Web sémantique et réseaux sociaux - Construction d'une mémoire collective par recommandations mutuelles et représentations

Tuan Anh Ta

EDITE-ENST

Thèse présentée par : **TA Tuan Anh**

Spécialité : Informatique & Réseaux

Web sémantique et réseaux sociaux - Construction d'une mémoire collective par recommandations mutuelles et (re-)présentations

devant le jury composé de

M. Bruno Defude, Président

M. Vassilis Christophides, Rapporteur

M. Jean Charlet, Rapporteur

M. Alain Léger, Examinateur

M. Michel Scholl, Directeur de thèse

M. Jean-Marc Saglio, Co-directeur de thèse

Edition finale, juillet 2005

Remerciements

Cette thèse a été réalisée au Département Informatique & Réseaux (INFRES) de l'Ecole Nationale Supérieure des Télécommunications. Permettez-moi tout d'abord d'exprimer ma profonde gratitude à tous ceux qui ont contribué directement ou indirectement à l'aboutissement de ce travail.

Je tiens à remercier Professeur Michel SCHOLL, mon directeur de thèse, pour son soutien constant, ses précieux conseils et sa disponibilité tout au long de cette thèse.

Un grand spécial merci à Jean-Marc SAGLIO, mon codirecteur de thèse, qui m'a soutenu et encadré constamment pendant toute la durée de cette thèse, et avec lequel je me suis lié d'une amitié profonde.

J'adresse mes sincères remerciements aux rapporteurs Vassilis CHRISTOPHIDES et Jean CHARLET pour leurs commentaires extrêmement enrichissants.

Je remercie également Messieurs Bruno DEFUDE et Alain LEGER d'avoir accepté d'être membres du jury de cette thèse.

Un remerciement tout particulier s'adresse à Michel PLU, avec qui j'ai passé de bons moments de recherche dans le cadre du contrat avec FT R&D.

Un grand merci à mon ami DAO Viêt Phuong pour sa collaboration et son aide à implémenter le prototype dans cette thèse.

Je tiens à remercier tous les membres du département INFRES pour la solidarité et la compréhension qu'ils m'ont accordées.

Mes remerciements vont à mes collègues à la Faculté des Technologies d'Information de l'Institut Polytechnique de Hanoi, qui m'ont encouragé à poursuivre cette thèse.

Je ne pourrai pas oublier mes remerciements à ma grande famille, tout particulièrement ma fille Vân Nhi et sa maman émerveillée, qui avaient eu beaucoup de patiences et m'ont beaucoup manqué.

Résumé

Le Web est un vaste espace d'échange d'informations où les systèmes de recherche d'information (SRI) jouent un rôle important. Un SRI peut être simplement un moteur de recherche textuelle, mais aussi un système plus « intelligent » qui peut diriger ses recherches sur la base des connaissances associées à l'information comme dans le Web sémantique. Dans un paradigme plus social, l'intelligence humaine peut être exploitée également pour la construction d'un nouveau SRI. Les méta-informations peuvent être fournies par les utilisateurs eux-mêmes et partagées dans une sorte de « mémoire collective » pour aider à l'accès à l'information. Cette thèse présente deux approches de construction d'une telle mémoire collective par multiplication de recommandations mutuelles et par représentation de connaissances dans des parcours de découverte.

Pour la première, nous avons développé un protocole d'échange de méta-informations dans le cadre du réseau « web-of-people ». Le principe de fonctionnement de ce réseau est d'utiliser les capacités sociales des utilisateurs pour filtrer, indexer et recommander l'information grâce au développement de relations de confiance entre les personnes. Ce réseau repose sur une architecture pair à pair (P2P) dans laquelle on reprend les usages de partage et d'échange qui ont été popularisés autour des weblogs. Comme application du web sémantique, nous avons appliqué le standard RDF à l'échange de méta-informations dans un tel réseau. Notre contribution offre, d'une part, un cadre conceptuel pour le web-of-people (schéma de métadonnées, services d'interrogation et de notification) dont la cohérence et la minimalité ont été validés par un prototype et, d'autre part, quelques techniques (codage, algorithme) pour optimiser les requêtes en réseau.

La deuxième approche est un nouveau mode de présentation pour les résultats de recherche produits par un SRI. En effet, à la différence d'un SRI classique, qui répond aux questions posées par une liste plus ou moins pertinente de ressources (documents), nous proposons de répondre par un itinéraire que l'utilisateur devra parcourir pour découvrir progressivement les ressources dont la pertinence n'apparaîtra qu'à la fin du parcours. Un tel itinéraire est généré à partir de méta-informations soigneusement fournies par des experts du domaine d'application concerné. Ces méta-informations constituent une forme très élaborée de mémoire collective dans laquelle des ressources sont sélectionnées selon les thèmes conçus pour un guide de découverte.

Abstract

The Web is a broad space of information exchange where information retrieval systems (IRS) play a significant role. An IRS can be simply a textual search engine, but also a more "intelligent" system, which can lead searching for information annotated with semantics as in the Semantic Web. In a more social paradigm, the human mind can be exploited as well for constructing a new IRS. Meta-information can be provided by users themselves and be shared as a kind of "collective memory" to facilitate information access. This thesis presents two approaches to build such a collective memory by multiplication of mutual recommendations and representation of knowledge in discovery itineraries.

Following the first approach, we developed a meta-information exchange protocol within the "Web of People" network's framework. The operational foundation of this network is the usage of users social capacities to filter, index and recommend information thanks to the development of relations of confidence between people. This network relies on a peer-to-peer architecture that takes the advantage of sharing and exchanging practices, which have been popularized in weblogs. Like any application of the Semantic Web, we applied the RDF standard to meta-information exchange in such a network. Our contribution offers, on the one hand, a conceptual framework for the Web of people (metadata schema, interrogation and notification services) whose coherence and minimality were validated by a prototype and, on the other hand, a few techniques (coding, algorithm) to optimize networked queries.

The second approach is a new mode to present searching results provided by an IRS. Indeed, unlike a traditional SRI, which answers queries by a more or less relevant list of resources (documents), we propose to answer by an itinerary in which one will navigate in order to step by step discover resources whose relevance will be fully acknowledged only at the end of the navigation. Such an itinerary is generated from meta-information carefully provided by application domain experts. The provided meta-information builds a sophisticated form of collective memory in which resources are selected according to topics designed for a discovery guide.

Table des matières

Introduction 13

Partie I : Etat de l'art

1. Web sémantique et applications pour la recherche d'information 23
1.1 Web sémantique 24
1.1.1 Couche de base 25
1.1.2 RDF 26
1.1.3 Ontologie 26
1.1.4 Logique et inférence 27
1.1.5 Preuve et confiance 28
1.2 Langages de description 28
1.2.1 Resource Description Framework - RDF 28
1.2.2 Langages d'interrogation 31
1.2.3 Les Topic Maps 34
1.3 Des SRI au Web sémantique 36
1.3.1 Moteurs de recherche et SRI à base d'ontologies 37
1.3.2 Vers le Web sémantique 38
1.4 Approche de recherche d'information dans un réseau social 40
1.4.1 Bookmarks partagés 42
1.4.2 Weblogs sémantiques 43

2. Technologies P2P pour le Web sémantique 47
2.1 Principes des systèmes P2P 48
2.1.1 Objectifs opérationnels 48
2.1.2 Modèles d'architecture 50
2.2 Combinaison P2P et Web sémantique 53
2.2.1 InfoQuilt 55
2.2.2 SWAP 56
2.2.3 Edutella 58

2.2.4 RDFPeers 60
2.3 Approche de médiation décentralisée pour le web sémantique 61
2.3.1 Motivations et recherches voisines 62
2.3.2 Notre contribution 63

Partie II : Web-of-people

3. Conception d'architecture 67
3.1 Architecture de référence 68
3.1.1 Identification sociale des utilisateurs 68
3.1.2 Personal Knowledge Base (PKB) 69
3.1.3 Interrogation et notification 72
3.2 Schéma de connaissances 74
3.2.1 Post 75
3.2.2 Topique 75
3.2.3 Accessibilité 77
3.2.4 Intégration 78
3.2.5 Recommandation active et abonnement 79
3.2.6 Personnalisation 80
3.3 URI des ressources et exemple de descriptions 81
3.4 Service d'interrogation 83
3.4.1 Protocole d'accès 84
3.4.2 Authentification et restriction d'accès 85
3.5 Service de notification 87
3.5.1 Format des messages 87
3.5.2 Evènements 88

4. Prototypage 91
4.1 Architecture et implémentation du prototype 92
4.1.1 Implémentation des services 93
4.1.2 Comment déployer les services avec l'annuaire ? 95
4.1.3 Principe d'authentification 96
4.2 Application pour l'utilisateur final 97
4.2.1 Editeur weblog 97
4.2.2 Gestionnaire de messages 98
4.2.3 Interface de navigation et de recherche 99
4.2.4 Scénarios d'application 100

4.3 Conclusion et perspectives 103

5. Evaluation efficace de requêtes réseau 105
5.1 Exemple introductif 106
5.2 Modèle de requêtes réseau et évaluation 109
5.2.1 Requêtes réseau 109
5.2.2 Stratégies d'évaluation 111
5.3 Implémentation efficace de requêtes en relationnel 114
5.3.1 Schéma relationnel 115
5.3.2 Evaluation de requêtes 116
5.4 Implémentation et expérimentation préliminaire 119
5.5 Conclusion 122

Partie III : E-Parcours

6. Vers des parcours sémantiques pour la recherche d'information 125
6.1 Métaphore de gestion documentaire 126
6.1.1 De la collection de ressources à la base de descriptions 126
6.1.2 Facilitation en recherche d'information 127
6.2 Conception des parcours sémantiques 128
6.2.1 Carte de présentation 129
6.2.2 Génération des itinéraires 131
6.2.3 Exemple d'application 133
6.3 Conclusion 134

Conclusion générale 137

Références 139

Annexes

A. Vocabulaire Webop 151
B. Service Webop 155
C. Benchmark 157

Introduction

Le Web est essentiellement un espace volumineux d'information accessible par le réseau Internet. Créé par Tim Berners-Lee, le web dès les origines se construisait sur le protocole HTTP qui permet d'échanger des ressources accessibles en ligne : documents, images, fichiers multimédia, etc. Le format de document utilisé le plus fréquent sur le web est HTML. Il permet de représenter des documents hypertexte avec des balises de marquage et ainsi des hyperliens entre eux. Néanmoins, le web ne se limite pas au protocole HTTP. Il peut s'étendre avec d'autres protocoles, par exemple, FTP, SMTP,… De même, HTML n'est pas le seul format de document du web. XML est actuellement un langage standardisé pour la représentation des données et documents structurés sur le web. Il a la particularité de n'avoir aucune sémantique formelle et peut donc s'appliquer à n'importe quelle application, pour représenter l'organisation de tout type de donnée ou de document [Quint, 2003].

Au-delà d'un simple espace d'information le web permettrait aussi de connecter mieux le monde virtuel. La naissance de la technologie Peer-to-Peer (P2P [Milojicic *et al.*, 2002]) a ouvert le véritable échange pair à pair de ressources entre personnes. Le principe fondateur du P2P est la communication directe entre les membres égaux d'un réseau, auxquels il procure par surcroît la puissance répartie entre les machines interconnectées. L'explosion des applications dans le domaine de partage de fichiers telles que Napster ou Kazza a fait le succès de cette technologie.

Malgré ses acquis déjà nombreux, selon [Spivack, 2004] le web pourrait encore évoluer dans deux axes - selon son degré de connectivité d'information et celui de connectivité sociale - pour devenir le web de demain (voir Figure 1). Une évolution majeure sur le premier axe est celle du Web sémantique [Berners-Lee, 2000 ; Berners-Lee *et al.*, 2001 ; Charlet *et al.*, 2003]. Si dans le web actuel, les machines ne sont pas capables d'interpréter l'information et ne fournissent donc que des outils de localisation, de transfert, de mise en forme et de présentation, dans le web sémantique, l'information aura une signification explicite, ce qui permettra aux machines de la traiter réellement. Cette nouvelle vision permettra de transformer le web en un vaste espace d'échanges de ressources entre machines, permettant l'exploitation de grands volumes d'informations et de services variés [Laublet *et al.*, 2002]. L'objectif du web sémantique est donc de libérer les utilisateurs d'une partie du travail de recherche et d'exploitation des résultats, grâce à des capacités accrues : de recherche d'information, d'intégration de sources d'information, de découverte, d'exploitation et de combinaisons de services, de raisonnement des machines.

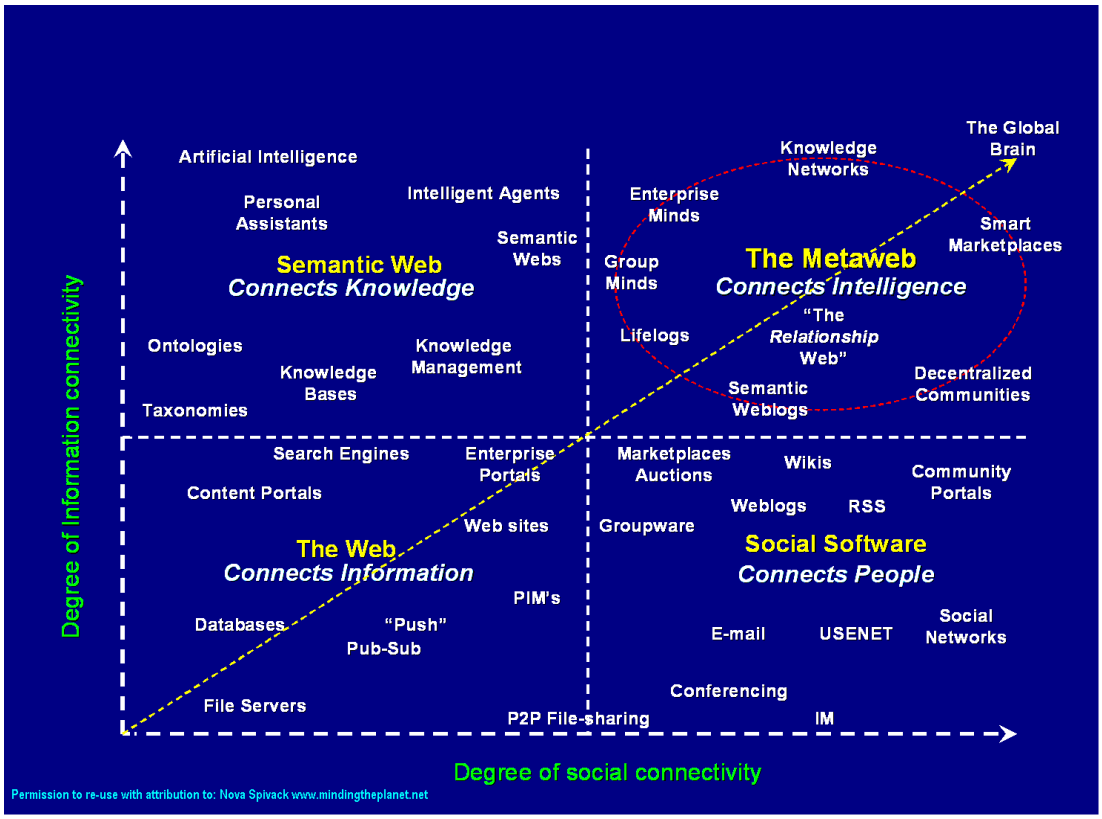


Figure 1 : Le web de demain (extrait dans [Spivack, 2004])

D'une autre manière, l'évolution du web sur le deuxième axe a pour but de devenir un moyen facile et rapide de développer les relations sociales que deux ou plusieurs individus peuvent avoir entre eux. Récemment, il est apparu un nouveau mode d'utilisation du web : le « blogging ». Comme une sorte du journal personnel, un weblog (ou blog) [Bausch *et al.*, 2002 ; Mortensen *et al.*, 2002 ; Rodzvilla, 2002] permet à son propriétaire de publier chronologiquement de brèves informations, ou notes documentaires, qui seront lues par les internautes. De plus, avec des techniques telles que le « trackback » [Trott, 2002] et la syndication [Hammersley, 2003], les weblogs permettent d'intégrer des informations provenant de divers sites et, ainsi, de rendre tangibles des liens de proximité entre les personnes qui produisent ou gèrent ces informations. Cette pratique manifeste donc l'idée d'un réseau social entre les « bloggers ».

Pour décrire la convergence du web sémantique et du réseau social dans l'évolution du web, Spivack emploie le terme Métaweb. Ce terme n'est pas neuf[1] et sa définition est loin d'être homogène. Lexicalement, un métaweb peut se comprendre comme une autre couche sur le web. Cette couche se construit naturellement à partir des métadonnées qui permettent d'accroître la probabilité d'accéder à une ressource pertinente, ainsi que d'améliorer la capacité d'un utilisateur à repérer les ressources pertinentes, parmi

[1] environ 55000 pages web trouvés pour ce terme sur Google

d'autres qui leurs sont proches. D'après Spivack le métaweb de demain pourrait être beaucoup plus intelligent. Il s'agirait d'un réseau social où les hommes et leur savoir seraient mis au centre de l'intérêt. Autrement dit, le métaweb annonce un mouvement du web centré sur l'information et les ressources numériques à un web centré sur les rapports entre celles-ci et les humains.

Recherche d'information dans un réseau social

Face à la complexification de l'accès à l'information recherchée sur le web, FT R&D a lancé un projet appelé le « web-of-people » [Plu *et al.*, 2003 ; Ta *et al.*, 2004] qui tire profit de nouvelles promesses technologiques telles que le web sémantique et le P2P afin de faciliter l'accès à l'information. Son objectif est de développer des technologies et des pratiques d'usage pour favoriser la création de réseaux d'échange de méta-informations sur des ressources accessibles à travers un réseau d'information. Le principe employé dans ce projet est d'utiliser les capacités d'individus en réseaux à filtrer, indexer et à recommander l'information au travers de réseaux de relations de confiance (d'où l'idée de réseaux sociaux). L'échange de méta-informations dans un tel réseau est caractérisé par des transactions entre les nœuds distribués. Chaque nœud correspond à un utilisateur. Il fonctionne comme un intermédiaire où des méta-informations concernant l'intérêt de l'utilisateur peuvent être stockées localement.

Le web-of-people repose sur un réseau social dans lequel les actes d'échange, de partage, de communication et de publication se rapprochent et s'organisent dans l'usage des weblogs. Ainsi, l'internaute est encouragé à communiquer à travers des « journaux individuels » publiés sur le web. Ces sites personnels se composent au fil de l'actualité, de « posts » le plus souvent enrichis de liens externes, faisant état de l'interaction de leurs auteurs avec le web (découverte de nouveaux sites, réflexions sur l'actualité internationale, publication professionnelle,…). Modélisés par métadonnées dans le cadre du web sémantique, les weblogs dans le contexte du web-of-people devraient apporter plus de sémantique que les sites personnels simples. En effet ils peuvent, d'une part, enrichir les contenus par des termes appartenant à des ontologies et, d'autre part, accroître les liens entre eux grâce à ces termes connectés.

En ce qui concerne la stratégie technologique utilisée pour l'accès à l'information, le web-of-people permet un filtrage collaboratif qui pourrait être obtenu à partir d'un réseau d'intelligence humaine. Un utilisateur pourrait exploiter les capacités cognitives d'un ensemble de personnes qu'il apprécie pour leur aptitude à identifier des informations intéressantes. Grâce à un mécanisme de notification automatique, l'utilisateur pourrait avoir conscience de nouvelles ressources que ses partenaires ont qualifiées sur un ou plusieurs sujets particuliers. A son tour, l'utilisateur partagerait aussi ce qui peut intéresser ses partenaires. Autrement dit, l'intelligence collective serait exploitée pour faciliter la recherche d'information dans le réseau web-of-people. En fait, cette idée est déjà connue dans littérature sous le nom de recommandation active [Maltz et Ehrlich, 1995]. Elle est une pratique très courante chez les utilisateurs où l'on envoie souvent des pointeurs sur des documents intéressants à des collègues ou des amis.

Toutefois, il faut pour le web-of-people une nouvelle exploitation de cette idée dans la mesure où tout envoi de recommandation peut être automatisé pour des contacts établis a priori dans le réseau social.

Ma contribution dans cette thèse est triple. Premièrement, j'ai spécifié un cadre de conception (framework) pour le web-of-people. A partir de ce cadre de conception, j'ai développé un prototype qui permet aux utilisateurs d'échanger des méta-informations dans un réseau vraiment distribué. Deuxièmement, j'ai proposé un modèle et une implémentation efficace pour des réseaux P2P qui s'appuient sur l'architecture de médiation du web-of-people. Enfin, j'ai étudié une nouvelle facilité pour aider la recherche d'information avec des parcours sémantiques. Cette facilité peut être fournie comme une fonctionnalité du système web-of-people ou des systèmes de recherche d'information en général.

Un cadre de conception pour le web-of-people

Comme une sorte de métaweb, le web-of-people combine les technologies du P2P et du web sémantique pour constituer un réseau social facilitant l'accès à l'information. La construction d'un tel réseau P2P repose sur un cadre de conception de trois piliers suivants : (i) identification sociale, (ii) interopérabilité pour l'échange, et (iii) communication par services entre pairs [Ta *et al.*, 2004].

Tout d'abord, nous voyons qu'il nécessite un système d'identification indépendant du réseau physique hébergeant les participants dans le réseau social du web-of-people. Sans savoir où se trouve un utilisateur sur le réseau, un partenaire peut créer un lien social avec lui à travers une identité sociale. Cette identité est un identifiant composé par domaine et sous-domaine pour faciliter l'identification de groupes de personnes dans le réseau social. Les domaines sociaux dans le web-of-people n'ont aucun rapport avec ceux du web actuel (le DNS). Un annuaire est mis à disposition pour permettre de consulter l'adresse d'hébergement d'un utilisateur à travers son identifiant. Bien évidemment, une telle adresse peut être changée dans l'annuaire sans que l'identité sociale de l'utilisateur change. Il peut donc migrer d'hébergement sans que les liens sociaux (liens entre personnes) deviennent obsolètes.

« Chacun chez soi » est un des principes pour les échanges dans le web-of-people. Chaque utilisateur doit pouvoir gérer son « site » de façon autonome. Il n'exige donc pas une implémentation homogène pour tous les pairs. Ce dont on a besoin est uniquement un schéma commun appliqué aux échanges de métadonnées entre pairs. Nous voyons qu'un schéma de description à la RDF (le langage de base du web sémantique [Lassila *et al.*, 2000 ; Candan *et al.*, 2001]) est nécessaire pour un tel format d'échange. L'exploitation de RDF offre quelques avantages. Premièrement, l'identification universelle des ressources par URI dans la représentation RDF permet d'unifier facilement des descriptions sur des ressources provenant de différents sites distribués. Puis, ces descriptions sont interprétables pour toute machine - une interopérabilité pour l'échange des métadonnées.

En pratique, RSS 1.0 (RDF Site Summary [RSS-DEV, 2000]) est actuellement recommandé par W3C comme format d'échange entre sites d'information. Ce format est une application du RDF qui s'applique à la syndication du contenu en ligne (articles, actualités,...) pour qu'il puisse être repris sur d'autres sites de manière totalement automatisée. RSS est aussi appliqué largement pour syndiquer des weblogs. Malheureusement, ce format n'est pas suffisamment riche pour exprimer les relations sociales dans le web-of-people. Pour cette raison, nous devons spécifier un schéma RDF particulier qui peut capturer des besoins spécifiques pour les échanges dans notre réseau social.

Deux services indépendants sont envisagés pour les échanges entre pairs dans le web-of-people. Le premier service est celui d'interrogation. Grâce à celui-ci, un pair peut interroger un autre pour obtenir des descriptions partagées par ce dernier. Ce type d'accès « on-the-fly » est de base dans tout système distribué. Néanmoins, la recommandation active dans le web-of-people réclame un mode d'échange autre que celui du service d'interrogation. Ce mode, dit « push » au contraire de « pull », permet à chaque pair de reconnaître la mise à jour de descriptions dans d'autres pairs sans besoin d'interrogation. Grâce au deuxième service, celui de notification, un pair peut informer des évènements qui nécessitent une réaction de son voisinage.

La notification peut se faire par un mécanisme de message où l'envoi et la réponse de messages fonctionnent de manière asynchrone. Ce mécanisme est très classique et il a déjà été utilisé dans les logiciels sociaux de communication tels que email, messenger,… Néanmoins, le service de notification dans le web-of-people ne sert pas à envoyer des messages apportant des contenus mais plutôt des messages apportant des évènements. Tout message doit être structuré pour apporter un évènement qui exprime un acte d'échange ou de partage de l'utilisateur à travers sa base de connaissances. Par exemple, la notification de la qualification d'une nouvelle ressource correspond effectivement à l'évènement d'ajout de nouvelles descriptions pour celle-ci dans la base de connaissances.

Comme nous l'avons vu, le principe fondateur du web-of-people est le réseau social. Pour assurer la qualité du service et stimuler la participation d'utilisateurs dans ce réseau, nous devons résoudre les trois problèmes suivants.

- *Décentralisation* : une gestion centralisée pour le web-of-people est impossible car elle peut limiter l'échelle du réseau social. La construction d'une ontologie commune partagée pour tous les membres est très difficile dans ce contexte. Le réseau social nécessite donc une approche décentralisée où chaque utilisateur doit pouvoir créer lui-même son ontologie personnelle, mais aussi pouvoir ensuite la connecter avec celles des autres, à travers des liens ontologiques.

- *Sécurité* : l'authentification est une exigence pour les échanges dans le web-of-people. Le cas où un utilisateur ne veut partager des informations qu'avec des personnes en qui il a confiance est, sans doute, bien plus fréquent que celui où il accepte de les partager avec tous, i.e., de les mettre au public. Pour cela, tout accès par service d'interrogation doit être authentifié pour identifier celui qui

demande l'accès. Nous utilisons donc l'identité sociale des utilisateurs pour effectuer le contrôle d'accès.

– *Confiance* : un réseau social ne peut que grandir lorsqu'il suscite la confiance des utilisateurs. Un utilisateur ne fait confiance naturellement qu'aux personnes qui peuvent contribuer à sa recherche d'information. Ainsi, les informations venant de celles-ci ont effectivement une certaine qualité et peuvent intéresser le récepteur. C'est pour éviter du « spam » qu'il peut filtrer des messages inappréciables venant d'autres personnes. De telles relations de confiance sont encouragées à s'établir pour le principe d'échange dans le web-of-people.

Un prototype a été réalisé pour le web-of-people en implémentant le cadre de conception général. Ce prototype permet de déployer un réseau d'échanges basé sur la technologie des services web. La base de connaissances d'un utilisateur (son weblog sémantique) est gérée en ligne sur un serveur qui fournit deux services de base, interrogation et notification, pour supporter les échanges pair à pair entre utilisateurs. Ce serveur (appelé Webop) peut héberger un ensemble d'utilisateurs et faire connecter ceux-ci avec les utilisateurs hébergés dans d'autres serveurs à travers les services web fournis par chacun. Cette approche d'hébergement est actuellement utilisée par beaucoup de logiciels sociaux (weblog, mail, forum,…) car elle n'exige pour l'internaute qu'un navigateur HTML pour participation. Toutefois, l'hébergement dans l'environnement distribué des serveurs webop est beaucoup plus flexible. Au lieu d'imposer un serveur centralisé pour tous, un réseau P2P de serveurs d'hébergement est mis en place. Un groupe de travail peut installer pour lui-même un serveur propre, mais rester toujours en contact avec d'autres personnes dans le réseau entier du web-of-people. Cette solution a tiré profit de la technologie P2P pour résoudre le passage à grande échelle.

Un réseau de médiation P2P pour le web-of-people

D'un point de vue technologique, le web-of-people est une exploitation des architectures P2P dans le cadre du web sémantique. Il constitue effectivement un réseau P2P de bases distribuées de connaissances. Chaque base du réseau peut fonctionner de manière autonome mais en gardant la connectivité pour le partage avec les autres bases. Au cours de ces années, plusieurs approches ont été proposées pour la combinaison d'une architecture P2P avec le web sémantique [Arumugam *et al.*, 2002 ; Cai et Frank, 2004 ; Ehrig *et al.*, 2003 ; Nejdl *et al.*, 2002]. Elles se distinguent essentiellement dans l'architecture et aussi le mécanisme de recherche adopté pour le réseau P2P. Dans ce contexte, le web-of-people s'appuie sur une architecture de médiation particulière qui permet de relier sémantiquement les pairs entre eux. Comme il existe une médiation de schéma entre pairs, une requête posée à un pair peut être reformulée dans la mesure où elle peut concerner plusieurs pairs reliés.

Nous avons choisi une approche de médiation P2P semblable à celle exploitée dans [Ives *et al.*, 2004 ; Rousset, 2004]. La médiation dans le web-of-people repose sur un modèle taxonomique comme celui défini dans [Tzitzika et Meghini, 2003]. La mémoire

persistante de chaque pair est une base de connaissances dans laquelle les métadonnées sont organisées par rapport aux termes d'une taxonomie. Pour cela, chacune définit localement une taxonomie qui pourra ensuite être reliée à celle d'autres bases par des liens entre termes. Des requêtes qui sont posées à un pair doivent utiliser sa taxonomie locale. Néanmoins, les réponses attendues de ces requêtes ne sont pas seulement des métadonnées locales, mais aussi celles trouvées dans d'autres pairs par inférence à partir de liens mémorisés dans la taxonomie locale. Nous estimons que le modèle de médiation que nous avons proposé est suffisant pour une classe d'applications qui demandent prioritairement simplicité, robustesse et fiabilité dans l'intégration entre pairs.

Notre contribution principale est de proposer une implémentation efficace pour un tel réseau de médiation P2P [Saglio *et al.*, 2005]. Nous montrons que les métadonnées gérées par chaque pair peuvent être stockées par un schéma qui peut s'exprimer simplement en relationnel. Ainsi, une requête réseau posée à un pair donné peut être évaluée efficacement grâce à un moteur SQL. Le processus d'évaluation de cette requête réseau nécessite un envoi de requêtes distribuées à des voisins qui sont reliés via la médiation entre pairs. Nous adoptons alors la stratégie suivante pour l'évaluation : (i) elle devrait être totalement décentralisée; aucun contrôle centralisé n'est nécessaire pour le processus d'évaluation; (ii) une requête réseau devrait pouvoir être exécutée en parallèle par tous les pairs concernés grâce à un envoi simultané de requêtes distribuées; (iii) il faudrait éliminer la plupart des redondances dans le calcul d'une requête en éliminant les termes qui peuvent être trouvés plusieurs fois à travers différents chemins dans le schéma de médiation; cette technique d'optimisation permet d'éviter des calculs inutiles dans le processus d'évaluation. Nous croyons que la stratégie d'évaluation adoptée dans cette thèse peut répondre aux exigences de robustesse et fiabilité dans un réseau à grande échelle comme le web-of-people.

Parcours sémantiques en recherche d'information

Dans la recherche d'information au sens classique, pour répondre aux requêtes d'un utilisateur, des ressources doivent être (re)trouvées à l'aide d'un moteur de recherche. Ce moteur peut s'appuyer sur des techniques purement linguistiques ou sur des techniques de sémantique formelle, à base d'ontologies. Dans ce dernier contexte, l'utilisateur est bien informé dans son domaine de connaissances pour préciser ses requêtes, et la qualité des réponses dépend certainement pour partie de la précision des requêtes soumises. C'est pour cela que ce type de recherche ne convient pas aux utilisateurs non-avertis qui souhaitent souvent découvrir d'étape en étape des ressources dans un thème plus ou moins général du domaine concerné. Nous constatons qu'une nouvelle forme de réponse où les documents s'organisent dans un parcours sémantique peut satisfaire un tel besoin d'exploration de ressources.

Dans le projet e-parcours [Picouet et Saglio, 2002 ; Ta et Saglio, 2002], nous nous sommes intéressés à une telle facilitation dans la recherche d'information. Pour une requête posée par un utilisateur, la réponse est formulée comme un itinéraire que

l'utilisateur va parcourir et qui va lui donner progressivement les éléments de réponse à sa question principale. Grâce à cette progressivité qui résulte en partie d'une démarche argumentative de lecture, l'utilisateur ne tombera jamais dans une « forêt » de ressources même si sa question est assez générale. Un itinéraire est donc une aide à navigation sur un thème choisi à l'aide d'une requête. Il suggère pratiquement un chemin de découverte, sorte de guidage de visite que nous rencontrons souvent dans les expositions [Ta, 2003].

La métaphore des parcours sémantiques nécessite une organisation de ressources en thèmes de lecture. Il s'agit d'une nouvelle couche de métadonnées que nous appelons « carte de présentation ». Chaque thème de lecture dans la carte de présentation est un « propos » (élément de base d'un discours) qui peut donner accès à un ensemble de ressources trouvées dans la base de descriptions. Un propos est mis en relation avec d'autres propos à travers des liens cognitifs tels que « prérequis de », « composé de », etc. A partir des relations entre propos, des itinéraires peuvent être générés en gardant le bon ordre de lecture entre eux.

Nous proposons, partant de cette vision, une conception des parcours sémantiques dans le cadre du web sémantique. Elle nécessite trois types de services à fournir pour supporter la navigation par itinéraires dans un système d'information : i) service de découverte de ressources en interrogeant la base de descriptions; ii) service de collaboration pour permettre aux spécialistes de rédiger des propos dans une carte de connaissances; enfin iii) service de navigation en générant des itinéraires adaptés à chaque question d'un utilisateur.

Organisation de la thèse

Cette thèse s'organise selon trois parties. Dans la première partie, nous donnons un état de l'art pour situer le contexte de cette thèse. Le chapitre 1 est une synthèse sur le web sémantique et les systèmes de recherche d'information. Cette première présentation nous permet de préciser l'approche recherche d'information dans un réseau social, et de présenter l'intérêt de l'idée du *web-of-people*. Le chapitre 2 présente les combinaisons possibles des technologies P2P et web sémantique. Une telle combinaison est exploitée pour la construction du web-of-people dans la deuxième partie.

La deuxième partie concerne notre contribution dans cette thèse liée au web-of-people. Premièrement, dans le chapitre 3, nous nous adressons au cadre conceptuel appliqué à la construction du web-of-people. Le chapitre 4 décrit ensuite le prototypage à partir de la conception générale pour le web-of-people. Enfin, le chapitre 5 traite du problème de l'évaluation efficace de requêtes réseau.

Le dernier et seul chapitre de la troisième partie est réservé à la présentation de notre contribution dans le projet e-parcours. Nous pensons avoir apporté une aide pour la recherche d'information par l'utilisation des *parcours sémantiques*.

I

Etat de l’art

Chapitre 1

Web sémantique et applications pour la recherche d'information

Proclamé prochaine *évolution du web*, le *Web sémantique* a attiré depuis ces dernières années l'attention de nombreux chercheurs. Il s'agit d'arriver à un *web intelligent*, où les informations ne seraient plus stockées mais comprises par les ordinateurs afin d'apporter à l'utilisateur ce qu'il cherche vraiment. Ce premier chapitre présente un état de l'art sur les travaux autour du web sémantique. Après un bref aperçu sur la vision du web sémantique, nous étudierons une des bases de son succès, le modèle RDF (Resource Description Framework [Lassila et Swick, 1999]). Ce modèle de métadonnées permet une certaine interopérabilité entre des applications échangeant de l'information non formalisée et non structurée sur le Web, en annotant des documents non structurés et en servant d'interface pour des applications et des documents ayant une certaine structure. RDF a été appliqué, jusqu'à aujourd'hui dans la recherche et aussi dans le domaine industriel, notamment pour la « syndication » de ressources.

D'un certain point de vue, le web sémantique est une évolution pour les systèmes de recherche d'information (SRI). Afin de donner la capacité de traitement automatique sur des documents non structurés, le web sémantique, en ajoutant aux informations existantes une couche de métadonnées, les rend exploitables par les ordinateurs. Les producteurs d'informations sont invités alors à ajouter à chaque ressource des métadonnées décrivant son contenu en respectant le modèle RDF. Ces métadonnées apporteront des sémantiques sans ambiguïté pour automatiser les traitements. Ainsi, une fois mis en place, le web sémantique enrichira l'exploration d'informations sur les moteurs de recherche.

1.1 Web sémantique

La notion de web sémantique fait référence à la vision du web de demain dans lequel les utilisateurs devraient être déchargés d'une bonne partie de leurs tâches de recherche et ainsi d'exploitation des résultats, grâce aux capacités accrues des machines à accéder aux contenus des ressources et à effectuer des raisonnements sur ceux-ci [Laublet *et al.*, 2002]. Dans cette section, nous ne donnons qu'une vision générale sur l'architecture du web sémantique. Pour une présentation détaillée, les lecteurs peuvent se rapporter à [Charlet *et al.*, 2003].

Concrètement, le web sémantique est une infrastructure qui permet l'utilisation de connaissances formalisées en plus du contenu informel que l'on peut trouver dans le web actuel. Cette infrastructure s'appuie sur un certain niveau de consensus portant, par exemple, sur les langages de représentation ou sur les ontologies utilisées. Ainsi, elle permet, le plus automatiquement possible, l'interopérabilité et les transformations entre les différents formalismes et les différentes ontologies. Grâce à la formalisation de connaissances, elle peut faciliter la mise en œuvre de calculs et de raisonnements complexes tout en offrant des garanties supérieures sur leur validité. Mais restreindre le web sémantique à cette infrastructure serait trop limitatif. Sur la base des sémantiques bien définies pour ses ressources, le web sémantique pourra fournir aux utilisateurs, par le moyen d'agents logiciels, des services automatiques et avancés [Laublet *et al.*, 2002]. Comme l'écrivent en substance [Berners-Lee *et al.*, 2001], « le web sémantique est une extension du web courant, dans lequel on donne à une information un sens bien défini pour permettre aux ordinateurs et aux personnes de travailler en coopération ».

L'architecture du web sémantique s'appuie sur une pyramide de langages proposée par Tim Berners-Lee pour représenter des connaissances sur le web en satisfaisant les critères de standardisation, interopérabilité et flexibilité. Cette architecture en couches (Figure 1.1) peut permettre une approche graduelle dans les processus de standardisation et d'acceptation par les utilisateurs. Un langage de la couche haute doit être une extension du langage de la couche au-dessous. Jusqu'à aujourd'hui seulement les couches basses sont relativement stabilisées. La liste suivante introduit la fonction principale de chaque couche dans l'architecture du web sémantique :

- XML est utilisé comme couche de base syntaxique du web sémantique. Le langage XML est actuellement considéré comme un standard de transport de données sur le web.

- La couche RDF représente les métadonnées pour les ressources web.

- La couche « ontologie », fondé sur une formalisation commune, spécifie la sémantique de métadonnées fournies dans le web sémantique.

- La couche « logique » s'appuie sur des règles d'inférence qui permettent le raisonnement intelligent exécuté par des agents logiciels.

- Les couches « preuve » et « confiance » supportent un mécanisme de communication inter-agents pour valider les résultats de raisonnement. Cela pourra assurer la fiabilité des services automatiques du web sémantique.

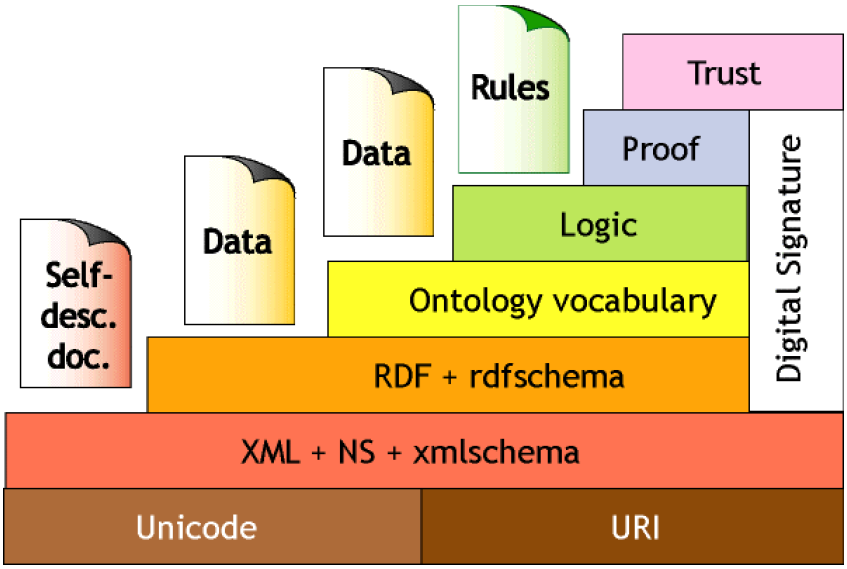


Figure 1.1 : Les couches du Web sémantique [Berners-Lee, 2000]

Reprenons et précisons les structures et fonctions de chaque couche.

1.1.1 Couche de base

Un aspect central du web est sa capacité d'identification et de localisation des diverses ressources. Elle repose sur la notion d'URI (Uniform Resource Identifier) qui permet d'attribuer un identificateur unique à un ensemble de ressources (virtuelles ou adressables). Autrement dit, l'URI est la clé pour que les choses soient identifiables dans la machine. Il en existe deux spécialisations : les URL (Uniform Resource Locations) qui supposent un système d'adressage universel, et les URN (Uniform Resource Names) qui supposent un système de nommage universel. Les ressources accessibles sur le web sont encore le plus souvent adressées par des URL.

XML (eXtensible Markup Language) est un langage de balises permettant de structurer des données et/ou des documents sur le web. Le langage XML est interopérable du point de vue syntaxique. Il sert à représenter des données échangeables sur le web. Ainsi, tous les langages du web sémantique sont systématiquement exprimables et échangeables dans une syntaxe XML. En bref, XML permet aux utilisateurs d'ajouter une structure arbitraire à leurs documents sans rien dire de la signification – de la sémantique – de ces structures [Bosak et Bray, 1999]. Notons que des documents XML peuvent être interrogés indépendamment de leur signification

suivant leur structure grâce à des standards comme XPath [Clark et DeRose, 1999] ou XQuery [Boag *et al.*, 2004].

1.1.2 RDF

RDF (Resource Description Framework [Lassila et Swick, 1999]) est une infrastructure permettant d'encoder, échanger et réutiliser des métadonnées sur le web. Cette infrastructure fournit la description du sens d'une ressource par des triplets, chaque triplet associant un sujet (la ressource), à un prédicat (la propriété descriptive) et à un objet (un littéral ou une autre ressource) dans une phrase élémentaire. On peut écrire ces triplets en utilisant des balises XML. Dans la modélisation RDF, on part du principe que des ressources (identifiées par des URI) ont plusieurs propriétés valorisées. Cette structure descriptive par triplets se révèle être une façon naturelle de décrire la plupart des données traitées par les machines. Les propriétés sont également identifiées par des URI. Elles doivent être utilisées comme des vocabulaires communs pour décrire les ressources.

Dans un langage naturel, un terme identique peut désigner des choses différentes. L'utilisation d'une URI différente pour chaque concept spécifique a pour but de résoudre le problème d'ambiguïté. Il faut alors un mécanisme qui permette de définir les vocabulaires en URI pour les utiliser dans les descriptions. C'est RDFS (RDF Schema [Brickley et Guha, 2004]) qui fournit un schéma de base permettant une telle définition de vocabulaires dans un modèle objet : des classes de ressources et des types de propriétés. Cependant, le modèle RDFS reste encore simple. Pour avoir une ontologie plus exhaustive, avec définition de classes par opérations ensemblistes (union, intersection, différence symétrique, etc.), il faut utiliser un langage de description de la couche plus haute (ontologie).

1.1.3 Ontologie

Une ontologie, en philosophie, est une théorie à propos de la nature de l'existant, des types de choses qui existent et des types de leurs liens. Les chercheurs dans le monde de l'intelligence artificielle ont adopté ce terme dans leur propre jargon, et pour eux une ontologie est la spécification explicite d'une conceptualisation partagée qui présente une vue du monde réel dans un domaine spécifique. L'usage d'ontologie s'est révélé utile dans différents domaines d'application : représentation d'informations et connaissances, intégration de systèmes d'informations, spécification de systèmes, etc. Plus concrètement, l'ontologie peut servir, selon [Uschold et Jasper, 1999], les objectifs de :

- *Communication* (entre humains et/ou organisations) : Le bénéfice d'usage d'ontologie ici est sans ambiguïté. Dans l'ontologie, il n'y a jamais deux termes ayant la même sémantique. Cette situation se produit souvent au contraire si l'on utilise un langage naturel pour la communication.

- *Interopérabilité* (machines et systèmes) : L'ontologie peut être utilisée comme un modèle intermédiaire pour la traduction entre les modélisations différentes collections d'objets. L'ontologie sert à définir le format d'échange entre les systèmes.

- *Ingénierie des systèmes* : L'ontologie peut servir divers aspects du développement des systèmes d'informations. Premièrement, elle peut assister le processus de construction de spécification de système. L'usage d'ontologie rend les documents du processus plus compréhensibles, évite l'ambiguïté dans la spécification. En outre, une représentation formelle d'ontologie permet un traitement automatique du développement. Elle soutient également l'automatisation du processus de vérification de la fiabilité de systèmes.

Dans le web sémantique, l'ontologie joue un rôle important pour faire communiquer les personnes et les machines, par échange de sémantiques et pas seulement de syntaxes. En effet, d'une part, une fois construite et acceptée par une communauté particulière, une ontologie doit traduire un certain consensus explicite et un certain niveau de partage qui sont essentiels pour permettre l'exploitation des ressources du web par différentes applications ou agents logiciels. D'autre part, la formalisation, autre facette des ontologies, est nécessaire pour que ces outils puissent être munis de capacités de raisonnement permettant de décharger les différents utilisateurs d'une partie de leur tâche d'exploitation et de combinaison des ressources du web.

Clairement, la sémantique du web est fondée sur des ontologies spécifiées explicitement dans un langage de représentation. Le W3C cherche à proposer un standard, connu actuellement sous le nom d'OWL (Ontology Web Language [Dean *et al.*, 2002]), dérivé de DAML+OIL [Horrocks *et al.*, 2001], un langage qui s'appuie sur la logique de description. Le langage OWL devrait être construit sur RDFS tout en disposant d'une syntaxe XML. Sans être exhaustif, disons au moins qu'il permet la possibilité de définir des classes de manière plus complexe correspondant aux connecteurs de la logique de description (intersection, union, restriction, etc.), des propriétés inverses ou transitives ou bien encore des restrictions de cardinalité sur les propriétés. En se fondant sur une logique de description, un tel langage a une sémantique formelle claire, ce qui permet de le doter de services inférentiels [Laublet *et al.*, 2002].

1.1.4 Logique et inférence

Afin de pouvoir résoudre toute variation de problèmes réels, le web sémantique devrait supporter un langage logique permettant des inférences plus ou moins complètes. En exploitant des connaissances disponibles sur le web sous forme de règles, les agents logiciels peuvent raisonner intelligemment et offrir des réponses automatiques à des questions posées par une personne. Cette possibilité devra s'appuyer sur la standardisation d'un langage de règles adopté pour le web sémantique. SWRL

(Semantic Web Rule Language [Horrocks *et al.*, 2003]), une combinaison d'OWL et RuleML (Rule Markup Language [Boley *et al.*, 2002]), est un tel langage.

1.1.5 Preuve et confiance

Comme le web est un espace d'information libre, la confiance d'une information trouvée n'est pas vraiment garantie dans tous les cas. Une telle confiance doit reposer sur des méthodes de qualification de l'origine de l'information, par exemple par des métadonnées et des annotations, éventuellement certifiées par des signatures électroniques. De ce fait, les (méta)données RDF pourront être chiffrées dans le web sémantique selon des techniques de signature connues [Alfred *et al.*, 1997].

De manière plus ambitieuse, la capacité de produire des preuves des déductions faites pourra augmenter le niveau de confiance des utilisateurs dans ces déductions. Un langage de preuve est simplement ce qui nous permet de prouver si un rapport est vrai. Un exemple d'un langage de preuve se composera généralement d'une liste de faits qui ont été employés pour dériver l'information en question, et la confiance pour chacun de ces faits qui ont été vérifiés.

1.2 Langages de description

Dans cette section, nous étudions le standard RDF du web sémantique pour la modélisation des métadonnées. Malgré sa représentation en XML, le modèle RDF repose sur une structure autre qu'un arbre XML. De nouveaux langages d'interrogation ont donc été développés pour la manipulation des métadonnées RDF. Enfin, il existe également une norme alternative au sein de l'ISO, les Topic Maps, pour l'annotation de ressources. Cette norme est comparable au RDF, mais le web sémantique peut s'élaborer sur l'un ou l'autre des standards acceptés au sein du W3C.

1.2.1 Resource Description Framework - RDF

D'un certain point de vue, le RDF est assez simple. Ce dernier ne consiste en effet en rien de plus qu'un moyen d'écrire d'une manière standardisée et compréhensible par les machines une assertion du type énoncé « ressource - attribut – valeur ». Un tel triplet est interprétable comme une déclaration (statement) de trois entités « sujet - prédicat – objet ». On notera que le modèle RDF n'est pas celui de la structure d'arbres d'XML même si une syntaxe XML existe. On est plutôt proche des premiers réseaux sémantiques.

Précisément, le sujet en RDF (i.e., ce sur quoi porte la déclaration) est nécessairement un objet de type ressource, entendons par là toute chose pouvant être référencée par une URI. Cela veut dire qu'une ressource peut être une page web mais

pas nécessairement, il peut s'agir également d'un objet non présent sur le web comme un livre imprimé, qui sera alors référencé par exemple par son numéro ISBN. Quant au prédicat, il doit être du type propriété. Il est lui-même identifié par URI. Chaque propriété possède une signification bien précise qui donnera la sémantique de description. Enfin, l'objet peut être une autre ressource mais aussi simplement une chaîne de caractères appelée littéral.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description rdf:about="http://www.infres.enst.fr/~saglio">
   <dc:title>Page personnelle</dc:title>
   <dc:creator>Jean-Marc Saglio</dc:creator>
</rdf:Description>
</rdf:RDF>
```

Figure 1.2. Un premier exemple de descriptions en RDF/XML

Une déclaration RDF est alors un graphe composé d'un arc (le prédicat) reliant deux nœuds (le sujet et l'objet), un ensemble de descriptions se traduisant par le graphe complexe ou la série correspondante de triplets. Par exemple, le graphe de deux triplets

```
(http://www.infres.enst.fr/~saglio, dc:creator, "Jean-Marc Saglio")
(http://www.infres.enst.fr/~saglio, dc:title, "Page personnelle")
```

exprime la description "*Jean-Marc Saglio est le créateur de la page personnelle http://www.infres.enst.fr/~saglio*", représentée en RDF/XML comme dans la Figure 1.2 (voir la syntaxe de RDF en XML dans [Lassila et Swick, 1999]).

Jusqu'à présent, on n'a dit nulle part quels étaient les attributs autorisés, à quelles ressources ils s'appliquaient, quelles étaient leurs valeurs admises... C'est le schéma RDFS qui précise tous ces points. Le schéma donne véritablement sa sémantique à la description RDF. Il permet de décrire ressources et propriétés dans un modèle objet : il existe des classes de ressources et des types de propriétés. Il est possible de définir des hiérarchies de classes et de propriétés dont l'applicabilité (i.e., domaine) et le domaine de valeurs (i.e., range) peuvent être contraintes.

En un mot, le schéma RDFS permet de définir un vocabulaire pouvant être utilisé pour décrire des ressources. On peut imaginer à loisir de nombreux vocabulaires différents, adaptés chacun à un domaine ou à une application spécifique. Notons que les vocabulaires, appelés aussi schémas de description, sont eux-mêmes écrits en RDF, en utilisant des balises de l'espace de nom RDFS (e.g., rdfs:Class, rdfs:subClassOf, rdfs:domain, rdfs:range,…). La Figure 1.3 présente un schéma de description écrit en RDF. Deux types de vocabulaire sont définis pour ce schéma. `#Personne`, `#Chercheur`, `#Doctorant` sont des classes de ressources d'un annuaire universitaire. `#Chercheur` et `#Doctorant` sont les sous classes de `#Personne`. `#nom` et `#email` sont des propriétés applicables aux ressources de la classe `#Personne` pour donner le nom et

l'adresse email de chacune. `#sousDirection` est une propriété d'association entre `#Doctorant` et `#Chercheur`.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<rdfs:Class rdf:ID="Personne"/>
<rdf:Property rdf:ID="nom">
   <rdfs:domain rdf:resource="#Personne"/>
   <rdfs:range
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="email">
   <rdfs:domain rdf:resource="#Personne"/>
   <rdfs:range
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdfs:Class rdf:ID="Doctorant">
   <rdfs:subClassOf rdf:resource="#Personne"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Chercheur">
   <rdfs:subClassOf rdf:resource="#Personne"/>
</rdfs:Class>
<rdf:Property rdf:ID="sousDirection">
   <rdfs:domain rdf:resource="#Doctorant"/>
   <rdfs:range rdf:resource="#Chercheur"/>
</rdf:Property>
</rdf:RDF>
```

Figure 1.3 : Un schéma de description défini en RDFS

Dans le modèle RDF de base, on a prévu également deux classes de ressources particulières : des réifications et des containeurs. Une réification est une ressource à propos d'une déclaration. Cette ressource doit être typée comme rdf:Statement et expliciter le triple de la déclaration impliqué par ses propriétés (rdf:subject, rdf:predicate, rdf:object).

Un container est une ressource de multiples valeurs (URI et/ou littéral). Il existe plusieurs genres de containers. Il s'agit des ensembles non ordonnés (rdf:Bag), des séquences (rdf:Seq) et des énumérations (rdf:Alt) comme étant les sous-classes. Par exemple, la description d'auteur d'une page web peut être un container qui donne une liste d'auteurs (voir Figure 1.4).

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description rdf:about="http://www.infres.enst.fr/~bd/webop">
   <dc:title>Web of people</dc:title>
   <dc:creator>
      <rdf:Bag>
         <rdf:li>Jean-Marc Saglio</rdf:li>
         <rdf:li>Tuan-Anh TA</rdf:li>
      </rdf:Bag>
   </dc:creator>
</rdf:Description>
</rdf:RDF>
```

Figure 1.4 : Description RDF avec un containeur

1.2.2 Langages d'interrogation

D'un point de vue général, on peut admettre que le web sémantique est un entrepôt virtuel et distribué de descriptions en RDF à exploiter par des applications. L'interrogation de ce type de données est devenue un objectif important des recherches autour du web sémantique. Par conséquent, différentes approches et différents langages de requête ont été proposés. Une première classe de langages de requête pour le web sémantique repose sur un modèle de triplets. Quel que soit le langage d'ontologie utilisé (RDFS, OWL,…) les connaissances sont interprétées, dans cette approche, comme un ensemble de triplets. La sémantique du langage de description se définit par les règles de déduction sur les triplets interprétés.

Supposons que l'on a créé une base d'annuaire en RDF de schéma défini dans la Figure 1.3. Le format RDF/XML de cette base est donné dans la Figure 1.5 qui peut être interprété en triplets comme ceci

```
(#saglio, rdf:type, #Chercheur)
(#saglio, #nom, "Jean-Marc Saglio")
(#saglio, #email, "saglio@enst.fr")
(#ta, rdf:type, #Doctorant)
(#ta, #nom, "Tuan-Anh TA")
(#ta, #email, "ta@enst.fr")
(#ta, #sousDirection, #saglio)
```

Pour trouver toutes les personnes dans l'annuaire, il faut chercher les ressources typées comme `#Personne` dans les triplets. Comme les règles de réduction du schéma RDFS sont appliqués sur la base de triplets, on peut y retrouver les deux nouveaux faits : `(#ta, rdf:type, #Personne)` et `(#saglio, rdf:type, #Personne)`. Ils sont impliqués à partir des triplets `(#ta, rdf:type, #Doctorant)` et `(#saglio,`

`rdf:type, #Chercheur)`, en raison que les instances de `#Doctorant` et `#Chercheur` sont aussi celles de `#Personne`.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
...
<!--Assertions pour les instances-->
<rdf:Description rdf:ID="saglio">
  <rdf:type rdf:resource="#Chercheur"/>
  <nom>Jean-Marc Saglio</nom>
  <email>saglio@enst.fr</email>
</rdf:Description>
<rdf:Description rdf:ID="ta">
  <rdf:type rdf:resource="#Doctorant"/>
  <nom>Tuan-Anh TA</nom>
  <email>ta@enst.fr</email>
  <sousDirection rdf:resource="#saglio"/>
</rdf:Description>
</rdf:RDF>
```

Figure 1.5 : Une base d'annuaire en RDF

L'approche à base de triplets a une caractéristique de simplicité, facile à implémenter. Mais la complexité des déductions peut limiter la vitesse de manipulation. Nous distinguons deux types d'inférence pour un langage de requête : i) inférence via des règles built-in qui modélisent la clôture du modèle, e.g., RDFPath [Kokkelink, 2001], Versa [Ogbuji, 2002] ; ii) moteur d'inférence avec des règles définis par les utilisateurs, e.g., SquishQL/RDQL [Miller *et al.*, 2002], TRIPLE [Sintek et Decker, 2001]. Un sommaire sur ces langages de requêtes est donné dans le Tableau 1.1. Les lecteurs peuvent se rapporter à [Magkanaraki *et al.*, 2002] pour une comparaison détaillée sur les langages de requêtes RDF.

Langage	Origine	Modèle de base	Inférence	Style du langage
SquishQL/ RDQL	ILRT/HP	Triplet	Règles	SQL
RDFPath	Kokkelink	Triplet	Built-in	XPath
Versa	Ogbuji	Triplet	Built-in	CG
TRIPLE	Standford, DKFI	Triplet	Règles	F-logic
RQL	ICS-FORTH	Graphe typé	Built-in	OQL

Tableau 1.1 : Sommaire des langages de requête RDF

De manière très différente des autres, le langage de requête RQL [Karvounarakis *et al.*, 2002] est un langage déclaratif qui permet de tirer avantage des techniques des bases de données pour pouvoir interroger efficacement de grandes collections de métadonnées RDF. Il repose formellement sur un modèle de graphe qui capture les primitives de modélisation de RDF(S) et permet l'interprétation des descriptions des ressources à travers un ou plusieurs schémas superposés. RQL adapte les fonctionnalités des langages de requêtes semi-structurés aux particularités de RDF, mais il étend également ces fonctionnalités afin de permettre l'interrogation uniforme des descriptions de ressources et de leur(s) schéma(s) associé(s).

D'un point de vue expressivité, le langage de requête RQL est puissant. Nous pouvons formuler des requêtes assez simples pour manipuler des descriptions au niveau d'instance ainsi que de schéma. Par exemple :

Q1. Tous les chercheurs et doctorants dans le département ? (on cherche les instances de la classe `Personne`)

```
Personne
```

Q2. Toutes les sous classes de `Personne` ?

```
subClassOf(Personne)
```

En utilisant des expressions de chemin, des requêtes plus ou moins complexes peuvent être facilement exprimées. Une requête RQL de style SELECT-FROM-WHERE (avec chemins) peut permettre de filtrer efficacement un sous-ensemble de données.

Q3. Les chercheurs qui encadrent au moins un doctorant ?

```
select x from Doctorant.sousDirection{x}
```

Dans cette requête, x est une variable de ressource. Le chemin Doctorant.sousDirection{x} permet de trouver toutes les instances de la classe `Doctorant`, x est l'objet de la propriété `sousDirection` pour ces instances.

Q4. Toutes les propriétés (ainsi que valeurs) utilisées pour décrire la personne identifiée par `infres.enst.fr/annuaire#saglio` ?

```
select @P, y from {x}@P{y} where x=&infres.enst.fr/annuaire#saglio
```

@P est une variable de propriété. Le chemin {x}@P{y} donne toutes les paires {x, y} pour n'importe quelle propriété.

De plus, RQL supporte une grande variété d'opérations (e.g., mathématique, ensembliste, chaîne de caractères,…) pour aider la manipulation de données. Pour ces

raisons, nous utiliserons ce langage pour montrer des exemples de requête dans un réseau de bases de connaissances (Chapitre 5).

Une différence majeure avec l'approche des autres langages est que RQL est un langage typé « à la » OQL [Alashqur *et al.*, 1989]. Il permet de distinguer clairement différents types pour les données d'entrée ainsi que de sortie d'une requête (e.g., classe, propriété, ressource, littéral,...). Grâce aux règles de typage dans le langage de requête, la sûreté des opérations, d'une part, et la validité de leurs compositions, d'autre part, peuvent être bien assurées. Par exemple, les utilisateurs ne peuvent pas formuler des requêtes avec des opérations arithmétiques (plus, minus,…) sur une classe. En outre, le typage de données permet de concevoir un meilleur stockage ainsi que des techniques d'optimisation (e.g., réécriture de requêtes) pour améliorer la performance de manipulation. Une comparaison sur la performance de RQL avec l'approche de stockage en triplets a été faite dans [Alexaki *et al.*, 2001].

1.2.3 Les Topic Maps

Une proposition concurrente à RDF(S) pour l'indexation de ressources est celle des Topic Maps standardisées par l'ISO. L'approche des Topic Maps repose sur les notions (i) de topic qui peuvent être n'importe quel sujet ou entité, (ii) d'associations qui étiquettent les relations entre topics et (iii) d'occurrences qui sont les ressources indexées par les topics. Les associations sont typées elles-mêmes comme des topics. Une syntaxe XML existe également pour les Topic Maps sous le nom XTM [Pepper et Moore, 2001].

Référence	RDF	Topic Maps	Correspondance
Chose	Ressource	Sujet	Exacte
Symbole	Nœud de ressource	Topic	Proche
Assertions	Triplets	Noms, occurrences, associations	Les assertions de topic maps sont plus complexes que celles de RDF
Identification	URI	Identificateur de sujet	Une ressource anonyme est similaire à un topique pour sujet non-adressable
Réification	rdf:Statement	Sujets de topic liés à des assertions	
Qualification	-	Scope	

Tableau 1.2 : Comparaison entre Topic Maps et RDF [Garshol, 2003]

Les deux technologies RDF et Topic Maps sont de fait compatibles. D'après [Garshol, 2003], nous pouvons trouver des solutions pour convertir des données en Topic Maps au format RDF et vice versa[2]. En principe, l'approche de RDF s'appuie sur des éléments appelés ressources tandis que les Topic Maps sont construits autour des topics. Il faut noter que le concept de ressource dans RDF n'exprime pas seulement des documents ou des données. Il peut être n'importe quelle chose identifiable.

```
...
<topic id="saglio">
    <instanceOf><topicRef xlink:href="#Chercheur"/></instanceOf>
    <baseName>
      <baseNameString>Jean-Marc Saglio</baseNameString>
    </baseName>
    <occurrence>
      <instanceOf><topicRef xlink:href="#email"/></instanceOf>
      <resourceData>saglio@enst.fr</resourceData>
    </occurrence>
    <subjectIdentity>
      <subjectIndicatorRef
        xlink:href="infres.enst.fr/annuaire#saglio"/>
    </subjectIdentity>
</topic>
...
<association>
    <instanceOf><topicRef xlink:href="#sousDirection"/></instanceOf>
    <member>
      <roleSpec><topicRef xlink:href="#professeur"/></roleSpec>
      <topicRef xlink:href="#saglio"/>
    </member>
    <member>
      <roleSpec><topicRef xlink:href="#eleve"/></roleSpec>
      <topicRef xlink:href="#ta"/>
    </member>
</association>
```

Figure 1.6 : Base d'annuaire en XTM

Une première différence entre RDF et Topic Maps se trouve dans la façon de faire des descriptions pour les topics et les ressources. Pour RDF, il y a un seul type d'assertions sous la forme de triplet « sujet - prédicat - objet ». En revanche, des assertions pour un topique impliquent ses caractéristiques : noms, occurrences et associations. En effet, des assertions pour des noms et des occurrences sont

[2] Mais il n'y a pas de correspondance 1:1 pour automatiser cette conversation.

structurellement identiques à des propriétés RDF tandis que des associations sont beaucoup plus complexes. Une association est bidirectionnelle et peut concerner de multiples topiques. Dans un graphe RDF, il existe des nœuds blancs (i.e., ressources anonymes) et des nœuds URI. Les topics avec sujet non identifiable sont exactement des nœuds blancs, et les autres correspondent à des topiques identifiés par URI. Par ailleurs, RDF et Topic Maps sont aussi différents dans l'approche de la réification et de la qualification. Le Tableau 1.2 résume les différences entre RDF et Topic Maps.

La Figure 1.6 illustre une base Topic Map équivalente à la base d'annuaire représentée en RDF. Chaque topic correspond à un nœud de ressource. Par exemple, le topic `saglio` représente une entrée dans la base d'annuaire. Il est instance d'un autre topic de type classe (`Chercheur`). Le nom du topic est une assertion particulière. D'autres valeurs de description (e.g., `email`) pour le topic sont décrites comme occurrences. Une association `sousDirection` permet de relier deux topics `ta` et `saglio`. Le sujet du topic rapporte l'URI de la ressource correspondante (i.e., `infres.enst.fr/annuaire#saglio`).

Notons que la notion de ressource dans les Topic Maps est plus restreinte que celle de RDF. Un topique peut avoir un sujet adressé par une URI (i.e., une ressource au sens de RDF). Si cette URI donne la localisation d'une ressource accessible sur le web (i.e., au sens des Topic Maps), le sujet du topic indique la ressource elle-même (`resourceRef`). Mais une URI peut être également une ressource symbolique (comme dans notre exemple ci-dessus). Dans ce cas, elle doit être spécifiée comme un indicateur de topique (`subjectIndicatorRef`). Autrement dit, si RDF considère toutes les choses comme ressources, les Topic Maps permettent de bien distinguer les ressources (documentaires) et les concepts ontologiques.

En conclusion, RDF et Topic Maps sont deux technologies qui ont beaucoup de similarités. On peut faire des conversions bidirectionnelles entre ces deux technologies. Néanmoins, une fusion entre elles est impossible. Elles peuvent exister concurremment pour développer différents types d'application.

1.3 Des SRI au Web sémantique

Le Web, grâce à sa simplicité d'édition et de consultation, a rendu l'Internet convivial et accessible à tous. Cependant malgré cet engouement qui a conduit à un développement énorme de l'offre, rien n'a été fait pour garantir qu'un document publié sera retrouvé, visible et lisible. Depuis une dizaine d'années des systèmes de recherche d'informations (SRI), nombreux et variés, destinés au grand public, ont été développés pour explorer le web. Dans cette section, nous donnons un bref aperçu de ces systèmes. Nous classons les SRI en trois générations : (i) les moteurs de recherche purement textuels, (ii) les SRI à base d'ontologies sans RDF, et (iii) ceux liés au web sémantique utilisant RDF.

1.3.1 Moteurs de recherche et SRI à base d'ontologies

De nos jours, les moteurs de recherche sont sans aucun doute la principale ressource à disposition des utilisateurs pour la recherche d'informations sur l'Internet. Grâce aux moteurs de recherche, il suffit d'écrire un ou plusieurs mots-clés concernant le sujet qui nous intéresse et en quelques secondes nous obtenons une liste de pages web qui contiennent les mots demandés. Des modèles de recherche d'information (booléen, vectoriel, probabiliste, etc.) ont été développés pour des documents textuels classiques depuis déjà plus de 30 ans [Salton et McGill, 1983]. Ces modèles ont été très étudiés dans le contexte de documents classiques : atomiques, plats et indépendants. Néanmoins, le web n'est pas simplement une collection de documents mais plutôt un réseau de documents fortement interconnectés par des liens hypertextes entre eux. Un axe de recherche prometteur consiste donc à étudier l'impact de la structure du web sur l'indexation et l'interrogation. La plupart de moteurs actuels considèrent le web comme un graphe orienté : les nœuds sont des pages HTML et les arcs sont des liens hypertextes. Ces liens permettrent d'évaluer le poids d'importance de chaque document pour un classement final de résultats (technique de « Pagerank » [Page *et al.*, 1998]).

De toute façon, les moteurs de recherche rendent souvent des centaines de documents pour chaque requête. La tâche la plus lourde revient à l'utilisateur qui doit fouiller dans cette masse de résultats pour sélectionner les documents qui lui seront les plus utiles. Les résultats ne sont pas tous pertinents et l'information retrouvée n'est pas complète. Autrement dit, la recherche plein texte n'est pas toujours efficace, ne serait-ce parce qu'il existe des variantes lexicales et des synonymes considérés comme étant des termes différents.

La problématique qui se pose est celle d'une recherche d'information intelligente où l'indexation devrait reposer sur la sémantique des ressources comme étant l'explication de structures et de concepts contenus dans les documents numériques ou qui leur sont associés. L'intérêt est d'une part d'apporter suffisamment de renseignements sur les ressources, en ajoutant des annotations sous la forme de métadonnées et d'autre part, de décrire leur contenu de manière à la fois formelle et signifiante à l'aide d'une ontologie pour être interprétables aussi bien par les humains que par les machines.

En principe, on peut distinguer au moins deux degrés dans les descriptions de ressource fondés sur les connaissances [Prié, 2000]. Il s'agit au premier degré d'éclairer un ensemble de ressources à l'aide d'une ontologie de concepts. Certains concepts sont repérés comme s'instanciant dans des documents, qu'il est alors possible de retrouver. Un monde est décrit, et ses concepts sont illustrés par des documents. Dans ce contexte, l'indexation doit suivre une phase de construction de l'ontologie du domaine d'application. Nous pourrions alors retrouver des documents sous leurs concepts associés en accédant à un portail sémantique qui nous permet de naviguer sur la carte de connaissances de la communauté. Ce portail distingue des portails d'information simples sous forme d'annuaire (e.g., dmoz) pour sa richesse et la précision de son ontologie utilisée. Par exemple, dans un tel portail sémantique nous pouvons formuler des requêtes exploitant des relations sémantiques entre termes d'une ontologie (cf. [Staab *et al.*, 2000]).

Le second degré, plus ambitieux, vise à décrire des faits relatifs aux contenus des documents, le plus souvent sous la forme d'assertions logiques. Cette approche donne plus de possibilités de finesse et de précision dans les descriptions. En se différenciant du premier degré, des descriptions par assertion permettent de déclarer de nouveaux concepts apparaissant dans les contenus des documents. C'est à dire que, les systèmes fondés sur cette approche ne supportent pas seulement l'accès intelligent aux ressources, mais aussi la recherche de connaissances. A l'inverse de la tâche d'indexation ci-dessus, l'annotation de ressources permet d'enrichir des connaissances lorsque de nouveaux concepts sont rencontrés.

Dans cette direction, Ontobroker [Decker *et al.*, 1999], WebKB [Martin et Eklund, 1999] et SHOE [Heflin et Hendler, 2000] sont les premiers parmi les systèmes à base d'ontologies appliqués à la recherche intelligente de ressources. Ces trois systèmes sont similaires par la façon d'imbriquer des connaissances dans les documents collectionnés a priori en utilisant un outil d'annotation particulier. Afin d'être exploitables et lisibles à la fois par l'homme et la machine, ces connaissances sont intégrées dans une ou plusieurs balises ad hoc dans chaque document. La différence entre ces systèmes se fait essentiellement sur le choix du langage de représentation de connaissances ainsi que sur leur modèle d'inférence sur des connaissances trouvées dans les documents annotés. Ontobroker et WebKB utilisent une représentation de connaissances connue, respectivement F-Logic [Kifer et Lausen, 1989] et Graphe Conceptuel [Sowa, 1998], tandis que SHOE propose lui-même un langage d'annotation. Comme les moteurs de recherche, la recherche d'information dans ces systèmes s'exécute également via un service centralisé. Cependant ce service utilisera des connaissances dans les pages annotées pour inférer et retrouver la réponse plus précise au besoin de l'utilisateur.

1.3.2 Vers le Web sémantique

Dans une vision de l'évolution, les systèmes présentés ci-dessus peuvent être considérés comme des prédécesseurs du web sémantique. De nombreux systèmes liés au web sémantique emploient maintenant le langage RDF pour l'annotation de ressources. Parmi eux on peut citer On2broker [Fensel *et al.*, 1999] (successeur de Ontobroker), CoMMA [Gandon *et al.*, 2002], C_Web [Amann et Michard, 2000], etc. Les lecteurs qui s'intéressent à des descriptions détaillées de ces systèmes peuvent se rapporter à [Gargantilla et Gómez-Pérez, 2004]. Le point commun de ces systèmes est de faciliter la gestion et/ou la recherche de ressources en utilisant des ontologies pour objectif de normaliser la sémantique des annotations. On construit alors des bases RDF pour stocker et manipuler des descriptions à base d'ontologies dans cet usage.

La Figure 1.7 donne une vision de l'utilisation des métadonnées sur le web sémantique. Des pages web sont annotées à partir de connaissances disponibles dans une ou plusieurs ontologies, et ces annotations, regroupées en entrepôts de métadonnées deviennent utiles pour des agents de recherche d'information, faisant ou non appel à des moteurs d'inférence permettant de déduire de nouvelles connaissances. KAON [Bozsak *et al.*, 2002], à notre connaissance, est actuellement le système le plus approprié à une

infrastructure pour le web sémantique. On y trouve des outils permettant aux utilisateurs de construire des ontologies, et d'annoter des pages web en utilisant ces ontologies. Le noyau du système est un serveur KAON considéré comme un entrepôt et un moteur d'inférence pour les connaissances à base de RDF. Supposons l'hypothèse que toutes les métadonnées sur les pages web annotées doivent être gérées par ce serveur. Nous pouvons alors développer des clients qui exploiteront ces métadonnées sur le serveur pour les besoins des utilisateurs. Pour cette raison, le serveur KAON supporte plusieurs API différentes pour la programmation : KAON-API et RDF-API. Les clients particuliers du projet KAON tels que le portail (KAONPortal) ou l'éditeur d'annotation (OntoMat) utiliseront son API spécifique (KAON-API) tandis que d'autres pourront utiliser celle du standard RDF (RDF-API). D'autre part, le serveur KAON a été conçu en prenant en compte les problèmes d'échelle, concurrence et sécurité.

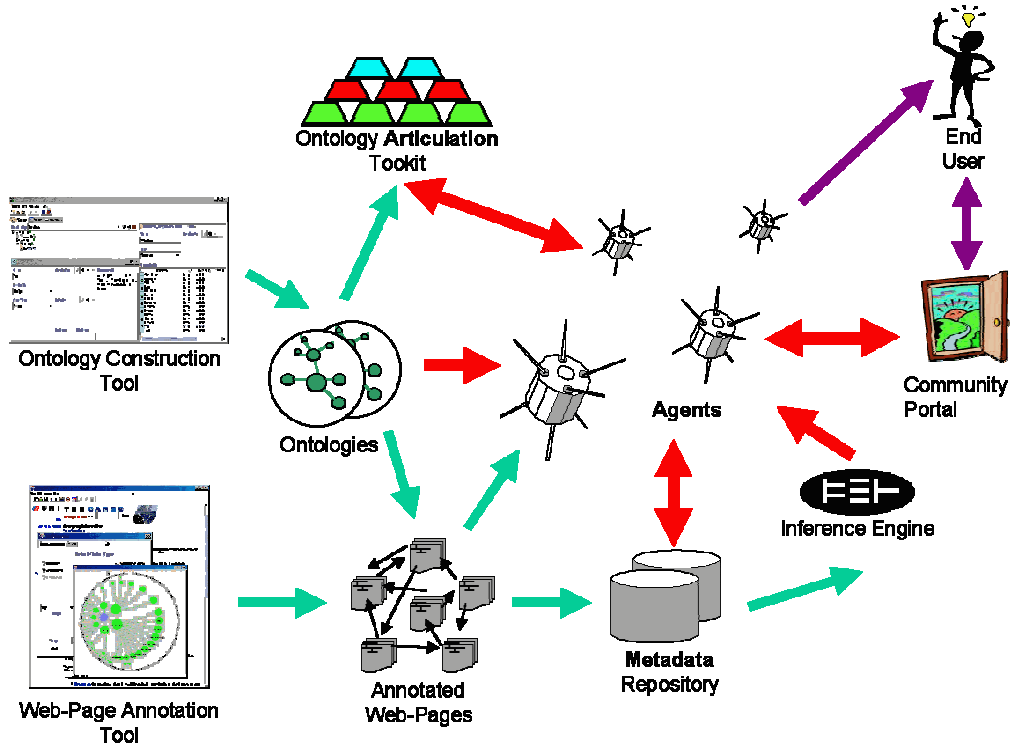


Figure 1.7 : Vision applicative du web sémantique (source semanticweb.org)

En guise de conclusion, les applications dans le cadre du web sémantique peuvent être valorisées de plusieurs façons. On peut utiliser des ontologies de façon simple pour améliorer la pertinence des recherches sur le web - le moteur de recherche peut ne rechercher que les pages faisant référence à un concept précis au lieu de celles qui utilisent des mots-clés ambigus. Mais des applications plus avancées utiliseront des ontologies plus complexes pour associer l'information d'une page web à des structures de connaissance et à des règles d'inférence. Elles peuvent aussi demander des garanties supérieures pour les traitements automatiques à base de connaissances qui se trouvent dans le web sémantique.

1.4 Approche de recherche d'information dans un réseau social

Les systèmes de recherche d'information classiques se fondent principalement sur une architecture centralisée où l'utilisateur exploite un service de recherche, publique ou communautaire, afin de trouver les informations correspondant à son intérêt. De tels services sont fournis par exemple par les portails sur l'Internet, soit pour la recherche publique soit pour le besoin d'une entreprise ou une communauté. Dans ce contexte les utilisateurs ne jouent que le rôle des consommateurs du service dans la recherche, aucune collaboration n'est nécessaire entre eux.

Dans la société de l'information à venir, le rôle humain reste important mais il le deviendra de plus en plus. Une personne ne sera plus seulement traitée comme un consommateur mais aussi comme une source de connaissances dans la mesure où elle collectionnera et qualifiera un ensemble d'informations selon son intérêt dans sa propre base de connaissances. Une telle base de connaissances pourra jouer alors le rôle d'un intermédiaire dans la recherche d'information par d'autres personnes. Ainsi, on voit apparaître une approche de recherche d'information qui exploite la collaboration entre les personnes dans un réseau social. Dans cette nouvelle approche, la recherche d'information est une pratique collective. Les utilisateurs participeront eux-mêmes au processus de valorisation d'informations dans la recherche. Cette valorisation sera transformée sous forme de connaissances individuelles et partagées avec les autres membres de la communauté. Une convergence de trois principes est envisagée dans une telle approche : *filtrage collaboratif*, *répartition de connaissances* et *centrage sur les personnes*.

Filtrage collaboratif

Le filtrage collaboratif, en principe, se base sur l'hypothèse que les gens occupés à la recherche d'information devraient pouvoir se servir de ce que d'autres ont déjà trouvé et évalué [Berrut et Denos, 2003]. Nous considérons que les appréciations/annotations des utilisateurs sur les informations trouvées sont modélisées comme des connaissances dans leur mémoire personnelle (dépendante de leur profil d'utilisateur). Pour chaque utilisateur, un ensemble de proches voisins est identifié à travers un réseau social, qui est soit calculable par un automatisme soit établi par un facteur humain. Ainsi la quantité et qualité d'informations, trouvées par une personne ayant lancé une recherche, dépendront des documents qualifiés par son voisinage.

Il existe effectivement deux approches utilisées dans les systèmes de filtrage d'informations collaboratifs. La première, *filtrage automatisé*, emploie des méthodes statistiques pour faire des prévisions basées sur des configurations des intérêts des utilisateurs. Les utilisateurs fournissent alors des évaluations des documents, sous forme

de notes, pour constituer leur profil. Ces estimations sont comparées à celles d'autres utilisateurs et les similitudes sont mesurées. Des prévisions sont calculées comme moyenne pondérée des avis d'autres utilisateurs avec des goûts soit semblables, soit complètement opposés. Ces prévisions sont ainsi exploitées pour faire des propositions à un individu sur ce qui a été apprécié par des personnes dont les goûts sont proches des siens.

L'autre motivation pour le filtrage collaboratif, appelée *recommandation active* [Maltz et Ehrlich, 1995], vient d'une pratique courante chez les utilisateurs où on envoie des pointeurs sur des documents intéressants à des collègues ou des amis. Cette fonctionnalité peut être intégrée à un système de recherche d'information et permet à ses utilisateurs d'adresser des pointeurs aux personnes qu'ils jugent intéressées. Nous suivrons cette approche pour construire un réseau d'échanges entre utilisateurs. Chaque utilisateur garde dans sa mémoire les contacts possibles avec des « proches » afin de faciliter ses échanges actifs de pointeurs. Lorsqu'un document est qualifié, un pointeur sur celui-ci est adressé automatiquement aux personnes qui sont restées en contact.

Répartition de connaissances

Opposée à la technique d'indexation dans les portails classiques qui reposent principalement sur un entrepôt centralisé de métadonnées, la recherche d'information dans un réseau social requiert une exploitation de multiples bases de connaissances. Le terme de métadonnée prend bien en compte la notion d'ajout d'information à une ressource, et on pourra a priori les utiliser indifféremment pour rendre celle-ci plus pertinente dans la recherche d'information. Chacun des utilisateurs dans le réseau social possède une base de connaissances personnelle pour stocker des métadonnées valorisées par lui-même. Une telle base de connaissances est partagée avec les autres utilisateurs sous contrôle de son propriétaire afin de faciliter l'accès collectif à l'information. Cette architecture distribuée pourrait procurer l'autonomie maximale de chacun des participants dans la communauté.

Centrage sur les personnes

Les systèmes d'information développés pour un support communautaire tels que le C_Web [Amann et Michard, 2000] reposent en pratique sur l'exploitation d'une ontologie standardisée. Cette ontologie impose de fait un schéma commun pour les connaissances fournies par différents utilisateurs. Ceci implique qu'il n'y a plus d'hétérogénéité dans l'échange de connaissances entre utilisateurs. Néanmoins, cette approche simple exige toujours des efforts importants de construction et d'entretien de l'ontologie commune parfois très complexe. De plus, elle peut apparaître encombrante et inflexible une fois appliquée à un problème personnel et spécifique. Donc au lieu de forcer l'existence d'une ontologie commune, nous suivrons l'approche dans laquelle chacun peut définir sa propre ontologie, mais en la reliant avec celles définies par d'autres à travers des « colles sémantiques ». Cette approche « bottom-up », d'une part,

semble bien adaptée à la grande échelle d'un système global comme le web, et d'autre part, met en pratique une compensation entre la flexibilité et la réutilisation dans l'ingénierie des connaissances.

Nous allons analyser maintenant les pratiques collectives dans l'usage des bookmarks et des weblogs. Sur la base de ces pratiques, des systèmes applicatifs ont été construits pour offrir un environnement collaboratif à la recherche d'information.

1.4.1 Bookmarks partagés

Avec l'émergence des « bookmarks » (marque-pages ou signets), les navigateurs actuels offrent aux utilisateurs un moyen limité pour collectionner et organiser personnellement des références, via leur URL, à des pages web choisies. Le besoin de partage de bookmarks apparaît rapidement entre utilisateurs ayant les mêmes intérêts, par email ou par un service approprié qui leur permet de s'échanger les URL de ressources intéressantes. Plusieurs formats de données (e.g., XML, HTML) peuvent être utilisés pour faciliter ces échanges de bookmarks entre utilisateurs. Dans un projet récent au sein du W3C, Annotea, on a proposé un schéma à la description RDF pour les bookmarks partagés [Koivunen *et al.*, 2003]. En effet, Annotea est un serveur RDF qui permet aux utilisateurs de se partager leurs annotations et ainsi bookmarks sur des ressources web. En utilisant un client, un utilisateur peut trouver et visualiser des bookmarks partagés sur ce serveur.

Le partage de bookmarks ne se limite pas aux serveurs centralisés, mais peut être appliqué dans un environnement pair-à-pair (P2P). WideSource[3] est un moteur de recherche P2P. Chaque utilisateur crée son annuaire de sites favoris et partage sa liste, tout ou partie, avec les autres utilisateurs du moteur de recherche. Il est donc possible à chacun de parcourir, en plus de son annuaire, les favoris d'autres utilisateurs, de visualiser la liste des utilisateurs connectés, ou encore de limiter l'accès (IP ou par nom). Traditionnellement, ce moteur propose à son utilisateur la recherche simple par mots clés ou la recherche avancée sur des critères tels que : la taille de la page, les critiques, la langue, etc. En se différenciant des moteurs de recherche classiques, ce moteur ne fait que des recherches sur les annuaires de ses utilisateurs plutôt que sur le web entier.

Le filtrage collaboratif peut être appliqué également aux bookmarks partagés. Siteseer [Rucker et Polanco, 1997] est un système de recommandation de pages web qui utilise les bookmarks personnels et leur organisation en répertoires pour prédire et recommander des pages pertinentes. Il considère chaque répertoire de bookmarks d'un utilisateur comme une déclaration implicite d'intérêt. Il consulte alors les bookmarks de chaque utilisateur et mesure le degré de chevauchement (le nombre de URL communes par exemple) de chaque répertoire avec les répertoires d'autres utilisateurs. Notons que le système ne tire aucune sémantique ni du contenu des URL ni du nom du répertoire

[3] http://www.widesource.com

dans cette mesure de chevauchement. En effet, le chevauchement permet de déterminer les similarités entre répertoires et ainsi de former dynamiquement des communautés virtuelles.

Dans tous les cas les bookmarks ont des limitations spécifiques [Berrut et Denos, 2003]. Tout d'abord, les utilisateurs ne marquent pas souvent des sites qu'ils trouvent intéressants parce qu'un site peut être accessible à travers d'autres chemins (via un hyperlien ou un moteur de recherche). Dans l'usage de bookmarks, il est difficile de reconnaître la raison pour laquelle une personne a marqué une page. C'est peut-être un véritable intérêt ou simplement un besoin de revisiter ou d'y retourner. Enfin, il n'y a pas de bookmarks partiels qui permettent d'indiquer un intérêt marginal, et il n'y a pas de moyen de montrer un manque d'intérêt du sujet, qu'un système explicite de feedback peut demander.

1.4.2 Weblogs sémantiques

Le développement des pages personnelles n'est pas neuf. Depuis la naissance du web, des chercheurs se sont saisis des pages web comme un nouveau mode d'expression, de partage d'idées voire de publication personnelle. La croissance exponentielle du nombre de sites personnels s'est accompagnée d'une multitude d'outils pour aider à leur réalisation, mais ceux-ci restaient encore difficiles à utiliser. Pour publier sur le web, il fallait apprendre le langage HTML, la synchronisation entre online et offline, etc. Désormais, avec les weblogs [Bausch *et al.*, 2002 ; Mortensen et Walker, 2002 ; Rodzvilla, 2002], l'internaute n'a besoin que d'un navigateur. Il peut publier, mettre à jour les informations dans son site personnel à partir de n'importe quel ordinateur, n'importe où dans le monde. Les weblogs permettent à tout un chacun de structurer son site, non plus sous la forme d'une collection de pages plus ou moins bien reliées entre elles, mais comme un site bien ordonné très accessible à tout système de navigation [Kaplan et Guillaud, 2003].

L'usage des weblogs n'arrête pas de s'élargir. Avec les weblogs, un nouveau web se dessine, qui n'est pas que de la technicité en moins et de la liberté en plus. C'est aussi un web ordonné différemment : là où régnait parfois le classement alphabétique ou l'ordonnancement thématique, s'impose plus volontiers le classement temporel. Quel que soit l'usage des weblogs, on les reconnaît à leurs caractéristiques spécifiques : (1) être dirigé par un individu, (2) consister en de courtes notes personnelles ou descriptions se référant à des ressources extérieures au site, (3) être mis à jour avec classement par ordre chronologique inverse, (4) être accessible pour le public, enfin (5) être archivé comme une mémoire persistante [Pepper, 2002]. On appelle « *post* » (fiche de lecture) l'élément d'écriture dans un weblog qui se doit, en principe, de rester « bref ». Un post est toujours archivé par son lien permanent (« permalink ») sur le site et classé éventuellement dans une ou plusieurs catégories s'il existe un ordonnancement thématique. On appelle « *topique* » une telle catégorie qui peut être mise sous une catégorie plus générale.

Comme les weblogs se multiplient très vite, des services de recherche autour de weblogs, fournis par les sites commerciaux, émergent sous deux types : l'un s'utilise comme un annuaire pour trouver des weblogs (e.g. bloghop.com, blogwise.com, blizg.com, blogazama.com, etc.) et l'autre comme un moteur de recherche pour trouver des posts dans les weblogs (e.g. bloogz.com, blogstreet.com, blogrunner.com, blogvision.com, etc.). De ce fait les weblogs ont fait naître une nouvelle opportunité dans le marché des moteurs de recherche.

Un aspect intéressant des weblogs est que l'auteur peut rendre disponible, via une technique de syndication comme RSS [RSS-DEV, 2000] ou Atom[4], tout ou partie du contenu de chaque article, pour une publication de la série (le « thread ») sur un autre weblog ou site web. Publication personnelle, le weblog peut devenir fortement communautaire et l'on voit se développer des communautés d'intérêt et/ou de pratique sous forme de webrings, de rencontres virtuelles, dans un souci de partage de connaissances ou de sources d'informations. C'est là l'une des caractéristiques essentielles des weblogs.

Une étude a été réalisée auprès de 177 weblogs, notamment des weblogs « professionnels » [Aimeur *et al.*, 2003]. Cette étude a fait apparaître que 75% des répondants consacrent plus de 10% de leur temps dévolu à des lectures professionnelles à la lecture des weblogs. Selon eux, certains de ces weblogs représentent les meilleures sources d'information sur un sujet particulier et diffusent des informations qui ne sont pas publiées dans d'autres médias. La lecture de ces weblogs permet ainsi de prendre contact avec des personnes dans le domaine professionnel qui partagent les mêmes centres d'intérêt.

Certes, la création et la consultation de weblogs par des professionnels pourraient ainsi être un moyen d'échanges et de rencontres entre collègues. Cela devient, sans doute, l'infrastructure du nouveau paradigme présenté au-dessus pour la recherche d'information. Paquet et Pearson ont proposé de construire un système d'échanges de posts entre utilisateurs à travers des canaux thématiques (appelés topiques) [Paquet et Pearson, 2004]. Lorsqu'un utilisateur publie un post sur son weblog, il doit également publier celui-ci sur le canal commun de son topique. En utilisant le mécanisme de Trackback (cf. [Trott, 2002]), après un « ping » sur le canal d'échange, le lien permanent de ce post peut apparaître, et donc être visible pour les autres utilisateurs qui découvrent le canal après cette publication. Cette approche exige donc une gestion centralisée des canaux d'échange. L'utilisateur ne peut découvrir que les posts inscrits dans les canaux existants.

FriendBlog[5] est un système fonctionnant en ligne comme étant un hébergeur de weblogs. En se différenciant par l'approche de canaux thématiques, FriendBlog permet de créer des canaux d'échanges directs entre amis. Un utilisateur peut inviter un autre à établir une relation afin que des posts publiés par son correspondant puissent apparaître

[4] http://www.atomenabled.org/

[5] http://www.funchain.com

sur son canal. Ainsi, il peut avoir toujours conscience de ce que celui-ci publie et vice versa. Une personne peut créer plusieurs canaux ; il y aura pour chacun un groupe d'invités. De tels liens entre personnes créeront un réseau social dans l'usage des weblogs.

Nous avons envisagé dans cette thèse un système d'échange basé sur l'usage des weblogs pour le web-of-people. Il devait reposer sur une architecture pouvant satisfaire les trois principes suivants visés pour la recherche d'information dans un réseau social. Premièrement, il n'existe pas de système commun de topiques d'échange comme celui proposé par Paquet et Pearson. Ainsi chaque utilisateur gèrera lui-même des topiques personnels pour classer/indexer ses posts. Mais il leur sera permis de lier leurs propres topiques à des topiques « étrangers » sémantiquement ressemblants. De ce fait, un utilisateur pourra lire des posts selon son intérêt à travers ces liens. Deuxièmement, le web-of-people s'appuie sur un réseau distribué de connaissances, non centralisé comme FriendBlog. Chaque utilisateur sera un nœud dans ce réseau. Une architecture P2P appliquée pour ce réseau permettra des communications directes entre utilisateurs. Enfin, un mécanisme de notification est disponible pour permettre des échanges actifs dans le web-of-people. Grâce à ce mécanisme, un utilisateur pourra notifier à ses amis des références de ressources en créant simplement chez lui de nouveaux posts sous les topiques appropriés. Cette pratique de recommandation active constituera une véritable méthode de filtrage collaboratif pour l'accès à l'information dans le web-of-people.

Le web-of-people pourra être déployé comme une application du web sémantique. Nous exploiterons un schéma commun de description RDF pour les échanges de connaissances entre utilisateurs. L'application de la représentation RDF au schéma d'échange de connaissances offrira quelques avantages : (i) assurer l'interopérabilité pour tout échange, (ii) s'adapter à la nature distribuée de connaissances dans un réseau P2P, et (iii) faciliter l'intégration du système avec d'autres dans le cadre du web sémantique.

Chapitre 2

Technologies P2P pour le Web sémantique

Les réseaux Pair-à-Pair (Peer-to-Peer en anglais - P2P) apparaissent actuellement comme un moyen populaire de communiquer et de partager facilement de l'information - des fichiers le plus souvent, mais également des calculs, des données plus structurées. Les technologies P2P se sont d'ailleurs montrées très efficaces dans une certaine philosophie de partage dans l'intérêt de chacun avec ou sans esprit communautaire. Elles permettent à différents participants (individus, organisations) de maintenir leurs propres ressources en les échangeant avec les autres participants dans un environnement vraiment distribué. Cependant, la plupart des systèmes P2P pour l'usage du grand public d'aujourd'hui s'appuient principalement sur la technique de recherche par mot-clés. C'est cette technique simple qui empêche l'exploitation efficace de tels réseaux P2P car il manque encore des descriptions sémantiques pour les ressources partagées.

Certes, la promesse technologique du web sémantique peut aider à régler cette limitation dans les réseaux P2P actuels. En exploitant des connaissances telles que des descriptions RDF, les recherches pair-à-pair pourront rendre des résultats plus pertinents pour les participants. Ainsi, une combinaison du P2P et du web sémantique émerge dans le contexte de nouvelles applications qui nécessitent la nature distribuée de la gestion de connaissances.

Dans ce chapitre, nous présentons un état de l'art des systèmes P2P en nous intéressant particulièrement au déploiement des réseaux P2P dans le contexte du web sémantique. Après un bref aperçu sur les principes des systèmes P2P, nous étudions quatre projets récents [Arumugam *et al.*, 2002 ; Cai et Frank, 2004 ; Ehrig *et al.*, 2003 ; Nejdl *et al.*, 2002] qui exploitent les technologies du web sémantique dans la construction des réseaux P2P. Enfin, nous soulignons l'approche de médiation à base d'architecture P2P pour le web sémantique [Ives *et al.*, 2004 ; Rousset, 2004].

2.1 Principes des systèmes P2P

Les technologies P2P sont aujourd'hui de plus en plus utilisées car elles constituent une méthode souple et efficace de partage de ressources entre membres d'une même communauté. Par ailleurs, en plus des utilisations grand public les plus connues (partage de fichiers et messagerie instantanée), ces technologies peuvent résoudre un ensemble de problèmes liés à l'utilisation optimale des ressources disponibles dans les réseaux et les entités constituant ceux-ci. De nombreux systèmes P2P ont été développés et avec plus ou moins de succès pour l'usage du grand public tels que Gnutella, Kazza, Avaki, Groove, FreeNet, SETI@Home, JXTA, .NET Services,… (cf. [Milojicic *et al.*, 2002]).

2.1.1 Objectifs opérationnels

Client-serveur et P2P sont effectivement deux modèles de base des systèmes distribués. Dans le modèle client-serveur, un réseau est constitué par plusieurs machines parmi lesquelles il existe au moins un serveur centralisé mis au service de ses clients. Chaque client exploite des données ou des services mis à disposition par le serveur. En se différenciant du modèle purement client-serveur, dans un réseau P2P les machines jouent à la fois le rôle d'un client (consommateur de services) et celui d'un serveur (fournisseur de services). Ceci constitue donc une égalité entre ces machines – les pairs du réseau.

Les technologies P2P peuvent s'appliquer à un grand nombre de domaines d'application. Elles permettent effectivement de résoudre un ou plusieurs des objectifs suivants :

- **Partage de coût**. A l'opposé d'un système client-serveur qui répartit le coût du système sur un seul ou plusieurs serveurs, un système P2P partage ce coût entre tous les clients (pairs). Par exemple, dans un réseau de partage de fichiers, nous pouvons télécharger des fichiers à partir de n'importe quel pair. Alors, il n'existe plus la notion du fournisseur unique dans ce réseau.

- **Agrégation de ressources.** Les technologies P2P peuvent aider à créer une grille informatique qui permet de bénéficier des ressources de différentes machines pour réaliser un but commun. Une telle ressource peut être la puissance de calcul ou l'espace de stockage libre sur une machine.

- **Grande échelle**. L'absence d'une autorité centrale forte dans un réseau P2P pourrait lui permettre de se développer à grande échelle. Nous verrons plus loin que l'échelle d'un tel réseau P2P dépend de fait du choix d'un modèle d'architecture et aussi d'un algorithme de routage appliqué sur ce réseau.

- **Autonomie**. Cet objectif assume que le fonctionnement d'un pair ne repose sur aucun fournisseur de services centralisés. Tout travail d'un utilisateur doit pouvoir s'effectuer localement dans le pair ou avec un réseau partiel de pairs.

- **Anonymat/intimité**. Lié à l'autonomie des pairs, l'architecture P2P peut permettre l'anonymat dans les activités de chaque pair. Elle pourrait garantir que le demandeur original d'un service peut ne pas être « chenillé » (même son adresse IP).

- **Dynamisme**. L'architecture P2P pourrait supporter un environnement fortement dynamique où un pair peut dynamiquement rejoindre ou quitter le système sans bloquer le fonctionnement du réseau.

Selon l'étude dans [Milojicic *et al.*, 2002], les systèmes P2P connus pour l'usage du grand public aujourd'hui peuvent être classés dans 4 catégories majeures : calcul distribué (e.g., SETI@home, Avaki, Entropia), partage de fichiers (e.g., Napster, Gnutella, Freenet, Publius, Free Haven), communication/collaboration (e.g., Magi, Groove, Jabber), et plate-forme (e.g., JXTA and .NET My Services). La Figure 2.1 montre que certains systèmes soulignent l'une des trois dimensions (le calcul, le partage ou la communication), tandis que les plate-formes soutiennent toutes ces dimensions.

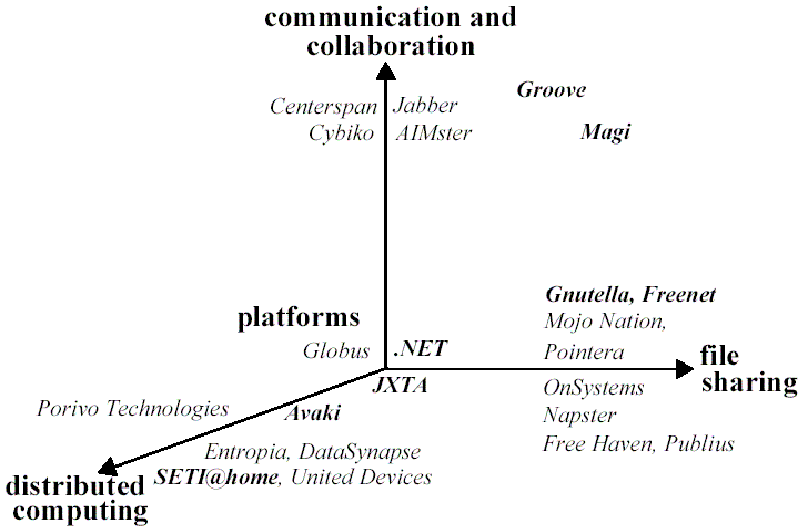


Figure 2.1 : Une classification des systèmes P2P [Milojicic *et al.*, 2002]

Dans la première dimension, les applications P2P reposent sur l'idée d'un calcul distribué où une grande tâche peut se diviser en plus petits morceaux pour s'exécuter en parallèle dans un certain nombre de pairs indépendants. Il s'agit donc d'une grille de calcul qui tire profit des ressources disponibles de beaucoup d'ordinateurs connectés à l'Internet pour résoudre des problèmes extrêmement difficiles.

Dans la deuxième dimension, les applications se concentrent effectivement sur une gestion de contenu dans l'environnement P2P. C'est l'objectif qui a motivé la naissance de la technologie P2P dans les systèmes de partage de fichiers tels que Napster, Gnutella, … Ils permettent pratiquement à un utilisateur de rechercher et de télécharger des fichiers que d'autres utilisateurs ont rendu disponibles. Cependant, les applications

ne se limitent pas au partage de fichiers. Une autre exploitation intéressante de la technologie P2P est de créer une grille de données dans laquelle les machines se connectent et offrent leur espace de mémoire inutilisé au stockage collectif des données.

Dans la dernière dimension, les applications P2P permettent à des utilisateurs de communiquer et collaborer, en temps réel, sans compter sur un serveur centralisé pour rassembler et transmettre de l'information. La transmission de messages instantanée (instant messaging en anglais) est une sous-classe de telles applications. De même, il apparaît également des applications comme les jeux collaboratifs qui permettent à des utilisateurs d'agir l'un sur l'autre à travers un réseau P2P.

2.1.2 Modèles d'architecture

Selon le modèle d'architecture appliqué aux systèmes P2P, nous avons quatre types de réseau P2P : service centralisé, service décentralisé, superpair (superpeer) et localisation par hachage (voir Figure 2.2).

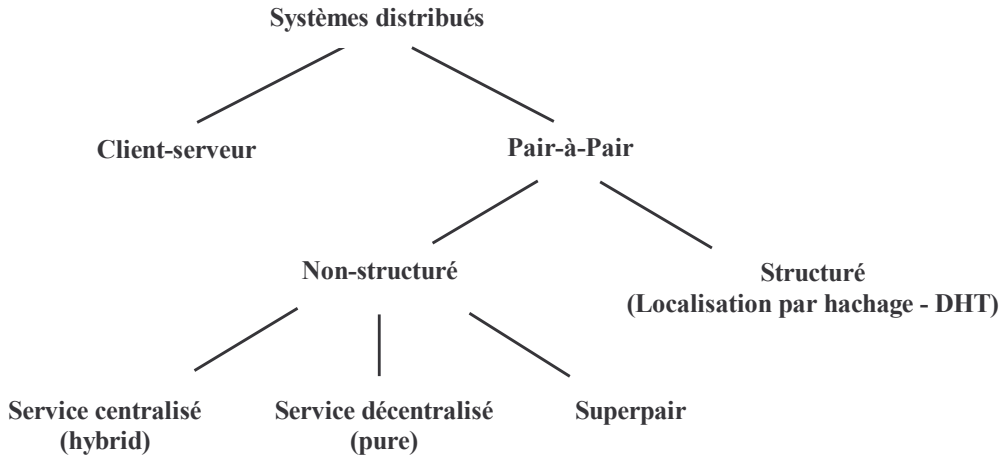


Figure 2.2 : Taxonomie des modèles distribués

Service centralisé

Il s'agit d'une architecture dans laquelle un serveur se charge de mettre en relation directe tous les pairs connectés. L'intérêt de cette technique réside dans l'indexation centralisée de toutes les ressources partagées par les abonnés sur le réseau. En général, la mise à jour de cette base s'effectue en temps réel, dès qu'un nouveau pair se connecte ou quitte le réseau. L'avantage de l'indexation centralisée réside dans le confort et l'efficacité des recherches, à condition que le service centralisé ne soit pas surchargé. L'algorithme de recherche est très simple parce que toute recherche passe par le serveur centralisé. Dès lors que des pairs sont retrouvés pour répondre à une recherche, les contacts directs seront établis entre eux.

Cependant, cette architecture introduit également deux défauts principaux. Elle ne propose d'une part qu'une seule porte d'entrée, son serveur centralisé, ce qui peut bloquer le fonctionnement de l'ensemble du réseau. D'autre part, le fait d'indexer toutes les ressources ne garantit pas évidemment le passage à l'échelle du système.

Service décentralisé

Aucun serveur centralisé n'existe dans cette architecture ; chaque machine dans son rôle est identique à une autre. C'est pour cela que l'on appelle ce type de réseau le modèle « purement » P2P. Contrairement aux réseaux centralisés, où il suffisait de se connecter au serveur pour avoir accès aux informations, le recherche dans ces réseaux purement P2P se base sur une propagation de recherche dont le mode le plus simple est une série de diffusions en broadcast. Cela a pour conséquence de foisonner et donc de ralentir les échanges de ressources entre les machines.

La performance d'un tel réseau purement P2P dépend donc essentiellement de l'algorithme de routage adopté pour propager une requête de recherche à tous ou seulement à un certain voisinage. Différents algorithmes de routage ont été proposés pour améliorer la recherche dans les réseaux purement P2P. On peut néanmoins les classer en deux catégories : algorithmes de routage aveugles et ceux informés. En principe, choisir un voisin dans un routage aveugle est complètement fait par hasard tandis qu'un routage informé ne fait le choix que des voisins pouvant bien répondre à la requête. L'évaluation pour ce choix repose, soit sur des informations synthétiques de documents stockées dans ses voisins, soit sur une statistique des résultats de chaque requête reçus par ceux-ci ; on peut alors propager prioritairement la requête vers les pairs qui ont répondu aux requêtes similaires. Pour avoir une comparaison détaillée de ces deux méthodologies de routage, les lecteurs peuvent se rapporter à l'étude dans [Tsoumakos et Roussopoulos, 2003].

Superpair

Le modèle superpair introduit une hiérarchie entre pairs appelés superpairs et pairs ordinaires connectés à un superpair. Le modèle a pour but d'utiliser les avantages des deux types de réseaux (centralisé et décentralisé). On a alors un réseau de superpairs où chacun est un serveur d'indexation de ses pairs. Cette topologie permet de diminuer le nombre de connexions sur chaque serveur d'indexation, et ainsi d'éviter les limitations de bandes passantes. Un mécanisme issu des réseaux décentralisés est utilisé pour tenir à jour dans chaque superpair un index des ressources à partir des informations provenant des pairs du groupe et aussi des autres superpairs.

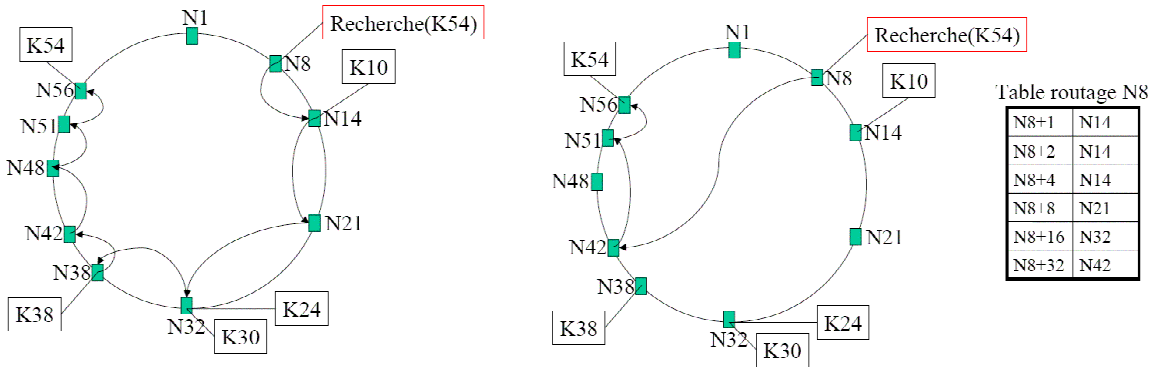
Le routage dans les réseaux de superpairs est évidemment plus efficace que dans les réseaux purement P2P. Un superpair agira comme un répertoire centralisé pour le compte d'un ensemble fini de pairs. Le routage se limitera aux superpairs dans ces réseaux. Cette solution permet de résoudre le problème d'extensibilité de l'approche purement distribuée tout en gardant l'efficacité de la solution centralisée. Cependant, de tels systèmes supposent l'existence de pairs pouvant jouer ce rôle de superpairs. Bien évidemment, de tels pairs existent mais les techniques mises en œuvre afin de désigner de tels pairs sont complexes : élire un pair comme superpair nécessite de disposer de

nombreux paramètres sur ses capacités propres mais aussi et surtout sur les capacités, notamment en terme de bande passante, des réseaux l'entourant.

Localisation par hachage (DHT)

Dans les réseaux P2P décrits ci-dessus, que ce soit avec service centralisé, décentralisé ou avec superpair, la localisation des ressources partagées n'est pas effectivement contrôlée et aucune garantie pour le succès d'une recherche n'est offerte à l'utilisateur. On appelle *non-structuré* ce type de réseaux P2P pour le distinguer des réseaux qui s'appuient sur une technique de routage par contenu, à savoir, Tapestry, Pastry, CAN, Chord,… Dans ces réseaux dits *structurés*, la localisation des ressources est contrôlée par une table de hachage distribuée (DHT - Distributed Hash Table) [Harren *et al.*, 02 ; Ratnasamy *et al.*, 2001 ; Stoica *et al.*, 2001 ; Zhao *et al.*, 2000]. C'est une technique d'indexation et de placement par mots clés qui s'applique principalement aux systèmes de stockage distribué à grande échelle.

Techniquement, un réseau P2P à la DHT se base sur un algorithme de routage par contenu qui peut garantir de retrouver, le plus efficacement possible, les ressources qu'un utilisateur a déposées sur le réseau. Un tel système de stockage distribué fonctionne indépendamment du réseau physique sous-jacent et constitue un réseau logique dans lequel chaque pair se voit assigné un identifiant aléatoire. Lorsqu'une ressource est déposée (partagée) sur ce réseau, un identifiant est assigné également à la ressource sur la base d'un hachage du contenu. Une copie de la ressource sera gardée alors dans le pair ayant l'identifiant qui est le plus proche de celui de la ressource. La technique de routage par contenu effectue une correspondance entre l'identifiant de ressources et l'identifiant de pairs afin de router les requêtes vers les pairs qui ont un identifiant le plus similaire à celui de la donnée recherchée.



(a) Recherche naïve (b) Recherche améliorée

Figure 2.3 : Exemple d'un réseau P2P structuré [Stoica *et al.*, 2001]

Prenons un exemple à base de Chord [Stoica *et al.*, 2001] pour mieux comprendre le routage DHT. La Figure 2.3 illustre un réseau structuré où les nœuds sont alloués sur l'anneau en fonction de hachage. Les ressources sont réparties sur les différents nœuds du réseau comme suit. Etant donné K la clé de hachage d'une ressource, cette ressource

doit être localisée sur le nœud (N) immédiatement supérieur ou égal à K. Par exemple, K24 et K30 sont localisées sur N32, tandis que K38 sur N38, K54 sur N56. Supposons que nous voulons chercher une ressource avec la clé de hachage K54 à partir du nœud N8. Dans le routage naïf (Figure 2.3.a), lorsqu'un nœud reçoit la requête, il propage la requête vers son successeur direct s'il ne possède pas la clé de hachage donnée dans la requête (e.g., N8 vers N14, puis N21, etc). Nous voyons clairement que la complexité de cette recherche est linéaire en nombre de nœuds.

En revanche, dans la Figure 2.3.b chaque nœud (i) a une table de routage. Cette table contient m entrées (2^m est le nombre maximum de nœuds dans le réseau). La $k^{\text{ème}}$ entrée de cette table doit indiquer le premier successeur dans l'anneau qui vérifie $(i + 2^{k-1}) \bmod 2^m$ (voir Figure 2.3.b). Pour chaque requête reçue, le nœud cherche l'entrée avec plus grande valeur inférieure à la clé recherchée. La requête est ensuite transmise au nœud sélectionné par cette entrée. Le processus est appliqué récursivement jusqu'à ce que la ressource soit trouvée. Le nombre moyen de messages transmis pour une requête dans ce cas est log(N), où N est le nombre maximum de nœuds.

2.2 Combinaison P2P et Web sémantique

Comme nous l'avons vu, le web sémantique est une évolution importante face à la difficulté de la recherche d'information sur le web. Grâce aux technologies du web sémantique, toute ressource échangée aura une signification explicite sous forme de métadonnées. Ceci permettra aux utilisateurs d'éviter une grande quantité d'informations inappropriées dans leurs échanges. Quant aux technologies P2P, elles permettront de créer un réseau d'échanges à grande échelle entre ces utilisateurs. Elles s'adapteront bien à la nature distribuée des organisations dans le futur. Pour ces raisons apparaît, pour le développement de nouvelles applications, une combinaison des technologies P2P et web sémantique.

Dans cette section, nous verrons plusieurs réseaux P2P développés dans le nouveau contexte du web sémantique, tels que InfoQuilt [Arumugam *et al.*, 2002 ; Sheth *et al.*, 2003], SWAP [Broekstra *et al.*, 2003 ; Ehrig *et al.*, 2003], Edutella [Brunkhorst *et al.*, 2003 ; Nejdl et el., 2002] et RDFPeers [Cai et Frank, 2004]. Ces réseaux se distinguent par leur modèle d'architecture et aussi par leur objectif d'application. Le premier système, InfoQuilt, est un réseau P2P du type centralisé. Il est conçu pour supporter l'intégration d'informations issues de plusieurs sources sémantiques. Comme d'autres systèmes d'intégration d'information à base d'ontologies [Bressan et al., 1997 ; Mena *et al.*, 1996 ; Stuckenschmidt *et al.*, 2000], InfoQuilt permet d'intégrer dans une vue uniforme les sources de connaissances fournies par plusieurs personnes. Chaque source est un pair pouvant être connecté à un espace global du réseau. Un mécanisme de recherche sémantique est mis à disposition pour permettre aux utilisateurs de découvrir les connaissances partagées par tous les pairs connectés à cet espace global.

Systèmes	Modèle d'architecture	Caractéristiques						Application
		Partage	**Agrégation de ressources**	**Echelle**	**Autonomie**	**Anonymat**	**Dynamisme**	
InfoQuilt	Centralisé	Ontologies (DAML+OIL)	Non	Petite	Oui	Oui	?	Intégration d'informations
SWAP	Décentralisé	Connaissances à base de RDF	Non	Grande	Oui	Non	Oui	Knowledge management
Edutella	Superpair	Métadonnées à base de RDF	Non	Grande	Oui	Oui	Oui	Echange de métadonnées
RDFPeers	Structuré (DHT)	Triplets RDF	Espace de stockage	Largement grande	Oui	Oui	Oui	Repository RDF à grande échelle

Tableau 2.1 : Comparaison des réseaux P2P

A l'opposé de l'architecture centralisée adoptée par InfoQuilt, le réseau P2P du projet SWAP repose plutôt sur une approche décentralisée pour créer un environnement de la gestion de la connaissance distribuée (cf. Distributed Knowledge Management [Bonifacio *et al.*, 2002 ; Elst *et al.*, 2003]). Plus concrètement, chaque pair de ce réseau est une personne qui rend visible à tous les autres sa base de connaissances. Les échanges entre les utilisateurs dans ce contexte se fondent essentiellement sur la *métaphore sociale de communauté de pratique*.

Edutella est un réseau superpair [Nejdl *et al.*, 2003] dont l'objectif est de créer un réseau distribué de bases de descriptions RDF autonomes en gardant toutes les caractéristiques de la technologie P2P telles que le contrôle local de données, l'organisation dynamique des pairs, etc. Edutella peut être donc considéré comme une infrastructure pour la recherche à partir de multiples sources de métadonnées. C'est le choix de modélisation de données qui a fait la différence de ce réseau avec les bases P2P qui s'appuient sur le modèle relationnel [Bernstein *et al.*, 2002 ; Ng *et al.*, 2003] ou le modèle semi-structuré XML [Papadimos *et al.*, 2003 ; Pitoura *et al.*, 2003 ; Sartiani *et al.*, 2003].

Enfin, RDFPeers vise à être un entrepôt (repository) distribué à grande échelle pour le stockage de métadonnées RDF. Il est en effet un réseau P2P du type structuré. Grâce au routage (placement et recherche) par contenu, la recherche d'un triplet sur le réseau est vraiment efficace en un temps logarithmique par rapport au nombre de nœuds. A notre connaissance, RDFPeers est le seul réseau qui applique la technique DHT au stockage des triplets RDF.

Le Tableau 2.1 résume les caractéristiques des quatre systèmes P2P selon leurs objectifs d'application (cf. section 2.1.1). Nous donnons par la suite des descriptions techniques pour chacun de ces systèmes.

2.2.1 InfoQuilt

Le système InfoQuilt développé au sein du laboratoire LSDIS (Université de Georgia) est un environnement permettant aux utilisateurs de formuler des requêtes complexes pour découvrir des connaissances à partir de multiples sources. Il existe plusieurs générations de ce système dont la dernière est une intégration des technologies appropriées du web sémantique et du P2P. Ce dernier système, appelé parfois PSW (Peer-to-peer Semantic Web), exploite DAML+OIL pour la représentation de connaissances et l'architecture P2P pour la distribution de connaissances dans des pairs différents. L'objectif d'InfoQuilt est de faciliter la création et l'entretien de multiples ontologies dans les pairs en gardant la capacité de recherche sémantique dans ce réseau.

InfoQuilt repose effectivement sur l'architecture P2P du type centralisé dans laquelle chaque client doit s'enregistrer dans l'espace global du système. Un tel espace permet à l'utilisateur de retrouver des contacts nécessaires avec d'autres pairs qui peuvent fournir des connaissances à son besoin. Du point de vue de l'architecture, cet espace est

l'équivalent d'un médiateur pour les ontologies créées et entretenues indépendamment par les individus. La Figure 2.4 illustre les composants principaux d'un pair InfoQuilt : la base locale stocke les ontologies créées par chaque individu ; l'IScape Builder est un composant important permettant à l'utilisateur de créer de nouvelles ontologies par liaison avec d'autres ontologies à travers l'espace global.

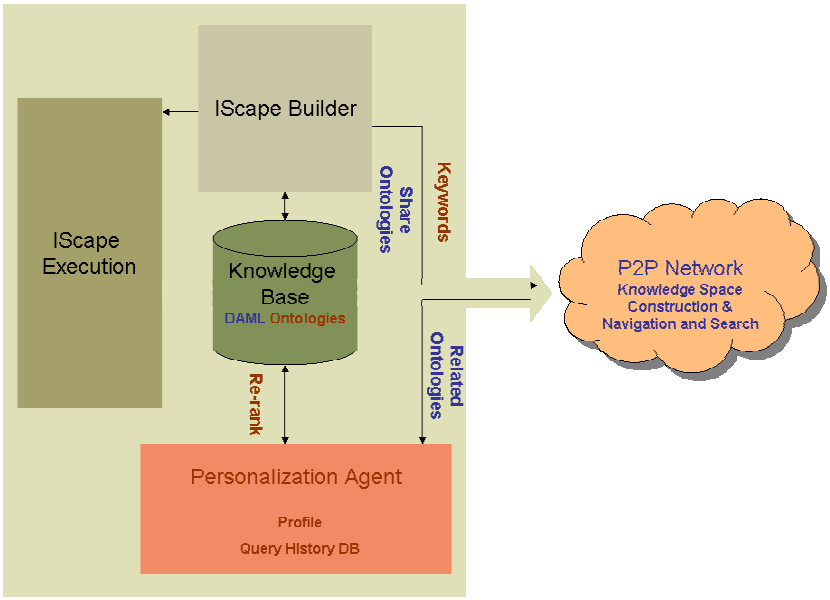


Figure 2.4 : Architecture d'un pair InfoQuilt [Arumugam *et al.*, 2002]

L'avantage du système réside dans le support de la recherche sémantique. Plutôt que d'utiliser un langage de requête spécifique, l'utilisateur formulera pour sa recherche sémantique des « IScapes » à l'aide de l'IScape Builder. Un IScape est de fait une structure sophistiquée (voir [Sheth *et al.*, 2003]) qui représente des concepts et des contraintes précises pour la recherche de connaissances à base d'ontologies. Malgré sa structure très complexe, l'utilisateur ne doit entrer que des mots clés pour sa recherche. A partir de ces mots clés, l'IScape Builder retrouve dans l'espace global des ontologies appropriées à ceux-ci. Ainsi, une liste d'ontologies appropriées sera suggérée à l'utilisateur, puis à son tour il choisira celles adaptées à son besoin. Après cette étape de sélection d'ontologies, l'utilisateur continuera avec d'autres étapes progressivement jusqu'à finir la construction d'un IScape (e.g., spécifier des agrégations entre les concepts trouvés, des contraintes de recherche, des paramètres,…). Dès lors que cet IScape est disponible, la recherche s'exécute pour celui-ci.

2.2.2 SWAP

SWAP est un réseau P2P complètement distribué. La Figure 2.5 présente l'architecture du système constituant un ensemble des pairs appelés les nœuds SWAP. Chaque nœud SWAP a un entrepôt local où sont stockées des connaissances extraites à

partir de multiples sources locales. La modélisation de telles connaissances repose effectivement sur un schéma RDF. L'utilisateur dispose d'une interface graphique pour éditer/consulter ses connaissances dans l'entrepôt local. Cette interface lui permet également de formuler des requêtes effectuant une recherche sur le réseau P2P. Etant donné une requête, les connaissances de l'entrepôt local seront d'abord utilisées pour répondre à la requête. Si les connaissances disponibles ne suffisent pas pour la réponse, la requête sera réécrite et puis distribuée aux pairs pouvant répondre. Les réponses retournées par ces pairs seront réunies et présentées à l'utilisateur. A son tour, l'utilisateur pourra décider selon son besoin comment intégrer les connaissances trouvées dans sa base locale pour les réutiliser après.

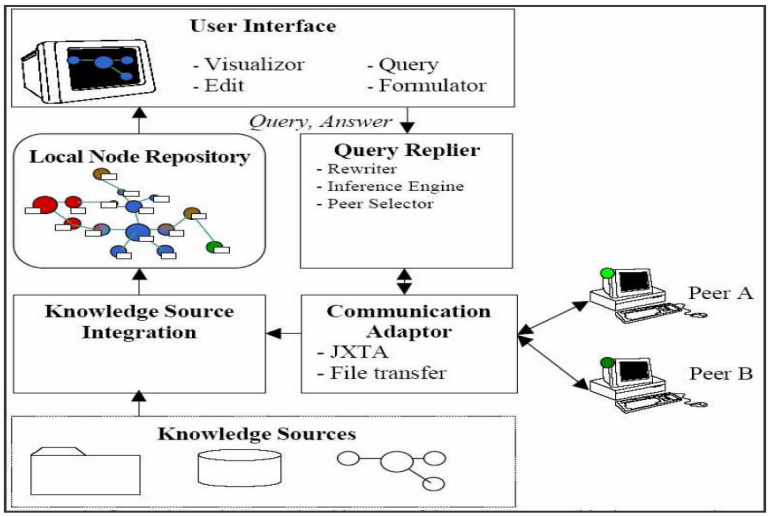


Figure 2.5 : Architecture d'un nœud SWAP [Ehrig *et al.*, 2003]

Le schéma de connaissances employé par chaque nœud SWAP est donné dans la Figure 2.6. Chaque déclaration (une instance de rdf:Statement ou rdfs:Resource) dans la base de connaissances s'associe à une méta-information pour mémoriser d'où elle vient (i.e., de quel pair). C'est une structure composée de deux classes, swap:Swabbi et swap:Peer (cf. [Broekstra *et al.*, 2003]). Une instance de swap:Swabbi représente des méta-informations relatives à la connaissance déclarée, telles que : sa localisation d'origine (swap:location), son étiquette d'affichage (swap:label), sa mesure de confiance (swap:confidence), … Le lien vers une instance de swap:Peer permet d'indiquer le pair dans lequel la connaissance a été trouvée. La valeur de swap:trust représente une mesure de confiance globale sur ce pair. Les méta-informations associées aux connaissances seront utilisées dans l'algorithme de routage du réseau SWAP. Cet algorithme repose sur les principes d'une métaphore sociale comme ceci :

1. On ne pose des questions qu'à des pairs que l'on juge de « bonne réputation ».

2. Une personne qui a répondu à une question peut donner de surcroît des connaissances sur le même domaine.

3. Il existe une possibilité qu'une personne ait la réponse à une requête si elle a échangé auparavant des connaissances similaires.

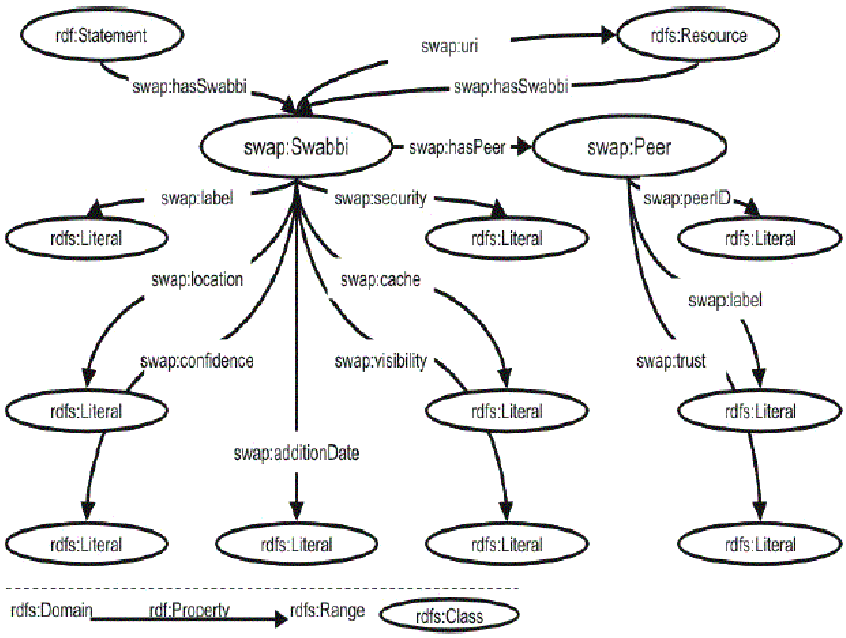


Figure 2.6 : Schéma de connaissances SWAP [Broekstra *et al.*, 2003]

Deux techniques de routage ont été développées selon cette métaphore sociale : la sélection de pairs et la relaxation de requêtes. La première technique permet de retrouver pour chaque requête les meilleurs pairs qui peuvent donner la réponse en respectant les règles 1 et 2 (i.e., les pairs ont été évalués avec une valeur importante de confiance globale et dans le domaine de la requête). Quant à la deuxième technique, elle permet d'étendre la recherche sur des connaissances similaires en appliquant la règle 3. Pour cela, les valeurs de confiance (swap:confidence, swap:trust) fondent le calcul de routage dans le réseau SWAP (voir [Tempich *et al.*, 2004]).

2.2.3 Edutella

L'objectif du projet Edutella est de créer une infrastructure pour les échanges pair-à-pair de métadonnées par descriptions RDF. Edutella spécifie un ensemble de services dans le but de bien connecter des pairs de (méta)données hétérogènes. Ces services sont prévus pour fournir au réseau les capacités d'interrogation, de réplication, de mapping entre schémas,… Néanmoins, dans l'étape actuelle de développement, seul le service d'interrogation est mis à disposition pour le réseau Edutella. Ce service permet une abstraction de contenu pour chaque pair dont les échanges reposent sur un modèle de données commun et une famille de langages de requête appropriée [Nejdl *et al.*, 2002].

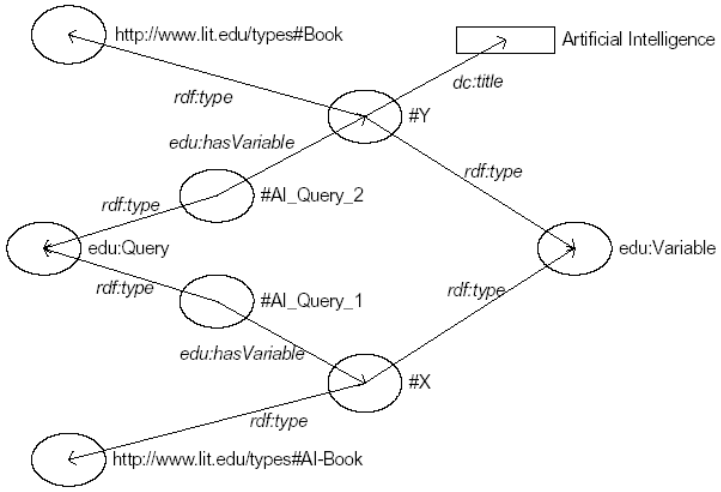


Figure 2.7 : Une requête au format RDF-QEL-1 [Nejdl *et al.*, 2002]

Le modèle commun, appelé ECDM (Edutella Common Data Model) n'est qu'une représentation formelle de type Datalog des descriptions RDF. Des requêtes Datalog seront ainsi formulées pour les échanges. On utilise, pour le service d'interrogation, la famille des langages de requête RDF-QEL-i basés sur la représentation RDF/XML. La Figure 2.7 donne un exemple de requête au format RDF-QEL-1. La requête permet de trouver des livres qui ont le titre « Artificial Intelligence » ou qui sont des instances de AI-Book. Comme le paradigme de QBE (Query By Exemple), le graphe RDF de la requête représente exactement la même structure du graphe de la réponse, avec des annotations complémentaires pour décrire les variables et les contraintes entre elles. Créer d'autres requêtes plus complexes de type Datalog est aussi possible en utilisant les langages de haut niveau (RDF-QEL-2, RDF-QEL-3, …). Les lecteurs peuvent voir dans [Nejdl *et al.*, 2002] la présentation de cette famille de langages de requête.

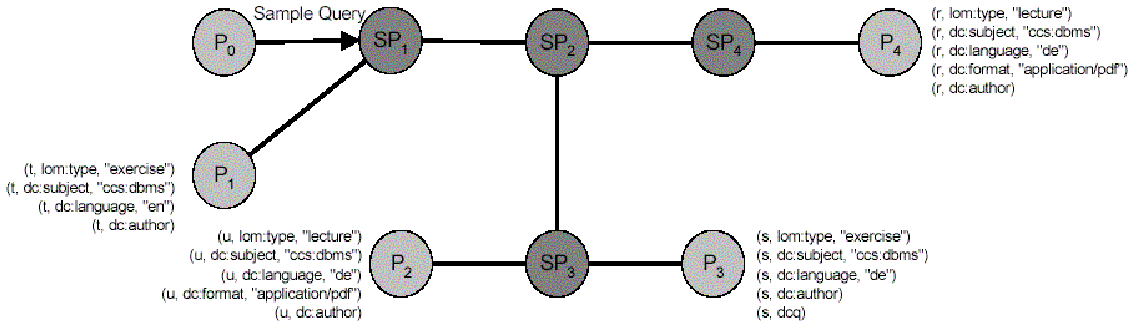


Figure 2.8 : Exemple d'un réseau P2P de superpairs [Brunkhorst *et al.*, 2003]

Granularité	Indices dans SP_2	
Schémas	dc lom dcq	SP_1, SP_3, SP_4 SP_1, SP_3, SP_4 SP_3
Propriétés	dc:subject lom:type dc:format	SP_1, SP_3, SP_4 SP_1, SP_3, SP_4 SP_3, SP_4
Propriétés/Ranges	(dc:subject, ccs:dbms)	SP_1, SP_3, SP_4
Propriétés/Valeurs	(lom:type, "exercise") (dc:language, "de")	SP_1, SP_3 SP_3, SP_4

Tableau 2.2 : Indices SP/SP dans SP_2

Du point de vue de l'architecture, le réseau Edutella se fonde sur un sous-réseau de superpairs auxquels chaque pair doit se connecter. L'algorithme de routage dans un tel réseau suit deux phases : router premièrement des requêtes dans le réseau (backbone) des superpairs et puis distribuer celles-ci seulement aux pairs connectés aux superpairs appropriés. Afin d'eviter la diffusion (broadcasting) dans le routage, des indices sont mis en œuvre dans les superpairs. Deux types d'indices sont utilisés pour ces deux phases de routage : superpair/superpair (SP/SP) et superpair/pair (SP/P). Les indices sont effectivement une récapitulation de métadonnées fournies par chaque pair/superpair. La granularité de ces indices est variable du niveau schéma au niveau valeur. Plus concrètement, au niveau schéma les indices déclarent les schémas qu'un pair a utilisés dans sa base de métadonnées ; donc le pair ne répondra aux requêtes qu'autour de ces schémas. Par exemple, le pair P_1 dans la Figure 2.8 ne supporte que le schéma Dublin Core (*dc*) et Learning Object Metadata (*lom*). D'autre part, les indices au niveau valeur donnent plus de renseignements précis sur les données. En résumé, il y a quatre niveaux d'indices dans le réseau Edutella : schémas, propriétés, propriétés/ranges, et propriétés/valeurs. Le Tableau 2.2 présente un exemple pour des indices enregistrés sur le superpair SP_2. Ces indices sont de type SP/SP. Notons que les indices sont mis à jour automatiquement dans les superpairs, lorqu'il y a un changement dans l'organisation du réseau (voir [Nejdl *et al.*, 2003 ; Brunkhorst *et al.*, 2003]).

2.2.4 RDFPeers

RDFPeers vise à être un entrepôt RDF distribué à grande échelle où chaque nœud peut stocker et interroger des triplets RDF de manière transparente. Les nœuds de cet entrepôt s'organisent dans un réseau P2P structuré dont les identificateurs de pair sont choisis aléatoirement. Quand un triple RDF est inséré dans le réseau, il sera stocké dans trois pairs par application d'une fonction de hachage sur le sujet, l'attribut, et la valeur d'objet. Ainsi, des requêtes de triplets peuvent être efficacement routées aux pairs dans lesquels les triplets devraient être stockés s'ils existent.

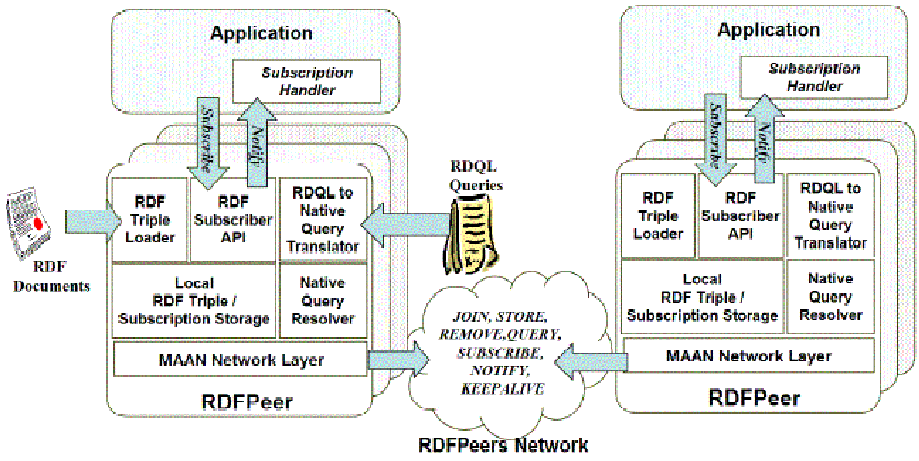


Figure 2.9 : Architecture du réseau RDFPeers [Cai et Frank, 2004]

La Figure 2.9 montre l'architecture du réseau RDFPeers. RDFPeers exploite MAAN [Cai *et al.*, 2003], une extension du réseau structuré Chord [Stoica *et al.*, 2001], pour sa couche de base de stockage de données. Ce sont l'enregistreur et l'interrogateur créés pour chaque nœud qui jouent le rôle d'interfaçage (enregistrement et interrogation respectivement) avec cette couche de stockage. Des messages particuliers du protocole MAAN sont utilisés pour manipulation. Un message STORE insère dans le réseau de stockage un ensemble de triplets, tandis qu'un message REMOVE permet de les supprimer. Un message QUERY permet de retrouver des triplets en visitant seulement les nœuds qui sont déterminés à stocker les triplets selon des critères de besoin. Il est possible d'avoir un interpréteur des langages de requête RDF de haut niveau tel que RDQL. Cet interpréteur pourra traduire des requêtes RDQL en requêtes natives du réseau RDFPeers (voir [Cai et Frank, 2004]).

2.3 Approche de médiation décentralisée pour le web sémantique

Grâce à l'existence des ontologies, la recherche d'information dans le web sémantique est plus pertinente en prenant en compte les liens unissant les éléments d'informations, de même que les relations portent les sens et contexte d'utilisation des informations. Aussi, il devient possible d'automatiser des tâches définies à partir des métadonnées décrivant explicitement la sémantique des ressources du web. Le problème qui se pose est celui de la manière de déployer des ontologies pour le web sémantique. Actuellement, on pense qu'il n'existera pas d'ontologie vraiment commune pour un domaine particulier, mais plutôt qu'il y aura plusieurs ontologies concurrentes et probablement des recouvrements entre celles-ci. Ceci stimule l'émergence d'une approche de médiation des ontologies qui se trouvent sur le web sémantique.

2.3.1 Motivations et recherches voisines

Les organisations comme les individus doivent développer eux-mêmes des ontologies pour leur usage dans le web sémantique pour deux raisons. Tout d'abord, il n'existe pas encore de véritables standards, de méthodologies éprouvées pour guider la création d'ontologies qui pourront s'adapter à la variation des besoins d'une grande communauté. Deuxièmement, il est souvent très difficile d'établir dans la communauté un consensus au sujet des terminologies et des structures à employer.

Dans l'approche de médiation décentralisée pour le web sémantique, l'architecture P2P devrait être exploitée pour permettre à chacun des participants de créer une base de connaissances autonome avec un schéma (ontologie) local. Une intégration sémantique des pairs permettra de répondre aux requêtes utilisateurs non seulement avec des données locales, mais en exploitant le réseau des pairs reliés sémantiquement. La Figure 2.10 illustre l'architecture de médiation dans laquelle chaque pair établit localement des mises en correspondance entre son schéma local et ceux de pairs reliés. Cette pratique est très différente de la médiation classique dans l'intégration de données d'où les sources sont mises en correspondance à travers un schéma global. Ceci est une limitation dans le cadre du web sémantique. Grâce à l'architecture P2P distribuée, le réseau sémantique peut facilement évoluer dans la mesure que chaque nouvelle source peut se relier avec n'importe quelle source qu'elle trouve appropriée. Ainsi, des relations sémantiques entre les schémas locaux seront utilisées pour constituer des recherches en enchaînant « les chemins sémantiques » existant entre les pairs.

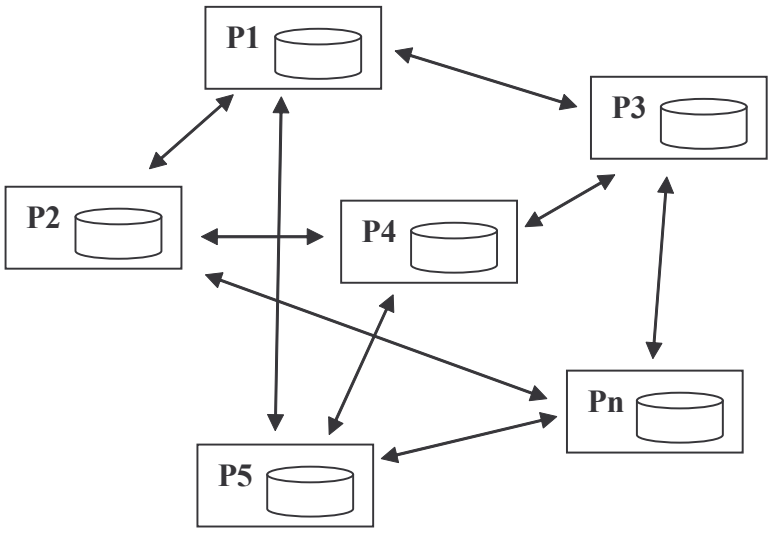


Figure 2.10 : Architecture de médiation de pairs

Dans ce contexte, il y a actuellement plusieurs systèmes définis comme infrastructures de médiation pour le web sémantique. Ils se différencient par la modélisation de données et par les relations sémantiques entre pairs. Premièrement, nous pouvons citer Piazza [Halevy *et al.*, 2003 ; Ives *et al.*, 2004], un système P2P

s'appuyant sur le principe des bases de données XML. Il fournit un langage de description à base de XQuery [Boag *et al.*, 2004] pour décrire des correspondances sémantiques entre sources de données. En exploitant un algorithme de reformulation de requêtes, une recherche dans le réseau Piazza peut se propager entre pairs reliés par des correspondances de schéma. Piazza permet également l'interopérabilité entre les sources XML et les sources RDF. C'est pour cette raison qu'il est considéré comme étant un pont qui pourrait relier le web sémantique à la richesse des données actuellement disponibles sur le web.

Un autre système fondé sur les ontologies simples à base de logique de description est SomeWhere [Adjiman *et al.*, 2004 ; Rousset, 2004] développé dans le laboratoire LRI. OWL PL est un sous-ensemble du langage de description OWL utilisé pour décrire des données dans ce système. Il permet de spécifier des classes en logique de description (avec des constructeurs de classe tels que équivalence, inclusion, union, intersection,...). Des objets (individus) sont associés à chaque classe de description. Une requête permettra de retrouver des objets sous une ou plusieurs classes combinées. Comme chaque pair du système définit lui-même des classes de description et qu'il existe des relations entre classes de différents pairs, la recherche d'individus sous une classe peut être étendue à plusieurs pairs reliés. La récriture de requêtes distribuées dans ce système repose sur la logique propositionnelle (voir [Goasdoué et Rousset, 2003]).

2.3.2 Notre contribution

Comme nous l'avons écrit en introduction, le web-of-people est une application particulière qui applique une architecture de médiation P2P pour la recherche d'information dans un réseau social. Dans cette application, chaque utilisateur définit lui-même une taxonomie locale de topiques pour indexer des posts qu'il peut publier. Des mises en correspondance entre les taxonomies personnelles permettent d'enchaîner des recherches de posts à travers plusieurs pairs dans le réseau. Un tel modèle très simple de médiation repose effectivement sur l'intégration taxonomique de multiples sources et a été proposé dans [Tzitzikas et Meghini, 2003]. Dans la partie II de cette thèse, notre contribution concernera les deux points majeurs suivants :

1. ***Architecture et protocoles de fonctionnement du web-of-people***. Nous proposerons un cadre de conception pour le web-of-people dans lequel chaque utilisateur sera identifié par une identité sociale indépendante de son adresse physique sur le réseau. Un schéma de connaissances RDF sera spécifié pour les échanges de métadonnées entre utilisateurs. Le fonctionnement du réseau devra se fonder sur des protocoles de base permettant d'établir des relations de confiance entre les utilisateurs. En utilisant ces protocoles, des utilisateurs pourront se mettre en communication directe par interrogation mutuelle et/ou par notification d'évènements justifiant des échanges entre eux. Dans la conception de ces protocoles, nous considérerons comme essentiels les problèmes de sécurité et de confiance entre utilisateurs.

2. ***Evaluation efficace de requêtes réseau***. Dans un réseau basé sur une architecture de médiation tel que le web-of-people, un utilisateur peut poser des requêtes réseau afin de chercher des posts dans des pairs reliés sémantiquement. Nous nous sommes intéressés à une implémentation qui permet de calculer efficacement de telles requêtes réseau. Dans cette implémentation (Chapitre 5), l'optimisation de requêtes sera considérée sous deux aspects : i) détecter et éliminer des calculs redondants dans le processus d'évaluation des requêtes, ii) accélérer des calculs en exploitant un codage particulier pour les hiérarchies taxonomiques.

II

Web-of-people

Chapitre 3

Conception d'architecture

Le Web-of-people a été présenté initialement dans [Plu *et al.*, 2003] comme un réseau d'échanges actifs dans lequel les participants peuvent adresser des revues (ou notes de lecture) de ressources à des connaissances (amis ou collègues). Chaque personne crée et organise par catégories (ou topiques) des revues sur des URL trouvées intéressantes. Ces revues sont échangées par un mécanisme de diffusion par listes d'utilisateurs attachées à chaque topique. Ainsi, un utilisateur peut recevoir, par ses connaissances, des notes sur des ressources web de son domaine d'intérêt. Toutefois les questions suivantes sont restées ouvertes :

- Quelle structure de connaissances doit être utilisée pour constituer la base de revues personnelles ?
- Quelle architecture et quels protocoles s'appliquent à ce système réparti ?
- Comment établir la confiance dans les échanges entre utilisateurs du web-of-people ?

Ce chapitre a pour objectif de répondre à ces questions en spécifiant une architecture à base de weblogs pour le web-of-people [Ta *et al.*, 2004]. C'est en publiant ses revues ou notes documentaires sous forme de posts dans une base personnelle de type weblog, qu'un utilisateur partagera avec ses connaissances des commentaires sur des ressources de commun intérêt. Dans un premier temps, nous décrirons l'architecture de référence du web-of-people. A partir de cette architecture, nous spécifierons notre cadre conceptuel général pour le web-of-people. Il comporte un mécanisme d'identification sociale d'utilisateurs, un schéma d'échange de connaissances et des services de base s'appliquant à la communication dans le réseau. Nous nous attacherons enfin à la conception détaillée des éléments de ce cadre de conception.

3.1 Architecture de référence

La Figure 3.1 illustre l'architecture de référence choisie pour le web-of-people. Il s'agit d'un réseau P2P dans lequel chaque pair est au service d'un utilisateur qui peut s'y connecter pour gérer des connaissances dans sa base personnelle, dit PKB (Personal Knowledge Base), et échanger ses connaissances avec les utilisateurs de son choix. Un weblog est généré comme une vue HTML de cette PKB pour que les connaissances personnelles soient accessibles et lisibles dans les navigateurs classiques. Afin de supporter des échanges actifs entre utilisateurs, chacun devra disposer d'une boîte de messages qui lui permettra de recevoir des notifications d'échange provenant de ses partenaires. Un annuaire d'utilisateurs est exploité pour la gestion des participants à travers leur identité dans le réseau social du web-of-people. Lorsqu'un utilisateur s'inscrit dans le réseau, une entrée doit être créée pour lui dans cet annuaire. Nous verrons ci-après les éléments nécessaires dans le cadre de conception pour web-of-people.

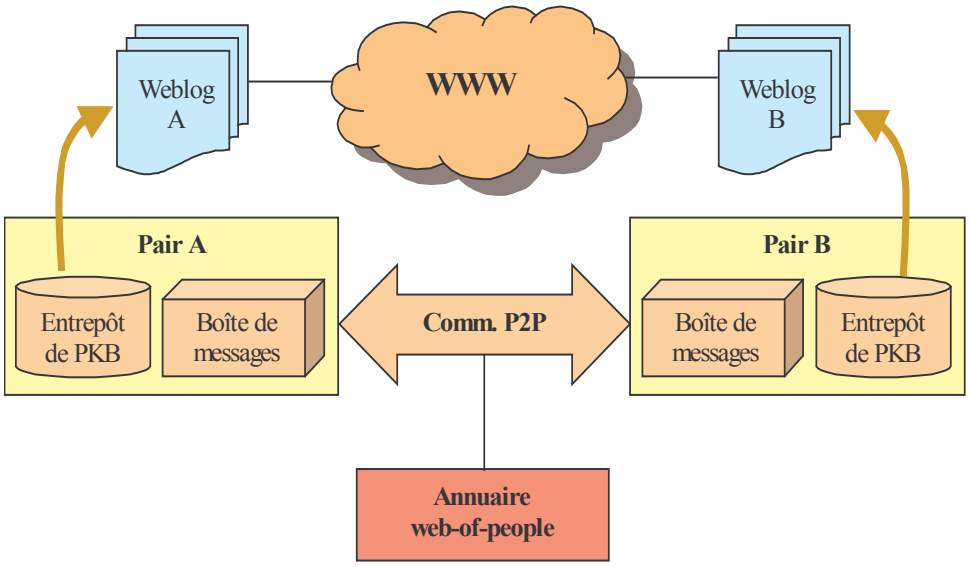


Figure 3.1: Architecture de référence pour le web-of-people

3.1.1 Identification sociale des utilisateurs

L'annuaire d'utilisateurs est un composant indispensable dans l'architecture du web-of-people pour séparer la vue sociale du réseau et l'organisation physique des pairs. Indépendamment de l'implémentation du réseau physique, chaque utilisateur se servira toujours d'une identité unique pour ses échanges sociaux. C'est l'annuaire qui permettra de retrouver les informations nécessaires pour établir des connexions avec un utilisateur dans le réseau P2P. Une telle information peut être simplement l'adresse IP du pair auquel l'utilisateur se connecte.

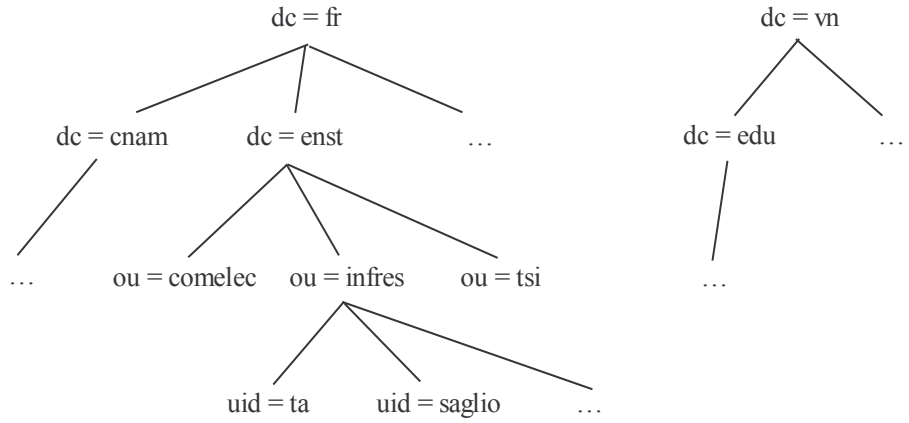


Figure 3.2 : Modèle DIT pour l'identification sociale des utilisateurs

L'identification des utilisateurs dans l'annuaire web-of-people se conforme au modèle d'arborescence hiérarchique DIT (Directory Information Tree) du protocole LDAP (RFC2251), où chaque entrée a un nom distinct. Chaque utilisateur enregistré dans le réseau doit avoir une entrée dans l'annuaire dont le nom distinct est son identifiant. La formation des noms distincts suit le modèle des DNS. Par exemple, le nom distinct (*uid=ta, ou=infres, dc=enst, dc=fr*), ou dans le format lisible *ta$infres.enst.fr*, réserve pour l'utilisateur *ta* dans l'unité *infres* de l'établissement *enst.fr* (cf. Figure 3.2).

Nous choisissons cette identification très similaire à celle des adresses email pour faciliter la gestion des accessibilités par sous-domaines. Cependant, l'identification dans le web-of-people repose plutôt sur le rôle social de chacun que sur l'adresse physique de son pair. L'identification dans le web-of-people exprime donc des informations personnelles ou professionnelles pour chacun, par exemple, son pays, son établissement professionnel, son groupe de travail, etc. Un utilisateur peut se déplacer sur le réseau physique sans changer son identification sociale.

N.B. Le caractère « $ », non « @ », est utilisé dans le format d'identification pour le distinguer de celui des adresses email. Un utilisateur ne peut s'enregistrer que sur un domaine pré-existant dans l'annuaire web-of-people. On ne permet qu'aux administrateurs du web-of-people de gérer ces domaines.

3.1.2 Personal Knowledge Base (PKB)

Comme nous l'avons évoqué en préambule, l'usage des weblogs est une pratique courante pour la publication personnelle. En exploitant cette pratique, nous définissons chaque PKB comme un weblog sémantique qui comporte des posts individuels. Chaque post dans le web-of-people est une revue personnelle sur une ou plusieurs ressources web collectionnées par un utilisateur. On considère ces ressources, dont les URL sont

référencées dans le post, comme une déclaration implicite d'intérêt de l'utilisateur. Les topiques du weblog expliquent donc un classement pertinent des ressources collectionnées à travers les posts de revue. Dans la Figure 3.3, un topique est dessiné par un cercle pour former un nœud tandis qu'un post l'est par un rectangle pour former une feuille. Les flèches entre topiques représentent des relations de subsomption entre eux. Un post contient des pointeurs sur le web, et doit être rattaché au moins à un topique du weblog.

Pratiquement, il existe différentes catégorisations et formats de weblogs. On peut construire effectivement des weblogs sans topiques, dans ce cas un weblog n'est qu'une liste de posts ordonnée chronologiquement. De même un post peut ne consister qu'en un simple texte de description sans pointeurs. Cependant, selon une étude de Bar-Ilan [Bar-Ilan, 2004] la plupart de posts (88,2%) dans les weblogs professionnels donnent des hyperliens vers les sources de référence. Dans le web-of-peole, on ne s'intéresse alors qu'aux weblogs dont les posts contiennent des pointeurs et sont classifiés selon leur(s) thème(s) abordé(s). Nous constatons que ce type de weblogs orientés topique sera une meilleure source de connaissances partagées, notamment pour des communautés professionnelles.

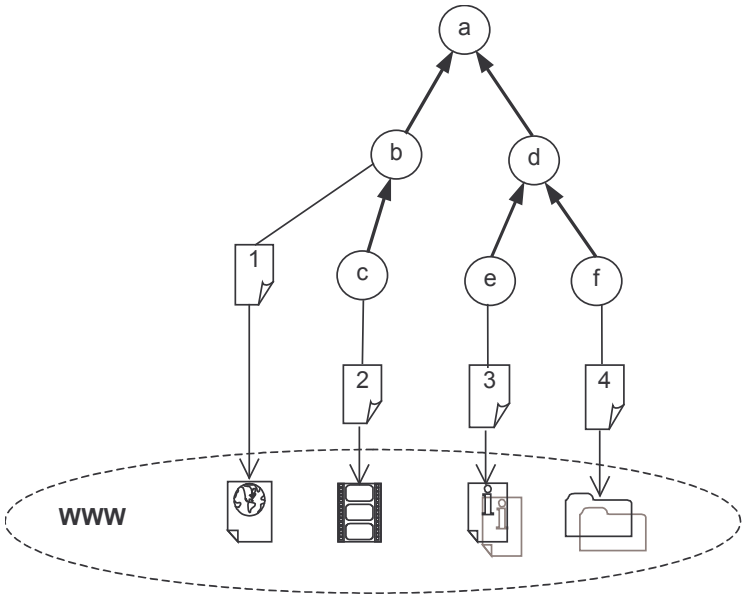


Figure 3.3 : La structure de connaissances dans une PKB

La Figure 3.4 donne un exemple de PKB qui est tiré des catalogues du site dmoz.org (Open Directory Project). Dans cet exemple, deux PKB ont été créées pour deux utilisateurs *u1* et *u2* (*u1$odp* et *u2$odp* leur sont assignés respectivement comme identifiant social). La PKB de chacun figure un sous arbre de topiques choisi pour sa taxonomie personnelle, par exemple les topiques sous *Software* pour la PKB de *u1*, ceux sous *Internet* pour la PKB de *u2*. Notons que tous les topiques sont décrits par métadonnées. Dans la figure ci-après nous ne donnons que pour exemple le titre et la date de création du topique *Internet-Soft*. Les flèches grasses entre topiques représentent

des relations hiérarchiques, l'un est un sous-topique de l'autre. Dans chaque PKB, des posts sont créés pour décrire des ressources qualifiées dans un ou plusieurs topiques. Un tel post consiste en descriptions pour son titre, la date de qualification, le commentaire de contenu, le pointeur de ressource,… Notons que les posts ne sont attachés qu'aux topiques locaux. Par exemple, *&1* et *&2* créés par *u1* ont été indexés sous le topique *Search*, qui se trouve dans la même PKB.

Une relation est possible pour les PKB. Une personne peut établir une correspondance entre sa PKB et celle d'autres en définissant des « intégrations » de topique. Une intégration constitue une sorte d'extension d'un topique local vers un topique étranger, i.e., le topique étranger est considéré comme un fils du topique local[6]. Par exemple, une flèche pointillée de *Internet-Soft* à *Internet* dans la Figure 3.4 signifie que *Internet-Soft* est inclus dans *Internet*. Cette intégration ouvrira un chemin pour la recherche vers *Internet-Soft* chaque fois que *Internet* est consulté.

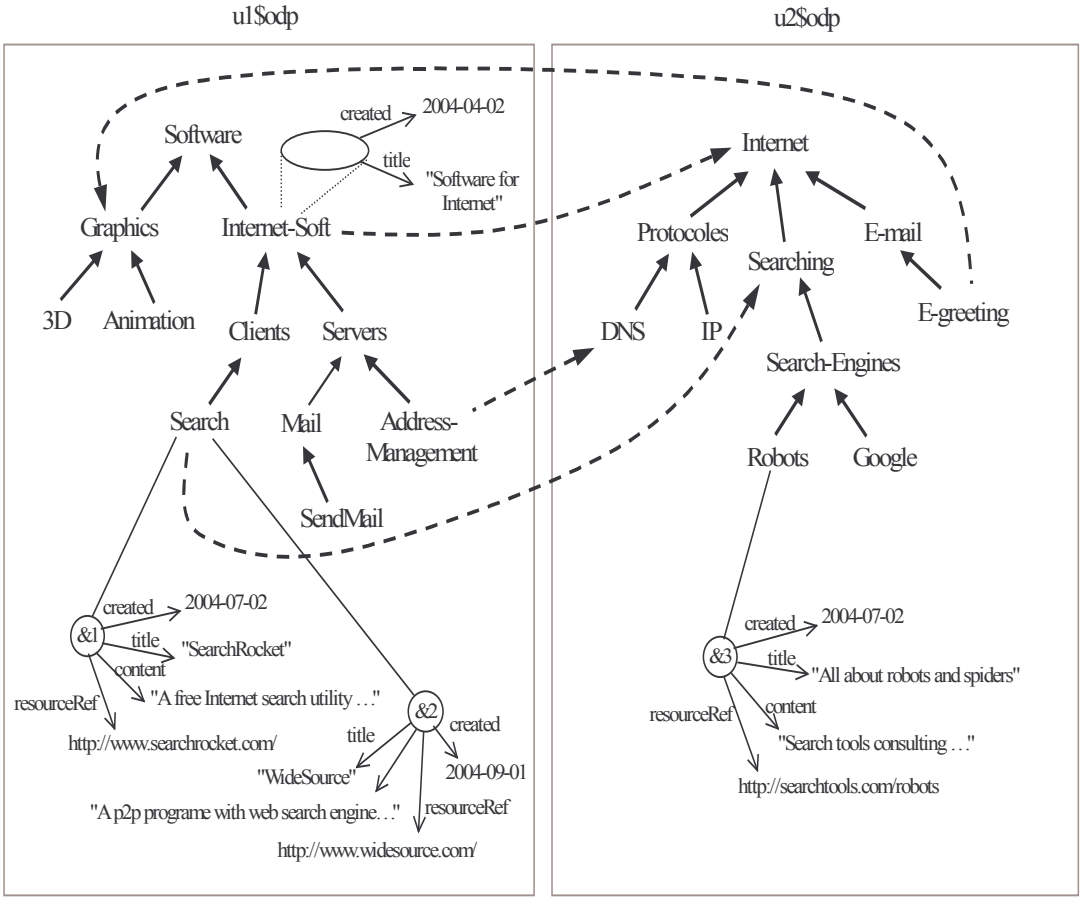


Figure 3.4 : Un exemple introductif pour le web-of-people

[6] Si l'étranger établit le même lien, il y a alors relation d'équivalence entre ces deux topiques

3.1.3 Interrogation et notification

Le web-of-people se fonde sur une distribution de PKB pour chaque pair. La communication entre utilisateurs dans ce système se fait uniquement par deux services de base fournis pour chaque pair : interrogation et notification.

Le service d'interrogation permet à un utilisateur de consulter des ressources dans d'autres PKB accessibles pour lui. Comme un mécanisme « pull », chaque requête formée selon son besoin lui rend un ensemble de descriptions. Ce service lui permet de chercher des posts en parcourant différentes PKB, en commençant par la sienne.

Mais s'il n'y avait que ce service, le web-of-people n'irait pas beaucoup plus loin que le web actuel face au problème de la quantité énorme de PKB et de topiques. Cela rend nécessaire un deuxième service pour permettre des échanges actifs entre utilisateurs. Supposons que chaque utilisateur a des contacts personnels avec un nombre limité de partenaires ou amis avec lesquels il souhaite échanger des connaissances sur sa PKB. En utilisant des messages particuliers, il peut notifier à ses amis les opérations les concernant, qui correspondent à des mises à jour dans sa PKB. Grâce à ces notifications, les récepteurs peuvent réagir et poursuivre les échanges. Par conséquent, un message transporte effectivement le sens d'un évènement de modification de relations fonctionnelles entre PKB. Autrement dit, on peut considérer la messagerie, au contraire de l'interrogation, comme étant un mécanisme « push » pour les échanges de connaissances.

Grâce à ces deux services, notre système peut se concevoir comme une organisation de multiples utilisateurs définissant, et modifiant par transactions, leurs PKB en relation les unes avec les autres.

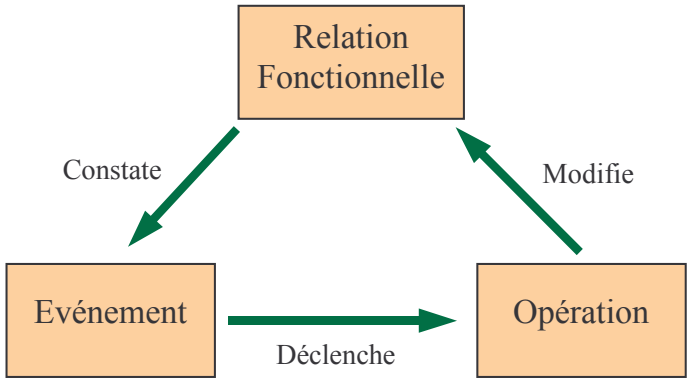


Figure 3.5 : Modèle des transitions dans une organisation selon la méthode REMORA

Selon le modèle présenté dans la méthode REMORA [Rolland *et al.*, 1988], tous les éléments d'une organisation peuvent être classés dans l'un des trois groupes suivants: *Relation Fonctionnelle*, *Evénement*, *Opération* (cf. Figure 3.5).

- Une relation fonctionnelle est un élément de connaissance dans une PKB qui concerne une autre PKB. L'extension de cette classe d'éléments définira à chaque instant l'état des liens entre toutes les PKB dans notre système. Une modification dans une PKB est un changement d'état.
- Un évènement est la constatation d'un changement d'état qui concerne une relation fonctionnelle. Les évènements sont ordonnables temporellement.
- Une opération est une action de modification qui peut être exécutée librement par un utilisateur ou déclenchée à la suite d'un ou plusieurs évènements. Cette action décrit un phénomène réel correspondant aux activités fournies dans l'organisation. Une opération peut modifier l'état d'une ou plusieurs relations fonctionnelles, les changements ainsi provoqués sont de plusieurs types : création ou suppression (passage de l'état inexistant à l'état existant ou vice versa), mise à jour (passage de l'état existant à un autre en attribuant de nouvelles valeurs).

Dans le web-of-people, une opération de modification dans une PKB peut provoquer, selon le modèle REMORA, des opérations de modification en cascade sur d'autres PKB à distance. La messagerie est utilisée simplement comme moyen de transport des évènements décrivant les changements pour les destinataires concernés. La Figure 3.6 illustre ce rôle de la messagerie. Pour chaque modification dans la *PKB1* qui pourrait provoquer une modification dans la *PKB2*, un message sera envoyé automatiquement à la boîte de messages liée à la *PKB2* afin de notifier cet évènement. En analysant les messages arrivés, des opérations standardisées seront proposées à l'avis du récepteur. Notons que le traitement d'un message arrivé peut être soit contrôlé par l'utilisateur (car il requière un choix de l'utilisateur pour réaction), soit automatique en fonction de l'application développée. Ce dernier choix aura l'avantage évident de vider automatiquement la boîte de messages.

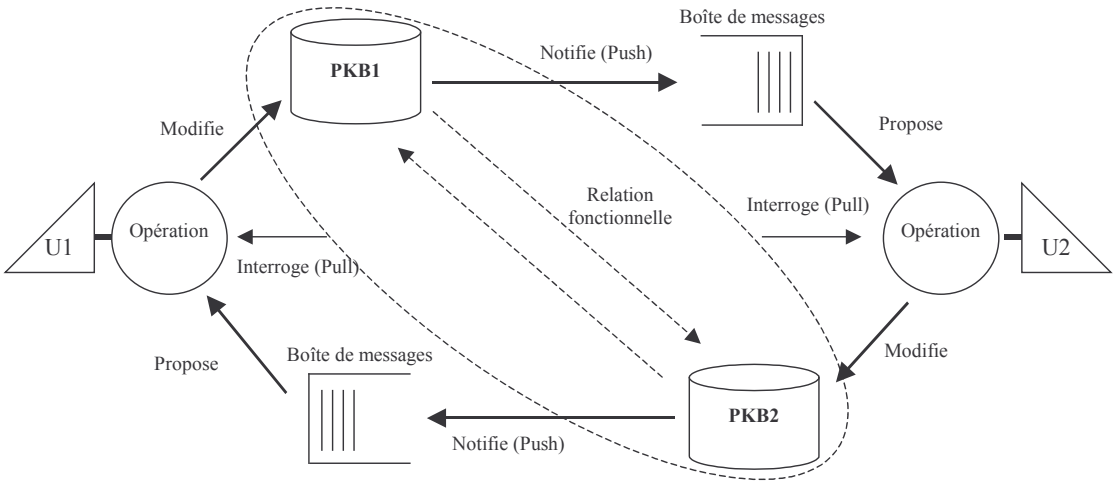


Figure 3.6 : Interrogation et notification dans le web-of-people

3.2 Schéma de connaissances

Dans cette section, nous spécifions un schéma d'échange pour les connaissances crées dans les PKB. Ce schéma commun permet effectivement d'assurer l'interopérabilité dans les échanges de connaissances. Il n'impose pas un choix de stockage dans l'implémentation de chaque PKB. Nous exploitons ici la représentation de RDF pour une telle modélisation de connaissances.

Basé sur l'usage d'un weblog pour chaque PKB, nous distinguons deux types de ressources : topique et post. Une telle ressource est identifiée par une URI et décrite sémantiquement à l'aide d'un ensemble de propriétés. La Figure 3.7 illustre l'ensemble du vocabulaire utilisé pour la description des PKB. Afin d'être lisible dans la présentation, nous utilisons d'ici et par la suite la syntaxe abrégée ns:vocab pour chaque vocabulaire. Par exemple, webop:Resource est la classe abstraite de ressources topique et post ; webop est le namespace du schéma[7].

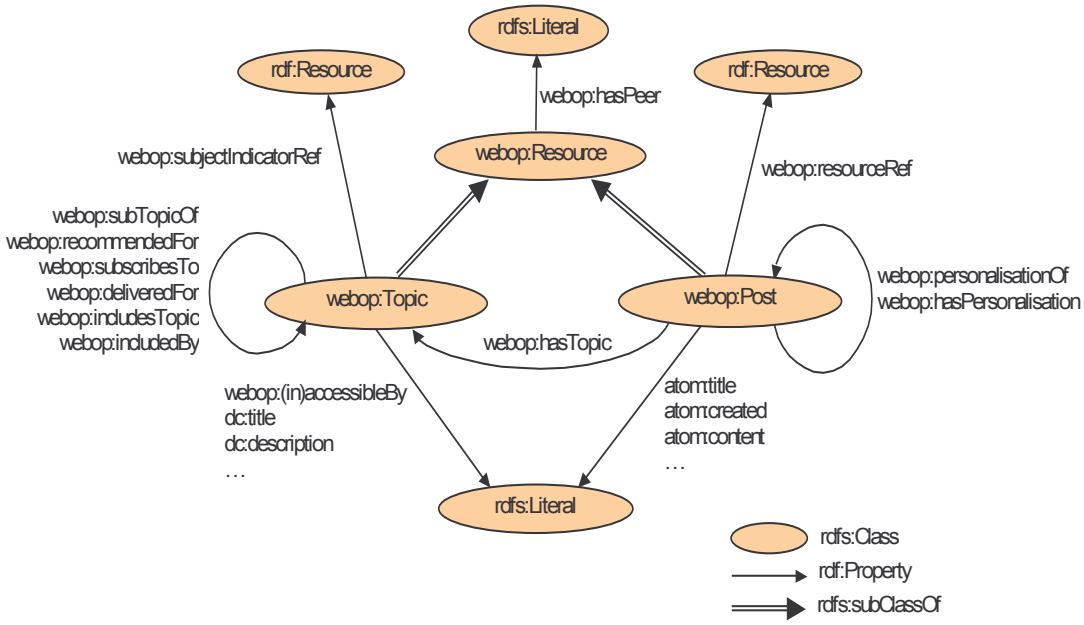


Figure 3.7 : Schéma de connaissances pour les weblogs partagés

Comme les topiques et les posts sont distribuées dans de multiples PKB, nous avons besoin d'une propriété pour décrire une telle distribution de ressources. C'est la propriété webop:hasPeer qui permet d'indiquer le pair (ou la PKB) dans lequel un

[7] webop="http://purl.org/webop/1.0/"

topique ou un post a été créé. La valeur de cette propriété est une identité sociale dans le web-of-people (e.g., *"ta$infres.enst.fr"*).

3.2.1 Post

Un post est décrit comme une instance de webop:Post. Nous employons un vocabulaire spécifique développé par AtomEnabled[8] pour adopter les descriptions de contenu d'un post. Atom est un format de syndication qui est encore en voie de maturation mais devrait prochainement être accepté par l'IETF comme une standardisation de publication personnelle sous forme de weblogs. Dans ce format de syndication, un post est décrit à l'aide des propriétés suivantes :

- atom:title - le titre du post
- atom:link - l'URL permanente du post dans le site web
- atom:created - la date et heure de création du post
- atom:modified - la date et heure de modification du post
- atom:issued - la date et heure de publication du post ; après cette publication le post ne pourra plus être modifié dans le web-of-people
- atom:summary - un résumé du contenu du post
- atom:content - pour décrire le contenu du post ; plusieurs contenus sont permis pour le post

Comme l'objectif du web-of-people est de faire échanger des méta-informations (i.e., information sur les ressources web) entre utilisateurs, un post peut référencer une ou plusieurs ressources par les descriptions webop:resourceRef. La valeur de cette propriété devrait être l'URL d'une ressource référencée. Etant donné un post *x* et une référence *url*, l'assertion (*x*, webop:resourceRef, *url*) signifie que *url* est référencée par *x*.

3.2.2 Topique

Un topique est une ressource instanciée du type webop:Topic. Chaque topique a une description par métadonnées pour capturer son titre, son texte de description, sa date de création,… Nous utilisons les propriétés au sein du vocabulaire Dublin Core (dc:title, dc:description, dc:date, …) pour cette description de métadonnées.

[8] http://www.atomenabled.org/

webop:subjectIndicatorRef est utilisé pour donner une description formelle d'un sujet de topique. Chaque topique peut avoir un indicateur de sujet qui le relie à un terme non ambigu dans une ontologie publique. Cette référence de sujet devrait être une URI telle qu'elle peut être définie dans le cadre de PSI (Published Subject Indicators) au sein d'OASIS [Pepper, 2003]. Dans le web-of-people nous pouvons exploiter de tels indicateurs pour retrouver des topiques qui parlent de même sujet.

Comme les topiques d'une PKB forment une hiérarchie taxonomique, entre eux il existe des relations de subsomption décrites par la propriété webop:subTopicOf. Supposons que *a* et *b* sont URI de deux topiques, l'assertion (*a*, webop:subTopicOf, *b*) dans la PKB indique que *a* est un sous topique de *b*, ou *b* subsume *a*. Notons que la subsomption entre topiques est réflexive et transitive. Lorsque *b* est subsumé par un topique *c*, *c* subsume également *a*.

Nous utilisons la propriété webop:hasTopic pour associer un post à un topique. Un post doit être associé avec au moins un topique de la taxonomie. Ainsi une instanciation webop:hasTopic permet à l'utilisateur de trouver un ensemble de posts interprétés sous un topique. L'interprétation d'un super topique (i.e., l'ensemble des posts instanciés) devrait inclure celle de ses sous-topiques. Par exemple, si *p* est un post associé avec *a* - (*p*, webop:hasTopic, *a*), *p* appartient alors à l'interprétation de *b* - (*p*, webop:hasTopic, *b*).

Créer puis mettre à jour sa PKB revient donc à gérer une base hiérarchique où les nœuds sont des topiques, les feuilles des posts, les uns et les autres décrits par les propriétés définies ci-dessus. Mais cette structure et ces propriétés ne sont a priori que des descriptions privées.

Aussi, afin de mettre les PKB – les weblogs – dans un contexte de partage et de réutilisation, nous spécifions maintenant des propriétés supplémentaires, à valeur dans des PKB étrangères, pour satisfaire les objectifs de :

(1) contrôle d'accès aux ressources partagées (webop:accessibleBy et webop:inaccessibleBy) ;

(2) intégration de topiques étrangers pour réutilisation (webop:includesTopic/ webop:includedBy) ;

(3) recommandation et abonnement (webop:recommendedFor, webop:subscribesTo/webop:deliveredFor) ; et enfin

(4) personnalisation de posts (revues) d'autres personnes pour la complémentarité de connaissances (webop:personalisationOf et webop:hasPersonalisation).

Nous voyons ci-après l'utilité et l'usage de ces propriétés.

3.2.3 Accessibilité

Partager des connaissances est une première fonctionnalité du web-of-people. Chaque personne peut ouvrir (accepter) pour d'autres l'accès en lecture de ses topiques par la propriété webop:accessibleBy. Bien que webop:accessibleBy ne s'applique qu'aux topiques, l'accès des posts, dès que la propriété atom:issued est renseignée (ce qui signifie qu'ils sont publiables), est aussi contrôlé via les topiques auxquels ils sont attachés. Notons que ce contrôle d'accès est pris en compte effectivement par le service d'interrogation du système qui ne pourra permettre la lecture que pour les propriétés des topiques et/ou posts accessibles.

Techniquement, la valeur d'une description d'accessibilité est une chaîne de caractères (string) indiquant l'identification de destinataires pour l'accessibilité, celle-ci peut être une seule personne (e.g., *"ta$infres.enst.fr"*), un (sous-)domaine (e.g., *"infres.enst.fr"*, *".enst.fr"*, *".fr"*), ou même tout le public (*"all"*). Le langage RDF permet de spécifier multi-valeurs par multi-descriptions sur l'accessibilité d'un topique. Par exemple, on peut rendre accessible un topique à deux personnes différentes, par exemple *ta$infres.enst.fr* et *saglio$infres.enst.fr*, par deux déclarions webop:accessibleBy dans la PKB.

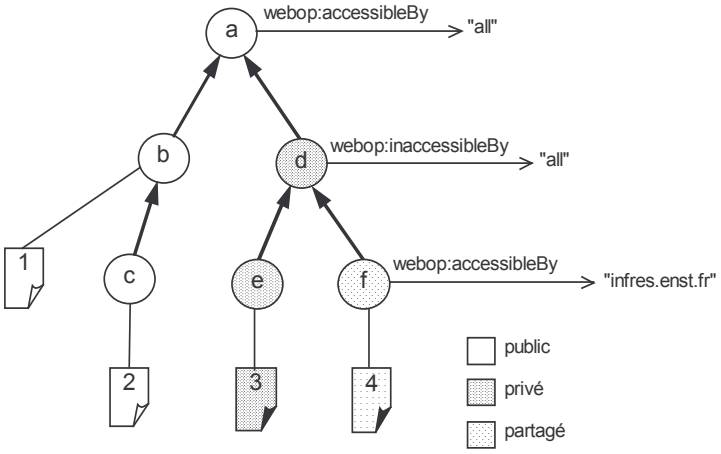


Figure 3.8 : Contrôle d'accès sur une PKB

Un topique héritera des valeurs de webop:accessibleBy de ses super-topiques. Ce topique sera donc accessible pour les utilisateurs qui peuvent accéder aux super-topiques. Pour en restreindre l'accès (refusé), on utilisera la propriété webop:inaccessibleBy. L'utilisateur ainsi désigné n'aura pas droit d'accès sur le topique même si son nom apparaît dans la liste de webop:accessibleBy. En pratique, le calcul pour l'ensemble de personnes pouvant accéder à un topique *a* se fait par la formule suivante.

$$access(a) = (access(parents(a)) \cup accessibleBy(a)) \setminus inaccessibleBy(a)$$

où *access*() est la fonction du calcul, *parents*() rend les pères du topique, *accessibleBy*() est la liste de personnes décries par webop:accessibleBy et *inaccessibleBy*() par webop:inaccessibleBy. Sur l'exemple dans la Figure 3.8, les topiques *a*, *b*, *c* et ainsi les posts associés avec eux sont accessibles pour le public, car $access(c) = access(b) = access(a) = all$; *d*, *e* sont privés, $access(e) = access(d) = access(a) \setminus all = \varnothing$; *f* est ouvert (partagé) pour les personnes limitées dans *infres.enst.fr*, $access(f) = access(d) \cup$ *infres.enst.fr*.

Bien que nous ne définissions que l'accessibilité sur les topiques, celle-ci est transparente également pour les posts. Un post est accessible si et seulement si l'un de ses topiques attachants est accessible.

3.2.4 Intégration

Dans la pratique d'une communauté, on essaie toujours de reprendre ce qui a été produit par d'autres afin d'éviter un double d'effort. Des techniques d'intégration peuvent permettre à l'utilisateur d'indiquer de différentes sources à consulter dans une recherche d'information. Comme une source de connaissances dans le web-of-people est toujours un topique dans une PKB, un utilisateur peut vouloir associer un topique de sa PKB avec celui d'une autre qui lui est accessible. Grâce à cette « intégration », toute recherche de topique pourra s'étendre à des topiques externes.

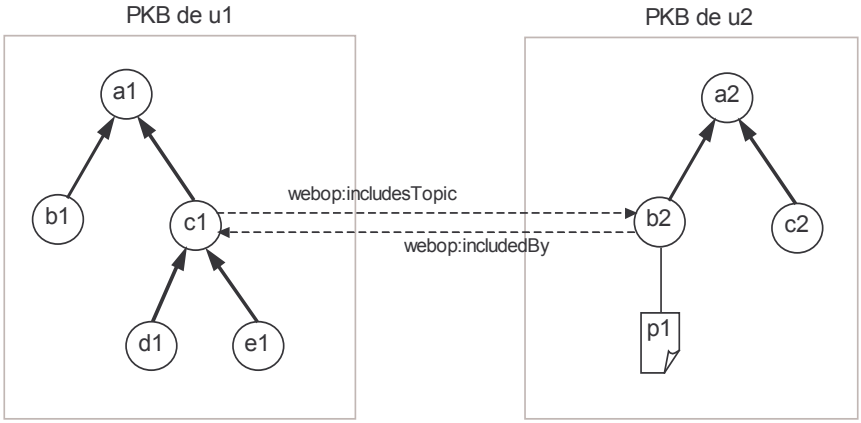


Figure 3.9 : Intégrer un topique d'une autre PKB

La Figure 3.9 illustre une articulation entre *c1* et *b2* par la propriété webop:includesTopic – (*c1*, webop:includesTopic, *b2*). Notons que *b2* est un topique étranger. Une interrogation de ce topique effectuera un appel à distance. Pour clarifier la présentation, nous illustrons d'ici et ci-après par flèches pointillées des descriptions dont l'objet est une ressource étrangère. Afin d'avoir un miroir de relation, dès que (*c1*, webop:includesTopic, *b2*) est inséré dans la PKB de *u1*, une notification

est envoyée à *u2* alors il pourra ajouter (*b2*, webop:includedBy, *c1*) dans sa PKB ; webop:includedBy est simplement la propriété inverse de webop:includesTopic. En fait, dans une métrique sociale, le couplage des propriétés permet de calculer la « notoriété » d'un topique.

En général, la recherche des posts sous un topique devrait s'exécuter jusqu'à la fermeture transitive des relations webop:subTopicOf et webop:includesTopic. Cette fermeture permet de parcourir pour une même recherche les sous-topiques qui ont été éventuellement intégrés de plusieurs sources différentes. Par exemple, dès lors que l'intégration se fait entre *c1* et *b2*, une recherche des posts sous *c1* se poursuivra par celle des posts attachés à *b2* (e.g., *p1*).

3.2.5 Recommandation active et abonnement

Une pratique courante chez les utilisateurs d'une communauté est d'utiliser l'email pour envoyer des pointeurs sur des documents intéressants pour des collègues ou des amis. On appelle cette pratique une recommandation active pour attirer l'attention à l'initiative de l'offreur [Maltz et Ehrlich, 1995]. Dans le web-of-people, on s'efforce à ce que, chaque fois qu'un nouveau post est créé, l'auteur l'adresse aux personnes qu'il sait être intéressées. Cependant, cette action requiert un effort relativement important de la part de l'expéditeur, et il arrive souvent que l'utilisateur n'envoie pas la référence à toutes les personnes qu'elle pourrait intéresser, ou qu'il oublie simplement de le faire.

Pour assister l'utilisateur dans cette action, nous utiliserons un mécanisme d'*adressage sémantique* qui exploitera automatiquement la mémoire des recommandations pour chaque topique de la PKB. Ainsi, lorsqu'un post sera créé associé à ce topique, le service de notification du web-of-people adressera automatiquement un message aux personnes qui ont reçu une recommandation de ce topique. Grâce à cet adressage sémantique, l'utilisateur est libéré du travail d'envoi de messages. Il doit simplement enregistrer dans sa PKB la destination des recommandations.

Dans notre conception, la cible pour la recommandation d'un topique n'est pas simplement une personne, mais aussi le topique du destinataire que l'expéditeur trouve le plus proche du sien. Une recommandation doit donc choisir parmi plusieurs topiques existant chez le destinataire. En conséquence, si l'expéditeur ne sait pas exactement pour quel topique il convient de donner sa recommandation, il ne peut que le lui rendre accessible [9].

[9] Toutefois, pour faciliter les échanges entre personnes, nous pouvons également envisager dans l'application un topique spécial - nommé par exemple « inbox » - créé pour chaque PKB, celui-ci serait public et prêt à recevoir tout type de recommandation. Mais il risquerait d'être la cible privilégiée du « spam ».

Supposons que les deux PKB dans la Figure 3.10 sont accessibles par l'un et l'autre. Une recommandation de la part de *u1* commence par une déclaration locale (*c1*, webop:recommendedFor, *b2*). Cette première étape déclenche une notification à *u2* pour dire que *c1* est recommandé pour *b2*. Si *u2* accepte la recommandation il répondra par une déclaration (*b2*, webop:subscribesTo, *c1*). Ce feedback suggérera en retour une action de la part de *u1*, pour qu'il crée un abonnement par adressage sémantique sur son topique – (*c1*, webop:deliveredFor, *b2*). Cet abonnement permettra l'envoi automatisé de notification à chaque création de posts dans le sous-arbre de *c1*, par exemple *p1*.

Pratiquement, la dialogue de création d'abonnement repose sur le mécanisme de « handshake » entre les utilisateurs. L'envoi de notification des nouveaux posts ne s'effectuera que si persiste l'acceptation du destinataire. La confiance sur les échanges entre utilisateurs devra donc être assurée par l'application pour que l'utilisateur ne reçoive que des notifications qui l'intéressent. Si l'expéditeur n'attendait pas l'acceptation de la part du récepteur, les envois suivants – qui seraient alors du « spam » - pourraient être filtrés automatiquement par l'application. L'application ne conservera que les messages concernant des abonnements à des topiques acceptés par une déclaration webop:subscribesTo.

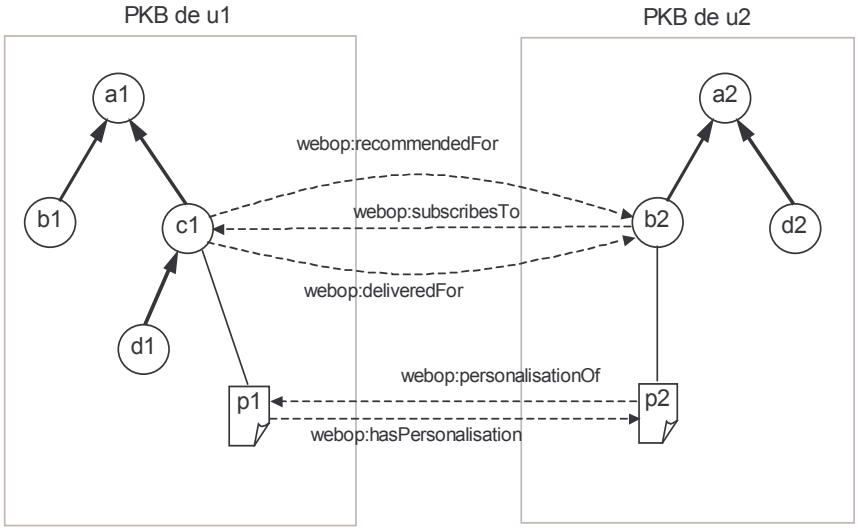


Figure 3.10 : Adressage sémantique pour une recommandation active

3.2.6 Personnalisation

L'échange le plus actif dans le web-of-people se fait par la personnalisation de post (la différence entre cette dernière et l'intégration de topiques est que l'on peut écrire des commentaires sur un post existant, non plus simplement le lire). Lorsqu'un post est une personnalisation d'un autre, on peut considérer ce dernier comme une citation du

premier avec de nouveaux commentaires. Par exemple, dans la Figure 3.10, dès que *u2* reçoit la notification concernant la création du post *p1*, il ajoute un commentaire en créant un post *p2* qui est une personnalisation de *p1* – (*p2*, webop:personalisationOf, *p1*). Comme webop:includedBy, la relation webop:hasPersonalisation – (*p1*, webop:hasPersonalisation, *p2*), est simplement la réciproque de webop:personalisationOf. Elle indique que le créateur d'un post a bien noté que son post a été personnalisé ailleurs. Dans une métrique sociale le couplage des propriétés webop:personalisationOf permet de calculer le « rayonnement » d'un post. En pratique, la personnalisation des posts peut être comparée à la technique du Trackback [Trott, 2002] utilisée dans les weblogs, par laquelle un post peut en référencer un autre et un lien inverse être posé sur le trackback du référent.

La personnalisation est une reconnaissance des sources (« crédit »). Sa déclaration automatique et persistante est une des bases de l'établissement de la confiance dans le réseau web-of-people. Grâce à elle un auteur peut mesurer sa popularité non plus seulement parce qu'il est lu par d'autres – en parcourant les pointeurs webop:includedBy – mais aussi parce qu'il est cité dans de nouveaux écrits – webop:hasPersonalisation.

Dans cette section, nous avons vu que tout échange possible dans le web-of-people doit être représenté par des relations constituées de liens sémantiques entre topiques de différentes PKB. La restriction des échanges, avec de tels liens sémantiques, permettait la confiance des intéressés dès le départ de communication, car seule la communication « honnête » et le respect mutuel assurent le bonheur partagé. Mais la complétude de ces liens sémantiques pour que les applications puissent assurer la communication honnête est une question qui reste encore ouverte.

3.3 URI des ressources et exemple de descriptions

Dans la perspective du web sémantique, les choses doivent être clairement identifiées. Les URI permettent de définir et de localiser sans ambiguïté sous une forme standard toutes les ressources physiques ou abstraites disponibles sur Internet. Les deux types d'URI sont les URL (Universal Resource Location) et les URN (Universal Resource Name).

En principe, une PKB peut exploiter n'importe quelle forme d'URI pour ses ressources, i.e., URN ou URL. Néanmoins, les URN ne conviennent qu'aux ressources sans rattachement à un endroit. Par exemple, un livre peut être identifié par un numéro ISBN et son URN devrait être de la forme `urn:isbn:xxx`. Quant aux ressources attachées à un endroit sur le web, leur URI prennent souvent un type d'URL. Une URL permet de déterminer précisément l'endroit et le protocole utilisé pour accéder à la ressource référencée.

Comme les ressources dans le web-of-people sont attachées à la PKB de l'auteur, nous proposons un format d'URL commun utilisé pour identifier les ressources. Supposons que toute ressource devrait être assignée un nom unique `<rid>` dans sa PKB. L'URL de chacune doit suivre la syntaxe :

```
<protocol>://<uid>/<rid>
```

où `<uid>` est le nom de la PKB et `<protocol>` est le nom du protocole utilisé pour le service d'interrogation. Revenons à l'exemple de PKB donné dans la Figure 3.4, `webop://u1$odp/Software` pourrait être généré comme l'URI du topique *Software* dans la PKB de l'utilisateur *u1$odp* ; *webop* est le nom du protocole d'accès.

Aussi, afin de faciliter la navigation dans le réseau des ressources, nous créons un topique spécial comme le racine pour chaque PKB. `<protocol>://<uid>` est l'URI de ce topique racine. De ce fait, on peut choisir facilement un point de départ pour navigation en composant l'identifiant d'une PKB donnée. Par exemple, `webop://u1$odp` sera l'URI du topique racine dans la PKB de l'utilisateur *u1$odp*. La Figure 3.11 présente l'ensemble des URI générées pour les topiques dans cette PKB. Nous appliquons ici la génération des noms locaux sous forme de chemin comme les noms des répertoires Unix.

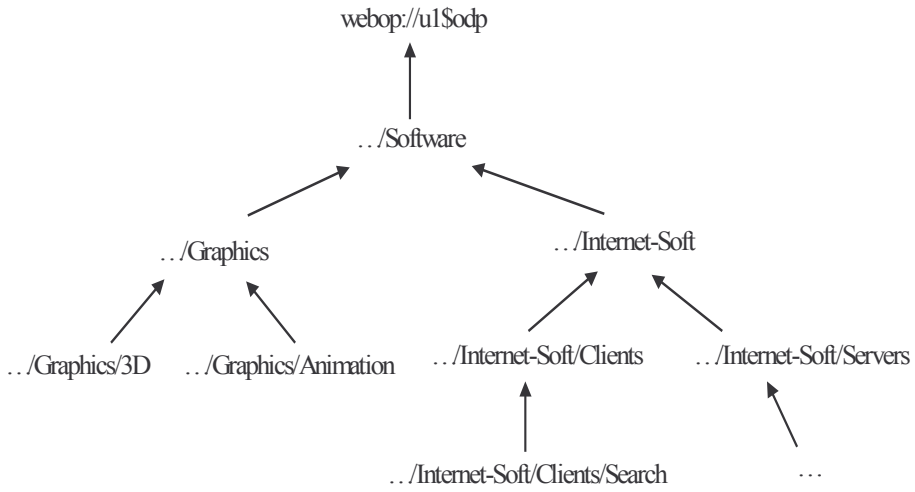


Figure 3.11 : Génération des URI de topiques

Les URI des ressources sont utilisées dans les descriptions RDF pour celles-ci. La Figure 3.12 donne une description RDF décrite pour le topique *Internet-Soft* dans l'exemple de la Figure 3.4 (`webop://u1$odp/Software/Internet-Soft` est son URI). La description doit se conformer au schéma de connaissances que nous avons spécifié au-dessus. Le schéma Dublin Core est exploité pour décrire le titre, la date de création du topique. La PKB à laquelle il appartient est celle de *u1$odp*. Il est un sous topique du topique local *Software* (`webop://u1$odp/Software`). Enfin, ce topique est inclus par le

topique étranger *Internet* (`webop://u2$odp/Internet`) ; i.e., il est intégré comme un sous topique de ce dernier.

```
<webop:Topic rdf:about="webop://u1$odp/Software/Internet-Soft">
  <dc:title>Software for Internet</dc:title>
  <dc:date>2004-04-02</dc:date>
  <webop:subTopicOf rdf:resource="webop://u1$odp/Software"/>
  <webop:hasPeer>u1$odp</webop:hasPeer>
  <webop:includedBy rdf:resource="webop://u2$odp/Internet"/>
</webop:Topic>
```

Figure 3.12 : Exemple de descriptions RDF pour un topique

La Figure 3.13 donne un autre exemple de descriptions RDF pour un post. Etant donné `webop://u1$odp/&1` est l'URI du post *&1* dans la Figure 3.4. Le post est décrit avec des propriétés atom pour son titre, sa date de création et son contenu de description. `http://www.searchrocket.com` est l'URL de la ressource référencée par ce post. Il est attaché au topique *Search* de la PKB (`webop://u1$odp/Software/Internet-Soft/Clients/Search`).

```
<webop:Post rdf:about="webop://u1$odp/&1">
  <atom:title>SearchRocket</atom:title>
  <atom:created>2004-07-02</atom:created>
  <atom:content>A free Internet utility...</atom:content>
  <webop:resourceRef rdf:resource="http://www.searchrocket.com"/>
  <webop:hasTopic
 rdf:resource="webop://u1$odp/Software/Internet-Soft/Clients/Search"/>
  <webop:hasPeer>u1$odp</webop:hasPeer>
</webop:Post>
```

Figure 3.13 : Exemple de descriptions RDF pour un post

3.4 Service d'interrogation

Nous spécifions dans cette section un protocole d'accès logique « à la LDAP » qui sert pour le service d'interrogation dans le web-of-people. Ce protocole doit être indépendant de l'implémentation de chaque PKB. Il définit les paramètres nécessaires pour accéder à une ou plusieurs ressources dans une PKB. Le résultat d'accès est un flux qui contient des descriptions des ressources accessibles. Ces descriptions doivent se conformer au schéma de connaissances spécifié au-dessus. Par contre, le développeur d'applications respectant le protocole d'accès garde la liberté du choix de stockage physique des PKB. Chaque accès à la LDAP sera transformé en une ou plusieurs

requêtes appropriées selon le modèle de stockage choisi et d'accès physique (relationnel ou semi-structuré).

3.4.1 Protocole d'accès

Pour accéder aux ressources dans une PKB, il faudra faire appel au service d'interrogation fourni par celle-ci. Chaque appel rendra comme résultat un flux RDF/XML qui contiendra des descriptions sur les ressources accessibles. Un accès devra porter sur les paramètres suivants :

- *URI* est l'URI d'une ressource dans la PKB prise comme point d'entrée pour l'accès. Ce point d'entrée peut être un topique ou un post.
- *Scope* détermine l'étendue de recherche relativement au point d'entrée. L'étendue doit avoir l'une des valeurs suivantes : **base** pour accéder seulement à la ressource d'entrée, **one** aux fils du point d'entrée, **sub** aux descendants dans le sous arbre du point d'entrée.
- *Filter* spécifie la sélection d'un sous-ensemble de ressources parmi toutes celles trouvées dans l'étendue de recherche, par exemple, seulement des posts créés depuis une année donnée.
- *Projection* est la liste de propriétés que l'on veut afficher dans le résultat. Par défaut, toute description visible pour les ressources accessibles sera affichée dans le résultat.

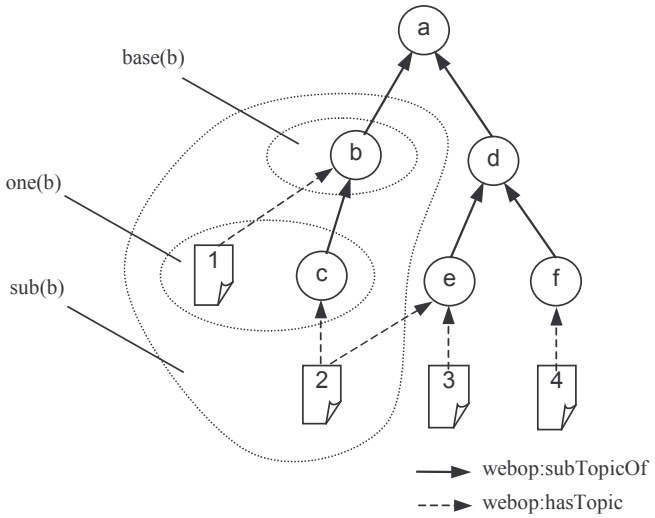


Figure 3.14 : Etendue de recherche sur la vue hiérarchique d'une PKB

Le protocole d'accès ci-dessus s'inspire du protocole LDAP (RFC2251) qui simplifie la navigation dans une PKB dans la mesure où toute PKB est une vue hiérarchique de ressources. La Figure 3.14 illustre trois étendues possibles sur un point d'entrée (*b*). Ces

accès simples permettent effectivement de descendre dans la vue hiérarchique de ressources. Par exemple pour trouver les sous topiques directs de *b*, on doit employer un accès à l'étendue **one** avec un filtre sur topiques. En revanche, pour monter dans la vue hiérarchique, il ne faut que regarder simplement des descriptions webop:subTopicOf si la ressource est un topique, webop:hasTopic si c'est un post. Dans l'exemple donné, l'accès à l'étendue **base** sur *b* permet de savoir son père (*a*).

Le protocole d'accès proposé peut servir essentiellement à la navigation mais aussi à la recherche simple de ressources. Le filtre optionnel d'un accès permet de ne retrouver dans l'étendue de recherche que des ressources choisies. Ce filtre est une expression booléenne dont les éléments sont des filtres atomiques. Le format de cette expression est conformé à celui du LDAP spécifié dans RFC2254. Un filtre atomique permet de sélectionner des ressources sur un critère de description. Par exemple, `([rdf:type]=[webop:Post])` indique une sélection des seuls posts tandis que `([atom:title]=*peer-to-peer*)` permet de trouver les posts ayant un titre contenant le mot clé « peer-to-peer ». La conjonction de ces deux filtres est écrite sous forme préfixe comme ceci

```
(& ([rdf:type]=[webop:Post]) ([atom:title]=*peer-to-peer*))
```

Dans le format de filtre, un attribut de sélection est toujours une propriété de description (i.e., une URI). Nous pouvons donc utiliser une syntaxe brève pour les vocabulaires standards.

```
[rdf:xxx] = http://www.w3.org/1999/02/22-rdf-syntax-ns#xxx
[rdfs:xxx] = http://www.w3.org/2000/01/rdf-schema#xxx
[atom:xxx] = http://purl.org/atom/1.0/xxx
[dc:xxx] = http://purl.org/dc/elements/1.1/xxx
[webop:xxx] = http://purl.org/webop/1.0/xxx
```

Nous pouvons appliquer également la syntaxe brève pour décrire des propriétés de projection. Par exemple, la liste de projection `{[atom:title], [atom:content]}` spécifie un accès uniquement pour le titre et le contenu de posts.

3.4.2 Authentification et restriction d'accès

Tout utilisateur demandant accès aux ressources d'une PKB doit être authentifié. Si l'authentification est impossible, on le considère comme un anonyme, i.e., il n'a que le droit du public pour l'accès aux ressources.

Un accès rend comme résultat seulement des descriptions accessibles par l'utilisateur. Nous distinguons trois types de descriptions sur les ressources :

- Les propriétés visibles – ce sont les métadonnées Atom pour les posts, Dublin Core pour les topiques et des propriétés particulières du schéma webop ; elles consistent pour les posts en webop:resourceRef,

webop:personalisationOf, webop :hasPersonalisation et webop:hasTopic ; pour les topiques en webop:subjectIndicatorRef, webop:referencingAllowed et webop:subTopicOf. Les descriptions pour ces propriétés seront rendues visibles si l'accessibilité est ouverte sur cette ressource.

- Les propriétés invisibles – ce sont les propriétés utilisées pour le contrôle d'accès webop:accessibleBy et webop:inaccessibleBy.

- Les propriétés « sociales » – il s'agit des relations qui impliquent des échanges sociaux entre topiques. Les propriétés webop:includesTopic et webop:subscribesTo décrivent un topique dans son rôle de « consommateur », tandis que webop:recommendedFor, webop:includedBy et webop:deliveredFor sont utilisés pour son rôle de « producteur ». Il peut exister des cas où l'on veut cacher les relations sociales sur un topique productif. Dans ce cas la description webop:referencingAllowed est mise à valeur FALSE pour ce topique, ce que devra avoir pour effet que toutes les descriptions sociales de topiques externes consommateurs reliées à celui-ci doivent devenir invisibles. Concrètement, on ne doit pouvoir accéder pour ces topiques externes aux descriptions webop:includesTopic et webop:subscribesTo que si webop:referencingAllowed est TRUE du côté du topique productif. Ces descriptions deviennent seulement alors des descriptions visibles.

```
<webop:Topic rdf:about="webop://u2$odp/Internet">
  <dc:title>Internet</dc:title>
  <dc:date>2004-02-02</dc:date>
  <webop:subTopicOf rdf:resource="webop://u2$odp"/>
  <webop:hasPeer>u2$odp</webop:hasPeer>
  <webop:includesTopic
         rdf:resource="webop://u1$odp/Software/Internet-Soft"/>
</webop:Topic>
```

Figure 3.15 : Flux de réponse à un accès

La Figure 3.15 illustre le flux de réponse à un accès pour le topique `webop://u2$odp/Internet`. Dans ce flux nous ne voyons que les descriptions visibles, et aussi celle pour la propriété sociale webop:includesTopic. Cependant, la description webop:includesTopic sera cachée pour cet accès si webop:referencingAllowed est mise à valeur FALSE pour le topique `webop://u1$odp/Software/Internet-Soft`. Cela implique que la PKB interrogée (*u2$odp*) doit savoir si cette inclusion est visible. Il n'y a pour cela que deux façons de faire : soit elle (re-)interroge à son tour la PKB du topique inclus (*u1$odp*), soit elle mémorise localement cette information. La deuxième stratégie offre plus de

performance d'accès aux propriétés sociales, mais elle nécessite un mécanisme de notification supplémentaire afin d'assurer la mise à jour de la mémoire locale en cohérence avec la PKB du topique inclus.

3.5 Service de notification

Dans cette section, nous définissons le format des messages qui est exploité pour le service de notification. Un message est structuré comme étant une ressource décrite sous forme de descriptions RDF. Grâce à cette représentation en RDF, le contenu d'un message est modélisé facilement pour rapporter un évènement de changement de PKB (i.e., un pointeur sur création ou suppression de triplets RDF). Nous verrons ensuite la liste des évènements à notifier par message dans le web-of-people. Ces notifications devront suffies pour assurer des transactions d'échanges entre PKB.

3.5.1 Format des messages

Chaque notification dans le web-of-people correspond à un message à envoyer de l'expéditeur au destinataire de la notification. Afin de faciliter le traitement de notifications dans l'application, les messages devront être structurés et bien représenter les évènements de notification. Nous spécifions ici une structuration des messages à base de RDF. Chaque message sera donc modélisé comme une ressource décrite avec les propriétés ci-dessous (cf. Figure 3.16) :

- webop:from - l'émetteur du message
- webop:to - le destinataire du message
- webop:date - la date d'envoi du message
- webop:body - le contenu du message, un évènement du type webop:Event
- webop:text - un texte alternatif attaché pour la notification
- webop:feedbackTo - liste optionnelle d'utilisateurs que l'émetteur souhaite faire informer de la réponse – autrement dit être destinataire de la notification de la réaction (feedback) – à ce message

Essentiellement, le contenu d'un message devra représenter l'évènement qu'il est nécessaire de notifier au destinataire. Cet évènement décrit un changement d'état effectif sur la PKB de l'émetteur qui peut concerner éventuellement le comportement du destinataire. Comme toute PKB peut être modélisée sous le schéma de connaissances spécifié dans ce chapitre, le changement d'état d'une PKB devra correspondre à une mise à jour de descriptions (triplets) RDF. Dans le format de message, une création d'un triplet est décrite en utilisant la propriété webop:createdStmt avec la valeur d'une réification. Cette réification permet d'indiquer précisément le triplet ajouté dans la PKB.

De même manière, la propriété webop:deletedStmt permet de décrire une suppression dans la PKB. On verra plus loin que d'autres réifications complémentaires puissent ajouter des informations utiles dans un message de notification.

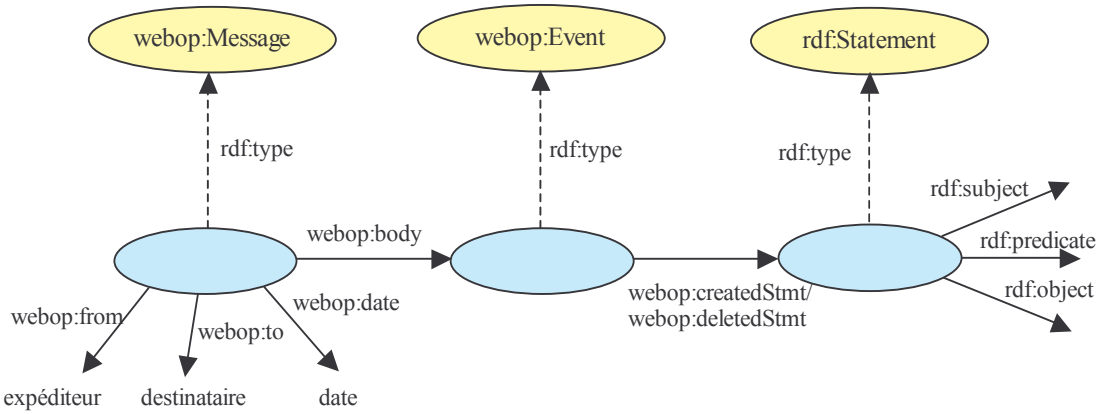


Figure 3.16 : Structure des descriptions d'un message

3.5.2 Evènements

Par définition, un évènement correspond à un changement d'état de la PKB d'une personne. Il s'agit d'une action de celle-ci qui peut provoquer la réaction d'une autre. Dans le Tableau 3.1 nous spécifions des évènements nécessairement notifiés dans le web-of-people. Chaque évènement se présente comme un trigger dans la PKB (i.e., un fait de mise à jour sans/sous condition). Le symbole "+" correspond une insertion de triplets dans la PKB, "-" une suppression. Pour simplifier la présentation, nous supposons topique *a1* et post *p1* créés par *u1* ; *a2* et *p2* par *u2*. Une notification de la part de *u1* appelle une réaction de la part de *u2* dans la mise à jour de sa PKB. Il appartient aux applications de choisir le mode de traitement de cette suggestion par l'utilisateur destinataire *u2*.

Dans ce tableau, nous avons indiqué les mises à jour proposées pour chaque notification reçue. A la réception d'un message, l'utilisateur décidera entre les mises à jour possibles dans sa PKB. Par exemple, si *u2* reçoit une recommandation de la part de *u1* – (*a1*, webop:recommendedFor, *a2*), il peut décider d'accepter l'abonnement pour cette recommandation – (*a2*, webop:subscribesTo, *a1*), et/ou d'intégrer seulement le topique reçu dans le sien – (*a2*, webop:includesTopic, *a1*). Finalement, le choix de réaction de l'utilisateur pour chaque évènement exprimera son comportement sur les échanges sociaux.

Toutefois, il y a également des mises à jour qui pourront être automatisées, sans intervention de l'utilisateur, par une routine de traitement de messages. Celles des réactions attendues que nous avons marquées d'une "*" dans le tableau sont les seules

que le système retient comme automatisées pour la cohérence des PKB en miroir les unes des autres.

Action de *u1*	**Notification de *u1* à *u2***	**Réaction attendue de *u2***
+(*a1*, webop:recommendedFor, *a2*)	Recommandation de lecture	+(*a2*, webop:includesTopic, *a1*) +(*a2*, webop:subscribesTo, *a1*)
+(*a1*, webop:includesTopic, *a2*)	Création d'une intégration	+(*a2*, webop:includedBy, *a1*)*
-(*a1*, webop:includesTopic, *a2*)	Suppression d'une intégration	-(*a2*, webop:includedBy, *a1*)*
+(*a1*, webop:subscribesTo, *a2*)	Acceptation d'un abonnement	+(*a2*, webop:deliveryFor, *a1*)
-(*a1*, webop:subscribesTo, *a2*)	Résiliation d'un abonnement	-(*a2*, webop:deliveryFor, *a1*)
+(*a1*, webop:deliveryFor, *a2*)	Création d'un abonnement	-
-(*a1*, webop:deliveryFor, *a2*)	Suppression d'un abonnement	-
+(*p1*, webop:issued, *d*) AND EXISTS (*p1*, webop:hasTopic, *a1*) & (*a1*, webop:deliveryFor, *a2*)	Nouveau post issued dans le cadre de l'abonnement de *a2*	+(*p2*, webop:personalisationOf, *p1*)
+(*p1*, webop:personalisationOf, *p2*)	Création d'une personnalisation	+(*p2*, webop:hasPersonalisation, *p1*)*

Tableau 3.1 : Liste des notifications nécessaires entre PKB

Notons que la notification de nouveaux posts dans le cadre d'un abonnement n'est pas effectuée dès sa création mais seulement pour les posts qui deviennent publiés. Un post n'est visible au public qu'après sa publication (i.e., la propriété atom:issued est mise en valeur pour celui-ci). Un post publié ne sera plus modifiable afin d'éviter le cas où une autre personne a personnalisé ce post sur la base de son contenu actuel alors qu'il continuerait d'évoluer. Nous considérons donc un post modifiable comme étant en cours de rédaction.

Un message de notification doit rapporter un évènement relatif à une des actions ci-dessus. Chaque évènement est décrit comme une création (webop:createdStmt) ou suppression (webop:deletedStmt) d'un triplet principal. Par exemple, la Figure 3.17 illustre un message de la part de *u1$odp* envoyé à *u2$odp* pour notifier la publication du post *&1* (`webop://u1$odp/&1`). Le triplet principal de l'évènement concerne donc la création de la propriété atom:issued pour ce post. Des descriptions, autres que le triplet principal, peuvent aussi être décrites à l'aide de la propriété webop:stmt dans le contenu du message. Ce sont des descriptions supplémentaires qui aident le récepteur du message à traiter l'évènement rapporté. Dans le message de l'exemple, une réification

sur la propriété webop:hasTopic a été ajoutée pour préciser le topique auquel le post est associé.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:webop="http://purl.org/webop/1.0/">
<webop:Message>
  <webop:from>u1$odp</webop:from>
  <webop:to>u2$odp</webop:to>
  <webop:date>2004-05-18T23:10:13-05:00</webop:date>
  <webop:body><webop:Event>
    <webop:createdStmt><rdf:Statement>
      <rdf:subject rdf:resource="webop://u1$odp/&1"/>
      <rdf:predicate rdf:resource="http://purl.org/atom/ns#issued"/>
      <rdf:object>2004-07-02</rdf:object>
    </rdf:Statement></webop:createdStmt>
    <webop:stmt><rdf:Statement>
      <rdf:subject rdf:resource="webop://u1$odp/&1"/>
      <rdf:predicate
           rdf:resource="http://purl.org/webop/1.0/hasTopic"/>
      <rdf:object rdf:resource=
           "webop://u1$odp/Software/Internet-Soft/Clients/Search"/>
    </rdf:Statement></webop:stmt>
    <webop:stmt><rdf:Statement>
      <rdf:subject rdf:resource="webop://u1$odp/&1"/>
      <rdf:predicate
           rdf:resource="http://purl.org/webop/1.0/hasPeer"/>
      <rdf:object>u1$odp</rdf:object>
    </rdf:Statement></webop:stmt>
  </webop:Event></webop:body>
</webop:Message>
</rdf:RDF>
```

Figure 3.17 : Un message de notification au format RDF/XML

Chapitre 4

Prototypage

Dans ce chapitre, nous décrivons le prototype que nous avons réalisé pour le web-of-people. Il s'agit d'un prototype de validation qui se conforme au cadre de conception spécifié dans le chapitre précédent. Le prototype nous a permis d'expérimenter différents scénarios d'échange dans le réseau social du web-of-people. Nous pensons avoir montré que les fonctionnalités supportées par les services de base d'interrogation et de notification sont suffisamment efficaces pour motiver des échanges de confiance dans le web-of-people.

D'un point de vue logiciel, nous avons adopté une implémentation à base de services web pour le prototypage. Au lieu de développer un pair client pour chacun, nous avons construit un serveur en ligne qui est capable d'héberger un groupe d'utilisateurs. Un nombre illimité de tels serveurs connectés est installé pour le fonctionnement du réseau d'échange. Un utilisateur ne peut s'enregistrer que sur un seul serveur d'hébergement, par défaut celui réservé à son groupe de travail. Une application web qui tourne sur ce serveur permet à l'utilisateur de s'y connecter pour gérer en ligne sa base de connaissances partagée. L'avantage de cette approche est double : (i) l'utilisateur final a besoin simplement d'un navigateur web pour pouvoir participer au web-of-people ; (ii) la participation de l'utilisateur est toujours « active » car sa PKB ainsi que sa boite de messages sont gérées en ligne. De plus, le mode d'hébergement est aussi celui le plus fréquent utilisé dans l'usage actuel des weblogs.

Cette implémentation a été faite dans le cadre d'un contrat avec FT R&D. Nous avons choisi Apache Axis pour la plate-forme de services web. Ainsi, chaque serveur d'hébergement est simplement un serveur HTTP. La PKB de chaque utilisateur enregistré sur ce serveur est stockée localement dans un modèle relationnel de triplets RDF. L'accès aux données basées sur ce modèle se fait à l'aide de l'API Jena [McBride, 2002]. Ce chapitre n'abordera que la conception générale de notre prototype. Pour une présentation technique détaillée les lecteurs peuvent néanmoins se rapporter à la spécification [Dao *et al.*, 2004]. Cette conception détaillée et le code sont propriété de FT R&D.

4.1 Architecture et implémentation du prototype

La Figure 4.1 donne l'architecture logicielle à base de services web adoptée pour le prototype de validation. Le prototype, nommé Webop, est un réseau distribué de multi-serveurs webop. Chaque serveur est au service d'un groupe d'utilisateurs. Comme serveur d'hébergement, il gère en ligne la PKB ainsi que la boite de messages de tout abonné qui peut s'y connecter via une application web. Cette application est une interface homme-machine qui permet à l'utilisateur de (i) gérer son weblog (sa PKB), (ii) naviguer et rechercher des connaissances dans le réseau de PKB, et (iii) gérer les notifications des autres utilisateurs avec lesquels il est en relation.

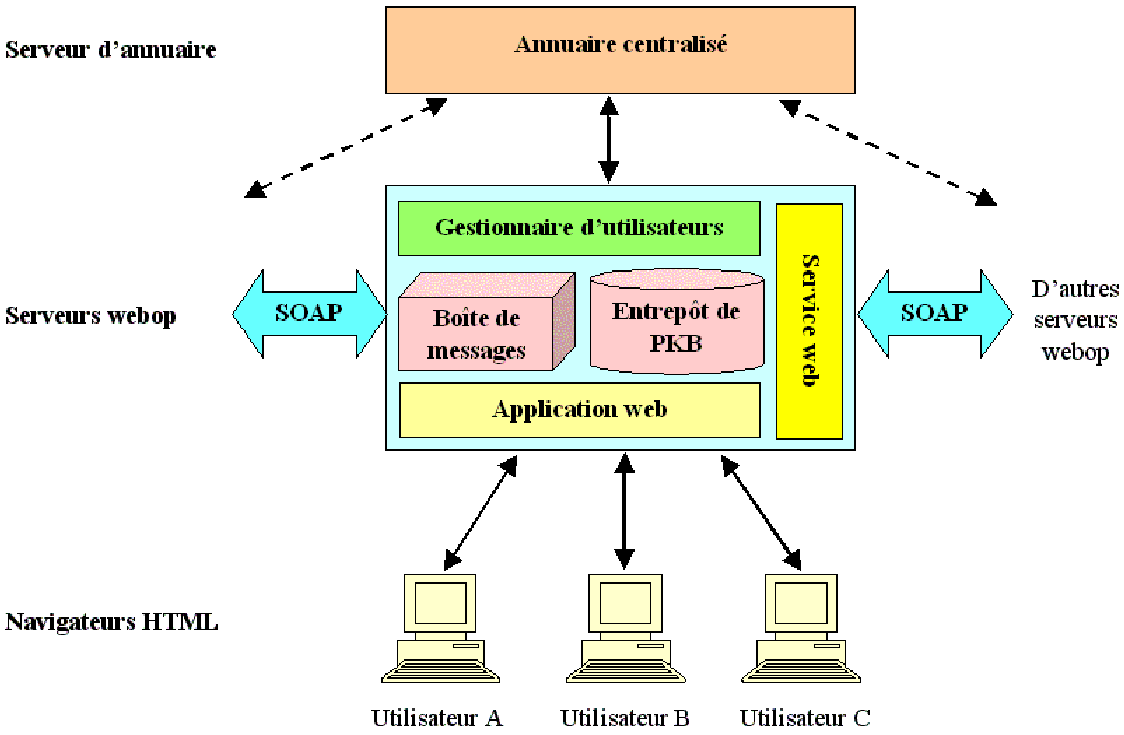


Figure 4.1 : Architecture logicielle du prototype

Bien qu'un utilisateur soit hébergé dans un serveur, son identité est transparente dans tout le réseau distribué des serveurs webop. Cette identité est gérée dans un annuaire centralisé assurant que chacun des utilisateurs est rattaché à un seul serveur d'hébergement. Chaque serveur webop dispose d'un gestionnaire d'utilisateurs connecté à cet annuaire. Dès lors qu'un nouvel utilisateur s'enregistre sur le serveur, le gestionnaire demandera la création d'une entrée dans l'annuaire pour celui-ci ; évidemment l'identifiant réservé doit être encore disponible.

Chaque serveur webop fournit un service web permettant les échanges entre serveurs webop. Ce service web constitue une implémentation basée sur le protocole SOAP pour les deux méthodes de communication entre pairs, l'interrogation et la notification. Pour

interroger la PKB de quelqu'un (ou lui notifier un évènement), il faudra invoquer le service web fourni à l'adresse de son serveur d'hébergement. Techniquement, tout appel de services web sera encadré dans un flux XML au protocole SOAP qui est indépendant de la couche de transport entre serveurs.

Dans le cadre de conception pour le web-of-people, on n'imposait pas un modèle de stockage pour les PKB et les boites de messages des utilisateurs. Comme un modèle de triplets est bien adapté à la fois au schéma RDF de PKB et à celui de messages, nous pouvons employer un entrepôt RDF pour gérer les PKB ainsi que les boîtes de messages dans un serveur webop. Nous avons choisi Jena pour implémenter ce type de stockage, car il fournit une API permettant de stocker des modèles RDF sur une base relationnelle, et un langage de requête pour ceux-ci. Cependant, d'autres types de stockage plus efficaces que les triplets peuvent être également envisagés. Nous croyons qu'un modèle de stockage typé par objet permettrait d'améliorer la performance d'accès aux PKB.

4.1.1 Implémentation des services

Un service web est un composant implémenté dans n'importe quel langage, déployé sur n'importe quelle plate-forme, mais enveloppé dans une couche de standards dérivés du XML. Il doit pouvoir être découvert et invoqué dynamiquement par d'autres services. Cette nouvelle technologie, initiée par IBM et Microsoft, puis en partie normalisée sous l'égide du W3C, est maintenant acceptée par l'ensemble des acteurs de l'industrie informatique sans exception.

Le concept de service web s'articule actuellement autour des trois acronymes suivants :

- SOAP (Simple Object Access Protocol) est un protocole d'échange inter-application indépendant de toute plate-forme, basé sur le langage XML. Un appel de service SOAP est un flux ASCII encadré dans des balises XML et transporté dans le protocole HTTP. Bien que la mission de SOAP soit de représenter la structure d'un message pour la communication avec un service web, celui-ci n'a aucune idée de ce que le message contiendra. La spécification SOAP actuelle ne définit que la façon d'envoyer des messages via HTTP, mais d'autres protocoles peuvent être utilisés également comme par exemple, SMTP. Cependant, la plupart des outils de développement de services web ne supportent actuellement que le HTTP.
- WSDL (Web Services Description Language) donne la description au format XML des services web en précisant les méthodes pouvant être invoquées, leur signature et le point d'accès (URL, port, etc..). C'est, en quelque sorte, l'équivalent du langage IDL pour la programmation distribuée CORBA.
- UDDI (Universal Description, Discovery and Integration) normalise une solution d'annuaire distribué de services web, permettant à la fois la publication

et l'exploration. UDDI se comporte lui-même comme un service web dont les méthodes sont appelées via le protocole SOAP.

Comme le développement des procédures distribuées RPC (Remote Procedure Call), l'implémentation d'un service web se divise en trois étapes. Premièrement, nous devons écrire une description en WSDL qui décrit les méthodes communes pouvant être évoquées par un client via le protocole SOAP. Dans cette description, nous donnons toute la définition concernant le prototype d'appel de chaque méthode (son nom d'invocation, ses paramètres d'entrée, la structure de données de sortie). A partir de la description commune, une génération automatique peut se faire pour créer des classes programmatiques pour les deux côtés : le serveur et le client. La deuxième étape concerne une implémentation spécifique en utilisant les classes générées pour chaque méthode d'appel sur le serveur. Dès lors que cette implémentation est finie, le service web peut être mis en place et prêt à invoquer par un client dans la troisième étape. L'appel du service peut se faire à travers les classes générées pour le client.

Le service web installé pour chaque serveur webop a deux méthodes correspondant l'une au besoin d'interrogation l'autre à celui de notification dans le web-of-people. Nous verrons ci-après le prototype d'appel de ces deux méthodes. Les lecteurs peuvent se rapporter à l'Annexe B pour une description complète du service web en WSDL.

La méthode d'interrogation est de la forme :

```
QueryResult queryPKB(String uid,
                     QueryParam param,
                     Certificate sign);
```

où `uid` est l'identificateur de la PKB pour interrogation, `param` les paramètres du protocole d'accès, `sign` le certificat d'authentification envoyé par le demandeur. Notons que l'authentification doit s'appliquer à la méthode d'interrogation ainsi que celle de notification. La technique adoptée pour cette authentification sera présentée plus tard.

Voici la structure de données pour les paramètres d'accès. Ils se conforment à la spécification générale pour l'interrogation d'une PKB (cf. section 3.4.1).

```
QueryParam {
   String uri; // l'uri de la ressource d'entrée
   String scope; // l'étendue de recherche
   String filter; // le filtre de sélection
   String[] projection; // la liste de propriétés de projection
}
```

Nous décrivons ci-dessous le retour de données pour chaque appel d'interrogation.

```
QueryResult {
   int code; // le code de résultat (valeur 0 si succès)
   String data; // le flux de données au format RDF/XML
}
```

Pour le besoin de notification, une deuxième méthode est utilisée pour déposer un message dans la boîte de chaque destinataire concerné.

```
int postMessage(String uid,
                String msg,
                Certificate sign);
```

où `uid` est l'identificateur du destinataire, `msg` est le message d'envoi au format RDF/XML (cf. section 3.5.1). Un appel de cette méthode donnera un code de retour pour notifier si la notification a été un succès.

4.1.2 Comment déployer les services avec l'annuaire ?

Dans l'architecture du prototype, un seul serveur est autorisé pour héberger un utilisateur.[10] L'adresse de son serveur de rattachement est mise sur l'annuaire commun du système. Grâce à cette information, on peut savoir où le service web doit être invoqué pour communiquer avec cette personne (i.e., interroger sa PKB ou déposer une notification dans sa boite de réception). La Figure 4.2 donne un exemple d'annuaire au format LDIF (RFC2849). Une entrée est créée pour l'utilisateur *u1$odp* avec le nom distinct `uid=u1, dc=odp`. En plus du nom (attribut `cn`) et de l'adresse email (attribut `mail`), l'attribut `host` indique l'adresse du service web à évoquer via le protocole SOAP. La valeur de cet attribut sera consultée pour établir des connexions au service web approprié à cette personne.

```
dn: uid=u1, dc=odp
objectclass: person
cn: TA Tuan Anh
mail: ta@enst.fr
host: http://ariane.enst.fr:8080/axis/WebopService.jws
```

Figure 4.2 : Exemple d'une entrée dans l'annuaire

Notons que l'architecture du web-of-people est très ouverte. Nous pouvons intégrer également des moyens autres que le service web pour les échanges entre personnes. Par exemple, nous pouvons envoyer des notifications via la messagerie classique (SMTP).

[10] mais une personne physique peut jouer le rôle de plusieurs utilisateurs

Mais l'automatisation du traitement des messages sera plus difficile. De même, d'autres protocoles utilisés pour l'interrogation peuvent être souhaités.

Cependant, l'exploitation de services web permettra d'intégrer facilement des sites weblog dans notre système. Un service web sera installé pour les sites weblog pouvant être accessibles dans le web-of-people. L'avantage de ce service est qu'il peut fonctionner même sur la couche http déjà supportée par les weblogs. De ce fait, le service web peut être considéré comme une sorte de wrapper qui transforme des pages HTML en descriptions au schéma RDF.

4.1.3 Principe d'authentification

L'authentification consiste à effectuer la vérification de l'identité du demandeur d'un service. Il existe plusieurs approches et techniques permettant l'authentification. Nous appliquons, pour notre service web, une technique d'authentification à base de certificats. Cette technique repose elle-même sur un algorithme de cryptographie à clés publiques [Alfred *et al.*, 1997]. Un certificat constitue une structure de données contenant des informations sur le propriétaire (en particulier son identité et sa clé publique), informations qui sont certifiées (i.e., signées) par une autorité de confiance appelée autorité de certification. La clé publique contenue dans ce certificat est associée à une clé privée que l'utilisateur doit garder secrète. Pour s'authentifier, un utilisateur signe avec sa clé privée sur une information choisie. C'est ce que fait en particulier l'autorité de certification. La clé publique permet au service utilisateur de vérifier que l'utilisateur possédait bien la clé privée (i.e., le décryptage de la signature doit donner l'autorité de certification). Voici la structure d'un certificat passé au service web du prototype pour s'authentifier.

```
Certificate {
  String uid; // l'identifiant du demandeur
  byte[] pkey; // la clé publique
  byte[] signature; // la signature (information certifiée)
}
```

Cette structure est utilisée pour les deux méthodes `queryPKB` et `postMessage`. L'information signée est celle du paramètre `param` dans la première méthode, `msg` dans la deuxième. L'algorithme de cryptographie RSA pourrait être employé pour le processus d'authentification.[11]

[11] Notons que nous pourrons appliquer également une fonction de hachage (e.g., MD5, SHA) sur l'information signée. Comme la taille d'un message à signer sera probablement assez gros, la fonction de hachage permettra d'obtenir un haché du message, i.e., une suite de caractères assez courte. L'expéditeur du message signera uniquement sur ce haché. Ainsi, lors de la réception du message il suffit au destinataire de calculer le haché du message reçu et de le comparer avec le haché signé après décryptage.

4.2 Application pour l'utilisateur final

Dans cette section, nous décrivons la conception de l'application pour l'utilisateur final du prototype. Il s'agit des interfaces correspondant à trois groupes majeurs de fonctionnalités offertes aux utilisateurs finals. Il nécessite tout d'abord un éditeur weblog sémantique qui permet à un utilisateur de gérer tous les éléments de sa PKB via une interface web en ligne. Cet éditeur doit lui faciliter toute action d'édition de posts et ainsi de topiques. Comme le web-of-people supporte des échanges actifs entre utilisateurs, un gestionnaire de messages est fourni à l'utilisateur pour recevoir des notifications. Via cette deuxième interface, des réactions appropriées aux notifications sont proposées à l'utilisateur. Enfin, la troisième interface offre à l'utilisateur les fonctionnalités de navigation, ainsi de recherche de connaissances dans le réseau de multi-PKB.

4.2.1 Editeur weblog

Cet éditeur est une interface classique permettant des actions de mise à jour sur les PKB d'utilisateur. En utilisant l'éditeur de weblog, un utilisateur peut éditer des posts/topiques, contrôler des accessibilités dans son weblog, intégrer des topiques de ses amis,… La Figure 4.3 illustre une interface développée en JSP pour cet éditeur. Dans cette interface l'utilisateur peut voir l'arborescence de ses topiques créés et, pour chaque topique sélectionné, la liste des posts attachés à celui-ci. Des « buttons » permettent d'invoquer des actions d'édition particulières.

Notons qu'un échange avec un ou plusieurs autres utilisateurs se fait implicitement comme modification locale automatiquement détectée comme un évènement à notifier aux destinataires concernés. Grâce à cette notification chaque destinataire pourra avoir la réaction appropriée à son comportement d'échange. Par exemple, une modification d'abonnement dans l'éditeur correspondra à un évènement de création/suppression d'une description webop:subscribesTo dans la PKB. Voir la liste des évènements nécessairement notifiés entre PKB dans la section 3.5.2.

Une autre remarque est qu'une action peut effectuer la mise à jour de plusieurs triplets dans la PKB. Cependant, le message de notification envoyé n'apportera qu'un évènement correspondant à cette action. Par exemple, une action de recommandation implique une création de deux triplets : webop:accessibleBy pour ouvrir le droit d'accès au destinataire, webop:recommendedFor pour constituer une recommandation entre deux topiques. Mais, l'évènement à notifier ne concernera que la deuxième description.

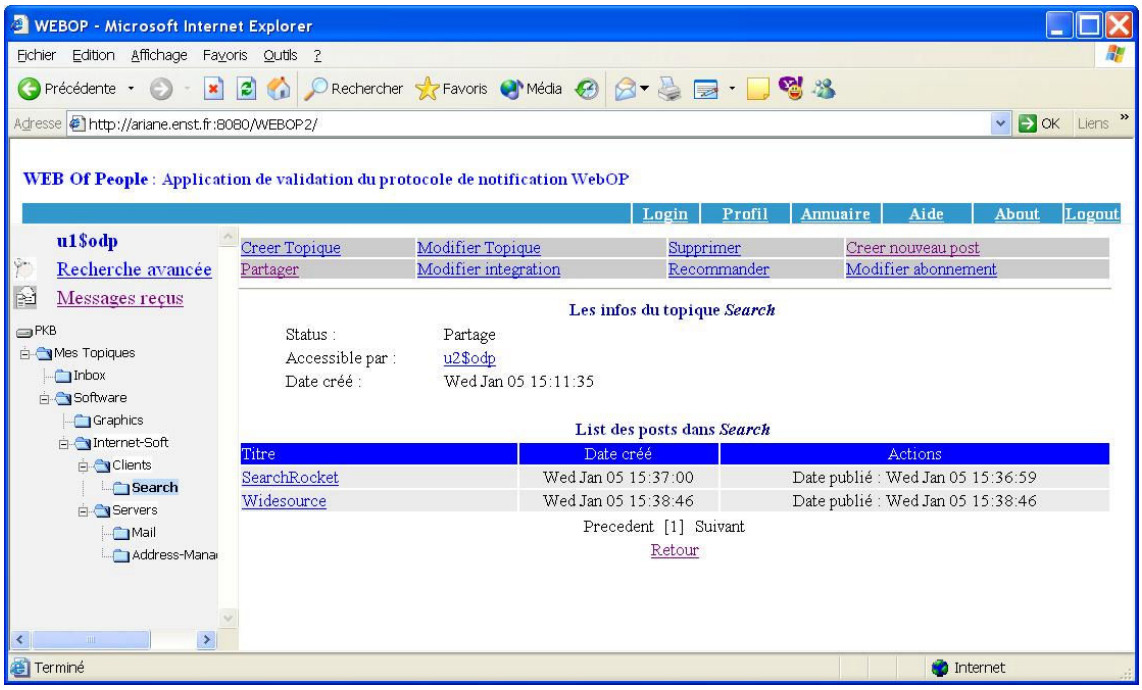


Figure 4.3 : Interface typique d'un éditeur de weblog

4.2.2 Gestionnaire de messages

Le rôle du gestionnaire de messages dans l'application est d'abord d'afficher les notifications reçues. Par son interface, il offre à l'utilisateur un choix de traitement pour chaque notification : i) simplement lire le message sans réaction ou ii) réagir par une modification de sa PKB. C'est pourquoi, ce gestionnaire est intégré avec l'éditeur de weblog pour offrir à l'utilisateur les réactions appropriées pour chaque message reçu. La Figure 4.4 donne un exemple d'interface tiré de l'application JSP développé pour notre prototype. Nous voyons que les messages reçus sont ordonnés chronologiquement. L'utilisateur peut choisir sa réaction pour chaque message. Par exemple, il peut décider d'abonner ou ne pas abonner (webop:deliveryFor) pour une réception d'une demande d'abonnement (webop:subscribesTo). Notons que, par ailleurs, l'éditeur weblog fournit l'interface pour que l'utilisateur puisse modifier plus librement sa PKB. Mais dans quelques cas, il pourrait être nécessaire de contraindre l'action suivant une notification reçue dans le gestionnaire de messages. Ainsi, si l'utilisateur ne pouvait créer un abonnement que s'il reçoit d'abord l'acceptation du demandeur alors cette application interdirait les « spams ».

Pour assister l'utilisateur dans le traitement de notifications, des routines de traitement automatisé de messages reçus peuvent être développées dans le gestionnaire. Cette automatisation permet de réduire des efforts de l'utilisateur s'il a toujours un même comportement sur un type de messages. Voici quelques traitements automatisés utiles dans l'application.

- *Anti-spams* : vider automatiquement toutes les notifications sur la création de nouveaux posts dont le topique n'est pas encore accepté pour un abonnement. Par exemple, la notification (*p1*, webop:issued, *d*) devrait être supprimé si *p1* est associé à un topique *a1* dans la PKB originaire, (*p1*, webop:hasTopic, *a1*), et il n'existe pas (*a2*, webop:subscribesTo, *a1*) où *a2* est un topique dans la PKB du récepteur.
- *Réponse automatique* : utiliser une règle définie par l'utilisateur pour traiter un type de messages ; par exemple, on peut créer une réponse automatique webop:deliveryFor pour toute demande d'abonnement webop:subscribesTo. C'est l'application qui spécifie des règles possibles pour le traitement automatisé des messages.

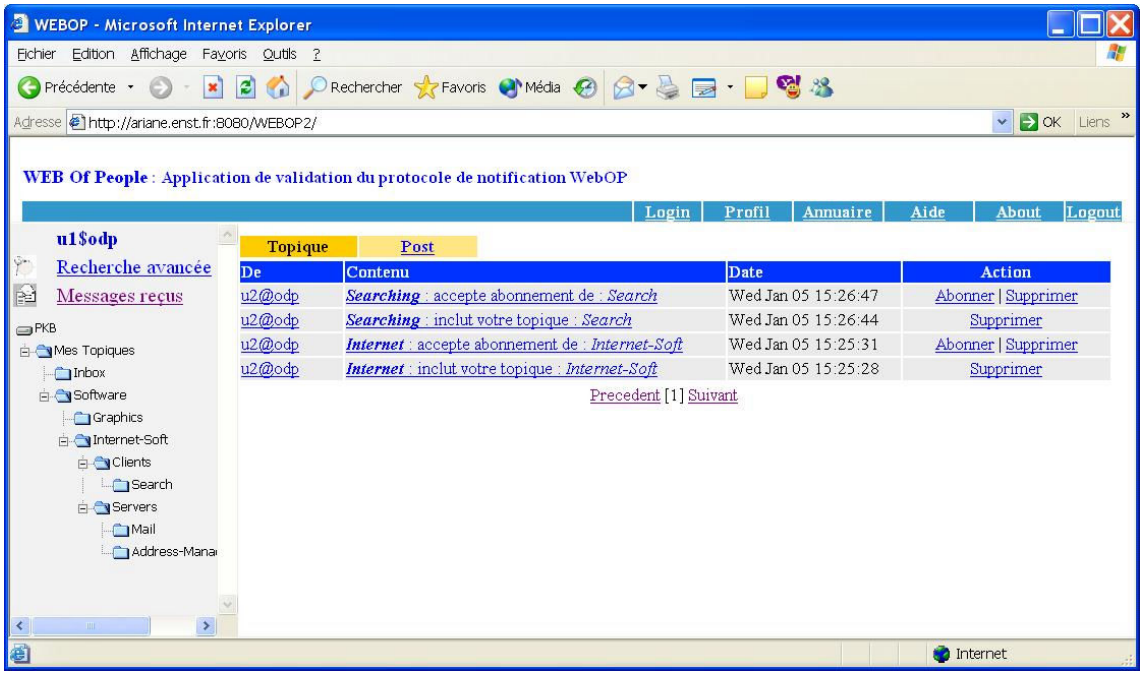


Figure 4.4 : Affichage de messages dans un gestionnaire

4.2.3 Interface de navigation et de recherche

D'un point de vue applicatif, le web-of-people est un réseau de weblogs sémantiques connectés. Le dernier composant de l'application permet à l'utilisateur de naviguer dans ce réseau. A travers des liens d'intégration entre topiques et/ou ceux de personnalisation entre posts, l'utilisateur peut passer d'un weblog à un autre en commençant le sien. La nature classique de l'interface de navigation est une arborescence de catégories. Elle permet effectivement les recherches où l'utilisateur peut retrouver des posts en descendant ou montant dans des hiérarchies de topiques.

De plus, nous pouvons supporter des recherches par requêtes distribuées dans le réseau des bases de connaissances. En formulant des requêtes selon des critères de besoin, l'utilisateur peut obtenir des résultats pertinents pour lui. Le type le plus simple pour une telle interface est un formulaire pour la recherche de posts. Grâce à celui-ci, l'utilisateur peut retrouver des posts sous un topique donnée satisfaisant un filtre par métadonnées (titre, date, …). Notons que cette recherche ne se limite pas dans la PKB locale de l'utilisateur, mais peut être étendue à d'autres PKB via un enchaînement de topiques intégrés.

Dans le cadre du projet avec FT R&D, nous avons étudié une interface de navigation plus graphique pour visualiser les PKB contenant un grand volume de topiques. Cette interface permet de projeter les topiques d'une PKB dans un espace bi-dimensionnel de type starfield[12] [Alberg et Shneiderman, 1994]. Les dimensions de projection possibles étaient à choisir entre les cinq que nous avons bien définies pour un topique : 1) « niveau de spécialisation » - profondeur du topique dans la hiérarchie, 2) « poids » - nombre total de posts dans le sous arbre, 3) « actualité » - nombre de jours écoulés depuis la dernière publication d'un post dans le sous arbre, 4) « magnitude » ou « rayonnement » - nombre d'intégrations dont le topique fait l'objet et 5) « notoriété » - nombre de posts dans le sous arbre ayant fait l'objet de personnalisation. L'interface graphique du type starfield convient bien pour donner la vue statistique d'une PKB.

4.2.4 Scénarios d'application

Nous présentons ici quelques scénarios d'application pour illustrer la variété des échanges sociaux entre utilisateurs dans le web-of-people. Un processus d'échanges entre deux utilisateurs est une suite d'actions, éventuellement avec des notifications, faites par ceux-ci. Les scénarios ci-après tirent des actions de mise à jour s'enchaînant entre *u1* et *u2*. Les actions seront présentées en ordre chronologique dans les scénarios.

Le scénario 1 présente une « recommandation active » de la part de *u1*. Tout d'abord, *u2* ouvre un accès partagé sur son topique *a2* à *u1*. Comme *u1* peut voir le topique *a2* et juger son topique *a1* éventuellement intéressant pour celui-ci, il recommande *a1* à *u2* pour son topique *a2*. Cette action de recommandation est notifié à *u2* - (*a1*, webop:recommendedFor, *a2*). Dans ce scénario, les mises à jour correspondant aux évènements à notifier sont marqués par une étoile (*). A présent *u2*, à la réception du message de recommandation, peut exprimer son appréciation par une intégration avec abonnement. S'il effectue cette action, celle-ci à son tour est aussi notifiée. Dès que *u1* reçoit l'appréciation positive de *u2* pour un abonnement, l'abonnement (*a1*, webop:deliveryFor, *a2*) peut être créé automatiquement dans la PKB de *u1*. Alors, *u2* peut recevoir ensuite des notifications pour tous les nouveaux posts dans le cadre de cet abonnement. Par exemple, des lors qu'un post *p1* est publié, (*p1*, webop:issued, *d*), un message de notification est envoyé à *u2*.

[12] Un topique est un point dans l'espace d'affichage

Ord.	Utilisateur		Action	Mises à jour
	u1	***u2***		
1		✕	Partager un topique	+(*a2*, webop:accessibleBy, *u1*)
2	✕		Recommander un topique	+(*a1*, webop:recommendedFor, *a2*)* +(*a1*, webop:accessibleBy, *u2*)
3		✕	Intégrer le topique recommandé avec un abonnement	+(*a2*, webop:includesTopic, *a1*)* +(*a2*, webop:subscribesTo, *a1*)*
4	✕		Traitement automatisé	+(*a1*, webop:includedBy, *a2*) +(*a1*, webop:deliveryFor, *a2*)*
5	✕		Publier un nouveau post	+(*p1*, webop:issued, *d*)* +(*p1*, webop:hasTopic, *a1*) +(*p1*, webop:ressourceRef, *url*)

Scénario 1 : « Recommandation active »

Ord.	Utilisateur		Action	Mises à jour
	u1	***u2***		
1		✕	Ouvrir un topique public	+(*a2*, webop:accessibleBy, *all*)
2	✕		Intégrer un topique public sans abonnement	+(*a1*, webop:includesTopic, *a2*)*
3		✕	Traitement automatisé	+(*a2*, webop:includedBy, *a1*)
4		✕	Publier un nouveau post	+(*p1*, webop:issued, *d*) +(*p1*, webop:hasTopic, *a2*) +(*p1*, webop:ressourceRef, *url*)
5	✕		Demander un abonnement	+(*a1*, webop:subscribesTo, *a2*)*
6		✕	Accepter l'abonnement	+(*a2*, webop:deliveredFor, *a1*)*
7		✕	Publier un deuxième post	+(*p2*, webop:issued, *d*)* +(*p2*, webop:hasTopic, *a2*) +(*p2*, webop:ressourceRef, *url*)

Scénario 2 : « Consommation active »

A l'inverse du scénario 1, le scénario 2 illustre une « consommation active » de la part de *u1*. Comme il trouve un topique public *a2* intéressant, il l'intègre d'abord dans son topique *a1*. Mais cette intégration ne lui permet pas d'être informé des nouveaux posts publiés sous ce topique public. Par exemple quand un post *p1* est publié sous *a2*, il n'y a pas de message de notification pour cet évènement. Si *u1* veut être par la suite automatiquement informé des nouveautés (mode « push »[13]), il demande alors un abonnement pour son topique en envoyant à *u2* un message de notification de souscription. A son tour, *u2* peut accepter cette demande d'abonnement comme réaction. Grâce à cet abonnement, *u1* peut recevoir la notification d'un nouveau post *p2* publié par la suite.

Ord.	**Utilisateur**		**Action**	**Mises à jour**
	u1	***u2***		
1		✕	Ouvrir un topique public	+(*a2*, webop:accessibleBy, *all*)
2	✕		Recommander un topique	+(*a1*, webop:recommendedFor, *a2*)* +(*a1*, webop:accessibleBy, *u2*)
3	✕		Publier un post et notifier pour *a2* même sans abonnement	+(*p1*, webop:issued, *d*)* +(*p1*, webop:hasTopic, *a1*) +(*p1*, webop:ressourceRef, *url*)
4		✕	Fermer le topique pour un spammer	+(*a2*, webop:inaccessibleBy, *u1*)
5	✕		Toute autre recommendation du spammer sera ignorée	+(*a3*, webop:recommendedFor, *a2*)* +(*a3*, webop:accessibleBy, *u2*)

Scénario 3 : Anti-spam

Dans le scénario 3, *u1* joue le rôle d'un « spammer ». Dans un premier temps, le topique *a2* est ouvert pour tout le public. *u1* recommande son topique *a1* pour *a2*. Comme *u2* trouve ce topique non intéressant, il ne fait aucune modification sur sa PKB. Néanmoins, *u1* continue d'envoyer, sans attendre une demande d'abonnement de la part de *u2*, une notification d'un nouveau post *p1* attaché au topique *a1*. Comme il n'y a pas d'abonnement pour *a1*, cette notification est facilement détectée comme un spam par l'application de l'utilisateur *u2*. Ainsi, *u1* est considéré comme un spammer et l'accessibilité du topique (ou même de toute PKB) est fermée automatiquement pour

[13] Néanmoins il peut voir toujours le nouveau post comme tout post attaché à un topique accessible (mode « pull »)

lui. Une telle fermeture d'accessibilité permet que toute autre recommandation venant de *u1* soit ignorée systématiquement par la suite.

Nous voyons déjà dans ces simples exemples tout le bénéfice sémantique que l'application peut retirer de l'ordre d'arrivée des messages de notification, émis dans l'ordre des mises à jour des descriptions/propriétés des topiques dans les PKB.

4.3 Conclusion et perspectives

Dans ce chapitre nous avons présenté le prototype webop qui implémente le cadre de conception général pour le web-of-people. Ce prototype permet de mettre en pratique un réseau d'échanges à base des weblogs partagés entre utilisateurs. Les problèmes suivants ont été considérés dans ce réseau social.

- *Décentralisation* : le réseau social du web-of-people repose sur l'architecture P2P dont chaque nœud est une personne. Ceci pose comme exigence l'autonomie des pairs. Au lieu d'imposer une gestion centralisée de topiques communs pour les échanges, les utilisateurs peuvent définir chacun des taxonomies individuelles ainsi que des liens d'intégration entre elles. Les échanges se constitueront sur les liens taxonomiques établis directement entre deux individus. Cette approche convient vraiment au développement d'un réseau à grande échelle comme le web-of-people.

- *Sécurité* : tout accès aux ressources partagées dans le web-of-people est sous contrôle de leurs propriétaires. L'identité sociale de chaque lecteur sera vérifiée dans le processus d'authentification. Grâce au contrôle d'accès, un utilisateur peut faire le choix de partager ses ressources soit au public soit uniquement avec une liste de personnes à qui il fait confiance.

- *Confiance* : le réseau d'échanges du web-of-people peut susciter la confiance des utilisateurs. Les échanges se fondent effectivement sur les dialogues établis progressivement entre les utilisateurs. Un mécanisme automatique est facile à mettre en place pour empêcher les tentatives de « spam » dans ce réseau. Autrement dit, seuls les dialogues qui assurent la confiance des utilisateurs sont considérés.

Le prototype webop, comme système de recherche d'information, est constitué d'un réseau distribué de bases personnelles de connaissances - dites PKB. Ce système favorise effectivement l'accès à l'information à base de connaissances partagées entre utilisateurs. En naviguant et/ou posant des requêtes sur ce réseau de connaissances, les utilisateurs peuvent retrouver des informations reliées à leurs intérêts.

Il est important de noter que la conception du système donne le principe de mise en cohérence des PKB du réseau distribué : la notification des changements d'états locaux par messages diffusés aux seules autres PKB concernées. Ces messages, qui sont journalisés dans les boîtes de réception, doivent pouvoir être effacés au fur et à mesure

de leur traitement, de façon asynchrone à leurs arrivées. Nous avons spécifié les évènements minimaux que constituent la notification automatique de certaines mises à jour dans les PKB, d'une part, et leur traitement de manière systématique ou en fonction de l'application, d'autre part. Seuls ces évènements sont à la charge du cadre de conception général. Mais bien sûr une application développée sur ce cadre de conception pourrait maintenir une consistance entre PKB de niveau sémantique supérieur. Donnons-en deux exemples : (i) si les liens de réciprocités sont établis entre topiques à leur niveau maximum, les topiques peuvent être considérés comme équivalents ; (ii) si un pair autorise l'utilisation de ses liens externes vers d'autres pairs, par des tiers, alors l'application peut permettre l'établissement de liens transitifs.

En ce qui concerne le choix d'implémentation, le prototype webop s'appuie essentiellement sur un réseau de services web. La technologie des services web retient actuellement l'attention de l'industrie car elle permet de réduire les difficultés et les coûts associés à l'intégration de différentes applications. Pour ces raisons, nous pourrions intégrer facilement d'autres applications dans notre prototype. Il y a deux applications pour le web-of-people que nous pensons très utiles. Dans un premier temps, nous pourrions développer un outil graphique qui faciliterait la navigation sociale dans le réseau des weblogs du web-of-people. Ce principe de navigation est de visualiser dans un diagramme des rapports entre les utilisateurs qui partagent de mêmes intérêts. Cette visualisation permettrait d'identifier facilement les groupes d'intérêt ainsi que leurs experts par domaines dans le web-of-people. Ceci aiderait alors les visiteurs à se concentrer seulement sur quelques sites qui peuvent leur donner de meilleures informations dans un domaine donné. Cette approche de visualisation sociale peut être comparée avec celle du projet ReferralWeb [Kautz *et al.*, 1997] appliquée dans la métaphore des ressources web.

Nous souhaiterions également intégrer un outil de recommandation sociale pour les utilisateurs du web-of-people. Il s'agirait d'un service de calcul d'agrégations possibles entre les topiques similaires de deux utilisateurs. Autrement dit, il proposerait à un utilisateur donné de nouveau contacts à partir des similitudes mesurées. Un tel service de recommandation sociale est à l'étude dans un projet au sein de FTR&D [Plu *et al.*, 2004]. Ce service a été conçu comme un composant de l'application SoMeONe [Agosto *et al.*, 2003], le prototype démontrant l'idée initiale du web-of-people [Plu *et al.*, 2003]. Notre souhait serait donc d'adapter cette étude au nouveau système du web-of-people.

Chapitre 5

Evaluation efficace de requêtes réseau

Dans le cadre de conception pour le web-of-people, nous avons spécifié un protocole de base permettant d'accéder à la PKB d'une personne quelconque. Néanmoins, ce protocole ne spécifie pas comment traiter efficacement des requêtes réseau qui exigent une traversée de plusieurs pairs pour récupérer récursivement des posts sous tous les topiques liés. Nous avons donc cherché une implémentation efficace pour l'évaluation de telles requêtes réseau dans le web-of-people [Saglio *et al.*, 2005].

Notre objectif dans ce chapitre est double. Dans un premier temps nous donnons un modèle pour les requêtes réseau. Comme suggéré dans [Tzitzikas et Meghini, 2003], des requêtes réseau pourraient être optimisées en choisissant une bonne stratégie d'évaluation. Nous avons choisi une stratégie décentralisée afin de bénéficier de l'exécution en parallèle chez différents pairs pour l'évaluation des requêtes réseau. D'autre part, nous avons mis en place un mécanisme de détection de redondances dans le processus d'évaluation. Ceci pour permettre d'éviter des calculs redondants où un post peut être trouvé plusieurs fois.

Deuxièmement, nous avons fourni une implémentation concrète en relationnel montrant ainsi que les requêtes réseau peuvent être évaluées efficacement à l'aide d'un moteur SQL. Nous employons dans cette implémentation un codage particulier des termes dans une hiérarchie. Grâce à un tel codage, le calcul de fermetures transitives dans les requêtes ancêtres/descendants peut être accéléré considérablement.

Afin de justifier la nécessité de l'élimination de redondances dans l'optimisation de requêtes réseau, nous rapportons une analyse expérimentale sur un échantillon de hiérarchies choisies dans l'ODP. Cette expérimentation préliminaire nous permet de confirmer qu'il existe naturellement des redondances que nous devons éliminer pour améliorer la performance du système.

5.1 Exemple introductif

Tout d'abord, voyons un exemple pour des requêtes posées dans un réseau P2P à base de taxonomies comme le web-of-people. Cet exemple est donné dans la Figure 3.4. Nous pouvons considérer tout le réseau comme une base distribuée de topiques et de posts. Chaque topique est modélisé par une classe dont les instances sont des posts. La relation isA entre de telles classes crée une hiérarchie de topiques. De la même façon, une relation isA permet également de modéliser une liaison entre deux topiques de différentes hiérarchies. Par exemple, dans la Figure 3.4, la relation isA entre *Internet-Soft* et *Internet* représente une inclusion du topique *Internet-Soft* dans *Internet*.

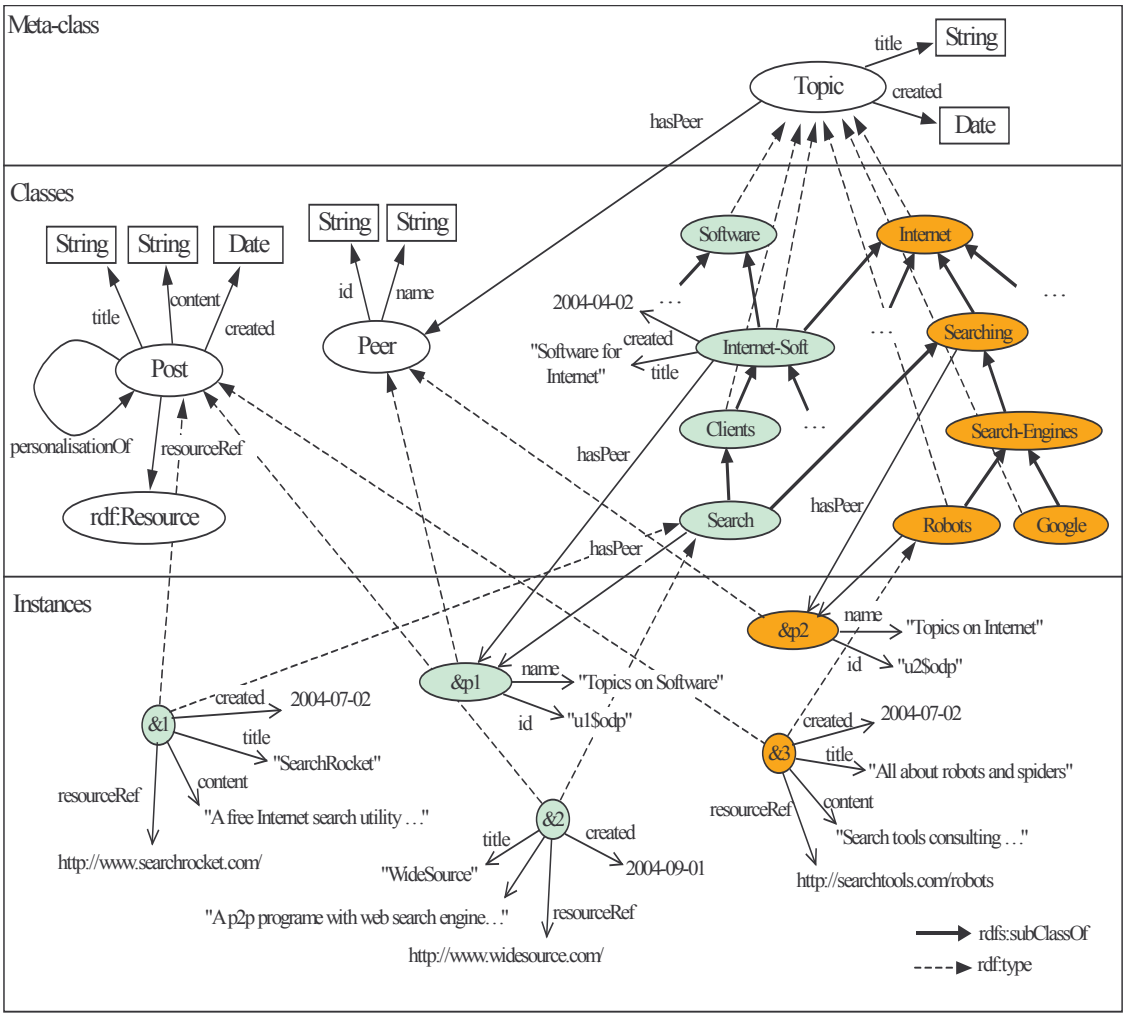


Figure 5.1 : Entrepôt virtuel au modèle du RQL

La Figure 5.1 illustre une base en RDF pour les topiques et les posts distribués pour l'exemple du web-of-people. Nous choisissons le langage de requête RQL [Karvounarakis *et al.*, 2002] pour exprimer des requêtes dans cette base distribuée, car

RQL apporte une puissance déclarative. Ceci permet d'employer une formulation simple pour exprimer des fermetures transitives entre topiques dans cette base. Un topique est une classe instanciée de la méta-classe *Topic*. Chaque topique est décrit avec un titre, une date de création et le pair auquel il appartient (propriété *hasPeer*). *Peer* est la classe des pairs du réseau. *&p1* et *&p2* sont les instances de pairs respectivement pour les utilisateurs "*u1$odp*" et "*u2$odp*". rdfs:subClassOf représente la relation isA entre classes topique soit dans un même pair (e.g., *Internet-Soft* isA *Internet*), soit dans deux pairs différents (e.g., *Internet-Soft* isA *Software*). rdf:type représente l'instanciation. Un post est une instance à la fois de *Post* et d'une classe topique. Un post appartient au même pair que le topique qu'il est instance. Par exemple, *&r2* appartient au pair *&p1* car la propriété *hasPeer* du topique *Search* pointe vers *&p1*.

Nous montrons maintenant sur quelques exemples comment RQL (dans sa dernière version [Karvounarakis *et al.*, 2004]) permet d'exprimer la recherche de posts et la navigation dans les hiérarchies. Nous verrons que certaines requêtes peuvent ne recevoir des réponses que localement, i.e., du pair auquel elles sont adressées (e.g., **Q4**, **Q5**, etc.). Certaines exprimées par des requêtes réseau nécessitent des réponses de plusieurs pairs reliés à travers des chemins isA (e.g., **Q1**, **Q2**, etc.).

Recherche de posts

Q1. Posts sous *Internet* ? (nous cherchons des instances du topique *Internet* ou de ses sous-topiques, qui soient locaux au pair *&p2* ou définis dans un autre pair mais reliés à *Internet* à travers un chemin isA)

```
select x from Internet{x}
```

Internet{x} est une expression de chemin à partir de la classe *Internet* se terminant par une instance *x*. L'interprétation d'un tel chemin est que *x* est une instance de la classe topique *Internet* ou de n'importe laquelle de ses sous-classes.

Nous pouvons restreindre également la recherche par des conditions sur certaines métadonnées telles que le titre, la date de création, etc.

Q2. Posts sous *Internet* créés depuis 2004 ?

```
select x from Internet{x}.created{d} where d>=2004-01-01
```

Internet{x}.created{d} est un raccourci syntaxique pour Internet{x}, {x}created{d}. Le deuxième chemin donne *d* en tant que date de création de l'instance *x*.

Une autre façon de restreindre la recherche de posts est de se limiter aux instances directes de certains topiques.

Q3. Posts sous *Internet*, instances directes d'un topique créé après le 2004-01-01 ?

select x from Topic{t}.created{d}, Internet{x; ^t} where d>=2004-01-01

t est une classe topique ayant la date de création après le 2004-01-01. Le chemin Internet{x; ^t} dénote que *t* est la classe directe de posts sous *Internet* ("^" indique « direct »).

Les requêtes ci-dessus sont forcement des requêtes réseau car elles sont satisfaites par des posts trouvés dans plusieurs pairs. Elles se distinguent des requêtes locales comme par exemple les suivants :

Q4. Posts locaux sous *Internet* ? (nous limitons la recherche au pair local *&p2*)

select x from Topic{t}.hasPeer{p}, Internet{x; ^t} where p=&p2

Q5. Posts sous *Internet*, *Searching* et *Search-Engines* ? (nous cherchons des posts dans une partie de la hiérarchie locale)

select x from Internet{x; ^$t} where $t >= Search-Engines

Q6. Posts sous *Internet* sauf *Searching* ?

Internet minus Searching

Internet est une autre expression plus brève de la requête **Q1**.

Q7. Couples de posts sous *Internet* ayant la même référence à une ressource ?

select x, y from Internet{x}.resourceRef{r}, Internet{y}.resourceRef{r} where x!=y

Requêtes de navigation

Q8. Fils locaux du topique *Internet* ?

select t from Topic{t}.hasPeer{p} where t in subclassOf^(Internet) and p=&p2

Q9. Topiques externes reliés au topique local *Internet* ?

select t from Topic{t}.hasPeer{p} where t in subclassOf^(Internet) and p!=&p2

Les requêtes **Q8** et **Q9** permettent une descente dans la hiérarchie de topiques. Tandis que **Q8** ne donne qu'une vue locale sur la hiérarchie, **Q9** navigue vers les hiérarchies des pairs voisins.

Q10. Topique(s) dont le post *&1* est une instance directe ?

select t from ^$t{x} where x=&1

5.2 Modèle de requêtes réseau et évaluation

Dans cette section, nous nous intéressons à la modélisation de requêtes réseau illustrées dans la section précédente. En général, une requête réseau permet de chercher des posts sous un terme (topique) donné. Comme il existe des mises en correspondance entre termes de différentes hiérarchies, la requête pourrait être calculée non seulement par le pair auquel la requête est posée, mais aussi par d'autres pairs ayant des termes reliés au terme donné.

5.2.1 Requêtes réseau

Définition 1: Chaque pair p possède une hiérarchie de termes $(T_p, \prec)$ où T_p est un ensemble fini et non vide de termes; $\prec$ est un ordre partiel représentant des relations *isA* entre termes. Soit I une intépretation qui détermine pour chaque terme t l'ensemble de posts instanciés de t ou n'importe quel successeur t' de t (i.e., $t \prec t'$). Nous appelons par la suite $I(t)$, en abrégé, l'interprétation de t.

Définition 2: Soit t, t' deux termes définis dans deux pairs différents p et p' ($t \in T_p$, $t' \in T_{p'}$). Avec l'autorisation du pair p', p peut déclarer t' « lié à » ou « inclus dans » t par un lien $[t, t']$. $Link_{p,p'}$ dénote l'ensemble de tels liens $[t, t']$ entre pair p et pair p'.

Comme dans [Tzitzikas et Meghini, 2003], nous pouvons modéliser un lien $[t, t']$ comme étant une relation isA : $t \prec t'$, avec la sémantique habituelle ; si un post est dans l'interprétation de t', il est aussi dans l'interprétation de t.

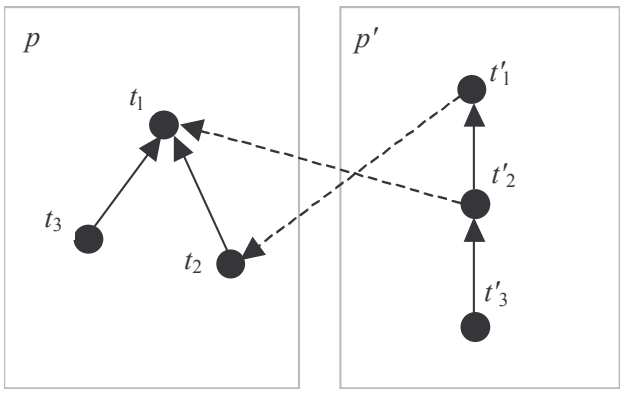


Figure 5.2 : Liens croisés redondants

Proposition 1 (liens croisés redondants): Etant donné deux pairs p et p', s'il existe $[t_1, t'_2]$, $[t_2, t'_1] \in Link_{p,p'}$, et $t'_1, t'_2 \in T_{p'}$, $t'_1 \preceq t'_2$ d'une part, et $t_1, t_2 \in T_p$, $t_1 \preceq t_2$ d'autre part, alors le lien $[t_1, t'_2]$ est redondant.

La Proposition 1 est illustrée par la Figure 5.2. Le lien $[t_1, t'_2]$ n'est pas nécessaire car t'_2 est accessible à partir de t_1 par le lien $[t_2, t'_1]$: $t'_2 \succ t'_1 \succ t_2 \succ t_1$.

Définition 3 : Etant donné un terme t dans un pair p,

$$L(p, t, p') = \{t' \mid t' \in T_{p'} \wedge \exists t_1 \in T_p (t \preceq t_1 \wedge [t_1, t'] \in Link_{p,p'})\}$$

dénote l'ensemble de termes dans un pair p' qui peuvent être atteints à partir du terme t et ses successeurs en traversant des liens déclarés dans p. Le pair p' est un voisin de p s'il existe au moins un lien $[t, t'] \in Link_{p,p'}$. $N(p)$ dénote l'ensemble de pairs voisins de p. Supposons que $I_p(t)$ est l'interprétation d'un terme t sans tenir compte de liens vers des voisins. $I_p(t)$ est donc l'ensemble de posts dans pair p, instances de t ou de n'importe lequel de ses successeurs. Si $I(t)$ est l'interprétation de t, y compris les posts existant dans d'autres pairs, nous avons :

$$I(t) = I_p(t) \cup \bigcup_{t' \in L(p,t,p_i), p_i \in N(p)} I(t') \qquad (1)$$

Définition 4 : $Q(p, t)$ dénote une requête posée à un pair p dont le résultat est $I(t)$, l'ensemble de posts sous le terme t dans le réseau. La requête $Q(p, t)$ est décomposée en une requête locale $Q_p(t)$ dont le résultat est $I_p(t)$, et un ensemble de requêtes envoyées aux voisins. A partir de l'équation (1), nous avons :

$$Q(p,t) = Q_p(t) \cup \bigcup_{t' \in L(p,t,p_i), p_i \in N(p)} Q(p_i, t') \qquad (2)$$

Considérons l'exemple de requête de la Figure 5.3. L'évaluation de $Q(p, t)$ implique celle de $Q(p', t'_2)$, $Q(p', t'_3)$, $Q(p', t'_4)$ et $Q(p', t'_5)$ dans le pair p' puisqu'il y a quatre liens entre les successeurs de t dans p et les termes dans p'. Néanmoins, les seuls $[t_1, t'_2]$ et $[t_2, t'_3]$ suffisent pour répondre à la requête $Q(p, t)$. Les liens $[t_4, t'_4]$ et $[t_3, t'_5]$ ne sont pas nécessaires (et donc redondants) pour cette requête; en effet, t'_4 et t'_5 et leurs successeurs peuvent être atteints en suivant le lien $[t_2, t'_3]$. De même, pour la requête réseau $Q(p, t_2)$, le lien $[t_3, t'_5]$ est redondant.

Proposition 2 : Etant donné $L_{\min}(p, t, p') = \{t' \mid t' \in L(p, t, p') \wedge \bar{\exists}\, t'_1 \in L(p, t, p') : t'_1 \prec t'\}$, nous avons:

$$Q(p,t) = Q_p(t) \cup \bigcup_{t' \in L_{\min}(p,t,p_i), p_i \in N(p)} Q(p_i, t') \qquad (3)$$

La Proposition 2 permet d'éviter des calculs redondants dans l'évaluation de requêtes réseau. Etant donné un terme t dans un pair p, un terme $t' \in L(p,t,p')$ est redondant dans le

calcul de la requête $Q(p, t)$ si $t' \notin L_{min}(p,t,p')$ puisqu'il existe déjà $t'_1 \in L_{min}(p,t,p')$ tels que $t'_1 \prec t'$.

$L_{min}(p,t,p')$ est calculé lors de l'exécution de la requête. L'algorithme suivant donne un calcul naïf de $L_{min}(p,t,p')$ (L_{min} en abrégé) à partir de $L(p,t,p')$ (L en abrégé). Dans la prochaine section, nous montrerons un algorithme efficace pour ce calcul en profitant d'un codage particulier des termes.

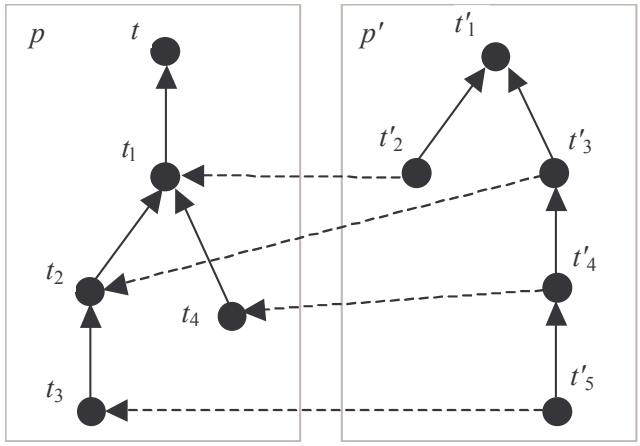


Figure 5.3 : Liens redondants détectés au temps d'exécution

Algorithme A – Calcul naïf de L_{min}

Entrée: L

Sortie: L_{min}

(1) $L_{min} := \varnothing$

(2) pour chaque $t_i \in L$ faire

(3) si $\bar{\exists}\, t_j \in L$ tels que $t_j \prec t_i$ alors $L_{min} := L_{min} \cup \{t_i\}$

(4) retourner L_{min}

5.2.2 Stratégies d'évaluation

D'après [Tzitzikas et Meghini, 2003], il existe un certain nombre de stratégies pour évaluer des requêtes réseau dans un réseau P2P de sources taxonomiques. Dans notre approche, l'équation (3) est utilisée pour l'évaluation de requêtes dans une architecture P2P complètement décentralisée. Dans chaque pair, la base de métadonnées ne contient que la taxonomie locale et les liens vers d'autres pairs. Une requête réseau posée à ce pair est évaluée avec une réponse locale et produit éventuellement des (sous)requêtes envoyées à certains pairs voisins.

L'évaluation de requêtes réseau selon l'équation (3) ne s'arrête qu'à condition qu'il n'existe aucun cycle en parcourant des liens entre pairs. En d'autres termes, il n'y pas de chemins isA, à partir d'un terme t_1 dans un pair, revenant à un terme t_2 dans le même pair tels que t_2 est un ancêtre de t_1. Nous prenons comme hypothèse que les cycles potentiels seront détectés lors de la création (insertion) des liens (cf. un algorithme d'élimination de cycles dans [Fahndrich *et al.*, 1998]).

Nous étudions dans la suite deux stratégies qui différent par leur façon de retourner les réponses au pair initial.

Stratégie 1

Lorsqu'il reçoit une requête $Q(p, t)$, le pair p évalue la requête locale $Q_p(t)$ et attend les réponses retournées par ses voisins pour les sous requêtes $Q(p_i, t')$. Dès lors que tous les voisins ont répondu, le pair p renvoie l'union des résultats à l'utilisateur ou au pair qui a posé la requête $Q(p, t)$. L'Algorithme 1 implémente cette stratégie.

Algorithme 1 - $Q(p, t)$

Entrée: p le pair auquel la requête est posée, t un terme
Sortie: Q un ensemble de posts
(1) $Q := Q_p(t)$
(2) pour chaque $p' \in N(p)$ faire
(3) pour chaque $t' \in L_{min}(p, t, p')$ faire
(4) $Q := Q \cup Q(p', t')$
(5) retourner Q

$N(p)$ dénote l'ensemble de pairs voisins de p. Pour évaluer l'ensemble de posts locaux $Q_p(t)$, tous les successeurs de t dans p doivent être parcourus et pour chacun d'eux l'ensemble de posts instances est rassemblé. Nous donnerons dans la prochaine section une implémentation efficace de cette stratégie.

Stratégie 2

Dans la deuxième stratégie, les pairs envoient des (sous)requêtes à leurs voisins, mais n'attendent pas les réponses. Les réponses locales sont toujours envoyées directement au pair p_0, auquel la requête initiale a été posée. L'idée de cette stratégie est qu'il pourrait prendre moins de temps de renvoyer directement les réponses au pair p_0, au lieu de les transférer via une chaîne de pairs intermédiaires. La comparaison de ces deux stratégies en utilisant un modèle de coût du réseau sous-jacent et de la bande de passant est au-delà de cette étude. L'Algorithme 2 implémente cette deuxième stratégie. Il nécessite deux procédures. La procédure $Q(p, t, p_0)$ traite une requête posée à un pair p pour un terme t. p_0 est le pair auquel la réponse devrait être renvoyée par la procédure

Send(R, p_0) ; un pair envoie au pair p_0 l'ensemble de posts R, réponse locale à la requête reçue.

Algorithme 2

***Q*(p, t, p_0)**

Entrée : p le pair auquel la requête est posée, t un terme,
p_0 le pair auquel la réponse devrait être renvoyée

Sortie : mettre à jour l'ensemble de posts Q si $p=p_0$

(1) $R := Q_p(t)$
(2) si $p=p_0$ alors $Q := Q \cup R$
(3) sinon *Send*(R, p_0)
(4) pour chaque $p' \in N(p)$ faire
(5) pour chaque $t' \in L_{min}(p, t, p')$ faire
(6) $Q(p', t', p_0)$

***Send*(R, p_0)**

Entrée: R un ensemble de posts, p_0 le pair auquel la réponse est renvoyée

Sortie: mettre à jour l'ensemble de posts Q

(1) $Q := Q \cup R$

L'évaluation de requêtes réseau peut être encore optimisée comme suit. Pour évaluer une requête réseau, un pair peut recevoir des requêtes venant de différents pairs. Par exemple, pour évaluer la requête $Q(p, t)$ donnée dans la Figure 5.4, le pair p'' reçoit deux requêtes pour les termes t''_1 et t''_2, venant de p et de p', respectivement. Clairement, l'évaluation de t''_2 est redondante si t''_1 a été déjà évalué.

Afin d'éviter une telle redondance, on garde trace des requêtes qui ont été traitées dans chaque pair. Supposons que chaque requête initiale a un identifiant unique q. Etant donné $Q(p, t_1)$ une (sous)requête à évaluer pour une requête initiale q, l'évaluation de $Q(p, t_1)$ peut être suspendue si une requête $Q(p, t_2)$ a été évaluée pour la même requête initiale q et t_1 est un successeur de t_2. Cette optimisation peut être appliquée pour les deux stratégies d'évaluation ci-dessus. L'Algorithme 1bis donné ci-dessous est une extension de l'Algorithme 1 prenant en compte cette optimisation. De la même manière, on peut écrire l'Algorithme 2bis à partir de l'Algorithme 2.

Algorithme 1bis - *Q*(p, t, q)

Entrée: p le pair auquel la requête est posée, t un terme,
q un identifiant de la requête

Sortie: Q un ensemble de posts

(1) $Q := \varnothing$
(2) si $\bar{\exists}\,[q_i, t_i] \in Trace$ tels que $q_i = q$ et $t_i \preceq t$ alors
(3) $\quad Trace := Trace \cup \{[q, t]\}$
(4) $\quad Q := Q_p(t)$
(5) $\quad$ pour chaque $p' \in N(p)$ faire
(6) $\quad\quad$ pour chaque $t' \in L_{\min}(p, t, p')$ faire
(7) $\quad\quad\quad Q := Q \cup Q(p', t')$
(8) fin si
(9) retourner Q

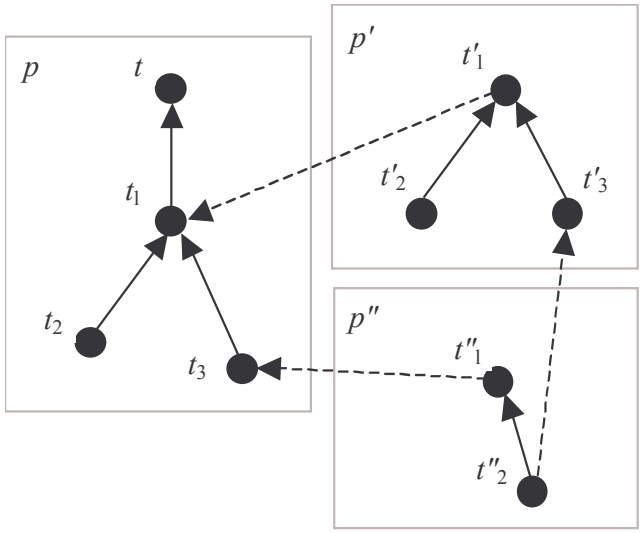


Figure 5.4 : Autre type de liens redondants

5.3 Implémentation efficace de requêtes en relationnel

Dans cette section nous présentons une implémentation utilisant un SGBD relationnel pour notre réseau P2P. Nous montrons que les requêtes RQL données dans la section 6.1 peuvent être efficacement calculées à l'aide d'un moteur SQL exploité dans chaque pair. Nous définissons d'abord le schéma relationnel adopté pour le stockage de métadonnées dans chaque pair.

5.3.1 Schéma relationnel

La Figure 5.5.a décrit un schéma relationnel utilisé pour l'exemple du web-of-people. La table Topic enregistre les topiques locaux à un pair. L'attribut turi est l'URI identifiant un topique, les attributs titre et created donnent le titre et la date de création. La table Link enregistre les liens taxonomiques vers d'autres pairs. turi est l'URI d'un topique local. targetturi et targetpeer donnent l'URI et le pair du topique cible du lien. Les posts sont stockés dans la table Post. Les attributs puri, titre, created, content, resourceRef dénotent respectivement l'URI, le titre, la date de création, la description de contenu et la référence de ressource pour un post. Afin de simplifier la présentation, le schéma donné n'est qu'une traduction réduite du schéma RDF général de la Figure 5.1. Sans perte de généralité, nous supposons qu'un topique a un seul père dans la hiérarchie locale (l'attribut parent dans la table Topic). De même, un post est classé dans un seul topique local (l'attribut turi dans la table Post).

Topic (turi, parent, title, created)
Link (turi, targetpeer, targetturi)
Post (puri, turi, title, created, content, resourceRef)
(a) *sans codage pour les termes*

Topic (turi, label, title, created)
Link (label, targetpeer, targetlabel)
Post (puri, label, title, created, content, resourceRef)
(b) *avec codage pour les termes*

Figure 5.5 : Schéma relationnel de métadonnées dans chaque pair

Du point de vue sémantique, un topique est équivalent à un terme au sens taxonomique. De nombreux codes (labeling schemes) ont été proposés dans la littérature pour coder des hiérarchies de termes, et notamment pour indexer des documents XML [Agrawal *et al.*, 1989 ; Dietz, 1982 ; Gavoille et Peleg, 2004 ; Kaplan *et al.*, 2002 ; Li et Moon, 2001]. Grâce à un tel codage, l'évaluation de requêtes ancêtres/descendants sur une hiérarchie peut être accélérée considérablement [Christophides *et al.*, 2003]. Nous remplaçons le schéma dans la Figure 5.5.a par le schéma dans la Figure 5.5.b où les termes sont étiquetés par un codage Dewey [Dewey, 1989]. L'attribut label dénote l'étiquette (code) d'un topique dans la table Topic. Cette étiquette est unique et utilisée pour identifier le topique dans d'autres tables. L'attribut turi (targetturi) est remplacé par label (targetlabel) dans Link et Post. Dans le codage Dewey, l'étiquette d'un terme est une chaîne de caractères. Un fils réutilise l'étiquette de son père comme préfixe de son étiquette. Nous montrerons plus loin que les requêtes réseau peuvent être

traduites efficacement en requêtes SQL à l'aide d'un tel codage appliqué pour les termes.

5.3.2 Evaluation de requêtes

Pour évaluer les requêtes RQL données, nous adoptons le schéma relationnel qui supporte le codage Dewey pour les termes (Figure 5.5.b). Selon la Proposition 1, dans tous les pairs, les liens redondants peuvent être éliminés a priori par une seule commande SQL comme ceci :

```
delete from Link r1
where exists (select * from Link r2 where r1.peer = r2.peer and
        not (r1.label = r2.label and r1.targetlabel = r2.targetlabel) and
        r2.label >= r1.label and r2.label < r1.label || 'xFF' and
        r1.targetlabel >= r2.targetlabel and r1.targetlabel < r2.targetlabel || 'xFF')
```

où || dénote une concaténation et label || 'xFF' est le code Dewey le plus grand pour un terme supérieur à label (voir [Christophides *et al.*, 2003] pour une discussion sur le codage de termes en SQL).

Nous donnons maintenant un plan d'exécution de la requête **Q1** (posts sous *Internet*) dans les deux stratégies d'évaluation (Algorithme 1 et Algorithme 2). Supposons que le terme *Internet* est identifié par l'étiquette *l* dans le plan d'exécution.

Plan d'éxécution Q1(*p*, *l*), stratégie 1

(1) Q := select puri from Post
        where label >= l and label < l || 'xFF'
(2) N := select distinct targetpeer from Link
        where label >= l and label < l || 'xFF'
(3) pour chaque $tp \in N$ faire
(4) L := select distinct targetlabel from Link
        where label >= l and label < l || 'xFF' and targetpeer = tp
(5) $L_{min} := Min(L)$
(6) pour chaque $tl \in L_{min}$ faire $Q := Q \cup$ **Q1**(tp, tl)
(7) fin pour
(8) retourner Q

$Min(L)$ est une fonction calculant L_{min} à partir de L par l'un des deux algorithmes suivants (A1 et A2). Grâce au codage des termes, l'ordre hiérarchique de deux topiques peut être vérifié par une comparaison de chaînes. Un terme t est un ancêtre d'un terme t'

si $l(t)$ est un préfixe de $l(t')$; $l(t)$ donne l'étiquette de t sous forme d'une chaîne de caractères.

Algorithme A1

Entrée : L trié
Sortie : L_{min}
(1) $L_{min} := \varnothing$
(2) *min* := 'xFF'
(3) pour $i := 1..|L|$ faire
(4) si $\neg$ *prefix*(*min*, $L[i]$) alors
(5) *min* := $L[i]$
(6) $L_{min} := L_{min} \cup \{min\}$
(7) fin si
(8) retourner L_{min}

Le calcul de L_{min} selon l'Algorithme A1 est linéaire dans la cardinalité de L. *prefix*(l,l') est une fonction booléenne, vraie si l est un préfixe de l'. Nous supposons que Link est trié physiquement sur l'attribut targetlabel et que le calcul de L par la commande SQL garde cet ordre dans le résultat.

Sinon, le calcul L_{min} est donné par l'Algorithme A2 qui ne suppose aucun tri sur L. La complexité de ce dernier est quadratique en $|L|$.

Algorithme A2

Entrée : L non trié
Sortie : L_{min}
(1) $L_{min} := \varnothing$
(2) pour $i := 1..|L|$ faire
(3) *min* := $L[i]$
(4) pour $j := 1..|L|$ faire
(5) si *prefix*($L[j]$, *min*) alors *min* := $L[j]$
(6) fin pour
(7) si *min* = $L[i]$ alors $L_{min} := L_{min} \cup \{min\}$
(8) fin pour
(9) retourner L_{min}

Notons que L_{min} peut être trouvé directement par la requête SQL ci-dessous, mais l'exécution de cette requête par un moteur SQL est moins efficace que l'implémentation directe de l'Algorithme A2.

L_{min} = select distinct targetlabel from Link r1
where r1.label >= l and r1.label < l || 'xFF' and r.targetpeer = tp and
not exists (select * from Link r2
where r2.targetpeer = tp and r2.label >= l and r2.label < l || 'xFF' and
r2.targetlabel > r1.targetlabel and r2targetlabel < r1.targetlabel||'xFF')

Voici le plan d'exécution de la requête **Q1** pour la deuxième stratégie.

Plan d'exécution de Q1(p, l, p_0), stratégie 2

(1) R := select puri from Post
where label >= l and label < l || 'xFF'
(2) si $p=p_0$ alors $Q := R$
(3) sinon *Send*(R, p_0)
(4) N := select distinct targetpeer from Link
where label >= l and label < l || 'xFF'
(5) pour chaque $tp \in N$ faire
(6) L := select distinct targetlabel from Link
where label >= l and label < l || 'xFF' and targetpeer = tp
(7) $L_{min} := Min(L)$
(8) pour chaque $tl \in L_{min}$ faire **Q1**(tp, tl, p_0)
(9) fin pour

De même manière, nous pouvons construire des plans d'exécution pour d'autres requêtes réseau. Par exemple, le plan d'exécution pour la requête **Q2** ressemble au plan **Q1**, mais la réponse locale est remplacée par la requête SQL suivante

Q := select puri from Post
where label >= l and label < l || 'xFF' and created >= 2004-01-01

La traduction pour les requêtes locales est encore plus facile. Par exemple :

Q4. Posts locaux sous *Internet* ? (l est l'étiquette du terme *Internet*)

select puri from Post
where label >= l and label < l || 'xFF'

Q8. Fils locaux du topique *Internet* ?

select turi from Topic
where label >= l and label < l || 'xFF' and length(label) = length(l) + 1

5.4 Implémentation et expérimentation préliminaire

Nous avons montré que l'évaluation de requêtes réseau pouvait être optimisée en éliminant des recherches redondantes renvoyant les mêmes posts. Cette section a pour objectif d'évaluer ce point à travers une expérience, en comptant pour chaque requête donnée le nombre d'appels (récursifs) économisés ainsi que le nombre de posts (redondants) écartés dans le résultat. Nous avons choisi, pour cette expérience, un échantillon réel basé sur 5 hiérarchies collectionnées à partir du site ODP (dmoz.org). Chaque hiérarchie correspond à un arbre de catégories avec des ressources indexées. Le Tableau 5.1 recueille des statistiques sur les 5 hiérarchies ODP collectionnées. Par exemple, la hiérarchie « Software » comporte 2316 topiques. La profondeur moyenne d'un topique de feuille est 4,79 et la profondeur maximale est 9. Le nombre moyen (maximal) de fils d'un topique est 4,44 (62). Le nombre moyen de posts attachés à un topique est 17,9 (41472 posts dans la hiérarchie entière). Le nombre de liens vers les pairs voisins est 124, c'est à dire que 5,3% des topiques dans la hiérarchie ont un lien.

Catégories	Topiques			Posts		Liens	
	#	Profond. moye. (max)	Largeur moye. (max)	#	Post /topique (max)	#	Lien /topique
Internet	1087	4.41 (8)	5.10 (65)	18307	16.8 (254)	73	0.067
Software	2316	4.79 (9)	4.44 (62)	41472	17.9 (469)	124	0.053
Hardware	226	3.71 (6)	4.26 (17)	6372	28.2 (255)	39	0.172
Systems	535	4.68 (8)	5.24 (51)	4642	8.7 (92)	38	0.071
Computer Science	190	4.18 (5)	6.78 (51)	1906	10 (121)	8	0.042
Total	4354	4.35 (9)	5.16 (65)	72699	16.7 (469)	282	0.065

Tableau 5.1 : Statistiques sur 5 ODP hiérarchies

Pour cette expérience préliminaire, nous avons fait les choix d'implémentation suivants :

1. Sans perte de généralité nous choisissons la requête **Q1** dont la performance est évaluée.

2. Tous les pairs ont été mis en oeuvre dans une seule base Oracle dont le schéma est conforme à celui dans la Figure 5.5.b, sauf qu'un attribut peer est ajouté dans chaque table (voir l'Annexe C). Le plan de requête **Q1**(p, l) est implémenté en PL/SQL. Cette implémentation simple procure le même

résultat en posts trouvés et le même nombre d'appels récursifs nécessaires qu'une implémentation où chacun des pairs aurait été mis dans une base séparée.

3. Comme nous n'avons pas considéré l'impact du trafic réseau dans cette évaluation, la stratégie 1 a été choisie pour l'implémentation en PL/SQL du plan de requête **Q1**(*p*, *l*).

4. Nous comparons trois implémentations du plan de requête. Dans la variante (1), on n'élimine aucun lien redondant. Tous les liens trouvés dans l'ensemble L sont suivis ($L_{\min}$ n'est pas calculé). Dans la deuxième implémentation (variante (2)), le calcul de $L_{\min}$ est fait selon l'Algorithme A2 (L n'est pas trié comme par l'Algorithme A1). Enfin, la variante (3) implémente l'Algorithme 1bis qui élimine d'autres calculs redondants comme illustré en Figure 5.4.

5. Nous exécutons des requêtes pour les topiques à différentes profondeurs de hiérarchie (niveaux). Les requêtes ont été posées sur toutes les hiérarchies. Le nombre de requêtes par niveau est égal au nombre de topiques dans les hiérarchies à ce niveau. Des requêtes qui n'exigent aucun lien à parcourir vers un pair voisin (requêtes locales) sont ignorées. Pour chaque requête en réseau, nous avons compté le nombre d'appels récursifs aux pairs voisins et le nombre de posts trouvés dans chacune des variantes d'implémentation. Une requête est *optimisable* s'il y a au moins un lien redondant éliminé au cours de l'évaluation par la variante (3). Avant d'exécuter les requêtes, nous avons pu éliminer définitivement 24 liens redondants (parmi 282) dans les hiérarchies selon la Proposition 1.

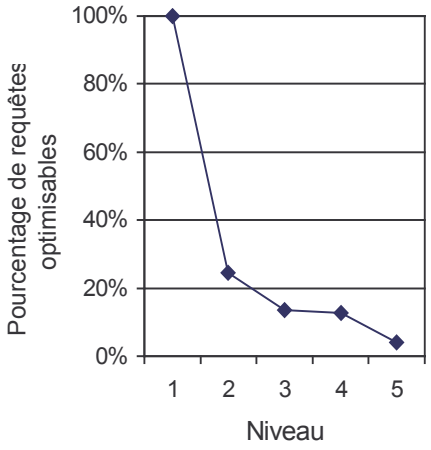
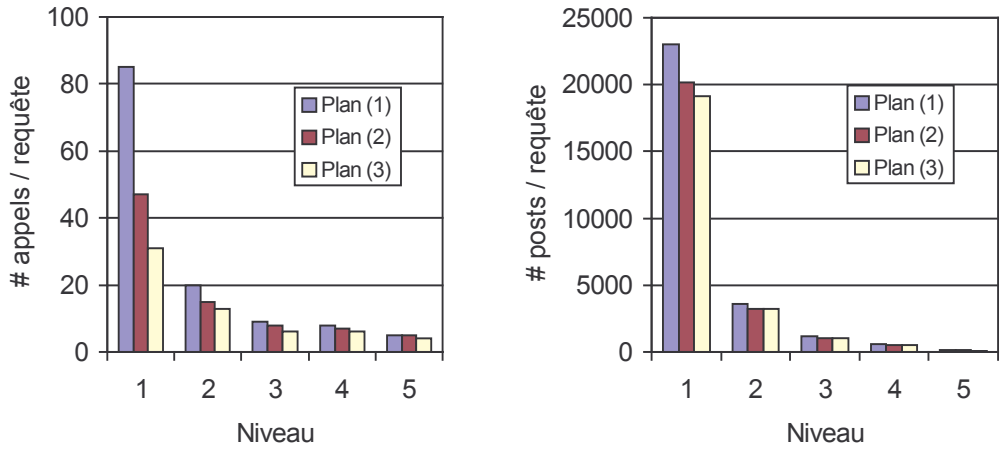


Figure 5.6 : Pourcentage de requêtes optimisables dans chaque niveau

La Figure 5.6 illustre le pourcentage de requêtes optimisables parmi les requêtes en réseau dans chaque niveau d'évaluation. Comme nous pouvions logiquement le prévoir, ce pourcentage diminue au fur à mesure que le niveau des topiques pour lesquels les requêtes sont posées augmente. Dans la Figure 5.7 nous donnons les résultats

d'évaluation pour toutes les requêtes optimisables classées par niveau : (a) le nombre moyen d'appels (récursifs) effectués, (b) le nombre moyen de posts trouvés dans chacun des trois variantes d'implémentation. La lecture de cette figure indique l'optimisation ne vaut pas le coût pour la recherche de posts en bas dans les hiérarchies (plus on est bas, moins il y a des liens à suivre dans notre jeu de données). Pour des recherches de posts en haut de la hiérarchie, en revanche, le plan 3 permet, par rapport au plan 1, d'économiser 64% des appels (Figure 5.7.a) et 17% des posts dans les résultats finaux (Figure 5.7.b).



(a) (b)

Figure 5.7 : Résultats d'évaluation pour les requêtes optimisables

Les résultats expérimentaux sont détaillés par les données du Tableau 5.2 qui affichent pour chaque niveau : a) le nombre de requêtes en réseau qui nécessitent au moins un appel d'un voisin et ceux pour lesquels un appel redondant existe, b) le nombre moyen d'appels aux pairs voisins, c) le nombre moyen de posts trouvés pour une requête optimisable dans chacune des variantes d'implémentation.

Niv.	Requête		Plan (1)		Plan (2)		Plan (3)	
	#	Optim.	#appels	#posts	#appels	#posts	#appels	#posts
1	5	5	85	22987	47	20132	31	19121
2	57	14	20	3620	15	3260	13	3196
3	88	12	9	1144	8	1061	6	1018
4	55	7	8	615	7	537	6	501
5	24	1	5	177	5	177	4	93

Tableau 5.2 : Résultats expérimentaux sur l'évaluation de requêtes

5.5 Conclusion

Dans ce chapitre, nous avons proposé une implémentation efficace pour un réseau P2P avec des taxonomies distribuées comme dans l'application web-of-people. Pour évaluer des requêtes dans un tel réseau, nous avons adopté une stratégie décentralisée pouvant bénéficier du parallélisme entre les différents pairs requis pour exécuter des calculs locaux. En outre, nous avons proposé une optimisation de requêtes avec (i) un mécanisme de détection de redondances pendant le temps d'exécution de chaque requête et (ii) un codage pour les hiérarchies de termes. Selon [Christophides *et al.*, 2003], un tel codage permet d'accélérer 3-4 fois, en ordre de grandeur, le temps de calcul pour les requêtes ancêtres/descendants dans une hiérarchie.

Comme il n'existe aucun jeu de données fourni par une application réelle comme le web-of-people, nous avons évalué le gain de performance obtenu par l'optimisation par une analyse expérimentale sur un ensemble de hiérarchies ODP. Cette évaluation préliminaire a montré qu'il existe un nombre considérable de redondances. Nous souhaiterions faire une évaluation plus approfondie sur le temps d'exécution de requêtes dans un vrai réseau P2P à grande échelle. Cette nouvelle expérience nous permettrait de valider l'hypothèse que l'optimisation de requêtes proposée pourrait améliorer la performance de requêtes en réseau. D'autre part, nous pourrions étudier comment deux stratégies choisies dans la section 5.2.2 affectent l'impact du trafic réseau dans l'évaluation de requêtes.

Dans la présentation de ce chapitre, nous avons assumé qu'aucun cycle n'a été créé dans les hiérarchies des pairs connectés. Cette condition assure que l'évaluation de requêtes selon l'une des trois variantes s'arrête dans tous les cas. Néanmoins, dans une application réelle telle que le web-of-people, il existe souvent des équivalences entre termes de différentes hiérarchies. Une telle équivalence entre deux termes t et t' ($t \equiv t'$) peut être modélisée par deux liens $[t, t']$ et $[t', t]$. Mais, le calcul de requêtes selon l'équation (3) risque de ne pas s'arrêter à cause d'un tel cycle entre t et t'. Dans ce cas, il faudrait d'autres algorithmes d'évaluation de point fixe.

III

E-Parcours

Chapitre 6

Vers des parcours sémantiques pour la recherche d'information

Dans ce chapitre, nous rendons compte d'une recherche au sein d'un projet interne au GET appelé « e-parcours » [Picouet et Saglio, 2003 ; Ta et Saglio, 2002]. Il s'agissait de l'étude d'une nouvelle voie en recherche d'information par l'utilisation de « parcours sémantiques ». Elle se distinguait de la recherche d'information au sens classique dans la mesure où l'ambition était de proposer, pour répondre (certes de manière la plus correcte possible) aux requêtes isolées d'un utilisateur, non pas par des documents candidats mais bien par un itinéraire que l'utilisateur devrait parcourir et qui lui donnerait « chemin faisant » les éléments d'une réponse « didactique » à sa question principale. Cette progressivité résulterait en partie d'une démarche argumentative et/ou pédagogique.

Pour les utilisateurs non-avertis du domaine d'application, une telle réponse sera pratiquement un guidage pour la découverte. Pour chaque question posée, ils pourront découvrir progressivement des ressources s'organisant par sous thèmes de lecture. Cette organisation est une nouvelle couche de présentation où, comme dans une métaphore routière, les thèmes de lecture sont reliés pour former des chemins « cognitifs ». Bien évidement, la rédaction d'une telle structure de présentation doit être mise sous la responsabilité de spécialistes du domaine d'application. A partir de cette présentation, des parcours sémantiques sont recommandés aux lecteurs finals pour qu'ils puissent naviguer dans les bons chemins de lecture.

Nous commencerons ce chapitre par la présentation d'une métaphore de gestion de ressources par laquelle nous situerons la nécessité des itinéraires de navigation pour faciliter la recherche d'information. Nous indiquerons ensuite comment nous concevons de tels itinéraires dans le cadre du web sémantique.

6.1 Métaphore de gestion documentaire

Depuis longtemps, la métaphore de la bibliothèque et/ou du musée est connue dans la gestion de ressources pour l'usage du grand public. Dans cette métaphore, le processus de qualification de ressources repose principalement sur trois étapes : (i) collectionner, (ii) relier, et (iii) présenter. Dans un premier temps, des ressources disponibles dans le monde entier sont collectionnées et enregistrées dans l'entrepôt avec des descriptions élémentaires telles que l'auteur de la ressource, la date de création, etc. Dans un deuxième temps, les ressources collectionnées sont évaluées et valorisées dans l'univers de référence de la communauté. Le modèle de l'univers peut être soit simplement celui d'une bibliothèque (i.e., théseaurus), soit une ontologie plus ou moins complexe. Enfin, les ressources qualifiées sont mises à disposition des utilisateurs pour usage ultérieur. Il peut exister plusieurs moyens pour la représentation, par exemple en donnant une interface de navigation par séquences hiérarchisées et indexées. Dans la suite, nous utiliserons cette métaphore de la gestion documentaire pour résumer les usages des technologies que nous voulons enrichir.

6.1.1 De la collection de ressources à la base de descriptions

Dans le web, les objets manipulés pour la recherche d'information s'appellent des ressources. Plusieurs modes de recherche d'information peuvent être offerts aux utilisateurs. La recherche documentaire basée sur les métadonnées décrivant ces ressources est le type le plus simple. Cette méthode classique privilégie l'indexation de ressources devant être collectionnées à partir de standards de métadonnées pour libraires tels que Dublin Core (titre, auteur, éditeur, date, etc.). La recherche d'information dans ce contexte est en générale exacte sur des valeurs d'attributs des métadonnées. Par exemple, nous pouvons chercher des documents pour un auteur, une date de publication,…

Les valeurs décrites par métadonnées sont souvent du type littéral. Elles peuvent être entrées librement par les utilisateurs sous formes de chaînes de caractères. Néanmoins, dans quelques systèmes les ressources pourraient être décrites à l'aide de vocabulaires plus contrôlés. Par exemple, les documents d'une bibliothèque peuvent être classifiés avec une taxonomie de catégorie pour faciliter la recherche par domaine. Les taxonomies sont les vocabulaires les plus simples. Elles représentent effectivement une classification hiérarchique de termes contrôlés. Le bénéfice des taxonomies est qu'elles permettent de grouper des termes reliés pour aider des utilisateurs à préciser leur besoin dans la recherche d'information.

Par rapport aux taxonomies, les thésaurus sont des objets plus riches (cf. les structures différentielles des taxonomies en sciences naturelles [Charlet, 2002]). Dans un thésaurus, les termes sont reliés par plusieurs types de liens : hiérarchique, équivalent, et associatif (relation). Selon les standards ISO (ISO2788, ISO5964) les relations majeures suivantes peuvent être définies entre termes : BT (Broader Term), NT (Narrower Term), SY (SYnonym), et RT (Related Term).

Afin d'offrir des recherches plus intelligentes, l'ontologie peut être utilisée dans une base de descriptions pour définir un schéma formel de connaissances dans lequel les ressources doivent se situer. Il existe de fait une grande variété d'approches de modélisation pour les ontologies dans la littérature de la représentation de connaissances. Dans le cadre du web sémantique le modèle RDF est la base pour les descriptions des ressources. Les ontologies peuvent être spécifiées à l'aide de RDFS ou OWL (cf. Chapitre 1).

6.1.2 Facilitation en recherche d'information

Quelle que soit l'approche utilisée pour la gestion de ressources documentaires, on utilise le plus souvent un « portail » jouant le rôle d'interface entre le système et les utilisateurs. Un portail peut fournir typiquement deux types de recherche : par navigation et/ou par requêtes. Chacun d'entre eux a son avantage particulier. L'interface de navigation sert souvent à visualiser le contexte général pour la recherche, tandis que les requêtes peuvent donner des résultats plus précis.

Classiquement, les ressources s'organisent en catalogue dans l'interface de navigation. Cette interface est utile pour les utilisateurs non-avertis qui ne savent pas encore exactement ce qu'ils doivent voir. En naviguant dans une hiérarchie de collections, les utilisateurs peuvent découvrir des ressources dans un contexte d'intérêt. Toutefois, les utilisateurs ont probablement un grand risque de tomber dans une forêt de répertoires et de ne rien trouver quand le volume de ressources dans un répertoire est incontrôlable.

Au contraire, pour pouvoir formuler des requêtes de recherche les utilisateurs devraient connaître plus ou moins le domaine de recherche. Grâce à leurs connaissances, ils peuvent entrer des critères de recherche appropriés à leur besoin via un formulaire fourni par l'application. La puissance de recherche par requêtes dépend effectivement de la capacité du système du point de vue sémantique.

Un parcours sémantique profite à la fois des avantages de la navigation et de la recherche par requête. Il permet aux utilisateurs non-avertis de découvrir ce qu'ils cherchent à travers des thèmes organisés. Au lieu de naviguer dans un espace de collections, les utilisateurs se voient recommander de suivre des chemins « faisant sens ». A chaque étape les utilisateurs peuvent se concentrer sur un nombre limité de ressources illustratives pour un thème particulier. La sélection de ressources pour ce thème peut être précisée à l'aide d'une requête dans la base de descriptions. L'organisation de tels thèmes significatifs en ordre cognitif aide les visiteurs à acquérir progressivement de nouvelles connaissances dans un domaine d'intérêt.

En fait, l'idée de parcours n'est pas neuve. Les expositions dans un musée sont exactement des parcours de présentation où les visiteurs découvrent des objets présentés sous des thèmes d'exposition. A partir d'un même entrepôt d'objets, on peut organiser la présentation dans beaucoup de parcours d'exposition différents. Ce type de présentation a été exploité dans un système d'information en ligne appelé

HyperMuseum [Stuer *et al.*, 2001]. Ce système repose sur une architecture de trois composants principaux : RDS (Resource Discovery Service), TCS (Theme Collaboration Service) et TRS (Theme Repository Service). Le service RDS offre l'interface pour interroger la collection virtuelle d'objets média. Plusieurs types de requêtes sont fournis, de la recherche plein texte à la recherche structurée. Le deuxième service (TCS) est conçu pour la communauté professionnelle. Il constitue une zone de travail commune permettant aux experts de créer des thèmes à partir des objets existants dans la collection de HyperMuseum. Les thèmes créés sont ensuite mis à disposition du public à travers le service TRS. Grâce à ce dernier service les visiteurs peuvent découvrir agréablement les ressources de la collection dans les thèmes prédéfinis par les experts.

6.2 Conception des parcours sémantiques

D'un point de vue conceptuel, nous distinguons trois « couches de données » dans la métaphore de gestion documentaire qui supporte des parcours sémantiques (cf. Figure 6.1). Dans la couche la plus base, les ressources collectionnés peuvent être décrites avec des métadonnées documentaires de type Dublin Core, ou bien en reliant à des concepts ontologiques définis dans la couche plus haute. Nous appelons *carte de connaissances* la couche qui contient des ontologies exploitées pour annoter/indexer les ressources collectionnées.

Afin de supporter des parcours sémantiques, il est nécessaire d'établir une nouvelle couche au-dessus de la carte de connaissances. Nous faisons l'hypothèse que les experts définiront des thèmes de lecture ainsi que des relations entre eux dans une structure de présentation appelée *carte de présentation*. Dans cette carte, un propos est une unité atomique de discours qui représente un thème complet avec une sélection de ressources dans la base de descriptions. Une telle sélection devrait utiliser des concepts dans la carte de connaissances pour donner un résultat pertinent. Par exemple, la requête « toute image de type jpg de la peinture La Joconde » permettra de trouver exactement les images pour illustrer un thème sur cette peinture. Un propos, en quelque sorte, peut être considéré comme une vue sur la carte de connaissances. Notons que notre métaphore rejoint la notion de thésaurus sémantique [Roussey *et al.*, 2002] où les thèmes de lecture correspondent aux termes d'un thésaurus qui peuvent renvoient à des concepts dans l'ontologie.

Pour pouvoir donner des chemins faisant sens, il existe entre propos des relations cognitives qui peuvent suggérer le bon ordre de lecture (e.g., prérequis, relié à, etc.). Comme dans un réseau routier, il est possible d'avoir différents itinéraires pour découvrir les thèmes dans une carte de présentation. Pour chaque question posée, un itinéraire est généré pour guider la lecture de l'utilisateur à travers des propos répondant à la question principale. La génération d'itinéraires de découverte peut être adaptée à la compétence de chaque utilisateur. Autrement dit, l'adaptation permettra de personnaliser le résultat de recherches en itinéraires selon le profil de chacun. Nous

précisons par la suite la structure de la carte de présentation et la méthode de génération d'itinéraires.

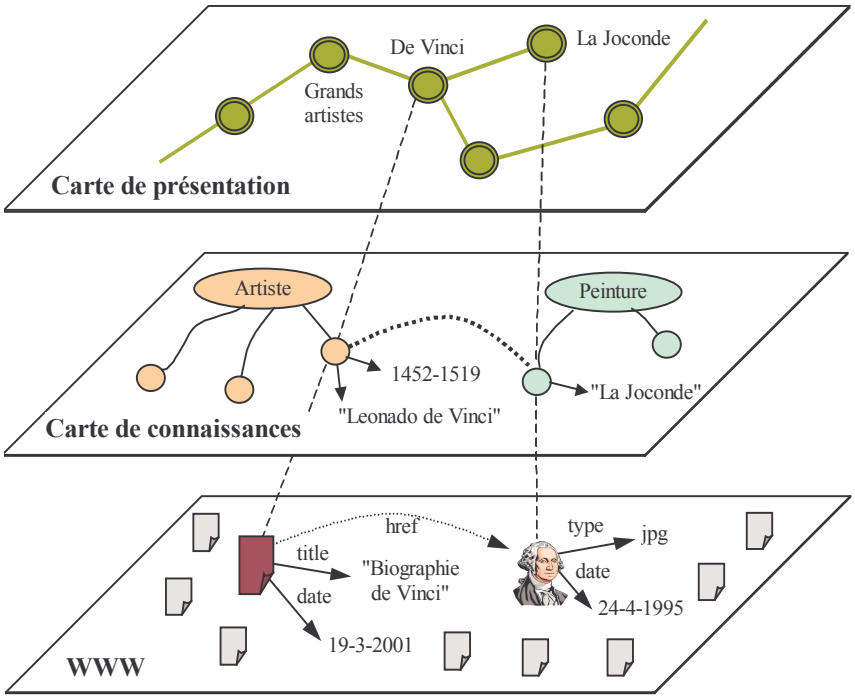


Figure 6.1 : Trois couches dans la métaphore de gestion documentaire

6.2.1 Carte de présentation

Une carte de présentation comporte des propos dont chacun représente une intention de la présentation. C'est l'équivalent d'un petit sujet ou d'un concept que l'auteur veut transmettre aux lecteurs dans un contexte particulier d'un domaine de connaissances. Les ressources illustrant le contenu dans chaque propos sont sélectionnées techniquement par une requête sur la base de descriptions. La Figure 6.2 donne un exemple avec les propos reliés au contenu de cette thèse. Imaginons que chaque propos dans cette carte donnera une vue bibliographique relative.

Comme nous l'avons vu, des relations cognitives sont nécessaires entre propos. En principe, ces relations peuvent donner une structure rhétorique plus ou moins complexe pour l'ordre de lecture des propos. De différents types rhétoriques tels que justification, re-formulation, démonstration, etc. ont été identifiés dans [Mann et Thompson, 1988]. Néanmoins, nous nous intéressons seulement aux quatre types les plus simples donnés ci-dessous. Ces relations sont très similaires avec les métalinks définis dans le cadre de la recherche de [Altıngövde *et al.*, 2001].

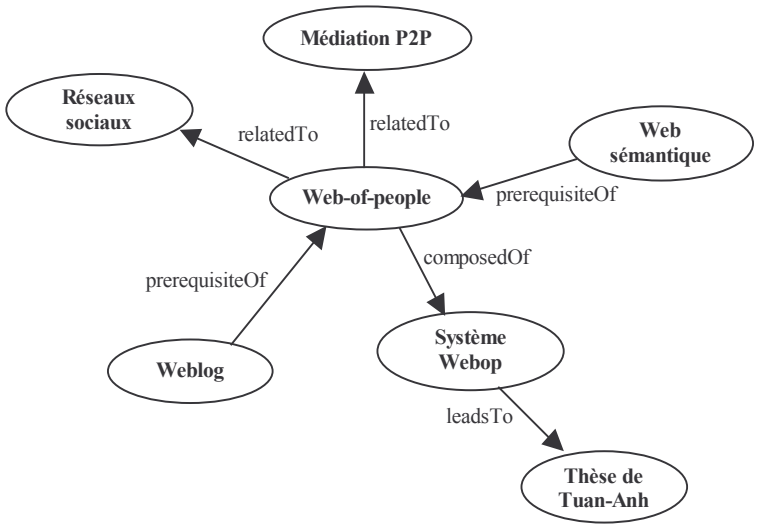


Figure 6.2 : Exemple des propos et relations cognitives

- ***leadsTo*** représente la recommandation la plus forte de l'auteur pour la lecture. Cette relation entre deux propos indique que l'un doit être visité lorsque l'autre l'a été. Par exemple, lorsque quelqu'un découvre « Système Webop », il est invité à se rapporter à « Thèse de Tuan-Anh ».

- ***prerequisiteOf*** exprime une condition de compétence pour pouvoir comprendre un propos. Par exemple, « Web sémantique » est un prérequis du propos « Web-of-people ». Bien évidemment, le prérequis doit être visité avant dans l'ordre de lecture pour ces deux propos.

- ***composedOf*** permet de proposer une présentation en détail pour un propos. Dans l'exemple donné, « Web-of-people » est composé de « Système Webop », car ce dernier précise le prototypage du web-of-people.

- ***relatedTo*** est la relation la plus faible entre propos. Elle ne représente pas une obligation en ordre et nécessité de lecture pour les propos reliés. Une personne s'intéresse à « Web-of-people » peut voir aussi « Médiation P2P ».

Pour aider le calcul dans le processus de génération d'itinéraires, chaque propos peut avoir des attributs complémentaires. Par exemple, on peut classer les propos dans plusieurs niveaux de difficulté comme « débutant », « intermédiaire », « avancé ». Une telle classification permet de privilégier des propos plus faciles à comprendre dans l'ordre de lecture. Cette classification peut être exploitée également pour générer des itinéraires adaptables à la compétence de chacun des utilisateurs. D'autres exemples pour les attributs de propos sont la fréquence de visite, l'appréciation du lecteur, …

Nous faisons l'hypothèse que les parcours sémantiques sont déployés dans le cadre du web sémantique. Un ensemble de vocabulaire particulier est utilisé pour décrire des propos dans une carte de présentation. Chaque propos est une ressource instanciée de la classe ep:Propos dont les propriétés sont :

- ep:title (string) : le titre assigné pour le propos ;
- ep:description (string) : la description textuelle du propos ;
- ep:query (string) : la requête qui donne accès à un ensemble des ressources gérées dans la base de descriptions ;
- ep:leadsTo, ep:prerequisiteOf, ep:composedOf, ep:relatedTo (propos) : les relations avec d'autres propos ;
- ep:level (number) : le niveau de difficulté du propos ; et d'autres propriétés,...

Nous venons de donner la liste préliminaire des propriétés utiles pour la description des propos. Il faudrait néanmoins une étude plus approfondie pour compléter cette liste.

6.2.2 Génération des itinéraires

Il est possible à partir des propos ainsi que des relations entre eux dans la carte de présentation de générer des scénarii de bonne lecture pour les visiteurs. Nous définissons un itinéraire comme étant chemin structuré traversant des propos choisis dans cette carte. La structure la plus simple pour un tel itinéraire est une succession des propos à visiter. Dans ce cas les visiteurs sont bien guidés sur un chemin d'étape en étape ; ils n'auront donc aucun choix de commutation dans la progressivité de lecture.

```
1 Weblog
2 Web sémantique
3 Réseaux sociaux
4 Web-of-people
  4.1 Système webop
  4.2 Thèse de Tuan-Anh
5 Médiation P2P
```

Figure 6.3 : Un itinéraire sous forme de table de matières

Une autre structure traditionnelle comme moyen de présentation est la table de matières. Dans ce cas, un itinéraire est un arbre ordonné de propos. La Figure 6.3 illustre un itinéraire sous forme d'une table de matières générée à partir de l'exemple donné. Cette table de matières favorise effectivement la lecture séquentielle des articles de haut en bas. Néanmoins les lecteurs peuvent sauter des étapes dans la progressivité de lecture. Par exemple, ils peuvent passer directement de « Web-of-people » à « Médiation P2P » sans découvrir les détails.

La Figure 6.4 donne une représentation en RDF de l'itinéraire de l'exemple ci-dessus. Cet itinéraire est décrit comme une instance de la classe ep:Itinerary qui

contient une séquence des étapes (objet de ep:steps). Chaque étape de l'itinéraire référence à soit un propos, soit un sous-itinéraire. Les propos sont identifiées par des URN (`urn:eparcours:xxx`).

```
<ep:Itinerary>
  <ep:title>Un exemple d'itinéraire</ep:title>
  <ep:steps><rdf:Seq>
    <rdf:li rdf:resource="urn:eparcours:weblog"/>
    <rdf:li rdf:resource="urn:eparcours:web-semantique"/>
    <rdf:li rdf:resource="urn:eparcours:reseaux-sociaux"/>
    <rdf:li rdf:resource="urn:eparcours:web-of-people"/>
    <rdf:li><ep:Itinerary>
      <ep:steps><rdf:Seq>
        <rdf:li rdf:resource="urn:eparcours:systeme-webop"/>
        <rdf:li rdf:resource="urn:eparcours:these-TuanAnh"/>
      </rdf:Seq></ep:steps>
    </ep:Itinerary></rdf:li>
    <rdf:li rdf:resource="urn:eparcours:mediation-P2P"/>
  </rdf:Seq></ep:steps>
<ep:Itinerary>

<ep:Propos rdf:about="urn:eparcours:web-semantique">
   <ep:title>Web Sémantique</ep:title>
   <ep:description>Introduction du web sémantique</ep:description>
   <ep:query>...</ep:query>
</ep:Propos>
...
```

Figure 6.4 : Représentation en RDF de l'itinéraire

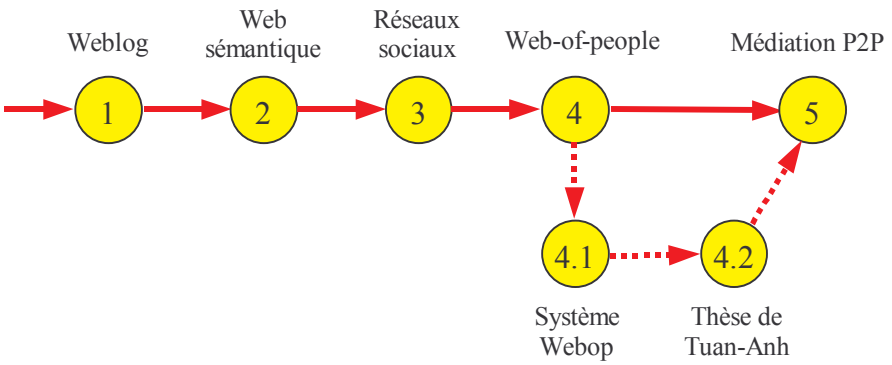


Figure 6.5 : Les étapes de lecture dans l'application

Notons qu'un itinéraire ne représente qu'un ordre de lecture. Cet ordre est visualisé dans une application pour donner concrètement les étapes de lecture. L'application peut

être conçue afin de pouvoir offrir des choix de commutation dans ces étapes. Par exemple, la Figure 6.5 illustre une interface qui permet à l'utilisateur de commuter pour voir le sous itinéraire d'une étape (s'il existe).

En principe, l'algorithme de génération d'itinéraires consiste en trois phases majeures. Nous décrivons brièvement ces trois phases ci-après.

- *Sélection de propos* : cette première phase permet de sélectionner des propos appropriés pour chaque question principale posée par un utilisateur. En se différenciant de la recherche d'information au sens classique, la requête d'utilisateur doit ne pas effectuer une recherche sur la base de descriptions, mais plutôt sur la carte de présentation (recherche de propos).

- *Relâchement* : après avoir sélectionné des propos, des relations entre propos sont considérées pour relâcher la sélection. Ceci permet de réunir pour le résultat final non seulement les propos trouvés par la requête, mais aussi des propos reliés à ceux-ci. Voici un exemple de relâchement dans le processus de sélection : si (*a*, ep:prerequisiteOf, *b*) et *b* a été sélectionné, on prend *a* pour la sélection. De même manière, nous traitons le relâchement pour les autres relations (ep:composedOf, ep:relatedTo, ep:leadsTo). En fait, l'objectif du relâchement est d'assurer que l'itinéraire généré sera complet pour la lecture.

- *Formation d'itinéraires* : cette dernière phase établit un chemin structuré sur les propos trouvés après la deuxième phase. L'algorithme de formation d'un tel chemin dépend parfaitement de la structuration choisie pour les itinéraires. Supposons que la table de matières est favorisée pour cette structuration des itinéraires, les règles suivantes peuvent être appliquées pour créer un arbre ordonné de propos: si (*a*, ep:prerequisiteOf, *b*), *a* doit être dans le même niveau et avant *b*; si (*a*, ep:composedOf, *b*), *b* se trouve dans le sous-itinéraire de *a* ; etc.

6.2.3 Exemple d'application

Pour illustrer le principe de parcours sémantiques dans une application concrète, revenons au prototype du web-of-people. Nous pouvons développer un serveur de parcours pour l'ensemble des weblogs partagés dans le réseau du web-of-people. En effet, il pourra jouer le rôle d'un portail pour les bases distribuées dans ce réseau. Une recherche dans ce portail donnera un guide de navigation permettant de découvrir des posts par thème de présentation.

Techniquement, le serveur de parcours consiste en une base de propos. Un outil d'édition collective de cette base est mis en place pour les experts de la communauté.

Chacun peut définir ses propos[14], mais doit les mettre en relation avec des propos qui existent déjà. Cela permet effectivement de générer des itinéraires qui peuvent traverser des propos créés par plusieurs experts. Comme la qualité des itinéraires dépend des propos produits, il est nécessaire de fournir une bonne interface pour faciliter l'édition collective de propos. En utilisant cette interface, un utilisateur se voit suggérer des propos similaires afin d'éviter la reproduction.

Dans le contexte de notre application, un propos ouvre l'accès à un ensemble de posts sélectionnés. C'est le service d'interrogation du web-of-people qui permet cette possibilité d'accès à travers le protocole à la LDAP (cf. section 3.4.1). Comme avec une requête LDAP (voir RFC2255), une requête du service d'interrogation peut être exprimée également sous forme d'une URL. Voici un exemple d'une telle URL

```
webop://u1$odp/Software??sub?([atom:created]>=2004-01-01)
```

qui donne accès à la PKB de l'utilisateur *u1$odp* avec les paramètres suivants :

- *Point d'entrée* : `webop://u1$odp/Software`
- *Etendu* : `sub`
- *Filtre* : `([atom:created]>=2004-01-01)`
- *Projection* : non

C'est l'équivalent d'une requête pour retrouver des posts créés depuis 2004 et indexés sous le topique *Software* (`webop://u1$odp/Software`). Cette URL doit être utilisée dans la description d'un propos pour spécifier la requête de sélection de posts.

6.3 Conclusion

Nous avons étudié, au cours de ce chapitre, une nouvelle facilité pour la recherche d'information autour du concept de parcours sémantique. Dans ce contexte, un système d'information devrait pouvoir fournir les trois services suivants : (i) accès aux ressources, (ii) édition collaborative de la carte de présentation et (iii) génération des itinéraires. Le premier service est connu comme la capacité d'interrogation permettant de chercher par métadonnées des ressources gérées dans la base du système. Les deux autres services permettent la construction d'une nouvelle couche de représentation afin que la réponse à une question posée par un utilisateur soit un itinéraire de navigation.

[14] Rappelons qu'un propos donne une vue sur les objets et relations de la carte de connaissances, donc sur les ressources qu'elle indexe.

Conceptuellement, un itinéraire est une composition représentant un ordre de lecture de propos. Chaque propos donne pratiquement une vue sur la carte de connaissances, à travers laquelle on trouvera les ressources de sélection. L'ordre de lecture généré doit respecter celui impliqué dans les relations cognitives entre propos qui ont été définis a priori par des experts-concepteurs. Nous avons bien défini quatre types de base de relations entre propos. Bien évidemment, il serait possible de définir d'autres types de relations qui représentent des structures rhétoriques plus complexes. Nous avons décrit également une méthode générale appliquée à la génération d'itinéraires. Il resterait néanmoins le problème plus difficile à traiter concernant l'adaptation des itinéraires aux profils de chaque utilisateur.

Afin de démontrer l'idée de parcours sémantique, nous n'avons implémenté que des maquettes dédiées à l'édition collaborative de parcours. Les fonctionnalités supportées dans ces maquettes sont encore très simples. Elles ne permettent que de créer des propos et puis générer des itinéraires sous forme de présentation HTML. Nous n'avons pas réalisé, au moment du projet, de couplage entre ces maquettes et un système d'information communautaire. C'est pourquoi nous aimerions dans un travail futur développer au moins un éditeur de parcours pour le web-of-people.

Conclusion générale

Dans cette thèse, nous avons traité de l'approche recherche d'information dans un *réseau social.* Le *web-of-people* est une application du *web sémantique* destinée à l'échange de connaissances et de documents entre personnes en réseau. Comme dans le principe du *filtrage collaboratif*, chacun des participants dans le web-of-people contribue par des revues personnelles sous forme de *posts* pour qualifier la collection de ressources de son choix. Une telle contribution est partagée avec tous ceux qui se trouvent dans le même groupe d'intérêt. Ainsi, une personne peut se servir de ce que d'autres ont qualifié pour son besoin de recherche d'information. D'un point de vue architecture, le web-of-people repose sur une architecture P2P dans laquelle chaque pair est propriétaire autonome d'une base de connaissances personnelles (sa *PKB*). Dans le modèle web-of-people de partage, une PKB peut être reliée sémantiquement à une autre à travers des propriétés décrivant le type de mise en relation entre leurs topiques. Ceci permet alors de constituer un réseau social de PKB connectées dans lequel des recherches d'informations peuvent être effectuées.

Notre premier effort a été de spécifier « à minima » le protocole de communication entre pairs dans le web-of-people. Ce protocole s'appuie seulement sur un mécanisme de notification pour assurer la cohérence entre les PKB mises en corrélation, mais en gardant l'autonomie de chacune. Pour chaque événement, défini comme le constat d'un changement sur une PKB, selon l'état des relations entre celle-ci et une autre PKB concernée par cet évènement, une notification doit être automatiquement envoyée au destinataire pour réaction. Dans le protocole, les réactions possibles sont spécifiées pour chaque type d'évènement à notifier.

Nous avons implémenté un prototype de validation du protocole général spécifié. Les suites de test faites avec ce prototype ont pu montrer l'intérêt à la fois technique (cohérence distribuée, intégration des vues distribuées) et social (confiance bien assurée) de ce protocole pour l'établissement d'échanges actifs entre personnes. Les principes suivants ont été choisis pour motiver la participation de chacun dans le réseau social.

i) Toute sorte de réutilisation, d'intégration de topique ou de personnalisation de post, doit être suivie d'une notification afin d'établir la reconnaissance de « crédits » dans la PKB du contributeur. De tels crédits établiront la « réputation » par laquelle un utilisateur sera consacré pour ses contributions.

ii) Le mécanisme de restriction d'accès est l'une des bases de l'établissement de la confiance dans le réseau social. Celui qui est un participant actif pourra être récompensé par l'ouverture d'accessibilité aux PKB de ses lecteurs.

iii) L'interdiction du spam peut être mise sous contrôle des dialogues de confiance dans les échanges actifs entre utilisateurs. Un utilisateur ne recevra la notification d'un nouveau post que si le post est dans le cadre d'un abonnement accepté a priori par lui-même.

Une fois que le réseau social est construit en conformité à ce protocole d'application, des requêtes peuvent être formulées pour effectuer des recherches de posts dans ce réseau. Au cours du chapitre 5, nous nous sommes centrés sur le problème de l'évaluation efficace de telles requêtes réseau. Notre apport dans ce chapitre est une implémentation efficace en relationnel pour chaque pair dans le réseau. Dans cette implémentation, l'optimisation de requêtes a été réalisée grâce à deux techniques : a) détection et élimination de calculs redondants pendant le temps d'exécution de requêtes ; b) codage de termes par un schéma d'étiquetage pour accélérer des calculs de fermeture transitive.

Enfin, la dernière partie de cette thèse a rendu compte de l'étude[15] d'une nouvelle fonctionnalité en recherche d'information autour du concept de *parcours sémantique*. Pour chaque question principale posée par un utilisateur, un itinéraire, comme aide à la navigation, est sélectionné pour lui donner un guide de lecture argumentative. Cette démarche convient bien aux utilisateurs qui sont des « débutants » pour leur « ouvrir » un domaine de connaissance. Comme les itinéraires sont générés à partir d'une représentation autour de thèmes soigneusement construits par les experts du domaine, les utilisateurs peuvent profiter de leurs connaissances pour être bien guidés dans les bons chemins de découverte. Notons que cette fonctionnalité ne remplace pas les techniques de recherche d'information au sens classique où chaque réponse est une collection de ressources trouvées par une requête posée directement sur la base de gestion. Il s'agit de fait d'une nouvelle forme de réponse qui présente des *parcours de découverte* plutôt que des listes de ressources sélectionnées.

[15] provisoirement achevée en 2002, avant que le projet web-of-people occupe tous nos efforts

Références

[1] Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.C., Simon, L., 2004. "*Somewhere in the semantic web*", LRI Technical Report.

[2] Agosto, L., Plu, M., Vignollet, L., Bellec, P., 2003. "*SomeOne: a cooperative system for personalized information exchange*", International Conference on Enterprise Information System (ICEIS), Angers, France.

[3] Agrawal, R., Borgida, A., Jagadish, H.V., 1989. "*Efficient management of transitive relationships in large data and knowledge bases*", in Proc. of the SIGMOD International Conference On Management Of Data, pages 253--262.

[4] Aimeur, E., Brassard, G., Paquet, S., 2003. "*Using personal knowledge publishing to facilitate sharing across communities*", in M. Gurstein (Ed.), Proceedings of the 3rd International Workshop on (Virtual) Community Informatics: Electronic Support for Communities - Local, Virtual and Communities of Practice.

[5] Alashqur, A., Su, S.Y., Lam, H., 1989. "*OQL: A Query Language for Manipulating Object-Oriented Databases*", in Proceedings of the International Conference on Very Large Databases, Amsterdam, 1989, pp. 433-442.

[6] Alberg, C., Shneiderman, B., 1994. "*Visual information seeking: Tight coupling of dynamic query filters with starfield displays*", Proceedings of CHI '94, Boston MA, ACM Press, 313-317.

[7] Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D., Tolle, K., Amann, B., Fundulaki, I., Scholl, M., Vercoustre, A.M., 2000. "*Managing RDF Metadata for Community Webs*", 2nd International Workshop on the World Wide Web and Conceptual Modeling, pp. 140-151.

[8] Alfred, J., Menezes, P.C., Oorschot V., Scott A.V., 1997. "*Handbook of Applied Cryptography*", CRC Press, Boca Raton, FL.

[9] Altıngövde, I.S., Özel, S.A., Ulusoy, O., Özsoyoğlu, G., Özsoyoğlu, Z.M., 2001. "*Topic-Centric Querying of Web Information resources*", DEXA Conf.

[10] Amann, B., Michard, A., 2001. "*The C_Web Architecture Specification V1.0*", technical report. INRIA.

[11] Arumugam, M., Sheth, A., Budak Arpinar, I., 2002. "*The peer-to-peer semantic web: A distributed environment for sharing semantic knowledge on the web*", in WWW2002 Workshop on Real World RDF and Semantic Web Applications. Honolulu, Hawaii (USA).

[12] Bar-Ilan, J., 2004. "*An Outsider's View on Topic-oriented Blogging*", In Alternate track papers & posters of the 13th international conference on World Wide Web.

[13] Bausch, P., Haughey, M., Hourihan, M., 2002. "*We blog: Publishing online with Weblogs*", Wiley Publishing, Indianapolis.

[14] Berners-Lee, T., 2000 "Semantic Web", Talk at XML 2000 Washington DC.

[15] Berners-Lee, T., Hendler, J., Lasilla, O., 2001. "*The Semantic Web*", Scientific American.

[16] Bernstein, P., Giunchiglia, F., Kementsietsidis, A., and L. Serafini, J.M., Zaihrayeu, I., 2002. "*Data management for peer-to-peer computing: A vision*", WebDB'02.

[17] Berrut, C., Denos, N., 2003. "*Filtrage collaboratif*", in Assistance intelligente à la RI, Hermes - Lavoisier, chapter 8, pp30.

[18] Boag, S., Chamberlin, D., Fernandez, M.F, Florescu, D., Robie, J., J. Simon, (eds), 2004. "*XQuery 1.0: An XML Query Language*", W3C Working Draft, http://www.w3.org/TR/xquery/

[19] Boley, H., Tabet, S., Wagner, G., 2001. "*Design Rationale of RuleML: A Markup Language for Semantic Web Rules*", in Proc. SWWS'01, Stanford.

[20] Bonifacio, M., Bouquet, P., Traverso, P., 2002. "*Enabling distributed knowledge management*", managerial and technological implications, Novatica and Informatik/Informatique, III(1), 2002.

[21] Bosak, J., Bray, T., 1999. "*XML and the Second-Generation Web*", Scientific American.

[22] Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Sure, Y., Tane, J., Volz, R., Zacharias, V., 2002. "*KAON - Towards a large scale Semantic Web*", in 3rd International Conference E-Commerce and Web Technologies, EC-Web 2002, Aix-en-Provence, France, September 2-6, 2002, Proceedings, volume 2455 of LNCS, pp. 304-313. Springer.

[23] Bressan, S., Fynn, K., Goh, C. H., Jakobisiak, M., Hussein, K., Kon, H., Lee, T., Madnick, S., Pena, T., Qu, J., Shum, A., Siegel, M., 1997. "*The COntext INterchange mediator prototype*", in Proc. ACM SIGMOD/PODS Joint Conference, Tuczon, AZ 1997, pp. 525—527.

[24] Brickley, D., Guha, R.V., 2004. "*RDF Vocabulary Description Language 1.0: RDF Schema*", W3C Recommendation, http://www.w3.org/TR/rdf-schema/

[25] Broekstra, J., Ehrig, M., Haase, P., van Harmelen, F., Kampman, A., Sabou, M., Siebes, R., Staab, S., Stuckenschmidt, H., Tempich, C., 2003. "*A metadata model for semantics-based peer-to-peer systems*", in Workshop on Semantics in Peer-to-Peer and Grid Computing, WWW'03 Budapest.

[26] Brunkhorst, I., Dhraief, H., Kemper, A., Nejdl, W., Wiesner, C., 2003. "*Distributed Queries and Query Optimization in Schema-Based P2P-Systems*", International Workshop On Databases, Information Systems and Peer-to-Peer Computing, VLDB'03, Berlin.

[27] Cai, M., Frank, M., 2004. "*RDFPeers: A Scalable Distributed RDF Repository based on a structured Peer-to-Peer network*", in 13th International World Wide Web Conference, WWW'04 New York.

[28] Cai, M., Frank, M., Chen, J., Szekely, P., 2003. "*MAAN: A multi-attribute addressable network for grid information services*", In 4th Int'l Workshop on Grid Computing.

[29] Candan, K.S., Liu, H., Suvarna, R., 2001. "*Resource description framework: metadata and its applications*", ACM SIGKDD Explorations Newsletter.

[30] Charlet, J., 2002. "*L'Ingénierie des connaissances: développements, résultats et perspectives pour la gestion des connaissances médicales*", Habilitation à diriger des recherches, Université Paris 6

[31] Charlet, J., Laublet, P., Reynaud, C. (Ed.), 2003. "*Web sémantique*", Rapport final, Action spécifique 32 CNRS / STIC.

[32] Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S., 2003. "*On Labeling Schemes for the Semantic Web*", 12th International World Wide Web Conference, WWW'03 Budapest.

[33] Clark, J., DeRose, S., 1999. "*XML path language. Version 1.0*", W3C Recommendation.

[34] Dao V.P., Ta T.A., Saglio J.M., 2004. "*Fonctions et Scénarios pour une application sur la plateforme Webop*", Spécification d'application, Projet Webop ENST-FTR&D.

[35] Dean, M., Connolly, D., Harmelen F. van, Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., 2002. "*OWL web ontology language 1.0 reference*".

[36] Decker, S., Erdmann, M., Fensel, D., Studer, R., 1999. "*Ontobroker: Ontology based access to distributed and semi-structured information*", in R. Meersman *et al.*, editor, DS-8: Semantic Issues in Multimedia Systems. Kluwer Academic Publisher.

[37] Dewey, M., 1989. "*Dewey Decimal Classification and Relative Index*", Forest Press, 20 edition.

[38] Dietz, P. F., 1982. "*Maintaining order in a linked list*", In Proc. of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC'82), pages 122--127.

[39] Ehrig, M., Tempich, C., Broekstra, J., Harmelen, F.v., Sabou, M., Staab, S., Stuckenschmidt, H., Siebes. R., 2003. "*SWAP - Ontology-based Knowledge Management with Peer-to-Peer Technology*", in Proceedings of the 1st National Workshop WOW.

[40] Elst, L., Dignum, V., Abecker, A., (eds), 2003. "*Agent-Mediated Knowledge Management*", International Symposium AMKM 2003, Stanford, CA, USA. Lecture Notes in Artificial Intelligence (LNAI) 2926. Springer, Berlin.

[41] Fahndrich, M., Foster, J.S., Su, Z., Aiken, A., 1998. "*Partial online cycle elimination in inclusion constraint graphs*", in Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation.

[42] Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H.-P., Staab, S., Studer, R., and Witt, A., 1999. "*On2broker: Semantic-based access to information sources at the WWW*", in World Conference on the WWW and Internet (WebNet99), Honolulu, Hawaii.

[43] Gandon F., Dieng-Kuntz R., Corby O. & Giboin, A., 2002. "*Web Sémantique et Approche Multi-Agents pour la Gestion d'une Mémoire Organisationnelle Distribuée*", Journées Ingénierie des Connaissances, p.15-26.

[44] Gargantilla, J.Á.R., Gómez-Pérez, A., 2004. "*A survey on ontology-based applications. E-commerce, knowledge management, multimedia, information sharing and educational applications*". OntoWeb Deliverable 1.6.

[45] Garshol, L.M., 2003. "*Living with topic maps and RDF*", http://www.ontopia.net/topicmaps/materials/tmrdf.html

[46] Gavoille, C., Peleg, D., 2003. "*Compact and localized distributed data structures*", Journal of Distributed Computing, Special Issue for the Twenty Years of Distributed Computing Research.

[47] Goasdoué, F., Rousset, M.C., 2003. "*Querying Distributed Data through Distributed Ontologies: a Simple but Scalable Approach*", IEEE Intelligent Systems.

[48] Halevy, A.Y., Ives, Z.G., Mork, P., Tatarinov, I., 2003. "*Piazza: data management infrastructure for semantic web applications*", in Proceedings of the Twelfth International World Wide Web Conference, WWW2003, Budapest, Hungary.

[49] Hammersley, B., 2003. "*Content Syndication with RSS*", O'Reilly

[50] Harren, M., Hellerstein, J. M., Huebsch, R., Loo, B. T., Shenker, S., Stoica, I., 2002. "*Complex Queries in DHT based Peer-to-Peer Networks*", in The 1st Interational Workshop on Peer-to-Peer Systems (IPTPS'02).

[51] Heflin, J. Hendler, J., 2000. "*Searching the web with shoe*", in Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01, pages 35--40. AAAI Press.

[52] Horricks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M., 2003. "*SWRL: A Semantic Web Rule Language Combining OWL and RuleML*", version 0.5.

[53] Horrocks, I., Harmelen, F.v., Patel-Schneider, P., (ed.), 2001. "DAML+OIL Specification", http://www.daml.org/2001/03/daml+oil-index.html

[54] Ives, Z.G., Halevy, A.Y., Mork, P., Tatarinov, I., 2004. "*Piazza: Mediation and Integration Infrastructure for Semantic Web Data*", Journal of Web Semantics, Vol. 1 No. 2, p. 155-175.

[55] Kaplan, D., Guillaud, H., 2003. "*Les weblogs : du phénomène à la révolution*", Paru le 27/05/2003 sur http://www.fing.org/index.php?num=3804,2

[56] Kaplan, H., Milo, T., Shabo, R, 2002. "*A comparison of labeling schemes for ancestor queries*", in Proc. of the Thirteen Annual Symposium on Discrete Algorithms (SODA'02).

[57] Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M., 2002. "*RQL: A Declarative Query Language for RDF*", The Eleventh International World Wide Web Conference, WWW'02 Hawaii.

[58] Karvounarakis, G., Magkanaraki, A., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M., Tolle, K., 2004. "*RQL: A Functional Query Language for RDF*", at The Functional Approach to Data Management: Modelling, Analyzing and Integrating Heterogeneous Data, P.M.D. Gray, L. Kerschberg, P.J.H. King, A. Poulovassilis (eds.), LNCS Series, Springer-Verlag.

[59] Kautz, H., Selman, B., Shah, M., 1997. "*ReferralWeb: Combining Social Networks and Collaborative Filtering*", Communications of the ACM, Vol. 40(3):pages 63--65.

[60] Kifer, M., Lausen, G., 1989. "*F-logic: A higher-order language for reasoning about objects, inheritance, and scheme*", in Proceedings of the ACM SIGMOD 1989.

[61] Koivunen, M.R., Swick, R., Kahan, J., Prud'hommeaux, E., 2003. "*Annotea Shared Bookmarks*", KCAP03 workshop, Sanibel, Florida.

[62] Kokkelink, S., 2001. "*Transforming RDF with RDFPath*", Working Draft, http://zoe.mathematik.uni-osnabrueck.de/QAT/Transform/RDFTransform.pdf

[63] Lassila, O., Swick, R.R., 1999. "*Resource description framework (RDF) Model and syntax specification*", W3C Recommendation, http://www.w3.org/TR/REC-rdf-syntax/

[64] Lassila, O., van Harmelen, F., Horrocks, I., Hendler, J., McGuinness, D.L., 2000. "*The semantic Web and its languages*", IEEE Intelligent Systems, Volume: 15 Issue: 6.

[65] Laublet, P., Reynaud, C., Charlet, J., 2002. "*Sur quelques aspects du Web sémantique*". Journées scientifiques Web sémantique, Paris.

[66] Li, Q., Moon, B., 2001. "*Indexing and querying XML data for regular path expressions*", in Proc. of 27th Inter. Conf. on Very Large Data Bases, VLDB'02.

[67] Magkanaraki, A., Karvounarakis, G., Christophides, V., Plexousakis, D., Ta. T., 2002. "*Ontology Storage and Querying*", Technical Report No 308, ICS-FORTH.

[68] Maltz, D., Ehrlich, K., 1995. "*Pointing the way: active collaborative filtering*", Proceedings of CHI'95, p. 7-11.

[69] Mann, W.C., Thompson, S.A., 1988. "*Rhetorical Structure Theory: Toward a functional theory of text organization*", Text 8 (3).

[70] Martin, Ph., Eklund, P., 1999. "*Embedding Knowledge in Web Documents: CGs versus XML-based Metadata Languages*", in: ICCS'99, 7th International Conference on Conceptual Structures, Springer Verlag, LNAI 1640, p. 230--246.

[71] McBride, B., 2002. "*Jena: A semantic web toolkit*", IEEE Internet Computing, 6(6):55--59.

[72] Mena, E., Kashyap, V., Sheth, A., Illarramendi, A., 1996. "*Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies*", in Proceedings 1st IFCIS

International Conference on Cooperative Information Systems (CoopIS '96). Brussels.

[73] Miller, L., Seaborne, A., Reggiori, A., 2002. "*Three implementations of SquishQL, a simple RDF query language*", in First Int'l Semantic Web Conference.

[74] Milojicic, D. S., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S., Xu, Z., 2002. "*Peer-to-peer computing*", Technical Report HPL-2002-57, HP Lab.

[75] Mortensen, T., Walker, J., 2002. "*Blogging Thoughts : Personal Publication as an Online Research Tool*", in Researching ICTs in Context, A. Morrison, Editor, Intermedia: Oslo. p. 249-279

[76] Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmr, M., Risch, T., 2002. "*Edutella: a p2p networking infrastructure based on rdf*", in Proceedings of the eleventh international conference on World Wide Web, pages 604-615, ACM Press.

[77] Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Loser, A., 2003. "*Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks*", in Proc. WWW2003, pages 536--543.

[78] Ng, W., Ooi, B., Tan, K., Zhou, A., 2003. "*Peerdb: A p2p-based system for distributed data sharing*", ICDE'03.

[79] Ogbuji, U., 2002. "*RDF Query using Versa*", in IBM developerWorks, Thinking XML: Basic XML and RDF techniques for knowledge management, Part 6.

[80] Page, L., Brin, S., Motwani, R., Winograd, T., 1998. "*The pagerank citation ranking: Bringing order to the web*", Technical report, Stanford Digital Library Technologies Project.

[81] Papadimos, V., Maier, D., Tufte, K., 2003. "*Distributed Query Processing and Catalogs for Peer-to-Peer Systems*", in: CIDR 2003, First Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA.

[82] Paquet, S., 2002. "*Personal knowledge publishing and its uses in research*", http://radio.weblogs.com/0110772/stories/2002/10/03/personalKnowledgePublishingAndItsUsesInResearch.html

[83] Paquet, S., and Pearson, P., 2004. "*A Topic Sharing Infrastructure for Weblog Networks*", to appear in Proceedings of the Communication Networks and Services Research (CNSR), IEEE Computer Society Press.

[84] Pepper, S. (ed.), 2003. "*Published Subjects: Introduction and Basic Requirements*", OASIS Published Subjects Technical Committee Recommendation, http://www.ontopia.net/tmp/pubsubj-gentle-intro.htm

[85] Pepper, S., Moore, G., 2001. "*XML Topic Maps (XTM) 1.0*". TopicMaps.org Specification.

[86] Picouet, P., Saglio, J.M., 2002. "*Définition de parcours sur un Web Communautaire*", Rapport GET-ENST.

[87] Pitoura, E., Abiteboul, S., Pfoser, D., Samaras, G., Vazirgiannis, M., 2003. "*DBGlobe: a service-oriented P2P system for global computing*", Sigmod Record 32, p. 77–82.

[88] Plu, M., Agosto, L., Bellec, P., Van De Velde, W., 2003. "*The Web of People: a dual view on the WWW*", Twelfth International World Wide Web Conference, WWW'03 Budapest.

[89] Plu, M., Agosto, L., Vignollet, L., Marty, J.C., 2004. "*A contact recommander system for a mediated social media*", International Conference on Enterprise Information System (ICEIS), Porto.

[90] Prié, Y., 2000. "*Sur la piste de l'indexation conceptuelle de documents*", Document numérique, vol 4.

[91] Quint, V., 2003. "*Documents structurés sur le Web*", Actes du congrès IDT/Net 2002, http://www.w3.org/2002/03/VQ-IdtNet2002.html

[92] Ratnasamy, S., Francis, P., Handley, K., Karp, R., Shenker, S., 2001. "*A scalable content-addressable network*", in Proceedings of ACM SIGCOMM 2001.

[93] Rodzvilla, J. (Ed.), 2002. "*We've got blog: How weblogs are changing our culture*", Perseus Publishing, Cambridge, MA.

[94] Rolland, C., Foucaut, O., Benci, G., 1988. "*Conception des Systèmes d'information, La méthode Remora*", Eyrolles.

[95] Rousset, M.C., 2004. "*Small Can Be Beautiful in the Semantic Web*", Proceedings of International Semantic Web Conference (ISWC 2004), pages 6--16.

[96] Roussey C., Calabretto S., Pinon J., 2002. "*Le thésaurus sémantique: Contribution à l'ingénierie des connaissances documentaires*", actes de la 13ème journée francophone de l'Ingénierie des Connaissances, Rouen, p 209-220.

[97] RSS-DEV Working Group, 2000. "*RDF Site Summary (RSS) 1.0*", http://web.resource.org/rss/1.0/spec

[98] Rucker, J., Polanco, M.J., 1997. "*Siteseer: personalized navigation for the Web*", Communications of the ACM, vol. 40, n° 3, p. 73-75.

[99] Saglio, J.M., Scholl, M., TA, T.A., 2005. "*Efficient query processing in P2P networks of taxonomy based systems*", to appear in International Workshop on Data Integration & the Semantic Web, CAiSE'05 Porto.

[100] Salton, G., McGill, M. J., 1983. "*Introduction to modern information retrieval*", NY: McGraw-Hill.

[101] Sartiani, C., Manghi, P., Ghelli, G., Conforti, G., 2004. "*XPeer: A Self-organizing XML P2P Database System*", in Proceedings of the First EDBT Workshop on P2P and Databases (P2P&DB 2004), Crete, Greece.

[102] Sheth, A., Thacker, S., Patel, S., 2003. "*Complex relationships and knowledge discovery support in the InfoQuilt system*", in The VLDB Journal, Volume 12, Issue 1.

[103] Sintek, M., Decker, S., 2001. "*TRIPLE-An RDF Query, Inference, and Transformation Language*", in Proceedings of the Deductive Databases and Knowledge Management Workshop, DDLP'2001, Japan.

[104] Sowa, J.F., 1998. "*Conceptual Graph Standard and Extensions*", in Lecture Notes in AI Vol. 1453, New York:Springer-Verlag

[105] Spivack, N., 2004. "*The Future of the Net*", http://novaspivack.typepad.com/nova_spivacks_weblog/2004/04/new_version_of_.html

[106] Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H.P., Studer, R., Sure, Y., 2000. "*Semantic community web portals*", Proc. of WWW9 / Computer Networks, 33(1-6):473–491.

[107] Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H., 2001. "*Chord: A scalable peer-to-peer lookup service for internet applications*", In ACM SIGCOMM'01.

[108] Stuckenschmidt, H., Wache, H., Vögele, T., Visser, U., 2000. "*Enabling technologies for interoperability*", in Workshop on the 14th International Symposium of Computer Science for Environmental Protection, pages 35–46, Bonn, Germany.

[109] Stuer, P., Meersman, R., Bruyne. S.D., 2001. "*The HyperMuseum Theme Generator System: Ontology-based Internet Support for the Active Use of Digital Museum Data for Teaching and Presentation*", Museums and the Web 2001, Sweden.

[110] Ta T.A., 2003. "*Navigation par itineraires dans un web communautaire*", Conférence de Recherche en Informatique Vietnam-Francophonie, RIVF'03, Hanoi.

[111] Ta, T.A., Saglio, J.M., 2002. "*Génération de parcours recommandés dans un Web communautaire*", Journées scientifiques Web sémantique CNRS, Octobre.

[112] Ta, T.A., Saglio, J.M., 2004. "*An object zooming model for the Semantic Web*", ICTTA'04 Syria.

[113] Ta, T.A., Saglio, J.M., Plu, M., 2004. "*An architecture based on semantic weblogs for exploring the Web of People*", Workshop Application of Semantic Web Technologies to Web Communities, ECAI'04 Valencia.

[114] Tempich, C., Staab, S., Wranik, A., 2004. "*REMINDIN': Semantic query routing in peer-to-peer networks based on social metaphors*", in International conference on World Wide Web, WWW'04, Newyork.

[115] Trott, B. and M., 2002. "*TrackBack technical specification*", movabletype.org

[116] Tsoumakos, D. Roussopoulos, N., 2003. "*A Comparison of Peer-to-Peer Search Methods*", in International Workshop on the Web and Databases (WebDB).

[117] Tzitzikas, Y., Meghini, C., 2003. "*Query evaluation in peer-to-peer networks of taxonomy-based sources*", Proceedings of the 10th International Conference on Cooperative Information Systems, CoopIS'03. Sicily.

[118] Uschold, M., Gruninger, M., 1996. "*Ontologies: principles, methods, and applications*", Knowledge Engineering Review, 11(2), 93--155.

[119] Uschold, M., Jasper, R., 1999. "*A Framework for Understanding and Classifying Ontology Applications*", in Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm.

[120] Zhao, B. Y., Kubiatowicz, J. D., Joseph, A. D., 2000. "*Tapestry: An Infrastructure for Fault-tolerant Widearea Location and Routing*", U. C. Berkeley Technical Report UCB//CSD-01-1141.

Annexes

A. Vocabulaire Webop

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >

<rdfs:Class rdf:about="http://purl.org/webop/1.0/Resource"/>

<rdf:Property rdf:about="http://purl.org/webop/1.0/hasPeer">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Resource"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdfs:Class rdf:about="http://purl.org/webop/1.0/Topic">
  <rdfs:subClassOf rdf:resource="http://purl.org/webop/1.0/Resource"/>
</rdfs:Class>

<rdfs:Class rdf:about="http://purl.org/webop/1.0/Post">
  <rdfs:subClassOf rdf:resource="http://purl.org/webop/1.0/Resource"/>
</rdfs:Class>

<rdf:Property rdf:about="http://purl.org/webop/1.0/subTopicOf">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Topic"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/hasTopic">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Post"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Topic"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/includesTopic">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Topic"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/includedBy">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Topic"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/recommendedFor">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Topic"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/deliveredFor">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
```

```
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Topic"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/subscribesTo">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Topic"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/personalisationOf">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Post"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Post"/>
</rdf:Property>

<rdf:Property
rdf:about="http://purl.org/webop/1.0/hasPersonalisation">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Post"/>
  <rdfs:range rdf:resource="http://purl.org/webop/1.0/Post"/>
</rdf:Property>

<rdf:Property
rdf:about="http://purl.org/webop/1.0/subjectIndicatorRef">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range
  rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource"/>
</rdf:Property>
<rdf:Property rdf:about="http://purl.org/webop/1.0/resourceRef">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Post"/>
  <rdfs:range
  rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Resource"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/accessibleBy">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/inaccessibleBy">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property
  rdf:about="http://purl.org/webop/1.0/referencingAllowed">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Topic"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

< !-- Vocabulary of message -->

<rdfs:Class rdf:about="http://purl.org/webop/1.0/Message" />

<rdf:Property rdf:about="http://purl.org/webop/1.0/from">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Message"/>
  <rdfs:range
```

```
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/to">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Message"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/feedbackTo">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Message"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/text">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Message"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/body">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Message"/>
  <rdfs:range
     rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag"/>
</rdf:Property>

<rdfs:Class rdf:about="http://purl.org/webop/1.0/Event" />

<rdf:Property rdf:about="http://purl.org/webop/1.0/createdStmt">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Event"/>
  <rdfs:range
 rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/deletedStmt">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Event"/>
  <rdfs:range
 rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
</rdf:Property>

<rdf:Property rdf:about="http://purl.org/webop/1.0/stmt">
  <rdfs:domain rdf:resource="http://purl.org/webop/1.0/Event"/>
  <rdfs:range
 rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
</rdf:Property>

</rdf:RDF>
```

B. Service Webop

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="WebopService"
        targetNamespace="http://purl.org/webop/1.0/"
            xmlns="http://schemas.xmlsoap.org/wsdl/"
            xmlns:apachesoap="http://xml.apache.org/xml-soap"
            xmlns:tns="http://purl.org/webop/1.0/"
            xmlns:typens="http://purl.org/webop/1.0/"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
            xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <types>
       <xsd:schema targetNamespace="http://purl.org/webop/1.0/">
          <xsd:complexType name="Certificate">
              <xsd:sequence>
                  <xsd:element name="uid" type="xsd:string"/>
                  <xsd:element name="pkey" type="xsd:hexBinary"/>
                  <xsd:element name="signature" type="xsd:hexBinary"/>
              </xsd:sequence>
          </xsd:complexType>
          <xsd:complexType name="QueryResult">
              <xsd:sequence>
                  <xsd:element name="code" type="xsd:int"/>
                  <xsd:element name="data" type="xsd:string"/>
              </xsd:sequence>
           </xsd:complexType>
          <xsd:complexType name="QueryParam">
              <xsd:sequence>
                  <xsd:element name="uri" type="xsd:string"/>
                  <xsd:element name="scope" type="xsd:string"/>
                  <xsd:element name="filter" type="xsd:string"/>
                  <xsd:element name="projection" minOccurs="0"
                       maxOccurs="unbounded" type="xsd:string"/>
              </xsd:sequence>
            </xsd:complexType>
       </xsd:schema>
    </types>
    <message name="QueryInput">
        <part name="uid" type="xsd:string"/>
        <part name="param" type="typens:QueryParam"/>
        <part name="sign" type="typens:Certificate"/>
    </message>
    <message name="QueryOutput">
        <part name="body" type="typens:QueryResult"/>
    </message>
    <message name="MessageInput">
        <part name="uid" type="xsd:string"/>
        <part name="msg" type="xsd:string"/>
```

```
        <part name="sign" type="typens:Certificate"/>
    </message>
    <message name="MessageOutput">
        <part name="body" type="xsd:int"/>
    </message>

    <portType name="WebopServer">
        <operation name="queryPKB">
           <input message="tns:QueryInput" name="QueryInput"/>
           <output message="tns:QueryOutput" name="QueryOutput"/>
        </operation>
        <operation name="postMessage">
           <input message="tns:MessageInput" name="MessageInput"/>
           <output message="tns:MessageOutput" name="MessageOutput"/>
        </operation>
    </portType>

    <binding name="WebopServerSoapBinding" type="tns:WebopServer">
        <wsdlsoap:binding style="rpc"
                  transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="queryPKB">
           <wsdlsoap:operation soapAction=""/>
           <input name="QueryInput">
               <wsdlsoap:body use="encoded"
                   namespace=" http://purl.org/webop/1.0/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
           </input>
           <output name="QueryOutput">
               <wsdlsoap:body use="encoded"
                   namespace="http://purl.org/webop/1.0/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
           </output>
        </wsdl:operation>
        <operation name="postMessage">
           <wsdlsoap:operation soapAction=""/>
           <input name="MessageInput">
                <wsdlsoap:body use="encoded"
                   namespace="http://purl.org/webop/1.0/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
           </input>
           <output name="MessageOutput">
               <wsdlsoap:body use="encoded"
                   namespace="http://purl.org/webop/1.0/"
           encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
           </output>
        </operation>
    </binding>

  <!-- service decln -->
  <service name="WebopServerService">
    <port name="Webop" binding="tns:WebopServerSoapBinding">
       <wsdlsoap:address
           location="http://localhost:8080/axis/WebopServer.jws"/>
    </port>
  </service>
</wsdl:definitions>
```

C. Benchmark

La Figure C.1 illustre la configuration du benchmark que nous avons effectué. Pour chaque niveau entré, le générateur de requêtes donne un ensemble de requêtes à évaluer. Chaque requête **Q1**(*p*, *l*) générée par ce générateur correspond à une recherche de posts sous un topique avec l'étiquette *l* dans le pair *p*. Le nombre de requêtes évaluées est donc égal au nombre total de topiques dans les hiérarchies du jeu de données. L'évaluateur évalue chaque requête donnée dans trois variantes du algorithme d'évaluation et enregistre les compteurs d'évaluation dans la table de résultat.

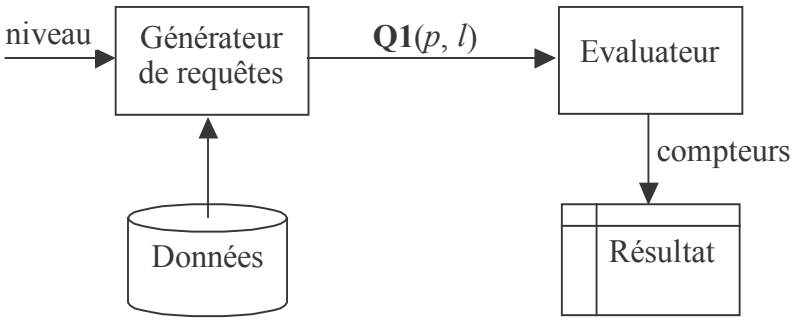


Figure C.1 : Configuration du benchmark

Topic (peer, turi, label, title, created)
Link (peer, label, targetpeer, targetlabel)
Post (peer, puri, label, title, created, resourceRef)

Trace (peer, label, id)
Result (peer, label, cnum1, pnum1, cnum2, pnum2, cnum3, pnum3)

Figure C.2 : Schéma de données du benchmark

La Figure C.2 donne le schéma du benchmark. Tout le jeu de données est stocké dans trois tables Topic, Link, Post. Elles se conforment au schéma relationnel proposé pour l'implémentation dans chaque pair sauf que l'attribut peer est ajouté dans chaque table. Cette modification permet de mettre tous les pairs dans une seule base. La table Trace est utilisée pour sauvegarder en mémoire des requêtes évaluées dans chaque pair. Elle est réservée pour l'implémentation de la variante 3 (avec toute optimisation) de l'algorithme d'évaluation. Le résultat d'évaluation des requêtes est sauvegardé dans la

table Result. Pour chaque requête **Q1**(*p*, *l*), nous comptons le nombre d'appels récursifs effectués (cnum1, cnum2, cnum3), et le nombre de posts retournés comme résultat (pnum1, pnum2, pnum3).

Nous donnons maintenant le pseudo-code de l'évaluateur. Nous assumons que #cnum1, #cnum2, #cnum3 sont trois variables globales dans l'évaluateur. Ils sont utilisé pour compter le nombre d'appels effectués dans l'exécution de chaque requête. gen_id() est une fonction qui génère un identifiant unique pour chaque requête évaluée.

```
// le pseudo-code de l'évaluateur
Procedure Evaluate(p, l)
begin
   #cnum1 := 0;
   r1 := Plan_1(p, l);
   #cnum2 := 0;
   r2 := Plan_2(p, l);
   #cnum3 := 0;
   q := gen_id();
   r3 := Plan_3(p, l, q);
   insert into Result values(p, l, #cnum1, r1.size, #cnum2, r2.size, #cnum3, r3.size);
end;

// variante 1 du plan d'exécution : sans optimisation
Function Plan_1(p, l) return table of puri
begin
   #cnum1 := #cnum1 + 1;
   Q := select puri from Post
         where peer = p and label >= l and label < l || 'xFF';
   N := select distinct targetpeer from Link
         where peer = p and label >= l and label < l || 'xFF';
   for each tp in N do
      L := select distinct targetlabel from Link
            where peer = p and label >= l and label < l || 'xFF' and targetpeer = tp;
      for each tl in L do Q := Q + Plan_1(tp, tl);
   end for;
   return Q;
end;
```

```
// variante 2 du plan d'exécution : avec optimisation Lmin
Function Plan_2(p, l) return table of puri
begin
   #cnum2 := #cnum2 + 1;
   Q := select puri from Post
         where peer = p and label >= l and label < l || 'xFF';
   N := select distinct targetpeer from Link
         where peer = p and label >= l and label < l || 'xFF';
   for each tp in N
      L := select distinct targetlabel from Link
           where peer = p and label >= l and label < l || 'xFF' and targetpeer = tp;
      Lmin := Min(L);
      for each tl in Lmin do Q := Q + Plan_2(tp, tl);
   end for;
   return Q;
end;

// variante 3 du plan d'exécution : avec optimisation Lmin et Trace
Function Plan_3(p, l, q) return table of puri
begin
   num := select count(*) from Trace
            where peer = p and id = q and l >= label and l < label || 'xFF';
   if num = 0 then
      insert into Trace values(p, l, s);
      #cnum3 := #cnum3 + 1;
      Q := select puri from Post
            where peer = p and label >= l and label < l || 'xFF';
      N := select distinct targetpeer from Link
            where peer = p and label >= l and label < l || 'xFF';
      for each tp in N do
          L := select distinct targetlabel from Link
                where peer = p and label >= l and label < l || 'xFF' and targetpeer = tp;
           Lmin := Min(L);
           for each tl in Lmin do Q := Q + Plan_3(tp, tl, s);
      end for;
   end if;
   return Q;
end;
```