

[XFPF10] Aime X., Furst F., Kuntz P., and Trichet F. Enrichissement semantique de requêtes au moyen d'ontologies de domaine personnalisées". actes de l'atelier "personnalisation du web". In *10ièmes Journées francophones d'Extraction et de Gestion de Connaissances (EGC'2010)*, Hammamet., Tunisie, 2010.

[ZPCS06] Corinne Zayani, Andre Peninou, Marie-Francoise Canut, and Florence SËdes. An adaptation approach : query enrichment by user profile. In *The International Conference on Signal-Image Technology & Internet–Based Systems, SITIS'06, Hammamet - Tunisie, 17/12/2006-21/12/2006*, pages 24–35, http ://www.ieee.org/, 2006. IEEE.

[ZWX+07] Qi Zhou, Chong Wang, Miao Xiong, Haofen Wang, and Yong Yu. Spark : Adapting keyword query to semantic search. In *6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, SITIS'06, ISWC/ASWC'07*, pages 694–707, 2007.

résumé de la thèse :

Cette thèse présente des approches et des outils d'aide à la recherche d'information. Notre travail s'inscrit dans le cadre d'un système de coopération basé sur des ontologies appelé OWSCIS (Ontology and Web Service based Cooperation of Information Sources). Nous traitons le problème de la recherche d'information en proposant une méthode d'enrichissement appelée QUEXME (QUery EXpansion MEthod)de requêtes basée sur l'analyse du comportement des utilisateurs et utilisant la notion d'importance d'un concept par rapport à une requête. Nous avons également abordé le problème de la visualisation dans le système OWSCIS en proposant une architecture du service de visualisation, composée de trois modules : requête, enrichissement et résultats. Les approches proposées dans cette thèse ont été prototypées et l'expérimentation de la méthode QUEXME a été réalisée en utilisant la base d'information (ontologie) développée dans le Système Euro-Méditerranéen d'Information sur les savoir-faire dans le Domaine de l'Eau (SEMIDE).
mots clés : Ontologie, Enrichissement de requêtes, Visualisation, Architecture de coopération, Système d'information, Web sémantique, Recherche d'information.

abstract :

This thesis presents approaches and tools for information retrieval. Our work is part of a ooperation system based on ontologies called OWSCIS (Ontology and Web Service based Cooperation of Information Sources). We treat the problem of information retrieval by providing an enrichment method called QUEXME (QUery EXpansion MEthod) of queries based on analysis of user behavior and using the concept importance notion with regards to query. We also discussed the problem of visualization in the OWSCIS system offering a architecture of service visualization. It is composed of three modules : the request, enrichment and results. The approaches proposed in this thesis have been prototyped and the testing of the QUEXME method was performed using the information base (ontology) developed in the Euro-Mediterranean Information System on know-how in the Water sector (EMWIS).
Keywords : Ontology. Query enrichment, Visualization, Cooperation architecture, Information system, Semantic Web, Information Retrieval.