EGC

SOS-DLWD

Des Sources Ouvertes au Web de Données

Atelier EGC 2012

Organisateurs :

- Stephan Brunessaux (Cassidian IPCC - EADS)
- Khaled Khelif (Cassidian IPCC - EADS)
- Nathalie Pernelle (LRI - Université Paris Sud, CNRS)
- Fatiha Saïs (LRI - Université Paris Sud, CNRS)
- Arnaud Saval (Cassidian IPCC - EADS)
- François Scharffe (LIRMM – Université de Montpellier 2, CNRS)

CASSIDIAN
AN EADS COMPANY

UNIVERSITÉ PARIS SUD
Comprendre le monde, construire l'avenir®

cnrs
dépasser les frontières

um2
UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

Atelier SOS-DLWD'12 : des Sources Ouvertes au Web de Données

Présentation

Cet atelier a pour objectif de réunir les travaux traitant d'une part, des problématiques liées aux sources ouvertes hétérogènes et indépendantes, et d'autres part, des problématiques concernant les liens sémantiques pouvant exister entre les données structurées afin de faciliter leur exploitation et leur intégration via le Web de données. Il est ainsi le résultat de la fusion de la troisième édition de l'atelier SOS (Sources Ouvertes et Services, RFIA'2010 et EGC'2011) et la première édition de l'atelier DLWD (Données Liées pour un Web de Données).

Le thème Sources Ouvertes et Services veut mettre en exergue les multiples problèmes liés au traitement de données disponibles en sources ouvertes (SO). Les SO désignent l'ensemble média accessibles librement, gratuits ou payants, tels qu'Internet, les bases de données publiques, les journaux, les chaînes de télévision et de radio, etc. par opposition aux sources fermées dont la consultation nécessite de disposer d'autorisations spécifiques. Ces SO fournissent d'importants volumes de données multimédia hétérogènes (images, texte, audio, vidéo, etc.) qui nécessitent des traitements adaptés afin de permettre leur exploitation. En plus des problématiques posées par l'hétérogénéité des données disponibles, l'enchaînement des traitements algorithmiques capables d'exploiter ces données représente un défi scientifique et technique. L'intérêt est porté sur toutes les étapes, partant de la phase de découverte des sources d'informations, en passant par la collecte et l'analyse des données collectées jusqu'à la phase de capitalisation et d'exploitation. L'intérêt est également porté sur les choix architecturaux retenus pour la réalisation d'applications exploitant les SO. En effet, ces applications tentent généralement de concilier plusieurs briques logicielles (COTS, logiciels open source, développements ad hoc, etc.) afin de les faire cohabiter en vue de la réalisation d'une tâche particulière. L'accent est mis sur les architectures orientées service (SOA) et sur l'utilisation des technologies du Web sémantique.

Dans le thème *Données Liées pour un Web de Données* nous avons souhaité aborder les problématiques liées à la publication des données structurées et à leur exploitation via le Web de données. Depuis les quatre dernières années, le nombre de sources de données structurées rendues disponibles sur le Web est en croissance fulgurante aboutissant à un espace global de données de l'ordre de milliards d'assertions (31 milliards en septembre 2011). Dans cet espace de données, des liens sémantiques peuvent être établis entre les données. Ces liens permettent aux robots d'exploration, aux navigateurs ou aux applications de naviguer parmi les sources de données et de combiner les informations provenant de sources différentes. Ces données liées sont nombreuses, distribuées, hétérogènes et peuvent être imprécises ou périmées. Aussi, différentes approches peuvent être définies en fonction des caractérisques des données et du domaine d'application concerné. Dans ce domaine, plusieurs initiatives sont menées au niveau national (comme le projet DataLift (http://datalift.org)) et au niveau international (comme les projets LOD2 (http://lod2.eu) et Planet Data (http://planet-data.eu)) afin d'amorcer et de regrouper les efforts pour résoudre les problèmes engendrés par la masse de données liées disponibles.

Les six articles articles présentés dans le cadre de cet atelier de la conférence EGC 2012 présentent des approches traitant des problème liés aux sources ouvertes et aux services permettant d'exploiter ces sources. Certaines de ces approches s'intéressent plus spécifiquement aux données liées disponble sur le Web. Le premier article présente deux approches de découverte de services web: (i) la première est fondée sur le traitement du langage naturel en analysant les informations pertinentes contenues dans les descriptions WSDL et sur l'annotation de services et (ii) la seconde est fondée sur l'analyse de la réputation obtenues à partir des différentes notations des utilisateurs pour décrire des recommandations. Dans le second article les auteurs s'intéressent à la sélection de services de recherche d'information adaptés au profil de l'utilisateur. Ce profil modélise les centres d'intérêt de l'utilisateur, selon ses connaissances ou selon ses besoins pour une session de recherche donnée. Le troisième article décrit l'implémentation réalisée, tant au niveau services de traitement qu'au niveau interface homme machine, d'un système d'indexation et de recherche de fiches d'anomalie du CEDIMAT utilisant la plateforme d'intégration WebLab. Le quatrième article propose un algorithme incrémental permettant de générer et de mettre à jour une représentation compacte d'une source RDF et qui pourra être utilisé pour identifier des sources pouvant contribuer à la résolution d'une requête distribuée. Dans le cinquième article, les auteurs proposent une approche d'extraction d'évènements d'intérêt, définis dans une ontologie de domaine, dans des sources textuelles

qui permet de combiner les résultats obtenus par différents outils d'extraction. Enfin, dans le sixième article les auteurs définissent un méta-modèle permettant à un utilisateur de créer et organiser ses informations personnelles, et d'obtenir des informations variant avec le contexte d'interrogation de ces informations personnelles.

Enfin, nous tenons à remercier les membres du comité de programme pour leur implication dans le processus d'évaluation des articles et pour la très bonne qualité des évaluations qui ont certainement aidé les auteurs à améliorer leur travaux.

Thèmes

- Identification, et découverte automatique de sources d'informations Accès et collecte d'informations à partir de sources ouvertes (Web, réseaux sociaux, flux RSS, etc.)
- Classification, filtrage d'informations d'intérêt, extraction d'informations à partir de textes non structurés et/ou utilisant des vocabulaires spécifiques (blogs, langage sms, forums, etc.)
- Extraction d'informations à partir de gros volumes de données multi-médias (texte, image, vidéo, audio)
- Modélisation et capitalisation des connaissances extraites à partir des sources ouvertes (ontologies, annotations sémantiques, etc.)
- Exploitation des connaissances extraites à partir de sources ouvertes : raisonnement, aide à la décision, visualisation, etc.
- Détection de signaux faibles
- Données publiques et gouvernementales
- Evaluation et qualification des sources d'informations
- Provenance et confiance des données et de leurs liens
- Evaluation et qualification des informations extraites à partir de sources ouvertes
- Inférence, fouille et validation de liens entre données.
- Intéropérabilité des sources de données et alignement d'ontologies
- Génération et publication des données
- Interrogation du contenu du LOD
- Développement de services pour les données liées
- Privacy / contrôle d'accès aux données liées
- Plateformes d'intégration de services de traitement hétérogènes : interopérabilité des services, orchestration sémantique, etc.
- Applications de veille stratégique ou économique à partir de sources ouvertes
- Application de renseignements d'origine sources ouvertes (ROSO)

Comité de Programme

- Yamine Ait Ameur (LISI/ENSMA, Université de Poitiers)
- Bernd Amann (LIP6–Université Paris 6)
- Florence Amardeilh (Mondeca)
- Nacéra Bennacer (E3S -Supelec)
- Patrice Buche (INRA SupAgro Montpellier)
- Gaël de Chalendar (CEA/LIST)
- Olivier Corby (INRIA Sophia)
- Madalina Croitoru (LIRMM–Université Montpellier 2)
- Mariana Damova (OntoText – Bulgarie)
- Jerôme David (LIG – UPMF)
- Valentina Dragos (Onera)
- Adil El Ghali (IBM)
- Jerôme Euzenat (INRIA - Rhone-Alpes)
- Christian Fluhr (Geol. Semantics)
- Patrick Giroux (Cassidian)
- Bruno Grilheres (Cassidian)
- Ollivier Haemmerlé (IRIT - Université Toulouse le Mirail)
- Nathalie Hernandez (IRIT - Université Toulouse le Mirail)
- Nicolas Hernandez (Université de Nantes)
- Michel Leclère (LIRMM–Université Montpellier 2)
- Amar-Djalil Mezaour (Exalead)
- Alexandre Pauchet (LITIS, Rouen)
- François Paulus (SemSoft)
- Yves Raimond (BBC)
- Chantal Reynaud (LRI – INRIA Saclay)
- Marie-Christine Rousset (LIG–Université de Grenoble)
- Mouhamadou Thiam (Université Gaston Berger – Sénégal)
- Raphael Troncy (EURECOM – Sophia Antipolis)
- Haïfa Zargayouna (LIPN, Université de Paris 13)
- Juliette Dibie-Barthélémy (INRA - AgroParis'Tech)
- Brigitte Safar (LRI - INRIA Saclay)

Liste des articles acceptés

- Découverte et recommandation de services web basées sur l'extraction d'information et la réputation symbolique
  Mustapha Aznag, Mohamed Quafafou, Nicolas Durand and Zahi Jarir

- Sélection adaptative de Services de Recherche d'Information web en fonction du besoin de l'utilisateur
  Aurélien Saint Requier, Gérard Dupont, Sébastien Adam, Yves Lecourtier and Stephan Brunessaux

- Intégration de services Web traitant les fiches d'anomalies du Simat dans la plateforme WebLab
  Olivier Bartheye, Stéphane Bellec, Gérard Dupont and Jetsadabodin Pintong

- Codage DFSR: Extraction de motifs de graphe pour une représentation compacte du contenu de sources RDF
  Adrien Basse, Fabien Gandon, Isabelle Mirbel and Moussa Lo

- Vers un système de capitalisation des connaissances : extraction d'événements par combinaison de plusieurs approches.
  Laurie Serrano, Maroua Bouzid, Thierry Charnois and Bruno Grilhères

- Modelling and Quering Context-Aware Personal Information Spaces
  Rania Khefifi, Pascal Poizat and Fatiha Saïs

Découverte et recommandation de services web basées sur l'extraction d'information et la réputation symbolique

Mustapha AZNAG*, Mohamed QUAFAFOU*, Nicolas DURAND*, Zahi JARIR**

*Laboratoire des Sciences de l'Information et des Systèmes (LSIS UMR 6168)
Université Aix-Marseille - Avenue Escadrille Normandie Niemen - 13397 Marseille
{mustapha.aznag, nicolas.durand, mohamed.quafafou}@univmed.fr
** Laboratoire d'Ingénierie des Systèmes Informatiques (LISI)
Université Cadi Ayyad - Boulevard Prince My Abdellah, B.P. 2390 | 40000 Marrakech
zahijarir@ucam.ac.ma

Résumé. Cet article montre que le problème de représentation de services web est crucial et analyse les différents facteurs qui l'influencent. Il discute une représentation classique et en propose deux nouvelles. La première représentation que nous proposons provient du domaine du traitement du langage naturel et est basée sur des règles pour annoter les descriptions de services et ainsi extraire les informations utiles pour l'indexation sémantique de services. La seconde méthode proposée, appelée réputation symbolique, est calculée à partir des relations entre les services considérés et est utilisée pour la recommandation de services web. L'impact de ces représentations pour la découverte et la recommandation est étudié et discuté à la lumière de nos expérimentations utilisant des services web réels.

1 Introduction

Un service web [1](W3C, 2004) est défini comme un programme modulaire mis à disposition sur le web par un fournisseur de services et destiné à supporter l'interaction machine-machine ou application-application dans des environnements distribués. Il est basé sur l'architecture orientée service (SOA). Notons que, dans le cas des services web, le mécanisme de communication est indépendant de tout langage de programmation et de toute plateforme d'exécution. Il existe plusieurs types de services web : (1) Les services web de type **REST**(Representational state transfer) et (2) et les services web de type **WS-***. Les services REST sont basés sur l'architecture du web et ses standards de base HTTP et URI. Les spécifications des services de type WS-* reposent sur les standards W3C SOAP (Simple Object Access Protocol) et WSDL (Web Services Description Language). Dans cet article nous avons étudié les services web de type WS-*. L'accès à ces services est réalisé par l'intermédiaire de messages SOAP. Ils utilisent une syntaxe basée sur la notation XML pour décrire les appels de fonctions distantes et les données échangées, et organisent les mécanismes d'appel et de réponse. Comme exemple de service web, nous pouvons évoquer les services météorologiques permettant à l'utilisateur

1. http ://www.w3.org/standards/webofservices/

de connaître la météo selon sa localisation. Les services web sont publiés dans des annuaires comme par exemple les annuaires UDDI (Universal Description Discovery and Integration).

Le nombre de serveurs offrant des services web, grâce à des moteurs spécifiques comme Axis [2], est en constante progression. Il est alors important d'automatiser l'indexation de ces services au niveau des annuaires. Pour cela, les documents WSDL décrivant les services peuvent être collectés et analysés automatiquement afin de générer des représentations structurées (vectorielles) de ces services. Les documents WSDL contiennent des descriptions textuelles, des opérations (au sens fonctions appelables à distance), des types de données simples ou complexes, etc. Les descriptions utilisées sont en langage naturel, multilingues et inter-domaines. Générer des représentations de services web est donc un défi majeur. De plus, la qualité des représentations a une forte incidence sur les annuaires qui offrent des mécanismes de découverte, de composition et de recommandation de services web. Ces différentes tâches ont été intensément étudiées ces dernières années (Maximilien et Singh, 2004; Dong et al., 2004; Xu et al., 2007; Yan et Piao, 2008; Petrova-Antonova et Ilieva, 2009; Omer et Schill, 2009; Elgazzar et al., 2010). Cependant, plusieurs problèmes restent posés : l'information décrivant les services est très hétérogène et ne respecte pas forcément les règles de la langue, de nombreux services souvent des services sans descriptions et enfin très peu de travaux ont porté sur l'impact des représentations utilisées pour la découverte et la recommandation.

Cet article présente une méthode classique de génération de représentations de services web, et en propose deux nouvelles. La première méthode, appelée *REI* (Reconnaissance et Extraction d'Information), se base sur des techniques de traitement du langage naturel et utilise des règles pour annoter les descriptions des services et ainsi extraire les informations pertinentes. La seconde méthode proposée, appelée *RS* (Réputation Symbolique), calcule les représentations à partir des relations entre les différents services web considérés. Nous présentons les algorithmes adéquats de découverte et de recommandation, et nous les utilisons pour confronter les différentes représentations.

La section 2 détaille le traitement appliqué aux documents WSDL. La section 3 introduit les différentes représentations de services web et explique comment chacune affine l'autre ou offre une nouvelle alternative. La section 4 est dédiée aux algorithmes adéquats de découverte et de recommandation de services web. Les expérimentations et les résultats sont présentés dans la section 5 avant de conclure en section 6.

2 Traitement des documents WSDL

Le langage standard WSDL se base sur XML et sert à décrire les services web. Le document WSDL d'un service web permet d'indiquer, grâce à des balises (*binding*, *message*, *operation*, *service*, etc.), le protocole de communication utilisé, le format de messages requis pour communiquer avec ce service, les opérations que le client peut invoquer et la localisation de ce service. Les versions 1.1 et 2.0 de WSDL [3] sont prises en compte dans nos travaux.

Après avoir vérifié la disponibilité d'un service web et validé le contenu de son document WSDL, ce dernier est analysé afin d'extraire toutes les informations potentiellement utiles. Ces dernières sont (a) le nom, la documentation et la version du document, (b) les types simples

2. http ://axis.apache.org/axis/
3. v1.1 http ://www.w3.org/TR/wsdl ; v2.0 http ://www.w3.org/TR/wsdl20/

ou complexes utilisés par les messages pour transmettre des informations, (c) le nom et la documentation du service, et (d) l'interface définissant les opérations disponibles. Pour chacune de ses opérations, le nom, la documentation, les arguments et les valeurs de retour sont récupérés. Remarquons que les "documentations" sont des descriptions textuelles correspondant à des commentaires.

Les informations extraites seront utilisées lors de la construction des représentations. Avant de présenter les méthodes permettant le calcul des représentations, nous évoquons quelques traitements textuels classiques utilisés par la suite. Ces traitements concernent : (1) **Suppression des balises :** les documentations peuvent être écrites en HTML. Nous supprimons donc toutes les balises HTML, les composantes CSS, etc. (2) **Séparation des mots :** certains termes sont composés de plusieurs mots séparés par une lettre majuscule, nous utilisons alors des expressions régulières pour extraire ces mots. Par exemple, l'application de *[A-Z][a-z]+* sur la chaîne *"GetAllCountryCurrenciesResponse"* produit *'Get'*, *'All'*, *'Country'*, *'Currencies'* et *'Response'*. (3) **Suppression des mots vides :** les mots vides de sens (articles, pronoms, etc.) sont supprimés. (4) **Lemmatisation :** l'algorithme de Porter (Porter, 1980) est utilisé pour calculer le *lemme* de chaque mot. Etant donné que les mots ayant le même lemme ont généralement le même sens, seuls les lemmes sont conservés. Par exemple, les mots '*computing*' et '*compute*' produisent le même lemme '*comput*'.

3 Différentes représentations de services web

Dans cette section, nous présentons une méthode classique de construction de représentations de services web et en proposons deux nouvelles. Notons que les représentations générées sont des représentations vectorielles.

3.1 Utilisation des Documentations et des Types WSDL (DT) :

La première représentation est centrée sur les descriptions textuelles. Pour un service web, toutes les documentations présentes dans le document WSDL (documentation générale, noms et documentations des opérations) sont concaténées et nous appliquons des traitements textuels classiques utilisés par la communauté *recherche d'information* et résumés dans la section 2 pour ensuite construire une description générale d'un service web donné, que nous appelons *descriptionGenerale*.

Notons que le principal inconvénient est que la plupart des services web ont des documentations de mauvaise qualité ne respecte pas les règles de syntaxe ou de grammaire ou n'en ont pas du tout. Pour faire face à ces limites nous rajoutons les types WSDL pour ensuite construire la représentation vectorielle pour un service web donné. Les types sont utilisés par les messages pour transmettre des informations entre services. Par conséquent, les types sont de pertinentes informations pour décrire la fonctionnalité d'un service web. Pour un service donné, tous les noms de types WSDL (types simples/complexes, etc.) sont alors ajoutés à son ensemble de descriptions *descriptionGenerale* formé précédemment. La représentation est construite donc après application des traitements 2, 3 et 4 de la section 2 sur les nouvelles données utilisées.

Cette approche est classique et est basée sur des traitements syntaxiques pour construire la représentation d'un service. Elle est bien connue par la communauté des services web. Nous introduisons à présent deux nouvelles méthodes de calcul de représentations notées *REI* et *RS*.

3.2 Reconnaissance et extraction d'information (REI) :

De nombreux services web ont des documentations trop détaillées. En particulier quand ils offrent beaucoup d'opérations et chacune avec sa propre description. Dans ce contexte, il est donc important de savoir reconnaître les parties ou les entités importantes dans ces textes afin de construire de meilleures représentations.

Notre approche consiste, grâce à des règles, à identifier, annoter et extraire les termes (simples ou multiples) pertinents à partir des descriptions textuelles disponibles pour un service web. Notons que nous avons utilisé auparavant l'approche dans le domaine de la génétique. Le lecteur intéressé peut se reporter à l'article (Charnois et al., 2006) pour plus de détails. Dans le cadre des descriptions de services web, nous avons écrit un ensemble de règles en Prolog traduisant la grammaire à clauses définies (DCG) mise en place. Ces règles sont déclaratives et rendent le traitement léger et rapide. Elles sont spécifiques au domaine mais elles ne sont pas spécifiques au corpus de services web utilisé. L'approche est endogène c'est-à-dire qu'aucune ressource telle qu'un dictionnaire n'est nécessaire. L'implantation a été réalisée avec la plateforme LinguaStream [4] (Bilhaut et Widlöcher, 2006). Cette plateforme est un environnement d'expérimentation intégré destiné aux chercheurs en traitement du langage naturel.

Nous considérons la description générale *descriptionGenerale* formée précédemment dans la section précédente 3.1 et nous appliquons sur cette description les étapes de traitement du processus de la méthode *REI*. Les étapes de traitement que nous appliquons ("tokenization", "Part-Of-Speech tagging" i.e. étiquetage des parties du discours, extraction avec les règles définies, etc.) ont été mises en oeuvre sous forme de modules. Notons que nous avons utilisé TreeTagger [5] pour l'étape "Part-Of-Speech tagging". Les règles fonctionnent de la façon suivante. A partir d'un *contexte*, une expression (en général un terme de plusieurs mots, une phrase nominale, etc.) est reconnue jusqu'à ce qu'une "phrase d'arrêt" soit rencontrée. Le contexte est un ensemble de mots déclencheurs. Les phrases d'arrêt peuvent être des mots, des symboles, des verbes, la ponctuation, etc.

Enfin, Nous rajoutons tous les noms de types WSDL à l'ensemble des termes construits par le processus précédent. La représentation vectorielle d'un service web donné est construite donc après application des traitements 2, 3 et 4 de la section 2 sur les nouvelles données considérées.

3.3 Réputation symbolique (RS) :

Toutes les représentations précédentes sont produites directement à partir du contenu de documents WSDL qui représentent le point de vue du fournisseur. Notre objectif, à présent, est de construire une représentation à partir de son contexte (i.e. les services voisins). Cela est particulièrement intéressant dans les cas où les documentations d'un service web sont absentes.

Pour cela, nous nous sommes basés sur la notion de réputation d'un service web qui reflète une perception commune des autres services web ou des consommateurs à l'égard de ce service. En d'autres termes, elle agrège les évaluations du service rendu par les consommateurs de service. Typiquement, une réputation sera construite à partir d'un historique de notations par les différents acteurs. Plusieurs systèmes de réputation ont été proposés dans la littérature (Xu

4. http ://www.linguastream.org
5. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

et al., 2007; Maximilien et Singh, 2002; Majithia et al., 2004; Goldbeck et Hendler, 2004; Nepal et al., 2011). Cependant, un système de réputation qui se base sur le retour des utilisateurs peut utiliser des évaluations malhonnêtes. Des modèles de Markov cachés (Hidden Markov Models) peuvent être utilisés pour prédire la réputation d'un service web (Malik et al., 2009).

Notre travail introduit un nouveau modèle, la *réputation symbolique*, en tenant compte de l'aspect qualitatif de la réputation des services web. Elle est produite à partir de la représentation (qui est une description symbolique) des autres services web. Afin de calculer la réputation symbolique, nous modélisons les relations entre les services par un graphe de dépendance. Ce dernier forme un réseau de services qui est notamment utilisé pour traiter différentes problématiques comme la composition de services web (Hashemian et Mavaddat, 2005; Omer et Schill, 2009). L'approche générale de création de ce graphe consiste à rechercher les dépendances entre les opérations disponibles pour les services en utilisant les paramètres d'entrée et de sortie. Ensuite, le graphe de dépendance des services web, présenté dans la définition 1, est induit. Nous considérons ici qu'une dépendance se produit entre deux services lorsqu'ils offrent deux opérations dépendantes : un service nécessite (resp. offre) des données à partir (resp. à) d'un autre service. La définition 2 présente plus formellement cette notion. Bien entendu, d'autres définitions de dépendances peuvent être utilisées sans affecter notre approche générale de calcul de la réputation symbolique.

Définition 1 *Graphe de dépendance des services : Soit* $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ *l'espace de services contenant tous les services web. On définit le graphe de dépendance des services comme un graphe orienté* $G = (\mathcal{S}, \mathcal{V})$ *tel que :* $\mathcal{V} = \{(s_i, s_j) \in S \times S, \exists f_i \in s_i, \exists f_j \in s_j : f_i \rightarrow f_j\}$. *Où* $f \in s$ *signifie qu'un service* s *offre l'opération* f.

Définition 2 *Dépendances des opérations : Soient* $\mathcal{F} = \{f_1, f_2, ..., f_n\}$ *l'ensemble des opérations offertes par tous les services web,* $In(f_i), Out(f_i)$ *représentent respectivement les paramètres d'entrée et de sortie de l'opération* f_i, $\alpha \in [0, 1]$ *est un seuil donné et* sim *la fonction de similarité. L'opération* f_j *dépend de l'opération* f_i, *noté* $f_i \rightarrow f_j$, *si et seulement si* $\forall p \in In(f_j), \exists q \in Out(f_i) : sim(p, q) \geq \alpha$.
Où $sim(p, q) = 1 - NGD(p, q)$ *est le facteur de similarité entre deux termes* p *et* q *calculé en utilisant NDG (Normalized Google Distance) (Cilibrasi et Vitányi, 2007) comme mesure de distance entre deux termes.*

Notre modèle de réputation symbolique est basé sur les marches aléatoires (Lovász, 1993) dans le graphe de dépendances entre les services web. Une marche aléatoire est une séquence de noeuds dans le graphe construit par le procédé suivant : sélectionner un noeud de départ de manière aléatoire dans le graphe, puis passer à une sélection au hasard d'un voisin de ce noeud et ainsi de suite. Les analyses de la marche aléatoire ont été appliquées à différents domaines (Lovász, 1993). Plus particulièrement, un modèle formel a été proposé dans (Rafiei et Mendelzon, 2000) pour permettre le calcul de la réputation d'une page web en utilisant la structure de lien hypertexte. Nous avons adapté ce modèle pour calculer la réputation symbolique de services web. Nous considérons le graphe de dépendances construit précédemment. La probabilité de visiter un service s pour le terme t à l'étape n, notée $P^n(s, t)$ est définie dans l'équation 1.

$$P^n(s,t) = (1-d) \sum_{q \rightarrow s} \frac{P^{n-1}(q,t)}{O(q)} + \begin{cases} \frac{d}{N_t} & \text{Si } t \in RV(q) \\ 0 & \text{Sinon} \end{cases} \quad (1)$$

La probabilité précédente est la base de l'algorithme qui calcule la réputation symbolique.

Algorithme 1 Calcul de la réputation symbolique

ENTRÉES : **S** (l'ensemble des services web), **k** (nombre maximum d'itérations), **d** (saut du marcheur aléatoire), **s** (un service).
SORTIES : $RS(s)$ (Réputation symbolique du service s).

$RS(s) \leftarrow \emptyset$
$RV(s) \leftarrow getVectorialRepresentation(s)$
pour tout $t \in RV(s)$ **faire** P(s,t) = $\frac{d}{N_t}$ **fin pour**
pour $l = 1, .., k$ **faire**
  si $l < k$ **alors** $d^{'} = d$ **sinon** $d^{'} = 1$ **fin si**
  pour chaque chemin $q_l \rightarrow ... \rightarrow q_1 \rightarrow s$ de longueur l et chaque terme t in $RV(q_l)$ **faire**
    $P(s,t) = 0$ si le terme t n'a pas été vu avant
    $P(s,t) = P(s,t) + ((1-d)^l / \prod_{i=1}^{l} O(q_i)(d^{'}/N_t)$
  fin pour
fin pour
pour tout terme t avec $P(s,t) > \frac{1}{N_t}$ **faire**
  $RS(s) = RS(s) \cup t$
fin pour
retourner $RS(s)$

En effet, supposons qu'avec la probabilité d le marcheur aléatoire passe uniformément à un service choisi au hasard parmi l'ensemble des services qui contiennent le terme t. Dans ce contexte, la probabilité qu'un marcheur aléatoire visite un service avec un saut aléatoire est $\frac{d}{N_t}$ (N_t désigne le nombre total de services dont la représentation vectorielle $RV(q)$ contient le terme t), 0 sinon. En outre, la probabilité que le marcheur visite un service s à l'étape n, après en avoir visité un parmi ces parents q est $\frac{1-d}{O(q)} P^{n-1}(q,t)$ où $P^{n-1}(q,t)$ désigne la probabilité que le marcheur aléatoire visite un service q pour un terme t à l'étape $n-1$ et $O(q)$ désigne le nombre de services parents de q dans le graphe de dépendance. L'algorithme 1 donne des détails de calcul de la réputation symbolique pour un service donné. Dans cet algorithme, $RV(s)$ correspond à la représentation vectorielle du service web s qui a pu être calculée avec l'une des méthodes introduites précédemment.

4 Système de découverte et de recommandation de services web

L'algorithme de découverte que nous proposons (cf. algorithme 2) se base sur les trois fonctions suivantes : appariement (*matching*), classement (*ranking*) et sélection. Il utilise les exigences du client : les besoins fonctionnels (la description textuelle du service web souhaité), et le score de la réputation numérique. Comme nous l'avons évoqué dans la section 3.3, cette réputation quantitative se base sur la rétroaction des consommateurs de services. Le processus d'appariement se base sur la représentation vectorielle des services web qui a pu être calculée avec l'une des méthodes introduites précédemment *REI* et *DT*. Le système de découverte se résume comme suit. Etant donné une requête client, il recherche les services qui répondent aux besoins fonctionnels du client. Si plusieurs services répondant aux besoins fonctionnels

sont retournés et si le score de la réputation numérique est spécifié, le système sélectionne les meilleurs services web et les retourne au client.

Nous proposons également un algorithme pour la recommandation de services basée sur la réputation symbolique (cf. algorithme 3). L'idée est d'enrichir les résultats retournés lors de la découverte des services. Pour chaque service découvert, l'algorithme recherche l'ensemble des services web à recommander. Les résultats ne se réduisent alors pas aux services qui correspondent à la requête, mais contiennent également les services recommandés pour chaque service découvert. Les étapes de l'algorithme sont les suivantes. La réputation symbolique $RS(s)$ pour chaque service découvert s est récupérée. Ensuite, l'algorithme trouve tous les services q qui ont une représentation vectorielle $RV(q)$ similaire à $RS(s)$. Remarquons que α désigne un seuil donné. Enfin, les services trouvés sont classés en fonction de leur score de réputation numérique et la similarité entre $RS(s)$ et $RV(q)$. Seuls les meilleurs services en fonction du résultat de ce classement sont recommandés.

Algorithme 2 Découverte et séléction de services web

ENTRÉES : **funReq** : Description textuelle, **repReq** : score de la réputation, **nbMax** : Nombre maximum de services à retourner.
SORTIES : select : la liste de services découverts.
1: $funMatch \leftarrow$ Matching(funReq)
2: $sRank \leftarrow$ Ranking(funMatch,repReq)
3: $select \leftarrow$ Select(sRank,nbMax)
4: **retourner** $select$

Algorithme 3 Recommandation de services web basée sur la réputation symbolique

ENTRÉES : **dS** : services découverts.
SORTIES : **L** : liste des services recommandés.
1: $L = \emptyset$
2: $RS(s)$ = getSymbolicReputation(s)
3: **pour** $q \in dS$ **faire**
4: $RV(q)$ = getVectorialRepresentation(q)
5: **si** $(\|RS(s) \cap RV(q)\| \geq \alpha)$ **alors**
6: L = $L \cup q$
7: **fin si**
8: **fin pour**
9: L $\leftarrow$ Ranking(L)
10: L $\leftarrow$ SelectBest(L)
11: **retourner** L

5 Expérimentations

Nous avons collecté 993 services web réels sur *service-finder.eu* [6] qui a l'avantage de classer les services en utilisant une ontologie (Funk et Bontcheva, 2010). Pour chaque service collecté, nous disposons donc de sa catégorie. Les catégories considérées sont : "Weather",

6. http ://www.service-finder.eu/

"SMS", "Currency Exchange", et "Jobs". Remarquons que d'autres sources de services web, à savoir *WebservicesX.net*[7], *xMethods.net*[8] et *seekda.com*[9] ont été envisagées mais elles ne permettent pas la classification en catégories.

Afin d'évaluer l'intérêt des représentations introduites précédemment (cf. section 3), nous avons procédé de la façon suivante :

- Pour chaque catégorie c, générer une requête req_c de découverte,
- Pour chaque requête req_c et pour chaque représentation rep, calculer les services web découverts $Q_{c,rep}$ avec l'algorithme 2,
- Calculer la précision et le rappel pour chaque catégorie et chaque représentation,
- Calculer la précision globale et le rappel global pour chaque représentation.

La *précision* (cf. équation 2a) représente le pourcentage de bonnes réponses parmi les services web trouvés ($Q_{c,rep}$) par notre système de découverte pour une requête. W_c représente l'ensemble des services web de la catégorie c. Le *rappel* (cf. équation 2b) représente, pour une requête, le pourcentage de services web pertinents trouvés par notre système parmi la totalité des services web de cette catégorie.

$$(a)Precision_{c,rep} = \frac{\|Q_{c,rep} \bigcap W_c\|}{\|Q_{c,rep}\|} * 100; (b)Rappel_{c,rep} = \frac{\|Q_{c,rep} \bigcap W_c\|}{\|W_c\|} * 100 \qquad (2)$$

Pour ces expérimentations, le calcul de la réputation symbolique (cf. algorithme 1) d'un service a été réalisé avec l'union des représentations vectorielles *DT* et *REI* obtenues pour ce service.

La table 1 regroupe les résultats, en termes de précision (*P*) et de rappel (*R*), obtenus pour chaque représentation et pour chaque catégorie de services web. D'une façon globale, la représentation *REI* est la meilleure. Ensuite nous avons *DT* et en dernier *RS*. Remarquons que le rappel global est moyen quelque soit la représentation. La représentation *DT* a obtenu des résultats moyens pour la catégorie *"Weather"*. *REI* est la meilleure représentation pour la catégorie *"Weather"*. Pour les autres catégories, nous remarquons que seule *REI* a obtenu des valeurs moyennes. Les autres représentations se sont écroulées. Nous pouvons dire d'une façon générale que REI est plus robuste que les autres représentations, au travers ces différentes catégories. En ce qui concerne la représentation *RS*, elle a obtenu des résultats moyens pour *"Weather"*. Notons que cette catégorie a obtenue de bons résultats avec chaque représentation. Nous observons que *RS* n'est pas adaptée à la découverte de services web car elle a des résultats très faibles pour toutes les catégories (e.g. *"SMS"*, *"Currency"* et *"Jobs"*) sauf pour *"Weather"*. Cela s'explique par le fait que pour un service donné, *RS* ne représente pas la description du service lui-même, mais représente la description de la relation du service avec les autres. Si ces autres services ne reflètent pas suffisamment la fonction et le but du service, alors sa découverte est compromise.

Etant donné nos observations sur *RS*, nous avons procédé à des expérimentations sur la recommandation de services web en utilisant la représentation RS. La table 2 présente un exemple de recommandations pour un service résultat d'une requête effectuée avec la catégorie *"Weather"*. Nous avons évalué les services recommandés manuellement et les premiers résultats sont très encourageants. Les services recommandés appartiennent, en effet, à la même catégorie que le service initial.

7. http ://www.webservicex.net/ws/default.aspx
8. http ://www.xmethods.net/ve2/index.po
9. http ://www.seekda.com

	DT		REI		RS	
Catégories	P%	R%	P%	R%	P%	R%
Weather	68,97	57,14	81,82	51,43	68,97	57,14
SMS	48,48	80,00	55,56	50,00	12,12	80,00
Currency	16,67	16,67	58,33	77,78	8,51	22,22
Jobs	04,35	08,33	57,14	33,33	6,67	41.67
Moyenne	34,61	40,53	63,21	53,13	24,06	50,27

TAB. 1 – *Évaluation des différentes représentations pour la découverte.*

Service découvert : http ://www.deeptraining.com/webservices/weather.asmx ?wsdl
Services recommandés :(1) http ://ws.cdyne.com/WeatherWS/Weather.asmx ?wsdl
(2)http ://rightactionscript.com/webservices/nusoap/server.php ?wsdl
(3)http ://www.deeptraining.com/webservices/weather.asmx ?wsdl
(4)http ://ws.soatrader.com/weather.gov/0.1/ndfdXML ?wsdl
(5)http ://ws.serviceobjects.com/fw/FastWeather.asmx ?wsdl

TAB. 2 – *Exemple de recommandations de services de la catégorie "Weather".*

6 Conclusion

Dans cet article, nous avons proposé deux nouvelles méthodes de calcul de représentations des services web. La première est basée sur la reconnaissance et l'extraction d'information à partir de descriptions de service web pour ne garder que les parties les plus importantes. La seconde représentation, la réputation symbolique, est plus contextuelle car elle ne considère pas le service lui-même, mais les services en liaison avec lui. Des algorithmes de découverte et de recommandations de services web, se basant sur ces représentations, ont aussi été proposés.

Dans les expérimentations sur des services web réels, nous avons confronté nos représentations avec une méthode traditionnelle de représentation. Nous avons alors observé les points suivants. La représentation classique *DT* ne produit pas des résultats corrects. La représentation basée sur l'extraction d'information (*REI*) est, quant à elle, plus robuste au travers toutes les catégories. Enfin, la réputation symbolique (*RS*) n'est pas appropriée pour la découverte et parait plus adaptée à la recommandation de services web.

Dans le futur, nous approfondirons nos travaux sur la réputation symbolique pour la recommandation. Nous augmenterons le nombre de services web réels utilisés pour les expérimentations. Enfin, nous rendrons disponible en ligne le portail correspondant à l'implantation complète de notre système.

Références

Bilhaut, F. et A. Widlöcher (2006). LinguaStream : An Integrated Environment for Computational Linguistics Experimentation. In *11th Conf. ECACL*, pp. 95–98.

Charnois, T., N. Durand, et J. Kléma (2006). Automated Information Extraction from Gene Summaries. In *DTMIB'06*, Berlin, Germany, pp. 4–15.

Cilibrasi, R. L. et P. M. Vitányi (2007). The Google Similarity Distance. *IEEE TKDE 19*(3), 370–383.

Dong, X., A. Halevy, J. Madhavan, E. Nemes, et J. Zhang (2004). Similarity Search for Web Services. In *VLDB'04*, Toronto, Canada, pp. 372–383.

Elgazzar, K., A. E. Hassan, et P. Martin (2010). Clustering WSDL Documents to Bootstrap the Discovery of Web Services. In *ICWS*, Miami, Florida, USA, pp. 147–154.

Funk, A. et K. Bontcheva (2010). Ontology-Based Categorization of Web Services with Machine Learning. In *LREC*, Valletta, Malta.

Goldbeck, J. et J. Hendler (2004). Inferring Reputation on the Semantic Web. In *WWW*.

Hashemian, S. V. et F. Mavaddat (2005). A Graph-Based Approach to Web Services Composition. In *SAINT*, pp. 183–189.

Lovász, L. (1993). Random Walks on Graphs : A Survey. *Bolyai Society, Mathematical Studies 2*, 1–46.

Majithia, S., A. S. Ali, O. F. Rana, et D. W. Walker (2004). Reputation-Based Semantic Service Discovery. In *WETICE)*, Washington, DC, USA, pp. 297–302.

Malik, Z., I. Akbar, et A. Bouguettaya (2009). Web Services Reputation Assessment Using a Hidden Markov Model. In *ICSOC)*, Stockholm, Sweden, pp. 576–591.

Maximilien, E. M. et M. P. Singh (2002). Reputation and Endorsement for Web Services. *ACM SIGecom Exchanges 3*(1), 24–31.

Maximilien, E. M. et M. P. Singh (2004). Toward autonomic web services trust and selection. In *ICSOC*, pp. 212–221.

Nepal, S., Z. Malik, et A. Bouguettaya (2011). Reputation management for composite services in service-oriented systems. *Int. J. Web Service Res. 8*(2), 29–52.

Omer, A. M. et A. Schill (2009). Dependency Based Automatic Service Composition Using Directed Graph. In *NWESP)*, pp. 76–81.

Petrova-Antonova, D. et S. Ilieva (2009). Towards a Unifying View of QoS-Enhanced Web Service Description and Discovery Approaches. In *YR-SOC)*, pp. 99–113.

Porter, M. (1980). An Algorithm for Suffix Stripping. *Program 14*(3), 130–137.

Rafiei, D. et A. O. Mendelzon (2000). What do the Neighbours Think ? Computing Web Page Reputations. *IEEE DEB 23*(3), 9–16.

W3C (2004). Web services architecture. Technical report, W3C Working Group Note 11 February 2004. Retrieved April 30, 2006 from http ://www.w3.org/TR/ws-arch.

Xu, Z., P. Martin, W. Powley, et F. Zulkernine (2007). Reputation-Enhanced QoS-based Web Services Discovery. In *ICWS*, pp. 249–256.

Yan, J. et J. Piao (2008). Towards QoS-Based Web Services Discovery. In *ICSOC*, pp. 200–210.

Summary

This paper shows that the problem of representation of web services is crucial and analyzes the various factors that influences it. It presents the traditional representation of web services and proposes two new representations. The first one comes from the natural language processing domain and is based on rules to tag web service descriptions and to extract significant information. The second method, called symbolic reputation, is computed from the relationships between web services. The impact of the use of these representations on web service discovery and recommendation is studied and discussed in the experiments using real world web services.

Sélection adaptative de Services de Recherche d'Information web en fonction du besoin de l'utilisateur

Aurélien Saint Requier*,**, Gérard Dupont**, Sébastien Adam*
Yves Lecourtier*, Stephan Brunessaux**

*Université de Rouen, LITIS BP 12, 76801 Saint-Etienne-du-Rouvray, France
aurelien.saint-requier@etu.univ-rouen.fr,
http://www.litislab.eu/
**EADS Cassidian, Information Processing Control and Cognition, Val de Reuil, France
aurelien.saint-requier@cassidian.com
http://www.cassidian.com/

Résumé. Dans le cadre de travaux de recherche sur la modélisation du besoin et du comportement de l'utilisateur, nous décrivons une approche de sélection de Services de Recherche d'Information (SRI) web adaptés au besoin de l'utilisateur. Un système expérimental intégrant une modélisation de l'utilisateur par un profil représentant ses centres d'intérêt, une modélisation du comportement par un mécanisme de récupération des interactions utilisateurs et une base de SRI généralistes et verticaux, est présenté. Nos axes de recherche portent sur la construction d'un modèle de sélection à partir de caractéristiques issues de la littérature et du profil utilisateur. La technique envisagée pour apprendre notre modèle repose sur la mise en œuvre d'un apprentissage par renforcement utilisant la théorie des processus de décision markoviens.

1 Introduction

Depuis plusieurs années, le web est devenu la première source d'information pour une majorité de personnes d'après Rainie et al. (2007). Les Services de Recherche d'Information (SRI) web ont donc été développés pour simplifier l'accès à cette masse d'information et Purcell (2011) montre qu'ils sont généralement utilisés par les internautes pour combler un besoin d'information. Malgré cette simplification, Holscher et Strube (2000) constatent que la Recherche d'Information (RI) sur le web est une tâche qui peut se révéler compliquée, particulièrement pour les utilisateurs novices, qui ne possèdent aucune connaissance du fonctionnement du web et des SRI. La difficulté provient d'une part de la nature du Web : une masse de données importante, un aspect dynamique et une hétérogénéité des données. D'autre part, en face du web, nous retrouvons l'utilisateur qui est caractérisé par son savoir, ses besoins d'information et ses connaissances du SRI. L'utilisateur a un comportement spécifique face à un SRI web. Premièrement, l'utilisateur exprime son besoin en peu de mots-clés (d'après une étude de Jansen et al. (2000)). Ce comportement montre une adaptation de l'expression du besoin de l'utilisateur au fonctionnement des SRI web. En effet, pour une majorité des SRI web, l'utilisateur est invité à exprimer son besoin sous forme de mots-clés. Cette transformation du besoin

est une première étape compliquée pour l'utilisateur ce qui explique la difficulté des SRI à comprendre le besoin réel de l'utilisateur. Deuxièmement, une étude de Keane et al. (2008) révèle que les utilisateurs favorisent les résultats de recherche en haut de liste tandis que l'étude de Jansen et Spink (2006) montre qu'ils sont peu enclins à visualiser les résultats passée la première page. L'utilisateur peut donc être rapidement frustré par les résultats de recherche proposés par son SRI favori et abandonner sa recherche par méconnaissance de SRI différents et performants disponibles sur le web.

Le constat de la spécificité du web et du comportement de l'utilisateur sur les SRI web nous amène à proposer une stratégie de RI spécifique pour le Web qui repose sur deux problématiques : la compréhension du besoin et la mesure de satisfaction de l'utilisateur dans le but de le guider vers un SRI adapté. Dans une première section, nous décrivons les approches de sélection de SRI figurant dans la littérature. Dans une seconde section, nous présentons nos travaux sur la réalisation d'un système expérimental de sélection de SRI et deux cas d'utilisation du système. Enfin, nous concluons sur l'approche proposée et détaillons les axes de recherche envisagés pour la construction d'un modèle de sélection de SRI .

2 État de l'art

À l'ombre des trois géants de la recherche d'information générale sur le web (Google, Yahoo ! et Microsoft Bing), il existe bien sûr d'autres SRI généralistes, mais aussi des SRI dits spécialisés ou verticaux, souvent peu connus des internautes. Un SRI vertical permet d'effectuer des recherches sur une thématique précise. Il peut s'agir d'un secteur d'activité, d'une spécialité professionnelle, d'un sport, d'un individu, d'un sujet ou d'un concept. Les SRI verticaux peuvent aussi se spécialiser sur un type particulier d'information ou de document (information statitique, juridique ou universitaire, articles, livres, photos, vidéos, ...). Plusieurs technologies permettent la création d'un SRI spécialisé. Certaines se basent sur une catégorisation automatique de pages web, avec une sélection de contenu opérée par un processus entièrement automatique. Mais souvent, la sélection des sites web est manuelle et spécifique au thème traité par le SRI. Pour cette seconde technique, soit on applique à l'ensemble des pages web sélectionnées le processus classique d'un SRI, soit on utilise la fonctionnalité *ad hoc* d'un SRI généraliste en limitant les résultats de celui-ci aux pages des sites sélectionnés. Les outils les plus avancés intègrent des ressources multimédia et offrent des interfaces évolutives. Ce type de SRI présente des avantages non négligeables, comme un accès plus rapide à l'information désirée et des résulats plus pertinents pour les spécialistes d'un domaine selon l'étude d'Econsultancy [1] ou encore une interface spécifique adaptée au type de contenu recherché. En effet, les sources clefs de leur domaine d'intérêt sont indéxées et les recherches se font directement dans le bon contexte. Le problème pour l'utilisateur est donc de sélectionner le SRI vertical adapté à son besoin d'information.

Dans la littérature, les travaux sur la sélection de SRI portent sur (i) la compréhension du besoin de l'utilisateur et (ii) sur l'analyse du comportement de l'utilisateur. Les premiers travaux sur la sélection de SRIs de Li et al. (2008) et Arguello et al. (2009) cherchent à prédire la performance d'une requête sur différents SRI afin de sélectionner le SRI le plus pertinent à partir de caratéristiques issues de la requête de l'utilisateur. Dans une étude portant sur plu-

1. http ://econsultancy.com/uk/reports/vertical-search-b2b-report

sieurs millions d'utilisateurs, White et al. (2008) montrent que l'utilisation de plusieurs SRI au cours d'une session de recherche peut améliorer l'efficacité de la recherche et proposent un modèle appris à partir des caractéristiques liées à la requête de l'utilisateur, la page de résultats et la correspondance entre la requête et la page de résultats. Guo et al. (2010) et Song et al. (2011) introduisent, en plus des caractéristiques précédentes, des caractéristiques issues de données d'interactions utilisateurs à l'échelle de la requête. Guo et al. (2010) montrent que le modèle basé sur des caractéristiques extraites à partir des interactions de l'utilisateur obtient des performances sensiblement équivalentes au meilleur modèle qui combine tous les types de caractéristiques. Afin de mieux interpréter les interactions de l'utilisateur, des travaux traitent de l'analyse des interactions utilisateurs à l'échelle de la session de recherche pour anticiper un changement ou un abandon du SRI courant. White et Dumais (2009) identifient des motifs à partir d'un ensemble de séquences d'actions utilisateurs caractéristiques d'un comportement précédent un changement de SRI lors d'une session de recherche. Ils construisent un modèle prédictif par regression logistique à partir d'une analyse de logs utilisateurs pour anticiper le changement de SRI dans le but de garder l'utilisateur sur le SRI. L'étude met en lumière cinq caratéristiques discriminantes pour la prédiction du changement de SRI. Dans une étude sur la prédiction de la frustration de l'utilisateur lors d'une session de recherche, Feild et al. (2010) confirment la pertinence de ces caractéristiques. Un début de réponse pour identifier les raisons du changement de SRI est donné dans les travaux de Guo et al. (2011) qui introduisent des caractéristiques postérieures au changement de SRI afin d'aider le système à prendre une décision adaptée à la raison qui a causé le changement de SRI.

Les approches décrites proposent plusieurs perspectives : (i) développer des outils d'aide à la recherche pour le garder sur le SRI courant, (ii) diversifier les résultats de recherche pour essayer de mieux cibler le besoin de l'utilisateur et (iii) proposer à l'utilisateur un SRI plus adapté à son besoin. Nos travaux proposent une approche dans le but de répondre à cette dernière problématique. La section suivante présente donc un sytème expérimental ayant pour objectif d'aider l'utilisateur dans sa RI sur le web par une sélection d'un SRI web adapté au besoin et au comportement de l'utilisateur.

3 Le système expérimental

3.1 Description générale

Le système mis en place pour l'expérimentation est basé sur la plateforme open source WebLab [2] de développement d'applications dédiées au traitement de documents multimédias. Celle-ci repose sur une décomposition en couches et sur une architecture orientée services permettant la construction d'applications à partir de briques élémentaires respectant des interfaces standardisées. Le système est donc une "application" basée sur le socle d'intégration WebLab. Il met en œuvre des composants en Web Services pour le traitement et des portlets intégrées dans le portail permettent la composition de l'interface utilisateur.

Le système expérimental proposé a pour objectif d'aider l'utilisateur dans sa stratégie de recherche. La figure 1 illustre le fonctionnement du système. Le système peut donc se décomposer en trois blocs afin (i) de comprendre le besoin de l'utilisateur, (ii) de suivre le comportement de l'utilisateur pour mesurer sa satisfaction et (iii) de lui recommander un SRI adapté.

2. http ://weblab-project.org

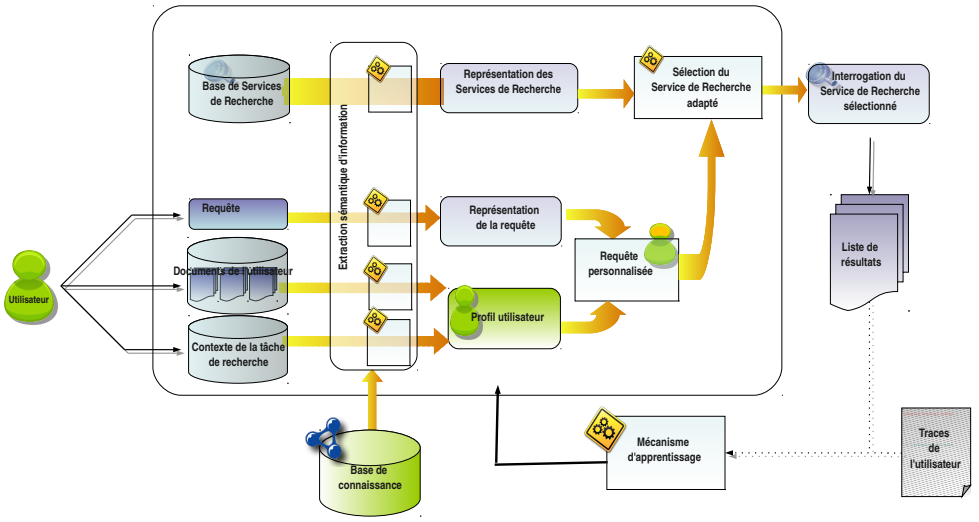


FIG. 1 – *Description générale du système expérimental.*

Afin de faciliter la compréhension du besoin de l'utilisateur par le système, nous utilisons un profil utilisateur pour modéliser ses centres d'intérêt. Pour la représentation et la construction du profil utilisateur, nous nous sommes appuyés sur les travaux de Daoud et al. (2009). En effet, nous distinguons le profil long-terme qui représente les connaissances de l'utilisateur et le profil court-terme qui correspond aux centres d'intérêt pour la tâche de recherche courante. Le profil est représenté sous la forme d'un vecteur de concepts pondérés issus de l'ontologie DBpedia [3]. Le profil long-terme est construit à partir de documents fournis par l'utilisateur qu'il juge représentatifs de ses centres d'intérêt. Le poids du concept correspond à sa fréquence d'apparition dans le corpus de documents fourni par l'utilisateur. Le profil court-terme est construit à partir de l'analyse des pages web visitées par l'utilisateur au cours la session de recherche.

Le profil utilisateur est utilisé par le système dans le but d'aider l'utilisateur à exprimer son besoin d'information. Le but est de transformer la requête mot-clés de l'utilisateur en une requête conceptuelle proche de ses centres d'intérêt. La transformation de la requête mots-clés en requête conceptuelle est semi-explicite. Une première liste de N concepts est récupérée en fonction des mots-clés de l'utilisateur. Ensuite, le système suggère à l'utilisateur une liste réduite de concepts correspondant aux concepts ayant une distance sémantique proche du profil utilisateur. La similarité sémantique entre un concept c et les concepts du profil utilisateur P se base sur l'approche de Milne et Witten (2008). Cependant, nous utilisons les catégories (correspondant au classement thématique des concepts) en commun des concepts pour déterminer si deux concepts sont proches sémantiquement. Le calcul de similarité sémantique est donné

3. http ://dbpedia.org/About

par la formule suivante :

$$Sim(c/P) = \sum_{i=0}^{N} W_{P(i)} * \frac{1}{|cat(c)||cat(P(i))|} \sum_{j=0}^{|cat(c)|} \sum_{k=0}^{|cat(P(i))|} Sim(cat_j(c), cat_k(P(i))$$

où :

$$Sim(cat_j(c), cat_k(P(i)) = \begin{cases} 1 & \text{si } cat_j(c) = cat_k(P(i)) \\ 0 & \text{sinon} \end{cases}$$

avec $W_{P(i)}$ le poids du concept $P(i)$ et $cat()$ un ensemble de catégories du concept correspondant. L'utilisateur sélectionne alors le concept traduisant le mieux son besoin d'information parmi la liste suggérée. La transformation de la requête mots-clés en une requête conceptuelle nous permet de mieux cerner le type de besoin de l'utilisateur en récupérant la propriété *rdf :type* du concept dans l'ontologie DBpedia pour le guider vers un SRI adapté.

Dans le but de recommander un SRI adapté au besoin de l'utilisateur, nous avons identifié 90 SRI web généralistes et verticaux (principalement à partir du site web Pandia [4]). Les SRI web généralistes identifiés sont les trois grands leaders de la RI sur le web, Google Search, Yahoo ! Search et Microsoft Bing. Les SRI web verticaux peuvent être divisés en deux sous-catégories : spécialisés sur un type de contenu (image, vidéo, blog, tweet, ...) ou traitant d'une thématique spécifique (juridique, économique, médicale, légal, gouvernemental, scientifique, droit de l'Homme, art de la table, ...). Un SRI web est décrit par les champs suivants : *id*, *title*, *url*, *text*, *cat*, *specialized*, *searchable* et *popularity*. Le champ *id* est un identifiant unique du service de recherche. Le champ *url* est l'url de la page principale du service de recherche avec le paramètre d'interrogation du moteur quand celui-ci est accessible. Le champ *searchable* indique si le service est interrogeable ou non via l'url indiquée. Les champs *title* et *text* correspondent respectivement aux propriétés *rdfs :label* et *dbpedia-owl :abstract* de l'ontologie DBpedia du SRI web correspondant. Le champ *cat* permet de décrire le type de besoin ou le domaine de spécialité traité par le SRI web. Plus précisément, le champ correspond aux valeurs de la propriété *rdf :type* du contenu des sources d'information traitées par le SRI. Un service de recherche peut être décrit par plusieurs champs *cat*. Le champ *specialized* permet de spécifier si le moteur de recherche est un moteur généraliste ou spécialisé. Enfin, le champ *popularity* attribue un indice de popularité au SRI compris entre 0 et 10. Une indexation de cette description basée sur le moteur de recherche libre Apache Solr [5] permet au système de recommander des SRI à l'utilisateur en fonction du type de besoin identifié dans sa requête.

Le système expérimental que nous avons présenté intègre des mécanismes pour comprendre le besoin utilisateur et recommander un SRI web adapté au besoin de l'utilisateur. La section suivante présente deux cas d'utilisation précis de notre système.

3.2 Cas d'utilisation

Pour une première validation de l'approche de notre système, nous présentons deux cas d'utilisation représentatifs des objectifs de nos travaux : aider l'utilisateur dans sa RI sur le web.

4. http ://www.pandia.com/
5. http ://lucene.apache.org/solr/

FIG. 2 – *Suggestion de requêtes pour le mot-clé « george » : (a) suggestions fournies par Google, (b) suggestions de notre système sans prise en compte du profil utilisateur, (c) suggestions de notre système avec un profil utilisateur orienté musique, (d) suggestions de notre système avec un profil utilisateur orienté histoire anglaise.*

Le premier cas d'utilisation montre l'apport du profil utilisateur pour la compréhension du besoin exprimé par l'utilisateur. La figure 2 illustre les suggestions de requêtes pour le mot-clé « george » provenant de différents modes d'outils d'aide à la formulation de requêtes. La figure 2(a) correspond aux mots-clés suggérés par Google dans un mode sans personnalisation. Les figures 2(b), 2(c) et 2(d) sont les suggestions de concepts DBpedia fournies par notre système avec différents profils utilisateurs. Le profil de l'utilisateur des suggestions illustrées par la figure 2(b) est vierge, les suggestions sont donc ordonnées par leur popularité donnée par DBpedia. Nous pouvons constater la personnalisation des suggestions proposées à l'utilisateur en fonction de son profil en comparant les suggestions des figures 2(c) et 2(d). Pour la même requête « george », les concepts « George Harrison », « George Michael », « George VI of the United Kingdom » et « George Martin » sont suggérés à l'utilisateur intéressé par la musique tandis que les concepts « George II of Great Britain », « George I of Great Britain », « George III of United Kingdom » et « George V of United Kingdom » sont proposés à l'utilisateur intéressé par l'histoire anglaise. Nous constatons donc l'intérêt d'un profil utilisateur pour aider l'utilisateur à formuler une requête représentant le plus fidèlement son besoin réel.

Le second cas d'utilisation montre la sélection d'un SRI web adapté au type de besoin exprimé dans la requête de l'utilisateur. Le scénario est le suivant : « Un utilisateur, expert dans l'optimisation pour les moteurs de recherche (SEO), doit s'informer des dernières actualités d'un algorithme d'ordonnancement des résultats de recherche nommé Panda. L'utilisateur ignore que l'algorithme en question est une mise à jour de l'algorithme de Google. ». La figure 3 montre les cinq premiers résultats retournés par Google pour la requête « news panda ». Nous constatons qu'aucun des résultats ne traitent de la problématique attendue, mais que les résultats portent essentiellement sur la thématique de l'animal.

La figure 4 illustre les résultats renvoyés par notre système. Dans un premier temps, le système identifie le type de besoin dans la requête de l'utilisateur et propose une liste de SRI web adapté à ce type de besoin. Pour notre exemple, le système propose trois SRI web spécialisés dans l'actualité. Ensuite, la requête de l'utilisateur est reformulée en fonction du profil de l'utilisateur (sur la gauche de la figure 4). Enfin l'utilisateur interroge un des SRI web de la liste

FIG. 3 – *Les cinq premiers résultats retournés par Google pour la requête « news panda », le 12 décembre 2011.*

proposée avec la requête personnalisée. Nous constatons que tous les résultats de la première page sont des articles récents traitant du nouvel algorithme d'ordonnancement de Google.

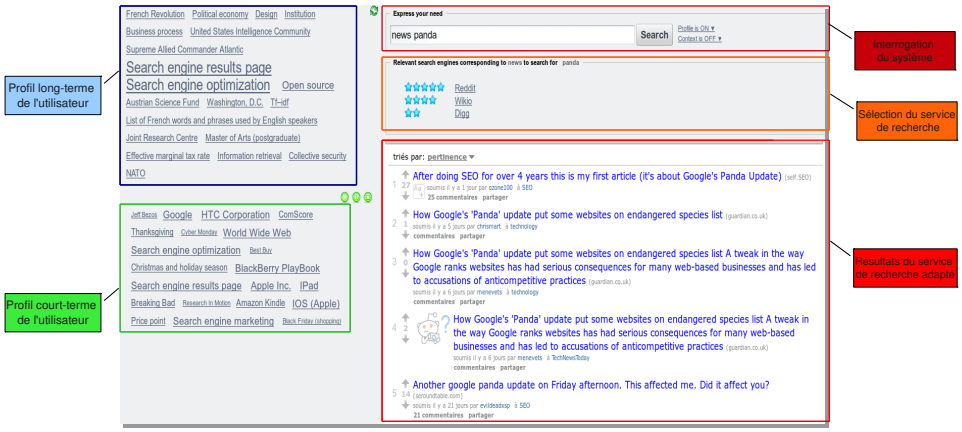


FIG. 4 – *La requête « news panda » traitée par le système proposé, le 12 décembre 2011.*

Les premières expérimentations de notre système sur des cas d'utilisation précis montrent des résultats probants. Afin de valider l'approche globale du sytème proposé, nous planifions pour le mois de février 2012 des expérimentations utilisateurs avec un objectif de 50 utilisateurs réalisant des tâches de recherche interactives adaptées à la problématique de sélection de SRI web. A partir de cette expérimentation, nous pourrons évaluer la pertinence de l'exploitation d'un profil utilisateur dans le processus de recherche et de sélection d'un SRI web en fonction du besoin utilisateur. Enfin, nous envisageons d'apprendre un modèle de sélection de SRI web

à partir des données d'interaction pour prendre en compte le comportement de l'utilisateur dans notre modèle de sélection.

4 Conclusion et perspectives

La spécificité du web et du comportement de l'utilisateur sur les SRI web nous a amené à proposer un système dans le but d'aider l'utilisateur dans sa tâche de RI. Nos axes de recherche portent sur l'analyse du besoin et du comportement de l'utilisateur dans le but de sélectionner un SRI adapté. Dans la littérature, deux types d'approches proposent des axes de recherche pouvant être adaptées au problème de sélection de SRI web : la recommandation du SRI en fonction de la requête de l'utilisateur et la prédiction du changement de SRI pendant une session de recherche. Le premier type d'approche permet de sélectionner un SRI adapté à une requête précise, mais ne prend pas en compte des caractéristiques propres à l'utilisateur pour comprendre son besoin d'information. Le deuxième type d'approche propose l'utilisation de modèles prédictifs du comportement de l'utilisateur dans le but de prédire un changement de SRI. Ces recherches montrent que les modèles prédictifs construits par régression logistique permettent de prédire le changement ou un abandon du SRI courant, mais ne proposent pas une solution adaptée à l'utilisateur aux raisons ayant entrainé le changement ou l'abandon du SRI. Nous avons donc proposé et implémenté un système expérimental intégrant des composants de modélisation des centres d'intérêts de l'utilisateur, de compréhension de son besoin et de recommandation de SRI. Nous avons présenté deux exemples de cas d'utilisation montrant l'apport de notre système d'aide à la RI web. L'objectif à court terme est d'organiser des expérimentations orientées utilisateur afin de valider notre approche globale d'aide à la RI web. De plus, ces expérimentations nous permettrons de collecter des données d'interaction dans le but de construire un modèle de sélection dynamique d'un SRI adapté.

La suite de notre travail consiste à intégrer au système un mécanisme apprenant capable de modéliser le comportement de l'utilisateur pour la sélection d'un SRI adapté. L'état de l'art montre l'efficacité des caractéristiques définies par White et Dumais (2009) pour la prédiction de changement de SRI. Cependant, les caractéristiques postérieures à un changement de recherche de Guo et al. (2011) permettent de comprendre la raison du changement et donc d'adapter la sélection du SRI. Afin de prendre en compte les intérêts (long et court termes) de l'utilisateur dans la sélection du SRI, nous proposons de combiner ces caractéristiques discriminantes avec un ensemble de caractéristiques issues du profil utilisateur pour apprendre notre modèle. Nous avons constaté, dans la littérature, que les comportements précédents un changement de SRI sont identifiés par de l'extraction de motifs à partir d'un ensemble de séquences d'action. Par exemple, un motif récurrent est la répétition de soumission de requêtes suivie d'aucun clic sur la page de resultats de recherche. Les différentes approches de l'état de l'art utilisent une technique de régression logistique pour modéliser les types de comportements précédents un changement de SRI. Cependant, la notion de séquences nous laisse penser que le problème de sélection de SRI peut-être formalisé par la théorie des processus de décision markoviens (MDP). En effet, les travaux de Dupont et al. (2010) ont montré que le cadre de l'apprentissage par renforcement utilisant la théorie des MDP est particulièrement adapté à la sélection dynamique d'outils de support à la RI. Nous proposons donc de transposer ces travaux à notre problématique de sélection de SRI adaptée au besoin et comportement de l'utilisateur.

Références

Arguello, J., F. Diaz, J. Callan, et J.-F. Crespo (2009). Sources of evidence for vertical selection. In *Proceedings of the 32nd international ACM SIGIR conference*, SIGIR '09, New York, NY, USA, pp. 315–322. ACM.

Daoud, M., L.-T. Lechani, et M. Boughanem (2009). Towards a graph-based user profile modeling for a session-based personalized search. *Knowl. Inf. Syst. 21*(3), 365–398.

Dupont, G., A. Saint Requier, S. Adam, Y. Lecourtier, B. Grilhères, et S. Brunessaux (2010). A step toward an adaptive composition of query suggestion approaches. In *Proceedings of the third symposium on Information interaction in context*, IIiX '10, New York, NY, USA, pp. 271–276. ACM.

Feild, H. A., J. Allan, et R. Jones (2010). Predicting searcher frustration. In *Proceedings of the 33rd international ACM SIGIR conference*, SIGIR '10, New York, NY, USA, pp. 34–41. ACM.

Guo, Q., R. W. White, S. Dumais, J. Wang, et B. Anderson (2010). Predicting query performance using query, result, and user interaction features. In *9th International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO 2010)*.

Guo, Q., R. W. White, Y. Zhang, B. Anderson, et S. T. Dumais (2011). Why searchers switch : understanding and predicting engine switching rationales. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, SIGIR '11, New York, NY, USA, pp. 335–344. ACM.

Holscher, C. et G. Strube (2000). Web search behavior of internet experts and newbies. *Computer Networks 33*(1-6), 337 – 346.

Jansen, B. J. et A. Spink (2006). How are we searching the world wide web ? : a comparison of nine search engine transaction logs. *Inf. Process. Manage. 42*, 248–263.

Jansen, B. J., A. Spink, et T. Saracevic (2000). Real life, real users, and real needs : a study and analysis of user queries on the web. *Inf. Process. Manage. 36*, 207–227.

Keane, M. T., M. O'Brien, et B. Smyth (2008). Are people biased in their use of search engines ? *Commun. ACM 51*, 49–52.

Li, X., Y.-Y. Wang, et A. Acero (2008). Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference*, SIGIR '08, New York, NY, USA, pp. 339–346. ACM.

Milne, D. et I. H. Witten (2008). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of AAAI 2008*.

Purcell, K. (2011). Search and email still top the list of most popular online activities. Technical report, Pew Internet & American Life Project.

Rainie, L., L. Estabrook, et E. Witt (2007). Information searches that solve problems. Technical report, Pew Internet & American Life Project.

Song, Y., N. Nguyen, L.-w. He, S. Imig, et R. Rounthwaite (2011). Searchable web sites recommendation. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, New York, NY, USA, pp. 405–414. ACM.

White, R. W. et S. T. Dumais (2009). Characterizing and predicting search engine switching

behavior. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, New York, NY, USA, pp. 87–96. ACM.

White, R. W., M. Richardson, M. Bilenko, et A. P. Heath (2008). Enhancing web search by promoting multiple search engine use. In *Proceedings of the 31st annual international ACM SIGIR conference*, SIGIR '08, New York, NY, USA, pp. 43–50. ACM.

Summary

As part of research on modeling needs and user behavior, we describe an approach to select web Information Retrieval Services (IRS) adapted to the needs of the user. An experimental system integrating a model of the user by a user profile representing these interests, modeling the behavior by a mechanism for retrieving user interaction and a database of general and vertical SRI is presented. Our research focus on the construction of a model selection with features from the literature and the user profile. We propose the use of reinforcement learning using the theory of Markov decision process for learning our model.

Intégration de services Web traitant les fiches d'anomalies du SIMAT dans la plateforme WebLab

Olivier Bartheye*, Stéphane Bellec**
Gérard Dupont***, Jetsadabodin Pintong*

*Département d'informatique,
Écoles Militaires de Saint-Cyr Coëtquidan, F-56381 Guer Cedex
olivier.bartheye@st-cyr.terre-net.defense.gouv.fr,
jetsadabodin.pintong@st-cyr.terre-net.defense.gouv.fr
**Cedimat,
Ecoles Militaires de Bourges,
avenue Carnot, BP 709, F-18015 Bourges Cedex
stephane.bellec@cedimat.terre.defense.gouv.fr
***Cassidian,
IPCC, 1 Bd Jean Moulin, CS40001, MetaPole, F-78996 Elancourt Cedex
gerard.dupont@cassidian.com

Résumé. Nous décrivons dans le présent article l'intégration et la mise à disposition des services d'indexation et de structuration de fiches d'anomalies du Système Informatique du Matériel de l'Armée de Terre (SIMAT) au sein de la plateforme WebLab.

1 Introduction

Nous abordons dans le présent article un exemple de traitement et d'exploitation des collections de données militaires non-structurées au travers d'une plateforme orientée service. D'un point de vue pratique, il s'agit de pouvoir extraire les saillances de ces documents, que sont généralement les mots-clés et les expressions définies à partir de ces mots clés. Ces saillances sont considérées significatives d'un point de vue des règles «métier» qui constituent le support fonctionnel et sont séparées de la sorte du reste du document qui n'est pas exploité et peut-être considéré comme du bruit de fond. Pour extraire ces saillances, de multiples méthodes existent et parmi celles-ci, on peut distinguer deux grandes approches.

La première nécessite des mécanismes d'indexation performants et permet à des utilisateurs et/ou à des experts fonctionnels du domaine de rechercher eux-mêmes les documents jugés intéressants par le biais de ces mécanismes.

La deuxième approche consiste à construire un ensemble de lexiques de termes considérés pertinents d'un point de vue de ces règles «métier» et de répartir de manière aussi discriminante que possible la connaissance dans ces lexiques. On obtient de la sorte des catégories lexicales où chaque lexique peut être étiqueté par une balise et le contenue de chaque balise de même nom renferme l'entité lexicale correspondante. Grâce à ces lexiques, il est possible de mettre

en œuvre des combinaisons d'entité lexicales par ce que l'on appelle des grammaires locales qui sont une version assouplie des grammaires globales et qui possèdent la propriété de ne pas traiter le bruit de fond présent entre deux expressions significatives (voir Gross (1997)). À partir d'un document électronique au format libre, on obtient à l'issue du traitement, un document semi-structuré au format XML qui caractérise la sémantique de ce document.

Les deux approches sont complémentaires et s'intègrent naturellement dans les plate-formes Web destinées à supporter l'accès et le traitement des documents dits de *source ouverte*. Ceux-ci ont en effet la particularité de posséder un contenu globalement non structuré (en faveur de la solution d'indexation classique) et quelques informations structurées sous forme de métadonnées. Or il apparaît que les données militaires de fiches d'anomalies du SIMAT ont finalement une structure similaire. C'est dans ce cadre que nous présentons notre objectif : intégrer dans la plate-forme WebLab des services d'indexations de fiches d'anomalies et des applications de semi-structuration sémantique de fiches d'anomalies à partir notamment de l'utilisation de lexiques et de grammaires locales.

La plate-forme WebLab[1] permet notamment, pour un domaine d'application particulier, d'associer de manière flexible et cohérente des composants logiciels issus du monde libre ou développés spécifiquement et d'adapter de la sorte le système d'information obtenu en fonction des formats de données échangées. L'objet de cette étude a été (i) de tester son applicabilité dans le domaine des documents techniques du SIMAT et (ii) de valider l'intégration distante de services existants par l'intermédiaire de l'architecture orientée service et le mécanisme de standardisation des interfaces de services proposé par le WebLab.

2 Recherche des fiches d'anomalies du SIMAT

Notre projet a pour but de construire un moteur de recherche documentaire afin de faciliter la démarche de recherche des fiches anomalies du *Centre d'Études et de Développements Informatiques du Matériel de l'Armée de Terre* (Cedimat), à partir d'informations collectées dans la base de données MySQL *Appoline*. contenant les «fiches anomalies» émises par les utilisateurs du *Système d'Information du Matériel de l'Armée de Terre* (SIMAT). Pour cela, il est nécessaire de développer des composants qui s'intégreront à la plateforme WebLab développée par CASSIDIAN[2].

Les composants développés devront assurer :

– l'indexation de la base de données Appoline.
– la recherche et la visualisation de données dans cette base.
– l'accès au moteur de recherche depuis la machine à distance.

Tous ces composants devront être indépendants et s'intégrer au sein de la plateforme WebLab sous forme de services Web. De plus la plate-forme doit être en mesure d'héberger les applications dites de source ouverte notamment Unitex[3] (voir Diewvilai (2009)) qui permet d'analyser des documents en utilisant les grammaires locales. La visualisation et la recherche de données devront s'effectuer au moyen de différents portlets intégrés au portail Liferay [4] ; ainsi il est possible de consulter la base de connaissances à distance via un navigateur Web.

[1] http ://weblab.ow2.org
[2] Une division du groupe EADS.
[3] http ://igm.univ-mlv.fr/~unitex/
[4] http ://www.liferay.com

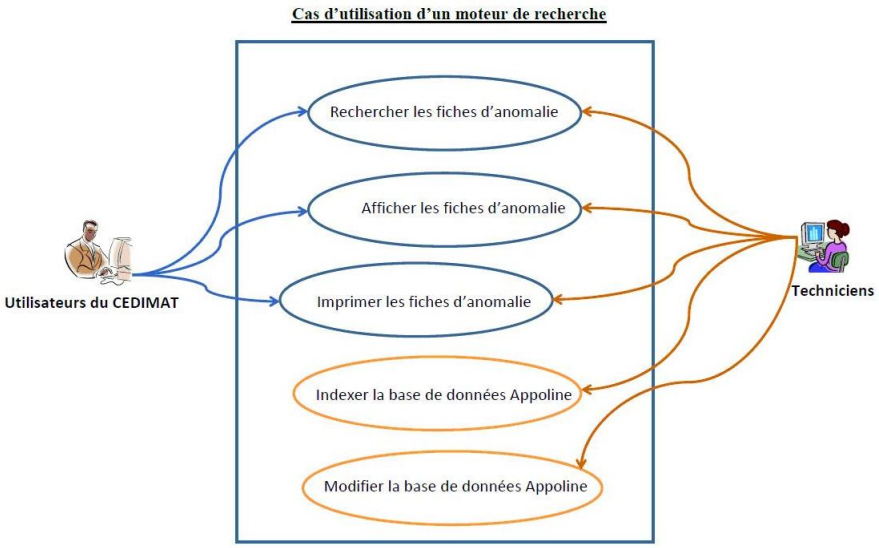


FIG. 1 – *Cas d'utilisation d'un outil de consultation de fiches d'anomalies.*

2.1 Structure d'une fiche d'anomalie

Le *Système Informatique du Matériel de l'Armée de Terre* (SIMAT) fait l'objet de déclaration de fiches d'anomalie de la part des utilisateurs, à raison d'une fiche Appoline par anomalie. Lorsqu'un utilisateur se retrouve face à un problème sur le SIMAT, il rédige un ticket d'incident, qui mène à la rédaction d'une fiche d'anomalie. Ainsi, la résolution d'un problème peut être suivie facilement depuis sa création jusqu'à sa résolution. Une fiche d'anomalie comporte de nombreuses informations où l'on retrouve la description du problème, les personnes ayant travaillé à le résoudre, ainsi que les solutions proposées.

Les trois principales informations portées par une fiche anomalie sont les suivantes :

– Le libellé du problème rencontré,
– La solution technique apportée au problème,
– La solution proposée à l'utilisateur pour le résoudre ou le contourner.

Toutes ces fiches sont archivées dans une base de données MySQL utilisée par l'application *Appoline* (actuellement la version de cette base est la version V5). Cette base de données constitue l'élément principal de la connaissance sur les actes de maintenance réalisés et peut être consultée par tous les intervenants de la chaîne des utilisateurs du SIMAT. Le travail effectué doit permettre de concevoir et de réaliser un outil de recherche afin d'automatiser cette tâche et de l'intégrer dans une plateforme WebLab afin de mieux exploiter le contenu des fiches *Appoline*, et plus particulièrement les libellés incidents, les solutions techniques et les solutions proposées par les utilisateurs.

Le rôle de l'outil *Appoline* est de permettre d'identifier, de conduire et d'archiver les anomalies survenant dans les différentes applications constituant le SIMAT. La persistance des données est assurée par le mapping objet/relationnel fourni par le «framework object-relational

Ano Ident :	80411	Ano Numéro :		Date d'Impression	10/06/2008
Gravité	Majeure	Statut :	INTEGRATION		
Date constatation	19-05-2008	Priorité :	Aucune		
Date de validation	19-05-2008	Depuis le :	09-06-2008		
Date de clôture	Inconnue	Type anomalie :	LOGICIEL TYPE SIMAT		
Personnel Demandeur	LTN A. P.		00.00.00.00.00		
Organisme Demandeur :	DCMAT				
Organisme maintenance	GQCV				
Logiciel	SIMAT	Fonction :	S32	Profil :	SC28
Version	08.14.03	Menu :	S_32401	Code ES :	042H196
Base de Donnée	scvsk00_scs01	Grille :	S_32401D		
Nom machine :	scvsk00	Type matériel	NEANT	Marque :	NEANT
N°Série :	NEANT				

Libellé incident

Lors de la modification du prix unitaire du lot d'une commande BDC dans la grille S_32401B, la mise à jour du prix unitaire et des montans correspondants (montant ht cadence, montant tt cadence) n'est pas faite jusqu'à la cadence.

Aprés vérification, il s'avere que la modification du prix va jusqu'à la cadence quand le numéro de lot papier du lot de commande n'est pas renseigné.
Quand ce numéro de lot papier est renseigné, la modification du prix ne va pas jusqu'à la cadence.

Il est necessaire de permettre la prise en compte de la modification du prix jusqu'à la cadence même que le numéro de lot papier est renseigné.

Libellé technique

09-06-2008 - TSEF T. - CEDIMAT - Transfert de la fiche anomalie
09-06-2008 - TSEF T. - CEDIMAT - Modification de la fiche anomalie
*** Cette fiche concerne la grille S_32401D ***

09-06-2008 - TSEF T. - CEDIMAT - Changement de statut
21-05-2008 - TSEF T. - CEDIMAT - Modification de la fiche anomalie
Concerne la grille S_32401D.
Dans le trigger LMO de F_LIGCOM 2 procedures (L_MAJPRIX et L_MAJCAD) interfèrent. En effet, aucune des deux procédures ne contiennent un store. Donc les modifications de L_MAJPRIX sont perdues lors de l'execution de L_MAJCAD (un clear est présent).

21-05-2008 - TSEF T. - CEDIMAT - Changement de statut
20-05-2008 - TSEF G. - CEDIMAT - Transfert de la fiche anomalie

Libellé utilisateur

09-06-2008 - TSEF T. - CEDIMAT - Transfert de la fiche anomalie
09-06-2008 - TSEF T. - CEDIMAT - Modification de la fiche anomalie
09-06-2008 - TSEF T. - CEDIMAT - Changement de statut
21-05-2008 - TSEF T. - CEDIMAT - Modification de la fiche anomalie
21-05-2008 - TSEF T. - CEDIMAT - Changement de statut
20-05-2008 - TSEF G. - CEDIMAT - Transfert de la fiche anomalie
19-05-2008 - LTN A. - DCMAT - Validation
Validé par le fonctionnel ACHAT.

FIG. 2 – *Exemple de fiche d'anomalie.*

mapping» (ORM) *Hibernate*[5] qui permet en outre de formuler des requêtes pour extraire les données stockées dans une base de données MySQL de plus de 120000 occurrences. Le moteur de recherche fourni par *Appoline* permet d'interroger cette base par des requêtes de type SQL. Chaque fiche anomalie précise des données statiques telles le numéro identifiant, l'utilisateur, la gravité, etc. Elle contient aussi le libellé incident et la solution technique de l'anomalie qui a permis de la résoudre. Le cycle de vie d'une fiche anomalie dans la chaîne de maintenance corrective est évaluée à chaque étape.

- *Libellé incident* : le texte saisi par l'utilisateur de SIMAT. Ce sont des commentaires libres.
- *Libellé technique* : le texte explique les méthodes et les solutions utilisées par les mainteneurs pour traiter le problème. Ce sont souvent des détails techniques.
- *Libellé utilisateur* : Ce champ sera envoyé à l'utilisateur quand une fiche anomalie est

[5] http ://www.hibernate.org/

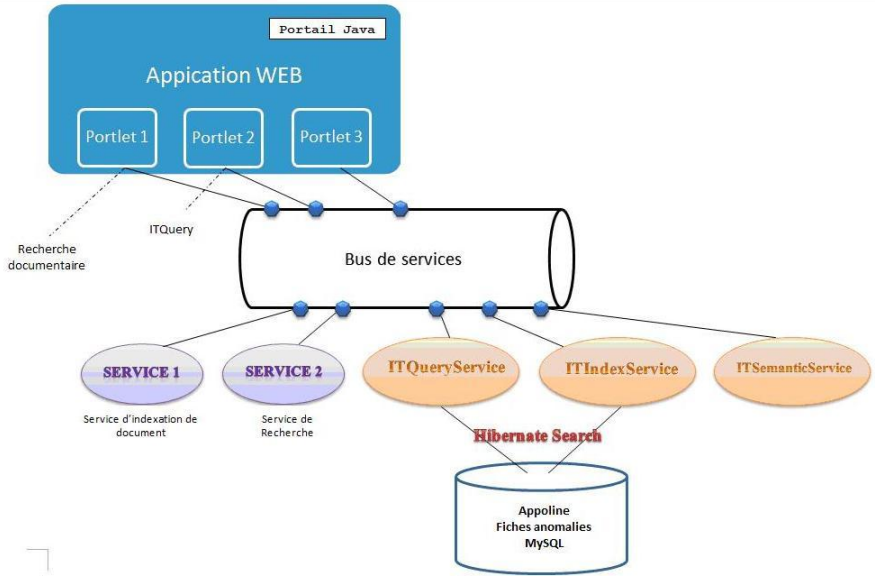


FIG. 3 – *Architecture des services.*

clôturée. Il contient une explication rapide et compréhensible de traitement, sans prendre en compte les techniques informatiques utilisées.

2.2 Une architecture de services Web traitant les fiches d'anomalies

En 2010, une architecture d'analyse et de diagnostic des fiches anomalies (figure 3) a été réalisée à partir de la technologie Hibernate Search[6] ; Hibernate Search est un moteur de recherche utilisant une capacité d'indexation de Lucene[7] et Hibernate. Trois services ont été développés :

1. *ITIndexService* qui permet de lancer une indexation,
2. *ITQueryService* qui permet de faire une demande de recherche avec ses options,
3. *ITSemanticService* qui permet de classifier le contenu des données fournies par *ITQueryService* à l'aide d'un analyseur sémantique développé en utilisant Unitex (voir Diewvilai (2009)) qui est une application de traitement de documents utilisant notamment des dictionnaires au format normalisé et des grammaires locales.

La solution pourra être réalisée avec les technologies de programmation Java, Web service et Portlet. À ce titre, la plateforme WebLab (voir section suivante ou encore Giroux et al. (2008); Canet et al. (2011)), développée par CASSIDIAN, a été sélectionnée. La fiche d'anomalie est accessible via un portlet de recherche de WebLab qui permet à un utilisateur ou un technicien d'accéder à distance à une fiche anomalie et qui place ces services de recherche

[6] http ://www.hibernate.org/subprojects/search.html
[7] http ://lucene.apache.org/java/docs/index.html

documentaire sur le réseau informatique. Le Cedimat disposera alors d'une plateforme indépendante du SIMAT permettant d'accéder à toutes les ressources utiles au suivi et à la résolution des problèmes en utilisant les technologies Web et Java actuelles.

3 Intégration des services dans la plate-forme WebLab

3.1 Présentation de la plate-forme WebLab

La plateforme WebLab a été développée par le département IPCC du System Design Centre d'EADS/Cassidian en collaboration avec des partenaires industriels et académiques. Après avoir été conçue pour répondre aux besoins de différents projets de recherche collaboratifs, elle s'est enrichie progressivement et a été industrialisée par EADS/Cassidian qui en assure depuis 2008 la diffusion en open-source par l'intermédiaire du consortium OW2 [8]. WebLab est fondée sur une architecture orientée service et sémantique et facilite l'intégration de composants commerciaux ou open-source qui fournissent des fonctionnalités telles que l'extraction d'informations dans des contenus non structurés, la classification, l'indexation, etc.

La plateforme WebLab peut être représentée selon un modèle logique incluant trois niveaux :

- le «*WebLab Core*» est le socle technique dit de *source ouverte* qui fournit un environnement dédié à l'interopérabilité des services intégrés.
- les «*WebLab Services*» constituent un ensemble cohérent de services logiciels et des composants d'IHM intégrés par l'intermédiaire du «*WebLab-Core*». Ces services sont implémentés avec des logiciels commerciaux, open-source ou de laboratoire.
- les «*WebLab Applications*» dépendent en général d'un domaine particulier et résultent de l'assemblage d'une sélection de «*WebLab Services*» permettant de répondre à un besoin «métier».

La présente étude exploite donc le coeur de WebLab, fournit de nouveaux services spécifiques métier liés au domaine de la documentation de SIMAT et propose l'implémentation d'un application dédiée.

3.2 Intégration des différents portlets composant l'application

Dans un premier temps, des portlets de recherche qui sont les clients du service Web ont été développés ; ces portlets sont déployés sous le portail Liferay (figure 4). L'objectif est de pouvoir transformer un service Web d'un moteur de recherche en service Web de la plateforme WebLab. Le client est un portlet déployé sous Liferay Portal connecté au service web via le bus de services Petals ESB[9].

1. Portlet *Search-portlet*
   - **Rôle** : effectue une recherche dans la base de données Appoline MySQL précédemment indexée à partir de mots-clés et envoie la saisie de la recherche au service web *ITImplSearcher* qui implémente l'interface Java *Searcher* ; récupère les résultats de la recherche et émet un événement dans le portail.

[8] http ://forge.ow2.org/projects/weblab
[9] http ://petals.ow2.org/

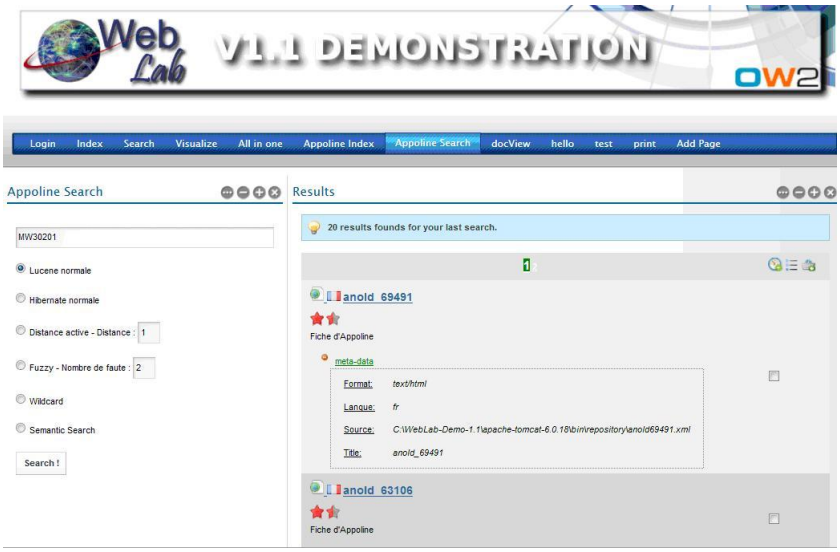


FIG. 4 – *Exemple de portlet sur le portail Liferay.*

- **Dépendances** : envoie un événement dans le portail pour *Result-portlet* avec la liste des résultats.
- **Service offert** : recherche des documents par mots-clés avec des options de recherche

2. Portlet *Result-portlet*
   - **Rôle** : affiche les résultats de la recherche effectuée dans *Search-portlet* ; l'utilisateur sélectionne ensuite le résultat souhaité pour l'afficher dans *Document-viewer-portlet*.
   - **Dépendances** : récupère l'événement émis dans le portail par *Search-portlet* avec la liste des résultats et envoie un événement dans le portail pour *Document-viewer-portlet* lorsque l'utilisateur choisit une fiche anomalie.
   - **Service offert** : affiche les résultats de la recherche en récupérant les documents de la recherche via le service web *Filerepository-Service* qui implémente l'interface Java *ResourceContainer* et sélectionne un résultat.
3. Portlet *Document-viewer-portlet*
   - **Rôle** : affiche la fiche anomalie que l'utilisateur a sélectionnée dans *Result-portlet* et récupère dans le portail l'événement émis par *Result-portlet* avec la fiche anomalie à afficher. Les documents affichés sont des documents normalisés au format WebLab. Le portlet met en évidence les instances repérées dans le texte en analysant les annotations.
   - **Service offert** : affiche le contenu d'une fiche anomalie.
4. Portlet *Index-portlet* (prépare la base indexée pour *Search-portlet*)
   - **Rôle** et **Service offert** : effectue une indexation de la base de données Appoline MySQL afin de permettre de faire des recherches par mots-clés et indexe la base de données Appoline MySQL.

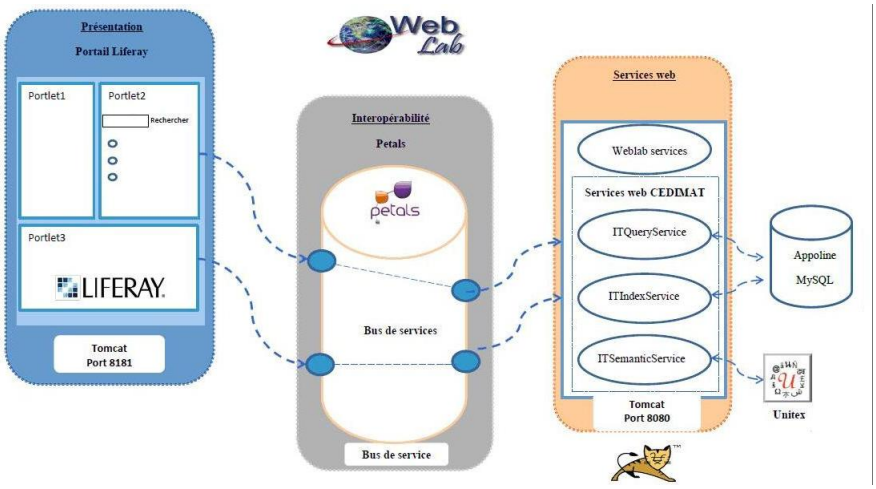


FIG. 5 – *Interopérabilité entre les différents composants du projet.*

3.3 Intégration des service Web dans la plate-forme WebLab

La procédure de fonctionnement d'un service web est la suivante :

1. le service Web définit un format pour les requêtes et les réponses ;
2. un ordinateur demandeur effectue une requête ;
3. le service Web effectue une action, et renvoie la réponse à l'ordinateur demandeur.

Le service Web devient un service de recherche qui permet d'effectuer la recherche dans une base de données *Appoline* MySQL. Le code Java est ajouté directement dans le code du service Web, c'est-à-dire dans la classe Java du projet.

1. Service *ITImplSearcher* appelé par *Search-portlet*
   - **Rôle** : gère les accès à la base de données *Appoline* en lecture et implémente l'interface *Searcher* de WebLab qui lui permet d'effectuer des requêtes dans la base ; peut recevoir des requêtes *Hibernate Search* qu'il exécutera avant de retourner des Resource WebLab.
   - **Service offert** : exécute une requête Hibernate Search ; créer et insére des Resource WebLab ; renvoye les résultats à *Search-portlet* pour l'affichage.
2. Service *ITIndexService* appelé par *Index-portlet*
   - **Rôle** : lance une indexation sur une base de données en utilisant la technologie Hibernate Search afin de permettre de faire des recherches par mots-clés.
   - **Service offert** : exécute une requête Hibernate Search et récupére les données à indexer ; indexe en utilisant Apache Lucene ; stocke les résultats d'indexation sous forme de fichiers index.
3. Service *ITSemanticService* appelé par *Search-portlet*

- **Rôle** : utilise la capacité d'Unitex pour traiter sémantiquement la liste de résultats d'*ITImplSearcher* puis renvoie la nouvelle liste classifiée,
  Service offert : récupère la liste de résultats d'*ITImplSearcher* ; effectue un traitement sémantique en utilisant Unitex ; renvoyer la liste de résultats annotés

4. Service *FileRepository* appelé par *Result-portlet* et *Document-viewer-portlet*.
   - **Rôle** : indique au *Result-portlet* un répertoire dans lequel il stocke des fichiers créés par le service *ITImplSearcher* et permet d'associer une URI unique à un fichier.

3.4 Intégration des services dans Petals ESB

Cette partie vise à fournir les informations nécessaires à l'intégration d'un service WebLab conformément à l'ESB (Enterprise Service Bus) et de mettre ensuite à la disposition des clients externes. Le bus de service est une solution d'intégration implémentant une architecture totalement distribuée, et fournit des services comme la transformation des données ou le routage ainsi qu'une interopérabilité accrue par l'utilisation systématique des standards comme XML et les services Web. Les données de configuration et d'administration sont alors distribuées sur les extrémités de l'ESB.

Le service sera hébergé sur un serveur d'application Tomcat et ensuite déployé sur un élément de liaison SOAP en utilisant l'unité de service (SU) et de l'assemblage de services (SA) pour le rendre disponible en tant que *endpoint ESB*. Ensuite, l'*endpoint* sera exposé par une autre unité de service afin de le rendre utilisable pour des clients externes. Dans notre cas, le client sera les portlets que nous avons développés. La figure 6 donne un aperçu complet de l'architecture technique de notre application.

4 Conclusion

L'objectif de ce projet de développement est de réaliser l'intégration des moteurs de recherche documentaire syntaxique et sémantique du *Centre de Développement Informatique du Matériel de l'Armée de Terre*, dans la plate-forme évolutive WebLab. Rappelons que WebLab fut développé par la société CASSIDIAN au travers de divers projets de recherche ANR (WebContent[10], e-Wok Hub[11]), européens (VITALAS[12], AXES[13], VIRTUOSO[14]), mais aussi au titre de Plans d'Etude Amont mandaté par la Délégation Générale pour L'Armement.

L'architecture en services distribués offerte par la plate-forme est particulièrement adaptée au travail d'intégration d'un outil de recherche d'incidents sur le système d'information du Cedimat. Sa flexibilité a permis d'obtenir un prototype fonctionnel qui s'appuie sur des services clés fournis en open source. Les technologies d'interopérabilité ouvertes fondées sur l'emploi de services Web et de Portlets ont facilité le travail d'intégration et présage à la fois d'un niveau de robustesse intéressant et de possibilités de passage à l'échelle. Le maintien en conditions opérationnelles d'une telle application semble donc possible de même que son évolution par l'intermédiaire de l'ajout de nouveaux services spécifiques.

[10] http ://www.webcontent.fr/
[11] http ://www-sop.inria.fr/edelweiss/projects/ewok/
[12] http ://vitalas.ercim.org/
[13] http ://www.axes-project.eu/
[14] http ://www.virtuoso.eu/

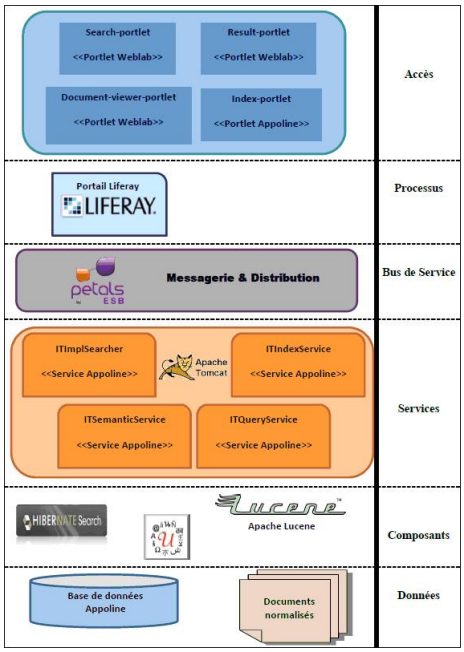


FIG. 6 – *Architecture technique de la plateforme WebLab.*

Références

Canet, G., G. de Chalendar, L. Dubost, G. Dupont, A. Dyevre, et K. Khelif (2011). Toward a versatile information toolkit for end-users oriented open-sources exploitation : Virtuoso. In *"Sources Ouvertes et Services" - SOS'2011, En association avec EGC'2011*.

Diewvilai, T. (2009). Classification automatique de fiches anomalies du système d'information simat. Technical report, CEDIMAT.

Giroux, P., S. Brunessaux, S. Brunessaux, J. Doucy, G. Dupont, B. Grilheres, Y. Mombrun, et A. Saval (2008). Weblab : An integration infrastructure to ease the development of multi-media processing applications. *ICSSEA*.

Gross, M. (1997). The construction of local grammars. In E. Roche et Y. Schab (Eds.), *"Finite-State Language Processing, Language, Speech, and Communication"*, pp. 329–354. MIT Press.

Summary

In this paper, we present the integration process and the activation of the indexation service, the search service and the structuration service devoted to anomaly reports inside the WebLab platform.

Codage DFSR: Extraction de motifs de graphe pour une représentation compacte du contenu de sources RDF

Adrien Basse*, Fabien Gandon*
Isabelle Mirbel**
Moussa Lo***

*Edelweiss, INRIA, Méditerranée, France
Adrien.Basse, Fabien.Gandon@inria.fr,
**I3S Labotary (CNRS, UNS)
Isabelle.Mirbel@unice.fr
***LANI, UFR SAT, Université Gaston Berger, Saint-Louis, Sénégal
moussa.lo@ugb.edu.sn

Résumé. Parmi les applications web sémantique, certaines manipulent des données issues de sources RDF distribuées. Pour identifier les sources qui contribuent à la résolution d'une requête distribuée, ces applications ont besoin de connaitre le contenu de chaque source. C'est dans ce cadre que nous proposons un algorithme incrémental qui génère et maintient une représentation compacte d'une source RDF. Cet algorithme améliore celui proposé par Basse et al. (2010) par la prise en compte des cycles, nœuds vides ou multi typés, par l'ajout d'un algorithme incrémental et par la génération d'un format RDF pour l'index. L'index généré est constitué d'une hiérarchie de graphes RDF interrogeables en SPARQL.

1 Introduction

Nombreuses sont les applications web sémantique qui s'intéressent à la problématique d'intégration de données issues de sources RDF [1] distribuées. Pour l'exécution de requêtes distribuées, par exemple, plusieurs solutions sont ainsi proposées (cf. Battle et Benson (2008) et Stuckenschmidt et al. (2004)) comme SPARQL 1.1 Federation [2] qui définit un ensemble d'extensions au langage de requête SPARQL [3]. Si ces extensions permettent de déléguer une partie de requête à un ensemble de services un problème majeur subsiste encore pour déterminer automatiquement en amont les services disposant de réponses pour une requête donnée. La résolution de ce problème est particulièrement cruciale dans un contexte comme le Linking Open Data où de nombreuses sources RDF hétérogènes sont publiées. Pour pouvoir envoyer une requête uniquement aux sources de données concernées, une description complète mais compacte (index) de chaque source est nécessaire. Pour connaitre le contenu d'une source RDF nous pouvons nous baser sur les motifs de graphes qui la composent. Dans cet article

1. Resource Description Framework, http ://www.w3.org/RDF
2. Federated SPARQL, http ://www.w3.org/2007/05/SPARQLfed/
3. SPARQL query language for RDF, http ://www.w3.org/TR/rdf-SPARQL-query

nous nous intéressons à l'extraction de ces motifs de graphes. La section 2 présente l'extension DFSR (Depth-First Search coding for RDF) qui adapte le code DFS (Depth-First Search) de Yan et Han (2002) pour représenter les motifs de graphes RDF. La section 3 détaille notre algorithme de construction de l'index. Les résultats et les expériences réalisées sont présentés à la section 4. La section 5 détaille notre algorithme incrémental de mise à jour de l'index. Un état de l'art sur la construction d'index et le graphe mining est proposé dans la section 6.

2 Concepts et principes de base du DFSR

Pour expliquer le codage DFSR, nous utiliserons l'exemple en figure 1 qui montre un graphe RDF contenant entre autres des cycles, des nœuds vides et des nœuds multi-typés.

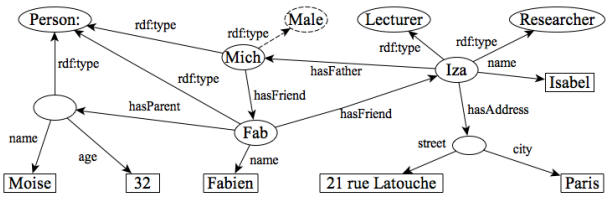


FIG. 1 – *Exemple de source RDF*

En utilisant les définitions de graphes de Baget et al. (2008) nous proposons ces définitions :

Définition 1. (Graphe RDF). Un graphe RDF G est un 4-tuple $G = (N_G, E_G, n_G, l_G)$ où

- N_G et E_G sont respectivement deux ensembles finis de nœuds et d'arcs. L_{GN} est un ensemble fini de labels de nœuds. T_G est un ensemble fini de types RDF de nœuds ou d'arcs identifiés par des URIs.
- $n_G : E_G \rightarrow (N_G \times N_G)$ associe à chaque arc le couple de ses nœuds sommets. Soit $e \in E_G, n_G(e) = (n_1, n_2)$, n_1 est appelé le sujet de e et n_2 est appelé l'objet de e. Le sujet est une URI ou un nœud vide et l'objet est une URI, un nœud vide ou un littéral.
- $l_G : N_G \rightarrow L_{GN}$ renvoie le label d'un nœud.
- $t_G : N_G \cup E_G \rightarrow T_G$ renvoie le type d'un nœud ou d'un arc.

Si le triplet (n_1, p, n_2) représente l'arc e alors $n_G(e) = (n_1, n_2)$ et $t_G(e) = p$.

Définition 2. (Taille d'un graphe). La taille du graphe $G = (N_G, E_G, n_G, l_G)$ est la cardinalité de E_G et représente le nombre d'arcs de G.

Définition 3. (Motif de graphes). Soit $G = (N_G, E_G, n_G, l_G)$ un graphe RDF. Un graphe RDF $P = (N_P, E_P, n_P, l_P)$ est le motif du graphe P si et seulement si

- $E_P = E_G, N_P = N_G, T_P = T_G$,
- $\forall e \in E_P, n_P(e) = n_G(e)$ et $t_P(e) = t_G(e)$,
- $\forall n \in N_G$ alors $l_P(n) = *$ et $t_P(n) = t_G(n)$ (* est le label d'un nœud anonyme).

Dit autrement un motif de graphes est un graphe dont les labels de nœuds sont remplacés par un caractère générique. Si P est le motif de G alors G est une instance de P.

Définition 4. (Type conjonctif). Quand un nœud est typé par n classes $c_1, \ldots, c_n$ nous définissons son type conjonctif comme l'intersection des classes $c_1, \ldots, c_n$ notée $c_1 \wedge c_2 \wedge \ldots \wedge c_n$.

Dans la suite de cet article nous utilisons une notation textuelle et une notation graphique pour

représenter les motifs de graphes. Dans un motif, un nœud a pour label son type concaténé au caractère générique *. La figure 2 montre un motif issu de la figure 1 et sa sérialisation linéaire.

FIG. 2 – *Notation graphique et sérialisation linéaire d'un motif de graphes*

Han et al. (2007), Yan et Han (2002, 2003) et Yan et al. (2004) utilisent le codage DFS pour des graphes non orientés et Maduko et al. (2008) pour des graphes orientés. En nous appuyant sur ces travaux nous proposons le codage DFSR pour les motifs de graphes RDF.

Définition 5. (Code DFSR). Soit $P = (N_P, E_P, n_P, l_P)$ un motif de graphes. $D = (id, k_D, C_D)$ est le code DFSR de P, avec

- id l'identifiant du code,
- k_D la concaténation des identifiants de codes DFSR joints pour obtenir D,
- C_D un ensemble de 6-tuples T.

A chaque arc $e \in E_P$ avec $n(e) = (n_i, n_j)$ correspond $T = (i, j, l_G(n_i), t_G(e), l_G(n_j), ke_T)$, avec i et j les temps de découverte des nœuds n_i et n_j suivant un parcours en profondeur. $l_G(n_i)$ et $l_G(n_j)$ sont respectivement les labels de n_i et de n_j. $t_G(e)$ est le type de e. ke_T est la concaténation des identifiants des codes DFSR H_i joints pour obtenir D avec $T \in C_{Hi}$. Dans la suite de l'article nous représentons les codes DFSR sans ke_T.

Pour construire les codes DFSR les types de propriété, sujet et objet de la source RDF sont d'abord triés suivant l'ordre lexicographique puis affectés à des identifiants entiers croissants. A partir de la source de la figure 1 nous obtenons les affectations suivantes : *age=1, city=2, ..., Lecturer∧Researcher=9, ..., Person=11.* L'identifiant zéro est affecté au type *Literal*. La figure 3 montre un motif de la figure 1 et son code DFSR minimum. Pour choisir le premier arc du code nous prenons le plus petit dans l'ordre lexicographique parmi les identifiants des propriétés comme dans Maduko et al. (2008). Si deux ou plusieurs arcs partagent l'identifiant de propriété minimum ils sont départagés par un tri lexicographique sur leurs identifiants de sujets et si besoin d'objets. Si le motif contient $n(n > 1)$ arcs minimum nous générons n codes DFSR avant de choisir le plus petit suivant l'ordre lexicographique. L'ajout de test entre les sujets et les objets permet de réduire les cas où nous devons générer plus d'un code DFSR. Sur la figure 3 le nœud $Lecturer \wedge Researcher$ a pour temps de découverte 1 car il est le sujet de l'arc ayant la plus petite propriété ($hasFather$). À partir de ce nœud, un parcours en profondeur et un ordre lexicographique sur les propriétés nous permet d'obtenir les autres temps de découverte.

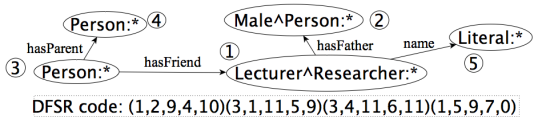


FIG. 3 – *Un motif de graphes et son code DFSR minimum*

3 Algorithme de génération de la structure d'index

Définition 6. (Noyau de motifs) Soient G et H deux motifs de taille $s > 1$ partageant $s - 1$ arcs. Nous appelons noyau de G et H le sous graphe de taille $s - 1$ qu'ils partagent.
Définition 7. (Jointure de motifs de graphes) La jointure $\bowtie$ de deux motifs de graphes $G = (N_G, E_G, n_G, l_G)$ et $H = (N_H, E_H, n_H, l_H)$ de taille $s > 1$ partageant un noyau de $s - 1$ arcs est le motif de graphes $J = (N_J, E_J, n_J, l_J)$ de taille $s + 1$ avec

- $E_J = E_G \cup E_H$ et $N_J = N_G \cup N_H$
- $\forall e \in E_J$,
  - si $e \in E_G \cap E_H$ alors $n_J(e) = n_G(e) = n_H(e)$,
  - si $e \in E_G \setminus E_H$ alors $n_J(e) = n_G(e)$,
  - si $e \in E_H \setminus E_G$ alors $n_J(e) = n_H(e)$.

Pour construire l'index, notre algorithme utilise ces deux définitions et respecte le principe suivant : Si un motif appartient à l'index alors tous ses sous graphes appartiennent à l'index. Une construction des différents niveaux de l'index est ainsi réalisée en trois grandes phases.
Phase 1 : Initialisation et énumération des motifs de taille 1. L'initialisation consiste à attribuer à chaque type de nœud et de propriété un identifiant entier suivant l'ordre lexicographique. Par exemple, les identifiants 1 et 11 sont respectivement attribués à *age* et *Person*. Pour construire le niveau 1 de l'index, nous exécutons une requête SPARQL pour retrouver l'ensemble des motifs de taille 1 de la source RDF. À partir de ces motifs et des identifiants de type précédemment créés, les codes DFSR de longueur 1 sont construits. D'abord, nous cherchons les identifiants correspondant au sujet, propriété et objet de chaque motif. Ces identifiants constituent les trois derniers éléments du 5-tuple représentant le code DFSR du motif. Ensuite les temps de découverte 1 et 2 sont attribués au sujet et à l'objet du motif. Ces temps de découverte constituent les deux premiers éléments du 5-tuple. Enfin un identifiant unique est attribué au code DFSR. Notons également que nous n'utilisons pas la notion de noyau pour la construction des niveaux 1 et 2.
Phase 2 : Construction des motifs de taille 2. Un code du niveau 2 est construit par jointure de deux codes de niveau 1 qui partagent un nœud. Nous distinguons trois cas de figure :
Cas 1 : Deux codes partagent le même sujet. Les temps de découverte du code de taille 1 minimum restent $(1, 2)$. Les temps de l'autre code sont $(1, 3)$. Après la construction du code nous vérifions si le motif correspondant dispose au moins d'une instance dans la source RDF. Seuls les motifs instanciés sont gardés dans l'index. Des informations complémentaires sont ajoutées à chaque code ainsi généré : (i) un noyau obtenu par concaténation des identifiants des deux codes de taille 1 joints ; (ii) un identifiant est associé au nouveau code ; (iii) les deux codes de taille 1 sont marqués comme étant inclus dans un code de taille supérieure. Il est ainsi possible à la fin de l'algorithme de proposer un affichage de tous les motifs ou seulement des motifs de couverture maximale. La figure 4 montre la construction d'un code de taille 2.

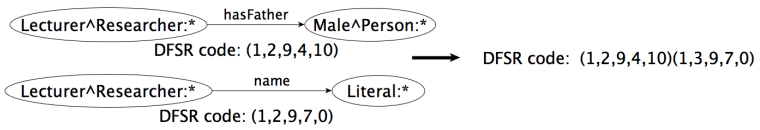


FIG. 4 – *Exemple de jointure de deux codes DFSR dont les sujets sont identiques*

Cas 2 : Le sujet d'un code est identique à l'objet de l'autre code. Les temps de découverte du code minimum restent $(1, 2)$. Les temps de découverte de l'autre code sont $(3, 1)$ s'il dispose du nœud objet partagé ou $(2, 3)$ sinon. Le reste du processus est similaire au cas 1.
Cas 3 : Les objets des deux codes sont identiques. Les temps de découverte du code minimum sont $(1, 2)$ et ceux du second code $(3, 2)$. Le reste du processus est similaire au cas 1.
Ces trois cas ne sont pas disjoints. Une jointure peut ainsi générer jusqu'à quatre codes.
Phase 3 : Construction récursive des motifs de taille *s*. La jointure de deux codes d_G et d_H de taille $s-1$ $(s > 2)$ qui partagent un même noyau ($s-2$ arcs) est effectuée pour obtenir un code de taille s. Avant de garder le code résultant et de lui attribuer un identifiant nous vérifions (i) qu'il n'est pas redondant dans l'index (ii) et que le motif auquel il correspond est instancié dans la source. Comme dans la phase précédente les codes joints sont marqués. Pour effectuer la jointure de deux graphes d_G et d_H de taille $s(s > 1)$ il faut d'abord extraire respectivement de chacun de ces graphes les arcs spécifiques e_G et e_H n'appartenant pas au noyau partagé. L'objectif ensuite est de rattacher l'arc e_G au graphe d_H en cherchant les temps t_{H1} et t_{H2} dans d_H correspondant repectivement aux temps t_{G1} et t_{G2} des noeuds de e_G dans d_G. Pour trouver t_{H1} nous cherchons s'il existe un arc unique dans d_G passant par le noeud de temps de découverte t_{G1}. Si tel est le cas, trouver l'arc unique correspondant dans d_H nous permet de determiner t_{H1}. Si aucun arc unique n'est trouvé une procèdure simillaire à un test isomorphe entre d_G et d_H est exécutée pour determiner t_{H1}. Un travail simillaire est effectué pour trouver t_{H2}. Contrairement à t_{H1}, t_{H2} peut disposer de deux valeurs selon que les noeuds de e_G et e_H non liés au noyau sont identiques ou pas. Si deux valeurs sont disponibles pour t_{H2} cela signifie que la jointure va generer deux motifs. Les temps de decouverte t_{H1} et t_{H2} ainsi trouvés sont affectés aux noeuds e_G. L'arc e_G est ensuite ajouté à d_H qui est enfin trié. La figure 5 montre un exemple de jointure de deux codes de taille 2.

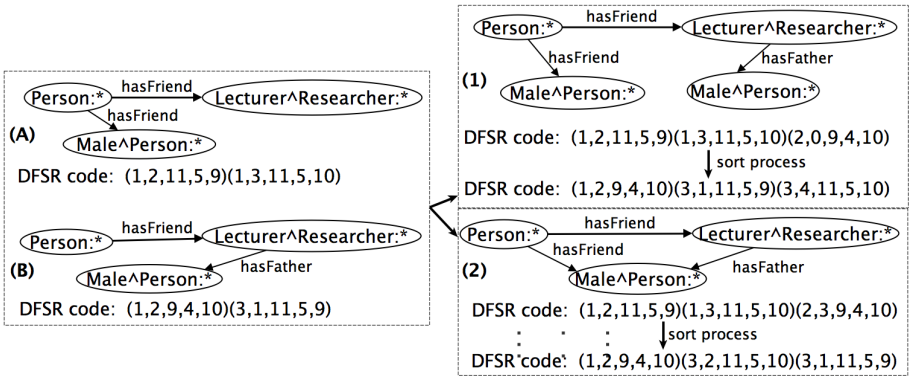


FIG. 5 – *Jointure de deux motifs de taille 2*

Notons finalement que nous combinons la génération et l'évaluation des candidats comme dans Yan et Han (2002) pour optimiser la construction de l'index. L'algorithme s'arrête au niveau s si aucun motif de cette taille n'est instancié dans la source RDF.

4 Expérimentations

Notre algorithme utilise la plateforme CORESE/KGRAM de Corby (2008) qui implémente SPARQL 1.1. L'index est construit après les inférences sur la source. Nous avons testé notre algorithme sur trois sources fusionnées : personData de DBPedia contenant des motifs en étoile (cf. Gandon et al. (2008)) ; une source basée sur foaf avec des nœuds anonymes et multi typés et des cycles ; une source de tags extraits de Delicious avec des motifs en forme d'étoiles et de chemins. La source résultante contient 149882 triplets. La figure 6 montre les résultats obtenus : *RP* représente le nombre de motifs non gardés car redondant dans l'index. *MP* est le nombre de motifs marqués car étant inclus dans des motifs appartenant à l'index et *JO* est le nombre de jointure effectuées. (*NF*) est le nombre de motifs supprimés car n'ayant aucune instance et (*CT*) est le temps d'exécution en seconde par niveau. La somme de *MP* et *UP* est le nombre de motifs dans l'index. Nous pouvons distinguer quatre périodes durant l'exécution :

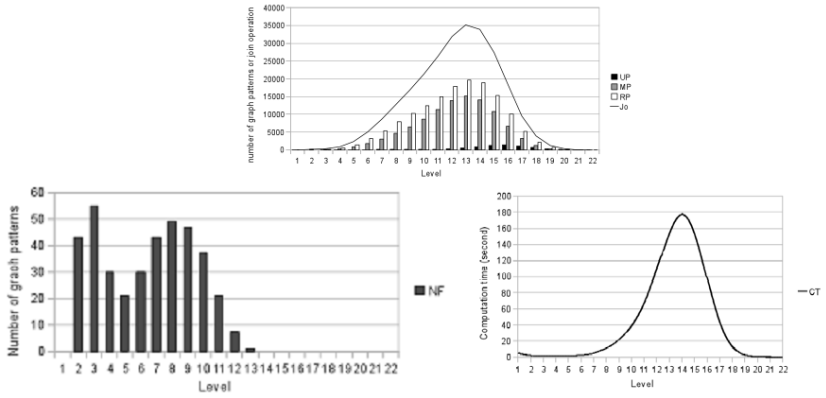


FIG. 6 – *Nombre total de motifs, NF, RP, JO et CT par niveau.*

Niveau 1 : Le temps d'exécution de la requête SPARQL du niveau 1 dépend fortement de la taille de la source. Combiné au nombre peu élevé de jointures pour les niveaux de 2 à 6 nous obtenons un temps d'exécution du niveau 1 supérieur à ceux des niveaux de 2 à 6. Pour améliorer le temps d'exécution du niveau 1, l'une de nos perspectives est d'utiliser les statistiques générées par les moteurs de recherche à la place de la requête pour construire le niveau 1.
Niveau de 2 à 6 : Le temps d'exécution est faible à cause du nombre peu élevé de jointures.
Niveau 7 à 13 : Le nombre de jointure augmente rapidement favorisant un nombre élevé d'accès à la source RDF, de tests isomorphes et donc l'augmentation du temps d'exécution.
Niveau 14 à 22 : $NF = 0$. Cela signifie que tous les motifs candidats sont instanciés. L'algorithme converge rapidement et le temps d'exécution est proche de zéro à partir du niveau 19. Les motifs marqués dans l'index représentent $90,44\%$ du nombre total de motifs. Le nombre de redondances est encore élevé et l'une de nos perspectives est de le réduire.
Complexité temporelle. La complexité du niveau 1 est de $\Theta(na) + Tq$ avec na le nombre de triplets et Tq le temps d'exécution de la requête Sparql du niveau 1 qui dépend également du nombre de triplets. La complexité du niveau 2 est de $\Theta(nb_1^2)(Tq_2 + O(ni_2))$ avec nb_1 le

nombre de motifs de taille 1, ni_2 le nombre moyen d'instances d'un motif de taille 2 et Tq_2 le temps moyen pour vérifier l'instanciation d'un motif de taille 2. La complexité du niveau $s > 2$ est de $\Theta(nb_{s-1}^2[s^s + Tq_s + ni_s])$.

5 Mise à jour incrémentale de l'index

L'ajout ou la suppression d'annotations de la source peut entrainer des changements dans l'index. A la place d'une reconstruction de l'index nous proposons un algorithme incrémental. Quand une annotation, représentée par un graphe contenant plusieurs triplets, est ajoutée nous distinguons quatre phases pour mettre à jour l'index :
Phase 1 – Initialisation : Nous construisons les codes correspondant aux motifs de taille 1 de l'annotation. Ces codes seront utilisés lors la phase 2.
Phase 2 – Mise à jour du niveau 1 : Nous vérifions la présence dans l'index des codes obtenus précédemment. Les codes absents de l'index sont ajoutés au niveau 1 avec de nouveaux identifiants. En parallèle tous les codes (absents ou non) sont répertoriés comme étant les seuls à pouvoir apporter des modifications au niveau 2.
Phase 3 – Mise à jour du niveau 2 : Les codes répertoriés au niveau 1 sont joints aux autres codes du même niveau avec qui ils partagent au moins un nœud. Les codes résultants sont classés en trois catégories :
Catégorie 1 : Le code résultant est déjà dans l'index et a été généré en utilisant la même jointure. Dans ce cas aucun changement n'est apporté au niveau 2.
Catégorie 2 : Le code résultant est dans l'index et a été généré en utilisant une autre jointure. Le code est ajouté au niveau 2 mais marqué redondant.
Catégorie 3 : Le code résultant n'est pas dans l'index. Ce nouveau code est alors ajouté au niveau 2 avec un nouvel identifiant. A la fin de chaque catégorie le code résultant est répertorié comme pouvant apporter des modifications au niveau 3.
Phase 4 – Mise à jour du niveau $n(n > 2)$: Les codes répertoriés au niveau $n-1$ sont joints aux autres codes de même niveau avec lesquels ils partagent un noyau. Les codes résultants sont classés en trois catégories comme dans la phase 3 et répertoriés pour le niveau suivant. L'algorithme s'arrête au niveau $s(s > 1)$ si aucun code n'a été répertorié au niveau $s-1$.
Nous distinguons trois phases pour mettre à jour l'index après une suppression d'annotations :
Phase 1 – Initialisation : Le processus est similaire à celui décrit lors de l'insertion.
Phase 2 – Mise à jour du niveau 1 : Nous vérifions dans la source RDF l'instanciation de chaque code issu de l'initialisation. Nous distinguons deux cas :
Cas 1 : Le motif dispose d'instances. Nous ajoutons l'identifiant du code correspondant dans une liste nommée *checkList* car ce code peut entrainer des suppressions au niveau 2.
Cas 2 : Le motif ne dispose plus d'instance. Le code correspondant est supprimé du niveau 1 et son identifiant est ajouté dans une liste nommée *delList* car tous les codes de niveau 2 générés à partir de ce code doivent être supprimés.
Phase 3 : Mise à jour du niveau $n(n > 1)$. Nous parcourons les listes *delList* et *checkList* précédemment créées.
Parcours de la liste *delList* : Chaque code de niveau n généré en utilisant un code de la liste *delList* du niveau $n-1$ est supprimé de l'index et son identifiant est ajouté à la liste *delList* du niveau n. Si le code supprimé n'était pas marqué comme redondant, nous enlevons le marquage effectué sur son premier code redondant s'il existe. Lors du processus de construction de

l'index si plusieurs codes sont redondants seul le premier n'est pas marqué comme redondant.
Parcours de la liste *checkList* : Nous vérifions que chaque code de niveau n généré en utilisant un code de la liste *checkList* est instancié dans la source. Nous distinguons deux cas :
Cas 1 : Aucune instance du motif. Le code est supprimé du niveau n et son identifiant est ajouté à la liste *delList* du niveau n. Si le code supprimé n'est pas marqué comme redondant alors le marquage est enlevé sur le premier code redondant correspondant si celui-ci existe.
Cas 2 : Le motif est instancié. L'identifiant du code est ajouté à la liste *checkList*.
L'algorithme s'arrête lorsque le niveau le plus élevé de l'index est mis à jour ou lorsque les listes *checkList* et *delList* du niveau inférieur sont vides.
Complexités temporelles. Pour l'ajout, la complexité du niveau 1 est de $\Theta(nb_1)$ avec nb_1 le nombre de motifs de taille 1. La complexité du niveau 2 est de $\Theta(nb_1 * nb_2 + nb_1(Tq_2 + ni_2))$ avec ni_2 le nombre moyen d'instances d'un motif de taille 2 et Tq_2 le temps moyen pour vérifier l'instanciation d'un motif de taille 2. La complexité du niveau $s > 2$ est $\Theta(nb_{s-1}(nb_{maj}$ $(s^s + nb_s + Tq_s + ni_s)))$ avec nb_{maj} le nombre de motifs répertoriés au niveau $s - 1$. Pour la suppression, la complexité du niveau 1 est de $\Theta(nb_1 * Tq_1)$. La complexité du niveau $s > 1$ est $\Theta(nb_s^2(nbd_{s-1} + nbv_{s-1}) + nbd_{s-1} * nbv_{s-1} * Tq_s)$ avec nbd_{s-1} et nbv_{s-1} le nombre respectif de motifs dans *delList* et *checkList* au niveau $s - 1$.

6 Etat de l'art

Les index sont généralement représentés comme une hiérarchie organisée en niveau selon la taille des éléments indexés. Dans la littérature les approches se différencient par la structure de ces éléments. En étendant la structure utilisée pour les bases de données relationnelles, Han et Xie (1994) proposent comme structure de base les paires d'identifiants d'objets de deux classes qui sont connectées entre elles par une relation logique. Stuckenschmidt et al. (2004) étendent cette approche en proposant une hiérarchie de chemins comme index. Gandon (2003) et Gandon et al. (2008) ajoutent à la structure de chemins celle d'étoiles. Si le problème de perte d'information structurelle est en partie résolu avec l'ajout de structures additionnelles aux chemins, d'autres inconvénients révélés dans Yan et al. (2004) montrent les limites des approches basées sur les chemins. Pour améliorer l'exécution de requêtes et prendre en compte des structures plus complexes que les chemins et les étoiles Yan et al. (2004) proposent d'utiliser les motifs de graphes comme structure de base de l'index. Les approches basées sur la découverte de motifs de graphes fréquents combinent une phase de génération de motifs candidats et une phase d'évaluation de ces motifs. Les problématiques principales étudiées dans ces approches sont (i) la gestion des redondances, (ii) la réduction de la taille de l'index et (iii) la proposition d'un opérateur de jointure entre motifs de graphes. Parmi les algorithmes de la littérature, nous distinguons principalement deux familles d'approches pour la gestion des redondances : (1) *des algorithmes qui utilisent une forme canonique pour comparer efficacement deux graphes et supprimer les redondances dans l'ensemble candidat*. Inokuchi et al. (2000) utilisent ainsi une matrice d'adjacence pour représenter un graphe et définit une forme canonique à partir de la matrice. Vanetik et al. (2002) utilisent également pour représenter un graphe une matrice dont les lignes sont les nœuds du graphe et les colonnes les chemins. Une séquence de chemins suivant un ordre bien défini permet dans Vanetik et al. (2002) d'établir une forme canonique du graphe. Han et al. (2007), Maduko et al. (2008), Yan et Han (2002) et Yan et al. (2004) utilisent une représentation sous forme d'arbre plus concise que les matrices et définissent le

code DFS minimum comme forme canonique ; (2) *d'autres algorithmes proposent un opérateur de jointure qui génère moins de redondances*. En effet les redondances découlent du fait qu'une opération de jointure peut générer plusieurs graphes parfois redondants et que plusieurs opérations distinctes peuvent proposer le même graphe. Huan et al. (2003) introduisent un opérateur de jointure générant au plus deux graphes et suppriment complètement les redondances lors d'une opération de jointure. Notons enfin que pour réduire les accès à la source RDF en éliminant les candidats, Kuramochi et Karypis (2001) utilisent la monotonie de la fréquence.
Comme la plupart des approches de découverte de graphes fréquents (Huan et al. (2003), Inokuchi et al. (2000), Kuramochi et Karypis (2001) et Yan et Han (2002) par exemple) nous générons les motifs candidats de taille s en joignant deux motifs de taille $s-1$. Pour éviter de joindre deux à deux tous les motifs nous avons ajouté des informations dans nos codes DFSR pour connaître les motifs qui partagent un même noyau et qui donc peuvent être joints.
Quand une classe ou une relation entre classes est mise à jour Han et Xie (1994) proposent une mise à jour incrémentale de la hiérarchie d'index. Les mises à jour affectent seulement le niveau 1 et des éléments de niveau supérieur précisément identifiés. La mise à jour de l'index lors d'ajouts dans la source consiste dans Yan et al. (2004) à mettre à jour la liste de graphes contenant les fragments ajoutés. Après un certain nombre de mises à jour Yan et al. (2004) constatent une dégradation de l'index et propose sa reconstruction totale
Nous utilisons les motifs de graphes comme structure de base de l'index comme Yan et al. (2004) mais étendu à RDF. Pour supprimer les redondances notre algorithme combine les deux solutions utilisées dans les différentes approches : (1) Utilisation d'une forme canonique (DFSR) pour faciliter la suppression des redondances, (2) proposition d'un opérateur de jointure entre codes DFSR pour générer au plus quatre motifs sans redondance. Durant la jointure nous identifions les arcs à ajouter et les nœuds de jointure. Une phase d'élagage n'est pas nécessaire dans notre cas car tous les motifs générés respectent la monotonie de la fréquence. Nous avons amélioré l'algorithme proposé par Basse et al. (2010) par la prise en compte des cycles, nœuds vides ou multi typés et par l'ajout d'un algorithme incrémental.

7 Conclusion

Dans cet article, nous présentons un algorithme pour construire et mettre à jour une représentation compacte d'une source RDF. Nous avons adapté pour cela le codage DFS aux motifs de graphes RDF et nous avons proposé un opérateur de jointure pour réduire significativement le nombre de motifs générés par jointure. La taille de l'index peut être significativement réduite en gardant uniquement les motifs de couverture maximale. La mise à jour incrémentale permet d'éviter une reconstruction totale de l'index à chaque mise à jour des sources. Une des principales perspectives de ce travail est la prise en compte d'autres caractéristiques des schémas RDFS pour réduire encore la taille de l'index.

Références

Baget, J.-F., O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, F. Gandon, A. Giboin, A. Gutierrez, M. Leclère, M.-L. Mugnier, et R. Thomopoulos (2008). Griwes : Generic model and pre-

liminary specifications for a graph-based knowledge representation toolkit. In *ICCS'2008*, Toulouse, France.

Basse, A., F. Gandon, I. Mirbel, et M. Lo (2010). Frequent graph pattern to advertise the content of rdf triple stores on the web. In *Web Science Conference*, Raleigh, NC, USA.

Battle, R. et E. Benson (2008). Bridging the semantic web and web 2.0 with representational state transfer (rest). *Web Semantics 6*, 61–69.

Corby, O. (2008). Web, graphs and semantics. In *ICCS'2008*, Toulouse, France.

Gandon, F. (2003). Agents handling annotation distribution in a corporate semantic web. *Web Intelligence and Agent Systems 1*(1), 23–45.

Gandon, F., M. Lo, et C. Niang (2008). Un modèle d'index pour la résolution distribuée de requêtes sur un nombre restreint de bases d'annotations rdf. In *IC'2008*, Nancy, France.

Han, J. et Z. Xie (1994). Join index hierarchies for supporting efficient navigations in object-oriented databases. In *VLDB'1994*, Santiago de Chile, Chile, pp. 522–533.

Han, S., W. Keong Ng, et Y. Yang (2007). Fsp : Frequent substructure pattern mining. In *ICICS'07*, Singapore, pp. 10–13.

Huan, J., W. Wang, et P. J. (2003). Efficient mining of frequent subgraphs in the presence of isomorphism. In *ICDM'03*, Melbourne, pp. 549–552.

Inokuchi, A., T. Washio, et H. Motoda (2000). An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD'00*, Lyon, France, pp. 13–23.

Kuramochi, M. et G. Karypis (2001). Frequent subgraph discovery. In *ICDM'01*, San Jose, CA, pp. 313–320.

Maduko, A., K. Anyanwu, A. Sheth, et P. Schliekelman (2008). Graph summaries for subgraph frequency estimation. In *ESWC'08*, Tenerife, SPAIN, pp. 508–523.

Stuckenschmidt, H., R. Vdovjak, G. Jan Houben, et J. Broekstra (2004). Index structures and algorithms for querying distributed rdf repositories. In *WWW'04*, NY, USA, pp. 10–14.

Vanetik, N., E. Gudes, et S. E. Shimony (2002). Computing frequent graph patterns from semistructured data. In *ICDM'02*, Maebashi, Japan, pp. 458–465.

Yan, X. et J. Han (2002). gspan : Graph-based substructure pattern mining. In *ICDM'02*, Maebashi, Japan, pp. 721–724.

Yan, X. et J. Han (2003). Closegraph : Mining closed frequent graph patterns. In *KDD'03*, Washington, pp. 286–295.

Yan, X., P. S. Yu, et J. Han (2004). Graph indexing : A frequent structure-based approach. In *SIGMOD'04*, Paris, pp. 335–346.

Summary

Many semantic web applications need to know what kind of content each triple store holds in order to assess if it can contribute to its queries. We present an incremental update algorithm to build and maintain indexes summarizing the content of triple stores.

Vers un système de capitalisation des connaissances : extraction d'événements par combinaison de plusieurs approches.

Laurie Serrano*,** Maroua Bouzid*
Thierry Charnois*
Grilheres Bruno**

*GREYC, Université de Caen Basse-Normandie
prenom.nom@unicaen.fr, http://www.greyc.fr
**IPCC, Cassidian
prenom.nom@cassidian.com, http://weblab-project.org

Résumé. Face à l'augmentation vertigineuse de l'information disponible librement (en particulier sur le Web), repérer efficacement les informations qui sont susceptibles de nous intéresser (dans un contexte professionnel ou personnel) s'avère une tâche longue et complexe. En réponse à cela, l'équipe IPCC [1] développe le WebLab [2], une plateforme d'intégration de différents services de « media mining » [3] pour la découverte de connaissances et l'aide à la décision. Dans cet article, nous présentons notre système d'extraction automatique d'événements pour le ROSO [4] fondé sur la combinaison de plusieurs approches actuelles en extraction d'information. Nous proposons, dans un premier temps, une modélisation du domaine et plus particulièrement des événements. Puis, nous décrivons de façon détaillée notre approche et les différentes techniques utilisées. Enfin, nous concluons en résumant l'avancée de nos travaux et les perspectives envisagées.

1 Introduction

Nos travaux se placent dans le cadre du WebLab, plateforme open source dédiée à l'intégration d'outils de « media mining » et exploitant les technologies du Web sémantique (Giroux et al. (2008), Brunessaux et al. (2011)). Cet article présente des recherches en cours visant à exploiter efficacement la masse croissante d'informations disponibles en sources ouvertes afin d'en extraire un ensemble de connaissances pertinentes. Il s'agit, plus précisément, de proposer un système de capitalisation des connaissances visant à faciliter et réduire le travail des opérationnels dans le cadre de la veille économique et stratégique et du ROSO. Pour cela, nos travaux s'organisent selon trois axes de recherche :

1. Information Processing, Control and Cognition
2. `http://weblab-project.org/`
3. Fouille de documents multimedia
4. Renseignement d'Origine Sources Ouvertes

– extraction d'information ;
– capitalisation et gestion des connaissances ;
– interaction homme-machine.

La figure 1 donne un aperçu du fonctionnement général du système de capitalisation des connaissances que nous envisageons. Pour résumer, un ensemble de documents est traité par le système d'extraction d'information, puis les informations extraites sont stockées sous la forme de triplets RDF[5] dans une base de connaissances. Celle-ci est régie par notre ontologie de domaine (cf. partie 2) et couplée à un moteur d'inférence permettant la découverte de nouvelles connaissances. Chaque événement est ensuite présenté à l'utilisateur sous forme d'une fiche de connaissances que celui-ci pourra compléter et modifier en fonction de son propre savoir (cf. partie 4). Enfin, nous envisageons d'utiliser les différentes actions de l'utilisateur afin d'améliorer les performances futures du système.

Cet article est centré sur notre modèle d'extraction automatique d'événements fondé sur la combinaison de plusieurs approches actuelles en extraction d'information. Ici, nous souhaitons extraire un ensemble d'événements d'intérêt (préalablement définis dans une ontologie de domaine) à partir de dépêches de presse (AFP par exemple) en anglais et en français. Nous proposons, dans un premier temps, une modélisation du domaine et plus particulièrement des événements. Puis, nous décrivons de façon détaillée notre approche et les différentes techniques utilisées. Enfin, nous concluons en résumant l'avancée de nos travaux et les perspectives envisagées.

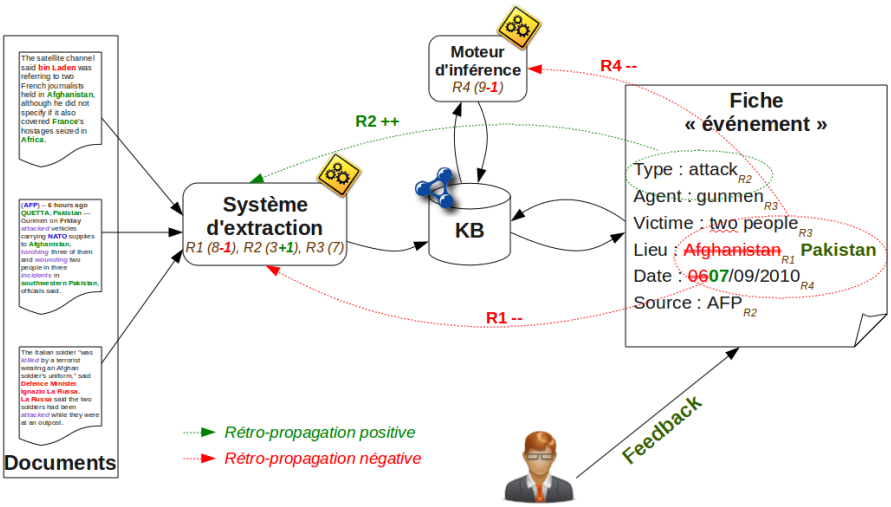


FIG. 1 – *Système de capitalisation des connaissances*

5. Resource Description Framework, `http://www.w3.org/RDF/`

2 Modélisation des connaissances

Afin de définir avec précision les informations pertinentes dans ce cadre d'application, celles-ci ont été modélisées sous la forme d'une ontologie de domaine.

2.1 WOOKIE : Weblab Ontology for Open sources Knowledge and Intelligence Exploitation

WOOKIE constitue l'ontologie de domaine qui sert de guide à notre système de capitalisation. Celle-ci est implémentée au format OWL [6] et suit les recommandations du W3C [7] concernant la représentation de la connaissance dans le cadre du Web Sémantique. WOOKIE a été élaborée suite à un état de l'art approfondi du domaine et des ontologies existant à l'heure actuelle. Nous avons examiné à la fois des ontologies dites de « haut niveau » mais également des modélisations spécifiques au ROSO telles que celles décrites par les standards OTAN. Suite à cet état de l'art, nous avons choisi de créer notre propre ontologie afin de répondre précisément aux besoins de notre application, en particulier pour la modélisation des événements. Toutefois, nous exploitons les travaux déjà réalisés par la création de liens (imports d'ontologies et équivalences de classes) entre WOOKIE et d'autres ontologies telles que OWL-Time [8] pour la représentation temporelle, Geonames [9] pour la représentation spatiale, FOAF [10], mais aussi The Event Ontology [11] et SEM [12] concernant les événements. WOOKIE est centrée sur le « pentagramme du renseignement » représentant les cinq entités centrales du ROSO (les événements, les personnes, les organisations, les lieux et les équipements) ainsi qu'un ensemble de liens entre ces entités (Serrano et al. (2011) pour plus de détails). Nous avons accordé une attention particulière à la modélisation de la classe « événement » compte tenu de l'importance de ces entités pour toute activité de veille (cf. partie 2.2). Précisons enfin que, même si WOOKIE est à l'heure actuelle utilisée dans le cadre du ROSO, nos choix de modélisation permettent d'éventuelles extensions à d'autres domaines d'application (intelligence économique, etc.).

2.2 Définition et modélisation des événements

L'événement étant l'objet central de notre système de capitalisation il est nécessaire de définir plus précisément ce concept. Considéré comme une entité aux propriétés bien spécifiques, l'événement a initialement été étudié par des philosophes (Davidson (2001)) puis par des linguistes (Van De Velde (2006)).

Nous avons retenu certains travaux tels que ceux de Krieg-Planque (2009). L'auteur donne une définition simple de l'événement mais qui nous paraît adaptée : « un événement est une occurrence perçue comme signifiante dans un certain cadre ». Ici, le terme « occurrence » met l'accent sur la notion de temporalité qui fait partie intégrante de ce concept. Cet aspect temporel de l'événement est au coeur de l'approche TimeML (Pustejovsky et al. (2003)), l'un des deux

6. Ontology Web Language, `http://www.w3.org/TR/owl-features/`
7. World Wide Web Consortium, `http://www.w3.org/`
8. `http://www.w3.org/TR/owl-time/`
9. `http://www.geonames.org/`
10. Friend Of A Friend, `http://www.foaf-project.org/`
11. `http://motools.sourceforge.net/event/event.html`
12. Simple Event Model, `http://www.few.vu.nl/~wrvhage/#research`

principaux courants dans le domaine de l'extraction automatique des événements. Le modèle utilisé dans le cadre des campagnes d'évaluation ACE (NIST (2005)) diffère de cette approche en définissant l'événement comme une structure complexe impliquant plusieurs arguments. Le « cadre » selon Krieg-Planque réfère à « un système d'attentes donné » qui « détermine le fait que l'occurrence acquiert (ou non) [...] sa remarquabilité [...] et, par conséquent, est promue (ou non) au rang d'événement. ». Dans nos travaux, ce cadre est défini par l'ontologie de domaine WOOKIE et plus précisément par la spécification de la classe « Event » à travers ses différentes sous-classes et propriétés. D'autre part, Neveu et Quéré (1996) s'attachent à décrire plus précisément la sémantique portée par les événements. Ils soulignent que l'interprétation d'un événement est étroitement liée au contenu sémantique des termes utilisés pour nommer cet événement. Pour plus de clarté nous reprendrons le terme de « nom d'événement » proposé par Krieg-Planque (2009). Ces « noms d'événement » transposent en langage naturel la « propriété sémantique » des événements mentionnée par Saval et al. (2009). De plus, cette description de l'événement est au centre d'un phénomène plus large, que Ricœur (1983) nomme « mise en intrigue », visant à organiser, selon le cadre mentionné plus haut, un ensemble d'éléments circonstants ou participants de l'événement.

En considérant ces différents travaux, nous présentons ci-après la définition d'un événement dans le cadre de nos recherches. Nous prenons pour point de départ la définition de Krieg-Planque citée précédemment. Toutefois, celle-ci étant très théorique, il convient d'expliciter comment un événement est exprimé au sein des dépêches de presse et comment est modélisé ce concept au sein de notre ontologie de domaine. Après observation de plusieurs dépêches, celles-ci semblent rapporter un événement principal, celui-ci étant le plus souvent résumé dans le titre et explicité tout au long de l'article (parfois en faisant référence à d'autres événements secondaires). En examinant de plus près cette description de l'événement tout au long de la dépêche, un certain nombre de « sous-événements » contribuent à la « mise en intrigue » mentionnée auparavant. Ces « sous-événements » correspondent à une association entre un « nom d'événement » et une ou plusieurs entités d'intérêt (une date, un lieu et des participants à l'événement). Dans le cadre de nos travaux, notre but est d'extraire ces « sous-événements » d'intérêt pour ensuite fusionner automatiquement ceux qui réfèrent dans la réalité à un seul et même événement. Chaque événement résultant de cette fusion sera présenté à l'utilisateur sous la forme d'une fiche de connaissances. Afin de proposer une représentation formelle d'un événement, nous nous appuyons sur les travaux de Saval et al. (2009) qui propose une extension sémantique pour la modélisation des événements de type « catastrophes naturelles ». Celui-ci définit un événement E comme la combinaison de 3 composantes : une propriété sémantique S, un intervalle temporel I, et une entité spatiale SP. Un événement est donc représenté sous la forme : $E\langle I, SP, S\rangle$.

Dans notre cas, la propriété sémantique d'un événement correspond aux différents types d'événement définis dans notre ontologie de domaine (les sous-classes de la classe « Event »), la composante temporelle constitue la date ou période d'occurrence de l'événement et l'entité spatiale équivaut au lieu d'occurrence de l'événement. Nous proposons d'adapter cette représentation à notre domaine d'application en l'enrichissant d'une composante supplémentaire A correspondant aux différents participants impliqués dans l'événement. Nous avons donc dorénavant $E\langle I, SP, S, A\rangle$ où A est un ensemble de participants jouant un ou plusieurs rôle(s). Un participant est noté P_i où $0 < i < n$ et un rôle est noté r_j où $0 < j < k$. La composante A est donc définie de la façon suivante : $A = \{(P_\alpha, r_\beta)\}$ tel que le participant P_α joue le rôle

r_β dans l'événement en question. La figure 2 illustre notre modélisation de l'événement dans l'ontologie de domaine (un exemple d'événement est proposé en vert).

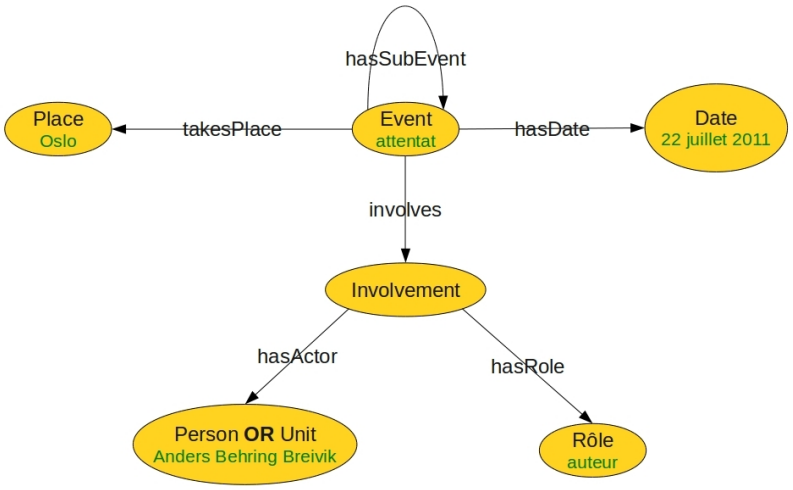


FIG. 2 – *Modélisation d'un événement dans WOOKIE*

3 Extraction d'information

3.1 Tour d'horizon

L'extraction d'information est une discipline assez récente qui consiste en une analyse partielle d'un texte afin d'en extraire des informations spécifiques. Celles-ci permettent de construire une représentation structurée (bases de données, fiches, tableaux) d'un document à l'origine non-structuré. Cela en fait une approche guidée par le but de l'application dans laquelle elle s'intègre, dépendance qui reste, à l'heure actuelle, une limite majeure des systèmes d'extraction. Les tâches les plus communes en extraction d'information restent la reconnaissance d'entités nommées (Nadeau et Sekine (2007)), de relations entre entités et d'événements (Hobbs et Riloff (2010)).

La reconnaissance d'entités nommées a été et est encore beaucoup étudiée car celles-ci constituent des éléments indispensables pour l'extraction d'entités plus complexes et reste une tâche difficile. Ces entités correspondent de façon générale aux noms de personne, organisation, lieu, mais aussi aux dates, unités monétaires, pourcentages, unités de mesure, etc. Par ailleurs, les différents outils d'extraction d'information s'attachent à extraire les relations entre entités et les événements. Les relations correspondent aux liens existant entre différentes entités repérées dans un texte : il peut s'agir par exemple de détecter les relations entre une personne et une organisation (appartenance, direction, etc.) ou encore d'extraire les attributs d'une personne (date de naissance, courrier électronique, adresse, etc.). Enfin, une dernière tâche est l'extraction d'événements, particulièrement utile dans les activités de veille économique et stratégique (Naughton et al. (2006)). Celle-ci peut-être conçue comme une forme

particulière d'extraction de relations où un « nom d'événement » est relié avec une date, un lieu et des participants (cf. partie 2.2).

Les dix dernières années ont vu apparaître un intérêt grandissant pour ce domaine avec notamment la création de campagnes d'évaluation telles que ACE [13], MUC [14], ESTER [15], CONLL [16], TAC [17], etc. Deux approches principales émergent alors : l'extraction basée sur des techniques linguistiques d'un côté et les systèmes statistiques à base d'apprentissage de l'autre. Celles-ci se basent, de façon commune, sur des pré-traitements linguistiques « classiques » comme la « tokenization » (découpage en mots), la lemmatisation (attribution de la forme non-fléchie associée), l'analyse morphologique (structure et propriétés d'un mot) ou syntaxique (structure d'une phrase et relations entre éléments d'une phrase). La première approche exploite les avancées en TAL [18] et repose principalement sur l'utilisation de grammaires formelles construites par la main d'un expert-linguiste. Les pré-traitements cités plus haut servent de base à la construction de règles et patrons linguistiques qui définissent les contextes d'apparition de telle entité ou relation. Notons ici l'importance particulière accordée à l'analyse syntaxique (en constituants ou dépendance) dans le repérage et le typage des relations et des événements. La seconde approche utilise des techniques statistiques pour « apprendre » des régularités sur de larges corpus de textes où les entités-cibles ont été préalablement annotées. Ces méthodes d'apprentissage sont supervisées, non-supervisées ou semi-supervisées et exploitent des caractéristiques textuelles plus ou moins linguistiques. Parmi celles-ci nous pouvons citer les « modèles de Markov Caché » (HMM), les « Conditional Random Fields » (CRF), les « Support Vector Machine » (SVM), etc. (Ireson et Ciravegna (2005) pour un état de l'art approfondi). Par ailleurs, de plus en plus de recherches portent sur l'apprentissage de ressources linguistiques ou encore sur l'utilisation d'un apprentissage dit « semi-supervisé » visant à combiner des données étiquetées et non-étiquetées (Nadeau (2007), Hobbs et Riloff (2010)).

Un nouveau type d'approche tend à se généraliser : ce sont les méthodes hybrides. Les acteurs du domaine choisissent de combiner plusieurs techniques face aux limites des approches symboliques et statistiques. Tout d'abord, celles-ci s'avèrent dépendantes d'un domaine ou d'un genre de texte particulier et cela nécessite une constante ré-adaptation des modèles d'extraction. Les approches à base de règles linguistiques souffrent également d'un développement manuel coûteux et de la nécessité d'une expertise en linguistique pour pouvoir les modifier et les adapter. Pour tenter de résoudre cela, les experts se penchent actuellement vers des méthodes d'apprentissage automatique de patrons linguistiques (Charnois et al. (2009)). Pour finir, les approches statistiques nécessitent, lors de la phase d'apprentissage, une grande quantité de textes pré-annotés et cela constitue une réelle contrainte car ces données ne sont pas toujours disponibles. Des recherches sont menées dans le sens d'un apprentissage dit « semi-supervisé » visant à mêler des données étiquetées et non-étiquetées (Nadeau (2007)).

13. Automatic Content Extraction, http://www.itl.nist.gov/iad/mig/tests/ace/
14. Message Understanding Conference, http://www-nlpir.nist.gov/related_projects/muc/
15. Évaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques, http://www.afcp-parole.org/camp_eval_systemes_transcription/
16. Conference on Computational Natural Language Learning, http://ifarm.nl/signll/conll/
17. Text Analysis Conference, http://www.nist.gov/tac/
18. Traitement Automatique des Langues

3.2 Modèle proposé

Dans la lignée de ces nouvelles approches, la combinaison de plusieurs techniques d'extraction nous parait être la solution la plus pertinente. En effet, élaborer un système composite permet de tirer le meilleur parti des différentes approches actuelles. Pour cela, nous avons choisi ici de combiner trois extracteurs :

- un extracteur symbolique, fondé sur des règles linguistiques écrites manuellement ;
- un extracteur statistique utilisant les CRF (Conditional Random Fields) ;
- un extracteur hybride basé sur l'apprentissage de motifs séquentiels fréquents.

3.2.1 Approche symbolique

Cette première approche a été développée grâce à la plateforme open source GATE [19]. Cet extracteur symbolique est composé d'un ensemble de règles de grammaire écrites manuellement et définissant les différents contextes d'apparition possible pour un type d'événement. Ces règles s'appuient sur un premier extracteur d'entités nommées que nous avons développé, puis sur une analyse syntaxique en dépendance (fournie par un module GATE) permettant d'établir des liens entre ces entités au niveau phrastique et de repérer les événements potentiellement pertinents pour notre domaine d'application. Une première évaluation en reconnaissance d'entités nommées a montré des résultats comparables à l'état de l'art avec de bonnes performances pour l'extraction des dates et des lieux mais aussi quelques faiblesses dans la délimitation des organisations ou des personnes. L'évaluation des événements extraits sera réalisée prochainement mais nous pouvons d'ores et déjà remarquer que les résultats de cet extracteur sont très fortement dépendants de l'exactitude de l'analyse syntaxique. Le fonctionnement et l'évaluation de ce système sont décrits plus en détails par Serrano et al. (2011).

3.2.2 Approche statistique

La seconde approche que nous avons choisie est basée sur l'utilisation des CRF [20], une technique statistique couramment utilisée en extraction d'information. Les CRF constituent un modèle probabiliste graphique discriminant et non orienté. Ce type de modèle et souvent utilisé pour annoter ou segmenter des séquences de données telles que des textes en langage naturel ou des séquences d'ADN. En traitement automatique des langues, les CRF trouvent des applications dans l'analyse syntaxique de surface (en constituants), la reconnaissance d'entités nommées et constituent une extension des Modèles de Markov Cachés (HMM). Dans le cadre de nos recherches, nous utilisons l'outil open source CRF++ [21] ainsi qu'un corpus annoté en entités nommées issu de la campagne d'évaluation TempEval [22]. L'apprentissage du modèle est réalisé en prenant en compte trois caractéristiques :

- la forme fléchie d'un mot ;
- sa catégorie grammaticale ;
- le type d'entité auquel il fait référence (le cas échéant).

19. General Architecture for Text Engineering, `http://gate.ac.uk/`
20. Conditional Random Fields
21. `http://crfpp.sourceforge.net/`
22. `http://timeml.org/site/timebank/timebank.html`

Cette partie de nos travaux est en cours et nous envisageons d'exploiter d'autres caractéristiques pour un apprentissage plus performant (forme lemmatisée, syntagmes, etc.). Les travaux existants ainsi que nos premières expérimentations ont montré que les résultats de ce type d'approche dépendent essentiellement de ces caractéristiques ainsi que de la quantité et de la qualité des données d'apprentissage. Nous sommes également en phase d'exploration des différentes techniques d'adaptation des CRF pour l'extraction de relations entre entités dans le cadre de l'extraction d'entités complexes telles que les événements.

3.2.3 Approche hybride

Enfin, nous nous intéressons à l'extraction d'événements par une technique hybride d'extraction de motifs séquentiels fréquents [23]. Ce type d'approche fait le lien entre les méthodes symboliques et statistiques en proposant d'apprendre automatiquement des patrons linguistiques.

La découverte de motifs séquentiels a été introduite par Agrawal et al. (1993) dans le domaine du « data mining » et adaptée par Cellier et Charnois (2010) à l'extraction d'information dans les textes. Ceux-ci s'intéressent en particulier à l'extraction de motifs séquentiels d'itemsets. Il s'agit de repérer, dans un ensemble de séquences, des enchainements d'items ayant une fréquence d'apparition supérieure à un seuil donné (dit « support »). Un certain nombre de paramètres peuvent être adaptés selon l'application visée : nature de la séquence et des items, nombre d'items, grain (mot, syllabe, paragraphe, etc.), support, etc. La fouille sur un ensemble d'items permet l'extraction de motifs combinant plusieurs types d'item et d'obtenir ainsi des patrons génériques, spécifiques ou mixant les informations (ce qui n'est pas permis par les motifs d'items simples), comme par exemple les patrons suivants : <homme de culture> <homme de N> <N PRP N> [24], etc. De plus, contrairement aux différentes approches que nous venons de mentionner, l'apprentissage de MSF ne nécessite ni corpus annoté avec les entités-cibles, ni analyse syntaxique. Cela constitue un réel avantage car, tout d'abord, l'annotation manuelle de corpus reste un effort important et l'analyse syntaxique est encore une technologie peu disponible librement et aux performances inégales selon les langues. Le point faible partagé par toutes ces méthodes d'apprentissage symbolique reste le nombre important de motifs extraits. Pour pallier ce problème, Charnois et al. (2009) propose l'ajout de contraintes pour diminuer la quantité de motifs retournés.

Dans la lignée de ces travaux, nous utilisons l'outil CloSpan (Closed Sequential Pattern Mining Package) fourni dans le cadre du projet open source IlliMine [25]. Celui-ci s'avère très utilisé dans la communauté et présente plusieurs points forts : il extrait uniquement des motifs dits « clos » (c'est-à-dire non redondants) et génère ainsi moins de motifs que d'autres systèmes. De plus, ce logiciel s'avère robuste et permet la fouille d'itemsets, fonctionnalité qui est rarement proposée par les outils existants. Nous sommes en train d'adapter la fouille de MSF avec CloSpan à notre domaine d'application et au traitement de dépêches de presse dans le but d'obtenir des patrons linguistiques permettant la détection d'événements. Tout d'abord, nous avons pré-traité notre corpus grâce à l'outil TreeTagger (Schmid (1994)) afin d'obtenir un découpage en séquences (ici en phrases) ainsi que différents types d'items : mot (forme fléchie),

23. MSF par la suite
24. N pour la catégorie nom, PRP pour préposition
25. `http://illimine.cs.uiuc.edu/`

lemme, catégorie grammaticale. Nous utilisons également notre système symbolique pour obtenir un item « entité nommée » (cela est provisoire, nous envisageons de réaliser une fouille de MSF dédiée à cette tâche). Enfin, nous effectuons un repérage lexical des potentiels « noms d'événement » et de leur type (sous-classes d'événement dans l'ontologie). Comme prévu, le nombre de motifs retournés par CloSpan s'avère élevé, nous devons donc introduire un ensemble de contraintes spécifiques à notre application. Dans un premier temps, seuls les motifs contenant au minimum un « nom d'événement » et une entité (lieu, date, personne, organisation) sont conservés. Les motifs sont ensuite regroupés selon le type de ce « nom d'événement » afin de dégager les régularités spécifiques à chaque type d'événement.

3.3 Qualité des extractions

Afin de combiner ces trois approches pour un système d'extraction plus performant, il devient nécessaire d'évaluer la qualité des extractions obtenues. En effet, cette évaluation servira, tout d'abord, en amont de la base de connaissances pour fusionner les résultats des différents systèmes d'extraction mais également lors de la présentation des fiches de connaissances à l'utilisateur (cf partie 4). L'évaluation de l'information est une problématique largement rencontrée dans la communauté du traitement de l'information et ceci à différents niveaux. Notre étude de la littérature à ce sujet a révélé d'une part des recherches portant sur la modélisation et d'autre part sur des techniques d'estimation de cette qualité.

Traitant des aspects modélisation, nous avons retenu ceux de Laskey et Laskey (2008) définissant une ontologie de « haut niveau » dédiée à la représentation de l'incertitude et visant à faciliter les mécanismes de raisonnement prenant en compte cette incertitude. Par ailleurs, Dedek et al. (2008) propose une extension de cette ontologie dédiée à la représentation de la qualité dans le domaine de l'extraction d'information. Nous avons également noté que la modélisation de la qualité de l'information est étroitement liée, dans beaucoup de travaux actuels, à la notion de provenance de l'information. En effet, dans beaucoup d'applications évaluer la qualité d'une information implique de connaître d'où elle provient, c'est-à-dire sa source mais aussi les différents traitements qui ont été opérés depuis sa collecte jusqu'à sa présentation à l'utilisateur. Davide Ceolin (2010) propose OPM [26], un modèle commun pour tracer et échanger la provenance d'une information. Dans la lignée de ces travaux, Van Hage et al. (2011) propose une combinaison des ontologies SEM et OPM dans le but d'estimer la confiance d'un événement. Sur des aspects plus pratiques, il existe des extensions de la syntaxe RDF par des « graphes nommés » permettant de faire des assertions sur des graphes [27] et, par ce biais, de représenter des méta-informations telles que la confiance et la provenance. Enfin, nous pouvons citer un ensemble de travaux portant sur différentes techniques d'évaluation de la qualité en extraction d'information. Van Keulen et Habib (2011) évalue la qualité des extractions grâce à un ensemble de règles de connaissance définies dans une ontologie combinée à une base de données probabiliste. Soderland et al. (2004) présente un module d'évaluation (intégré au sein de l'extracteur d'événements KnowItAll) fondé sur l'utilisation du Web combiné à un calcul d'information mutuelle (Pointwise Mutual Information). Nous nous sommes également intéressés aux recherches de Besombes et Revault D'Allonnes (2008) qui constituent une bonne

26. Open Provenance Model, `http://openprovenance.org/`
27. `http://www.w3.org/2009/12/rdf-ws/papers/ws06/`

approche générale de cotation de l'information prenant en compte les différents types d'incertitude mentionnés plus haut.

Au vu de ces différents travaux, nous sommes actuellement en cours de réflexion pour proposer une modélisation et une estimation de la qualité des extractions au sein de notre système de capitalisation des connaissances. Tout d'abord, la combinaison de plusieurs extracteurs est une particularité que nous devons prendre en compte dans nos choix. En effet, nous avons pu constater qu'il n'est pas évident d'évaluer et de modéliser la qualité d'une extraction de la même manière pour les 3 approches d'extraction que nous utilisons. Premièrement, l'extraction statistique à base de CRF est souvent qualifiée de « boite noire » car le modèle d'extraction appris est difficilement accessible et modifiable. Toutefois, ce système donne nativement une probabilité à chaque extraction retournée qui peut constituer un premier indice de qualité de l'information. L'extracteur symbolique, quant à lui, n'estime pas par défaut la qualité de ses résultats mais les règles d'extraction sont accessibles. Cette observation vaut aussi pour l'apprentissage symbolique car celui-ci aboutit à la création de règles du même type. Il est également possible d'évaluer les extractions produites par cette approche grâce au nombre d'apparitions de chaque motif dans le corpus d'apprentissage (dit « support »). Enfin, de manière générale, ces trois approches peuvent être jugées manuellement ou automatiquement par une évaluation « a priori » telle que celles menées par les campagnes d'évaluation du domaine (calcul de F-mesure, précision, rappel et autres métriques). Ces premières observations font émerger plusieurs questions importantes auxquelles nous devrons répondre : peut-on concevoir une méthode d'évaluation de cette qualité commune aux 3 extracteurs ? Devons-nous estimer la qualité du système d'extraction en lui-même (des règles ou des modèles appris) ou plutôt la qualité des extractions produites ? Un dernier problème est le mode de représentation de cette qualité : il s'agira de choisir la modélisation la plus proche des besoins du système mais aussi la plus compatible avec les technologies utilisées au sein du WebLab.

4 Fiches de connaissances

Toutes les informations extraites et stockées dans la base de connaissances sont présentées à l'utilisateur sous la forme de fiches de connaissances. Nous nous focalisons, ici, sur la construction automatique de fiches contenant les informations connues à propos d'un événement. Ce mode de visualisation apparaît comme le plus proche des besoins actuels des opérationnels du ROSO. Toutefois, il faut préciser que la fiche de connaissances est un moyen parmi d'autres de visualiser l'information et n'impacte pas la représentation interne de la base de connaissances. Pour construire ces fiches automatiquement, il est nécessaire de définir la nature des informations présentées à l'utilisateur mais aussi les différentes interactions possibles avec le système de capitalisation des connaissances.

Tout d'abord, la fiche comporte plusieurs champs : une description générale (type, nom, alias), la situation dans laquelle cet événement est survenu (lieu, date/période), les différents participants impliqués dans l'événement et enfin un certain nombre de liens avec d'autres événements découverts grâce au moteur d'inférence. Face à une fiche, l'utilisateur a la possibilité de modifier et/ou valider ces différents champs. Il peut alors effectuer plusieurs types d'action :

- valider un champ ou la fiche complète ;
- ajouter un nouveau champ ;
- corriger ou supprimer un champ.

Il faudra bien sûr mettre à jour automatiquement la base de connaissances selon ces modifications. Par ailleurs, l'utilisateur sera guidé dans son activité de veille par le système d'évaluation de la fiabilité des informations abordé plus haut (cf. partie 3.3).

5 Conclusions et perspectives

Nous avons présenté ici une approche d'extraction d'information fondée sur la combinaison de plusieurs extracteurs dans le but d'obtenir de meilleures performances. L'évaluation de notre extracteur symbolique ayant montré des résultats comparables à l'état de l'art mais encore imparfaits, nous attendons de meilleures performances grâce à l'apport des systèmes statistique et hybride. Notre approche se distingue par cette combinaison mais également par la phase de fusion des extractions suggérées par ces différents systèmes qui constitue une phase essentielle dans la capitalisation des connaissances. Nous explorons actuellement un certain nombre de travaux menés en fusion d'informations dans le but de les adapter à la fusion d'informations textuelles. De plus, nous voulons mettre l'accent sur l'importance de l'évaluation de la qualité des extractions pour aider à cette étape de fusion mais également pour guider l'utilisateur dans son travail de validation des fiches. D'autre part, le second axe de nos recherches porte sur les problématiques de capitalisation des connaissances à travers les mécanismes de raisonnement et d'inférence. Nous nous attacherons à définir des règles d'inférence permettant à la fois de regrouper différents « sous-événements » en un seul événement d'intérêt mais également de découvrir des liens entre événements. Il s'agira non seulement de raisonnement spatio-temporel mais aussi d'inférence tenant compte des relations entre un événement et ses participants, tous deux encadrés par notre représentation ontologique du domaine. Par ailleurs, le retour de l'utilisateur étant à l'heure actuelle indispensable face aux limites des technologies employées, nous exploiterons celui-ci afin de réévaluer l'estimation de confiance des extracteurs. Enfin, nous définirons une méthodologie d'évaluation de nos travaux qui comprendra, d'une part, une évaluation automatique du système d'extraction et, d'autre part, une évaluation globale du système de capitalisation par un ensemble d'utilisateurs.

Références

Agrawal, R., T. Imieliński, et A. Swami (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, SIGMOD '93, New York, pp. 207–216. ACM.

Besombes, J. et A. Revault D'Allonnes (2008). An extension of STANAG2022 for information scoring. In *Fusion 2008 - 11th International Conference on Information Fusion*, Allemagne, pp. 1635–1641.

Brunessaux, S., S. Cantarell, A. Giraudel, G. Patrick, et G. Bruno (2011). Herisson : a weblab-based platform for assessment and experimentation of information processing technologies. In *ICSSEA*.

Cellier, P. et T. Charnois (2010). Fouille de données séquentielle d'itemsets pour l'apprentissage de patrons linguistiques. In *Traitement Automatique des Langues Naturelles (short paper)*.

Charnois, T., M. Plantevit, C. Rigotti, et B. Cremilleux (2009). Fouille de données séquentielles pour l'extraction d'information dans les textes. *Revue Traitement Automatique des Langues (TAL) 50*(3), 59–87.

Davide Ceolin, P. G. (2010). Calculating the trust of event descriptions using provenance.

Davidson, D. (2001). *Essays on Actions and Events*. Oxford University Press.

Dedek, J., A. Eckhardt, L. Galambos, et P. Vojtás (2008). Discussion on uncertainty ontology for annotation and reasoning. In *URSW*.

Giroux, P., S. Brunessaux, S. Brunessaux, J. Doucy, G. Dupont, B. Grilheres, Y. Mombrun, et A. Saval (2008). Weblab : An integration infrastructure to ease the development of multimedia processing applications. *ICSSEA*.

Hobbs, J. R. et E. Riloff (2010). Information extraction. In *Handbook of Natural Language Processing, Second Edition*. Boca Raton, FL : CRC Press, Taylor and Francis Group.

Ireson, N. et F. Ciravegna (2005). Pascal challenge the evaluation of machine learning for information extraction. In *Proceedings of Dagstuhl Seminar Machine Learning for the Semantic Web*.

Krieg-Planque, A. (2009). *A propos des noms propres d'événement*, Volume 11, pp. 77–90. Les carnets du Cediscor.

Laskey, K. J. et K. B. Laskey (2008). Uncertainty Reasoning for the World Wide Web : Report on the URW3-XG Incubator Group. In *URSW'08*.

Nadeau, D. (2007). *Semi-Supervised Named Entity Recognition : Learning to Recognize 100 Entity Types with Little Supervision*. Ph. D. thesis.

Nadeau, D. et S. Sekine (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes 30*(1), 3–26. Publisher : John Benjamins Publishing Company.

Naughton, M., N. Kushmerick, et J. Carthy (2006). Event extraction from heterogeneous news sources. In *Proc. Workshop Event Extraction and Synthesis*. American Nat. Conf. Artificial Intelligence.

Neveu, E. et L. Quéré (1996). *Le temps de l'événement I*, Chapter Présentation, pp. 7–21. Number 75 in Réseaux. CNET.

NIST (2005). *The ACE 2005 (ACE05) Evaluation Plan*.

Pustejovsky, J., J. M. Castaño, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, et R. D. R. (2003). TimeML : Robust specification of event and temporal expressions in text. In *New Directions in Question Answering'03*, pp. 28–34.

Ricœur, P. (1983). *Temps et récit 1. L´intrigue et le récit historique*, Volume 227 of *Points : Essais*. Paris : Ed. du Seuil.

Saval, A., M. Bouzid, et S. Brunessaux (2009). A semantic extension for event modelisation. *21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2009)*.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Serrano, L., B. Grilheres, M. Bouzid, et T. Charnois (2011). Extraction de connaissances pour le renseignement en sources ouvertes. In *Atelier Sources Ouvertes et Services (SOS'2011) en conjonction avec la conférence internationale francophone EGC'2011*.

Soderland, S., O. Etzioni, T. Shaked, et D. S. Weld (2004). The use of web-based statistics to validate information extraction. In *Proceedings of AAAI 2004 Workshop on Adaptive Text Extraction and Mining (ATEM'04)*.

Van De Velde, D. (2006). *Grammaire des événements*. Presses Universitaires du Septentrion.

Van Hage, W. R., V. Malaisé, R. H. Segers, L. Hollink, et G. Schreiber (2011). Design and use of the simple event model (sem). *Web Semantics : Science, Services and Agents on the World Wide Web 9*(2).

Van Keulen, M. et M. B. Habib (2011). Handling uncertainty in information extraction. In *URSW*, Volume 778 of *CEUR Workshop Proceedings*, pp. 109–112.

Summary

Nowadays staggering increasing of the information available (specially on the Web) makes the discovery of relevant information (professionnally or personnaly speaking) more and more complex and time-consuming. Tackling this issue, the IPCC team develops the WebLab, a media mining platform aiming at integrating various tools to enhance knowledge discovery and decision making. In this paper, we present an automatic system to extract events dedicated to open source intelligence and based on the combination of multiple information extraction approaches. Firstly, we propose a quick state-of-the-art of the main scientific axes our researches tackle. Then, we describe more precisely our model and all the techniques we use. To conclude, we summarize what we have done so far and the future work we plan to do.

Modelling and Quering Context-Aware Personal Information Spaces

Rania Khefifi[*,**], Pascal Poizat[*,***], Fatiha Saïs[*,**]

[*]LRI, CNRS et Université Paris Sud,
[**]INRIA Saclay Île-de-France,
[***]Université d'Evry Val d'Essonne,
{rania.khefifi, pascal.poizat, fatiha.sais}@lri.fr

Résumé. The increasingly big amount of personal information (e.g., mails, contacts, appointments) managed by a user is characterized by their heterogeneity, their dispersion and their redundancy. The general goal of this work consists in designing a system, which allows providing the end-users personal data access with services that are relevant to his/her needs, and to access personal data both by mobile devices (smartphone) and Internet-connected Personal Computers. More specifically, we focus here on the problem of defining a common meta-model for a flexible and homogeneous personal information management. The meta-model that we propose allows users creating personal information and organizing them according to different points of view (ontologies) and different contexts. Contextual queries are defined to allow users to retrieve their personal information depending on context valuation. The semantic Web languages (OWL, RDF and SPARQL) are used to implement the approach.

1 Introduction

Personal information management (PIM) is the practice and analysis of the activities performed by people to acquire, organize, maintain, and retrieve information for everyday use. PIM is a growing area of interest beacause, everyone is looking for better use of our limited personal resources of time, money and energy. Several research on the topic is being done in different disciplines, including human-computer interaction, database management, information retrieval, and artificial intelligence.

The increasingly big amount of personal information (e.g., mails, contacts, appointments) managed by a user is characterized by its heterogeneity, its dispersion and its redundancy. Processing this amount of information is difficult and not obvious. Indeed, personal information is often managed by very specific and autonomous tools which deal with specific kinds of user data (***e.g.*** $iCal$ to manage appointments and meetings, $thunderbird$ and $outlook$ to manage e-mails). Despite the existence of various applications, there are no connections and no links between the pieces of personal information managed by the user applications. Indeed, many data present in one application can relate and concern other applications which leads to the redundancy of personal information. For example, someone receiving an e-mail asking to organize a meeting at a given date and in a given place with a colleague, would have if using

traditional systems, to add manually the main information concerning this meeting : add an event to a calendar tool, make a room reservation and possibly add the colleague contact to an address book. Some new mail tools can recognize dates and e-mail addresses and offer the user the ability to add them to the calendar and/or to the address book. This semi-automatic task helps users to create links between tools.

The need for automating users' tasks returns to the fact that the capacities of human mind are limited and the wish to extend human memory in order to reduce human effort and to improve the response time is increasing. Vannevar Bush, in Bush (1945), points to the fact that the human brain works by association. That is to say, he tries to combine objects together, based on the fact that there is no single object, it interacts at least with another object. The first system for personal information management MEMEX (Memory Extender) developed by Vannevar Buch in 1945 Bush (1945), allows automatic creation of references between library objects and establishing links between pieces of information. The two systems developed MyLifeBits Gemmell et al. (2002) and SEMEX Dong et Halevy (2005) recently are based on the MEMEX principles. MyLifeBits Gemmell et al. (2002) is a Personal Information Management System (PIMS) which extends MEMEX to allow media annotation, collection creation and multiple visualizations to increase the understanding level of data. SEMEX Dong et Halevy (2005) is a PIMS which offers users a flexible framework with two main goals : browsing personal information by semantically meaningful associations and enrich personal information space to increase users' productivity. This system uses data annotation, similarity computation from ontology-based framework and provides a unified interface for data access. Lifestreams of Freeman et Fertig (1995) presents several principles like storage should be transparent and personal data should be accessible anywhere. Also, it provides a unified framework that subsumes many separate desktop applications to handled personal information communication and stores every document created or received in a chronological order.

PIM users may create objects with the same name by referring to different entities and may create different objects with different names by referring them to the same entity. This kind of behavior may conduct to the semantic heterogeneity of personal information which makes the task of integration and interoperability between personal information more difficult. This is the reason why in this work we have developed an ontology based approach to deal with semantic heterogeneity as much as possible. COPIM system that we present in this paper, offers users the ability to organize their personal information in a way to ensure their semantic consistency, their reusability and their transparent querying. It is based on a meta-model allowing the user to create its own personal data model using different existing domain ontologies to describe and organize its personal information. Furthermore, user' personal information can be described according to different points of view thanks to the semantic Web technologies (***e.g.*** OWL, RDF, SPARQL) allowing a richer descriptions of personal data. The user needs can be expressed using conjunctive queries Chekuri et Rajaraman (1997) expressed in SPARQL.

The term of *context-aware* is introduced for the first time by Schilty and Theimer Schilit et Theimer (1994). The authors consider that a context-aware computer is able to discover and react to changes in the environment they are situated in. Context is referred to as location, nearby people identity, object identity, object change identity and object location. However, up to our knowledge there is no PIM system which is context-aware and ontology-based. In order to take into account the contextual information, in COPIM system we have proposed : (i) reifying the personal information descriptions by their geographical context and (ii) expressing

queries using a contextual parameter which can be evaluated in an equality operator or in a semantic similarity to obtain close answers. COPIM system have been implemented and evaluated on artificial personal information and gives very promising results.

The paper is organized as follows : a new meta model for personal information is given in section 2. We present, in section 3, a contextual querying for personal information system. In section 4, we give an illustrative example for the contextual querying and some first experimental results. We conclude in section 5 and give some future work.

2 PI2M : Personal Information Meta-Model

Each user has his own areas of interest and wants to manage his/her data in a way that satisfy the best his/her needs. There are several tools for managing personal data (***e.g.*** Outlook, iCal) in an independent way, which leads to the distribution, heterogeneity and the redundancy of personal data across the different tools. Using these tools, it is difficult for the user to have a homogeneous view on his personal information. The Personal Information Management (PIM) system that we plan to develop in the context of PIMI project [1] should allow the user having a uniform view of his/her personal information and to create and organize his personal data in the way that is most appropriate to his needs and habits.

In this section, we will present the meta-model which is used to express and formalize the personal information storing and management (see PI2M in Figure 1). It allows representing the personal information of a given user according to three levels : (i) description and structure, (ii) context validity and (iii) access policies.

In the following, we will describe the components of PI2M relative to description, structure and access policies. The access policies are not presented in this work.

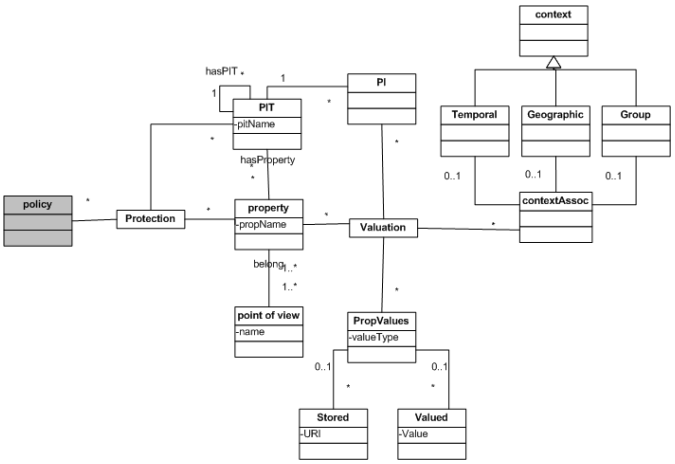


FIG. 1: PI2M : Personal Information Meta-Model in UML formalism.

1. This work is supported by project PIMI - Personal Information Management through Internet of the French National Agency for Research

2.1 Personal data description and structure

In a classical desktop, the user often describes his/here data and its structure using different names and different structuring mechanisms. In order to limit the semantic heterogeneity, we have chosen to restrict the user to describe data according to fixed vocabularies (concepts and properties). These vocabularies are represented in different ontologies which can be designed by the user itself or collected from the Web using search engines like `http://swoogle.umbc.edu/`.
As the same information can be represented according to different points-of-view, the meta model that we propose allows representing personal information in accordance with different points of view represented in different ontologies. For example, we may have information about an individual which is both a colleague, and, accordingly a friend and we might describe it differently depending on whether one takes the professional point of view or the personal point of view. Furthermore, to be able to exploit the different vocabularies defined in the different points-of-view describing the personal data, we define a Personal Information Type (PIT) in function of a set of concepts declared in the considered point-of-view ontologies. To this purpose, we consider that the ontologies are expressed using OWL (Ontology Web Language) `http://www.w3.org/2004/OWL/`. In the following, we will denote these ontologies as *PV-ontologies*.

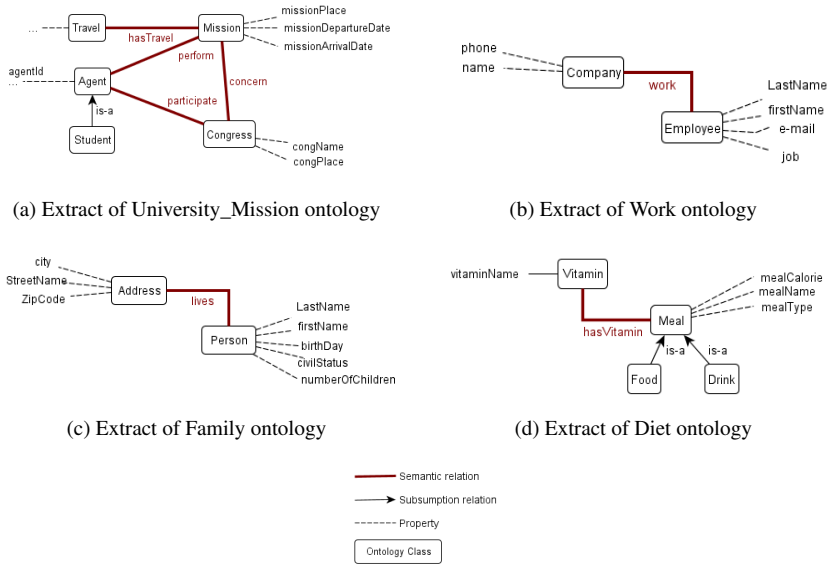
2.1.1 Point of View

A point of view is an ontology which can be either created by the user or collected from the Internet. It is composed of a set of classes (concepts) representing a specific domain. The classes are organized through the subsumption relation. Each class is defined by a set of properties.
We allow declaring disjunction constraints between classes of a same point-of-view as well as between classes of different points-of-view.
We can use ontology alignment tools, like TaxoMap Hamdi et al. (2009) and COMA++ Aumueller et al. (2005), to compute all the mappings of equivalence and subsumption between either the classes and the properties of the different points-of-view. We denote $\mathcal{M}_C$ the set of class mappings and $\mathcal{M}_P$ the set of property mappings.
In Figure 2a we show an extract of ontology for professional traveling and in Figure 2d, we show an extract of food preferences ontology. For example in first ontology, $Mission$, $Agent$, $Travel$, etc., represent classes, $participate$, $concern$, etc., represent object properties and $missionPlace$, $missionDepartureDate$, etc., represent data properties of the class $Mission$

2.1.2 Personal Information Type.

A *Personal Information Type* (PIT) is defined in function of a set of classes declared in the considered ontologies. A set of properties $\mathcal{P}$ is associated to each PIT. $\mathcal{P}$ is defined as an indexed (over ontology classes) set of properties. The PIT data model is represented as an OWL ontology.

Definition 1 ***–Personal Information Type (PIT).*** *A PIT is defined by a tuple* $(Name, \mathcal{C}, \mathcal{P}, M_p)$ *where :*



(a) Extract of University_Mission ontology

(b) Extract of Work ontology

(c) Extract of Family ontology

(d) Extract of Diet ontology

FIG. 2: Example of points of view.

- $Name$ *is the name of the PIT,*
- $\mathcal{C}$ *is the set of classes used to define the PIT. These classes should not be pairwise disjoint, w.r.t. the disjunction constraints that are declared between classes of the different ontologies.*
- $\mathcal{P}$ *an indexed (over ontology classes) set of properties.*
- $M_p = \{(c_1.p_1 \equiv c_2.p_2), (c_1.p_1 \preceq c_3.p_3), \ldots, (c_k.p_j \equiv c_n.p_m.)\}$ *is the set of mappings between the properties that are involved in the definition of the PIT. These mapping correspond to a subset of the whole set of mappings* $\mathcal{M}_P$.

Example 1 *Let there two ontologies : a* work *ontology* O_1 *and a* family *ontology* O_2. *The* $Person$ *concept is described by the set of properties* {lastName, firstName, e-mail, job} *and by the set of properties* {lastName, firstName, birthDate, civilStatus,numberOfChildren} *in the ontology* O_2. *In spite of the existence of two different descriptions in ontologies* O_1 *and* O_2, *the user can choose to create a new concept (**e.g.*** $Friend$) *represented by a new PIT having as name* $Friend$. *This new concept can be declared as having the same semantics than the concept* $Person$ *in* O_1 *and in* O_2 *thanks to a set of mappings* $\mathcal{M} = \{(Friend \equiv O_1 : Employee), (Friend \equiv O_2 : Person)\}$.
The new PIT $friend$ *can be described by a subset of properties derived from the two sets of properties of the concepts* $O_1 : Employee$ *and* $O_2 : person$, *(**e.g.** describe* Friend *by* $name$, $firstName$ *and* $e-mail$ *derived from* $O_1 : Employee$, *and* $civilStatus$ *derived from* $O_2 : Person$).

The meta-model that we have developed allows the user to declare subsumption relations between different PITs. In case of two duplicate properties, if there is a mapping of equivalence/subsumption between them, then one of them is arbitrarily chosen.

Example 2 *Let consider two PITs that are already created in the system : a* $Colleague$ *described by the set of properties* (lastName, firstName,e-mail, officeNumber) *, and a PIT* $friend$ *described by the set of properties* (lastName, firstName, birthDate, phoneNumber)*. The user may want to create a new PIT* $Colleague - Friend$ *which can be declared as more specific than the two PITs* $Friend$ *and* $Colleague$*. The set of properties of the PITs* $Friend$ *and* $Colleague$ *should be included in the set of properties of the new* $Colleague - Friend$ *PIT.*

In Figure 3, we show an example of PITs that instantiates the meta model given in Figure 1.

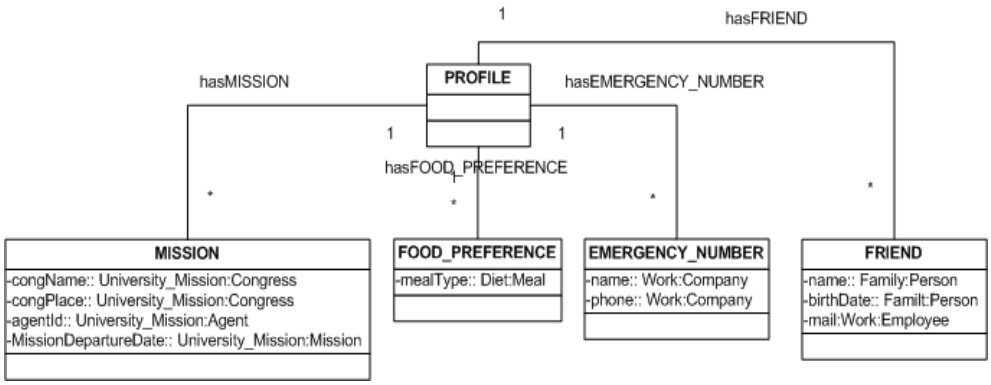


FIG. 3: Example of PITs (UML class diagram).

2.1.3 Personal Information.

Once the different PITs needed by the user are defined, the personal information (***e.g.*** contacts, mails, publications, appointments) can be attached to these PITs and then instantiate the different properties describing these PITs.

Definition 2 ***–Personal Information(PI)*** *is an instance of a PIT which refers to a user information. Each PI is also represented by a subset of the property instances which describe its corresponding PIT. We consider a PI as defined by a tuple* $(T, Reference, \mathcal{P}_{inst})$*, where :*

- T*, represents the PIT that is instantiated by the PI.*
- $Reference$*, is an identifier used to represent the Personal Information in the system.*
- $\mathcal{P}_{inst}$*, is a set of pairs* $(property, value)$ *which instantiates the PIT T properties.*

In COPIM we define two kinds of property values, to support PI management locally or over the Internet (***e.g.*** on a smartphone) : $valued$ and $stored$.

Valued : represents properties with atomic values that are stored in a local RDF data repository. These properties correspond to those frequently used by the user. Indeed, giving the

capability to store property values in RDF files, reduces the response time of the system and reduces memory access.

Stored : represents properties that have digitized values which are either : (*i*) contained in a static document that cannot be parsed by the system, from where no information can be automatically extracted (***e.g.*** pdf, jpeg, jpg,...). Hence, we keep the URI to access the document and show it to the user, or (*ii*) represented in a semi-structured document (***e.g.*** XML, ...) for which a query (***e.g.*** XPath) is declared to return the property value(s).

The personal information are represented in RDF as a set of triples. A PI with a reference $pi2m : ref$ is represented in the form of an RDF triple as follows : $<$ pi2m :ref onto :p value $>$

We note that the prefix $onto$ expresses the URI of the ontology where the property p comes from. It can also be represented in the form of RDF facts as follows : *onto :p(pi2m :ref, value)*.

In Figure 4, we show an example of PIs that instantiate the user PIT data model shown in Figure 3. For clarity reasons, we present the PIs using a UML object diagram, but they are represented in our COPIM system using RDF triples and stored in an RDF repository.

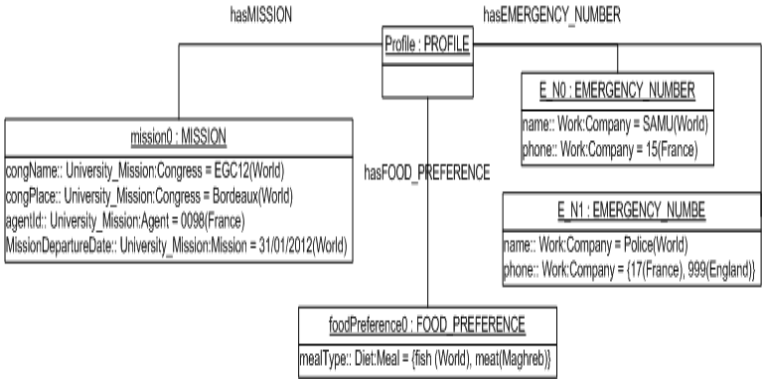


FIG. 4: Example of PI.

2.2 Personal Information Context

Personal Information can be sensitive to the context (geographical, temporal and social) ***e.g.*** a person leaving partly in France and in the UK may have two Social welfare numbers (SSN).

In the following, we will focus our study on the geographical context of personal information. We represent the geographical context of personal information in a geographical ontology expressed in OWL. We consider that the geographical ontology is composed of a set of classes that are organized in a hierarchy thanks to the *partOf* relation. We assume that one class cannot be *partOf* more than one class. We consider also the declaration of disjointness constraints and equivalence constraints between the geographical context classes.

In Figure 5, we show an example of a geographical ontology. There, one can see that France is partOf Europe. Disjointness is not represented in this figure for clarity purposes : each two classes not related by the transitive closure of partOf are disjoint (***e.g.*** $Paris$ and $London$)

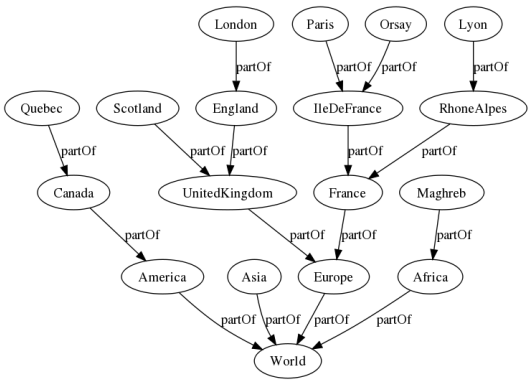


FIG. 5: Example of geographical context ontology.

2.3 Context-based Personal Information Representation

In subsection 2.1.3, we have considered that a property instance of a Personal Information is represented using an RDF fact, *onto :p(pi2m :ref, value)*

In order to extend this representation for expressing the context c where the value is valid we use a predicate with three arguments in the form of, *onto :p(pi2m :ref, value, c)* .

We restrict the representation of context information for only the *owl :DataProperty* values, because we consider that context information in case of *owl :ObjectProperty* is not meaningful.

In case of N-Triple RDF representation, the context information of a property value is expressed thanks to the reification mechanism (for the reification semantics, see `http://www.w3.org/TR/rdf-mt/#Reif`) which allows one to add new elements to RDF declarations dec, like data author, creation date and so on.

In our case, the reification consists here in adding to the triples of the form $<$ pi2m :ref onto :p value $>$ the geographical context property $contextG$ that has as domain a declaration dec and as range a class c from the geographical context ontology GO.

<dec rdf :type rdf :Statement > .
<dec rdf :subject pi2m :ref > .
<dec rdf :predicate onto :p > .
<dec rdf :object value > .
<dec contextG GO :c>

Example 3 *For example the declarations :*

onto :SSN(pi2m :person33,“29979352312”, GO :France)
onto :SSN(pi2m :person33,“44751032565”, GO :England)

declare the SSN number of person 33 in France and in England. These reified personal information are represented in plain RDF as follows :

Thanks to this reified representation, contextual personal information can be queried using the SPARQL language without any extension.

<dec1 rdf :type rdf :Statement > .	<dec2 rdf :type rdf :Statement > .
<dec1 rdf :subject pi2m :person33> .	<dec2 rdf :subject pi2m :person33> .
<dec1 rdf :predicate onto :SSN > .	<dec2 rdf :predicate onto :SSN > .
<dec1 rdf :object "29979352312" > .	<dec2 rdf :object "44751032565" > .
<dec1 contextG GO :France> .	<dec2 contextG GO :England>

3 Contextual Personal information Query Language

The interaction with the system is crucial for the user. It is important to provide the user a query interface with his PIMS in a transparent way. In this work, we will focus first on the development of the query engine over the personal information system by taking into account the context.
We allow the user expressing several kinds of contextual-queries :

1. *Strict-context query* : a query where the user asks for answers having the same context than the one given in the query ;
2. *Disjoint-context query* : a query where the user asks for answers having a context that is distinct from the one given in the query ;
3. *Similar-context query* : a query where the user asks for answers having a context that is similar to the one given in the query.

3.1 Conjunctive contextual queries

A queries are easily represented in the form of conjunctive queries(Chekuri et Rajaraman (1997)) with function symbols.
A strict-context query is expressed by the following conjunctive query : $Q(X)^{=c} : p(Y, X, c)$
A disjoint-context query is expressed by the following conjunctive query :
$Q(X)^{\neg c} : p(Y, X, Z) \wedge (Z \neq c)$
A similar-context query is expressed by the following conjunctive query :
$Q(X)^{\sim c} : p(Y, X, Z) \wedge contextSim(Z, c, Th)$, with $contextSim$ is a function that computes the semantic similarity between two contexts (Z, c) and returns $true$ if the similarity score is greater than a threshold Th and returns $false$ otherwise. The similarity computation is detailed in the subsection bellow.

We note that more complex queries where several predicates are involved can be expressed.

3.2 Similarity computation between contexts

In case where there are no answers for the user query, because of the absence of user context in the personal data, the user query is rewritten into a set of queries where the user context c is replaced by all the contexts c' that are similar to. To compute this similarity between two contexts, we use the principle of Wu and Palmer Wu et Palmer (1994) which is based on the distance of two contexts in the hierarchy of the context ontology.
Given an ontology $\mathcal{O}$ composed of a set of nodes and a root node R. A and B represent two ontology elements of which the we aim to compute the similarity. The similarity score is

computed as function of the distances : (i) $N1$ respectively $N2$ which represents the distance of A from their LCS (Least Common Subsumer), respectively, the distance of B from their LCS, and (ii) the distance N which computes the distance between the LCS of A and B and the hierarchy root R. The similarity measure of Wu and Palmer is computed by the following formula : $Sim_{WP}(A,B) = \frac{2\times N}{N1+N2+2\times N}$.

The similarity score between $\delta(A,B)$ between two classes A and B is obtained as follows :

$$\delta(A,B) = \begin{cases} 0 & \forall A \not\equiv B \\ 1 & \forall A \equiv B \ \vee \ (B \leq A) \vee \ (A \leq B) \\ Sim_{WP}(A,B) & otherwise \end{cases}$$

4 Illustrative example and first experiment results

To evaluate the efficiency of our approach, we have implemented a prototype that can be used by a novice user. This prototype allows the user to create his specific PITs and then to enrich the RDF repository if PIs using contextual data. In our framework, the user contextual queries are formulated in SPARQL and their evaluation uses, in case of no answers similarity computation between different contexts.

COPIM system has been tested on artificial data describing personal information on people going for professional traveling to attend research congresses. The considered dataset contains personal information like $missionDepartureDate$, $missionArrivalDate$ and $emergencyNumber$ (see Figure 4 for more details).

4.1 Illustrative example

To illustrate how contextual queries are processed in COPIM, let us consider a user query asking for emergency numbers and the following personal information. The values in parentheses represent the geographical context expressing the validity context of the values. For clarity reasons, we have chosen to use a relational representation of personal data.

	Name	**Phone**
1	SAMU (World)	15 (France)
2	Police (World)	17 (France)
3	Police (World)	999 (England)
4	EuropeanEmergency (World)	112 (Europe)

TAB. 1: Example of emergency numbers.

Let Q1, Q2, Q3, Q4 and Q5 a set of conjunctive queries using geographical context for Q2, Q3, Q4 and Q5 and without geographical context for Q1.

1. Query without context :
   $Q1(name, phone)$ = $Emergency\_Number(X) \wedge name(X,Y) \wedge phone(X,Z)$
2. A strict-context query where the context exists in personal information data.
   $Q2(name, phone)^{=France}$ = $Emergency\_Number(X) \wedge name(X,Y) \wedge phone(X,Z) \wedge context(Z, France)$

3. A strict-context query where the context does not exist in personal information data. $Q3(name, phone)^{=Paris}$ = $Emergency\_Number(X) \wedge name(X, Y) \wedge phone(X, Z) \wedge context(Z, France)$
4. A similar-context query where the query context exist in personal information data. $Q4(Y, Z, W)^{\sim France}$ = $Emergency\_Number(X) \wedge name(X, Y) \wedge phone(X, Z) \wedge context(Z, W) \wedge contextSim(W, France, 0.5)$
5. A similar-context query where the context asked in the query does not exist in personal information data. $Q5(Y, Z, W)^{\sim UnitedKingdom}$ = $Emergency\_Number(X) \wedge name(X, Y) \wedge phone(X, Z) \wedge context(Z, W) \wedge contextSim(W, UnitedKingdom, 0.5)$

After the query evaluation on personal data of table 1, we obtain the following answers by using the geographical context (Q2, Q3, Q4 and Q5) and without geographical context (Q1) in the query.

1. $Q1(n)$= { (SAMU,15), (Police,17), (Police,999), (EuropeanEmergency,112) }
2. $Q2$ = { (SAMU,15), (Police,17) }
3. $Q3$ = $\emptyset$
4. $Q4$ = {(SAMU, (15, France, 1)), (Police, (17, France, 1)), (EuropeanEmergency, (112, Europe, 1))}
5. $Q5$ = {(EuropeanEmergency, (112, Europe, 1)), (Police,(999, England, 0.8))}

The evaluation of Q1 needs scanning the whole RDF repository of PIs and returns all the stored information, even those that can be irrelevant for the user. In order to return more relevant and more accurate answers to the user the contextual queries are needed. The query Q2 presents the case where the user is interested in personal data that are valid in the user context expressed in the query (***i.e.*** returns answers containing the same context as the user one). For Q3, there are no answers, because of the non existence of personal data that is defined for $Paris$. The queries Q4 and Q5 are examples of queries where the user looks for personal data that are valid in his context as well as in those that are valid in similar contexts.

4.2 First experiment results

We have performed several tests to evaluate the execution time of the queries in the cases of strict-context query and similar-context query (see Table 2). In the case of similar-context query, we distinguish two cases : (i) *Similar-context*$_{qt}$, where the similarity between contexts is computed in query time and (ii) *Similar-context*$_{ol}$ where the similarity between contexts is computed off-line before the query evaluation.

As shown in table 2, the difference between the execution time of *Strict-context* and the *Similar-context*$_q t$ is negligible. However, the difference between execution time of *Similar-context*$_{qt}$ and the *Similar-context*$_{ol}$ is important which is due to the similarity computation time needed in case of *Similar-context*$_q t$.

5 Conclusion

In this article, we have presented COPIM a personal information management system which is context-aware and ontology-based. Our meta model is defined to deal with several

	France	**UnitedKingdom**	**Paris**	**Europe**
Strict-context	19	18	16	16
Similar-context$_{qt}$	682	540	150	789
Similar-context$_{ol}$	67	59	25	55
number of similar context	8	6	5	12

TAB. 2: Evaluation of query execution time (ms) according to the context and the query type

problems : (*i*) heterogeneity, (*ii*) redundancy and (*iii*) dispersion. It exploits ontologies to deal with semantic heterogeneity and to take benefits from the intensive work on knowledge representation and reasoning in this domain. The proposed contextual querying allows the system to give the user more relevant answers.

In the future, we plan to continue the work over personal information querying, and precisely using simple query to perform basic user workflows. Furthermore, we will study how to integrate the temporal and social contexts in COPIM.

Références

Aumueller, D., H. Do, S. Massmann, et E. Rahm (2005). Schema and ontology matching with coma++. In *ACM SIGMOD international conference on Management of data*, pp. 906–908.

Bush, V. (1945). As we may think. *The Atlantic Monthly 176*, 101–108.

Chekuri, C. et A. Rajaraman (1997). Conjunctive query containment revisited. In *Database Theory-ICDT'97*, pp. 56–70.

Dong, X. et A. Halevy (2005). A platform for personal information management and integration. In *VLDB-PhD Workshop*, pp. 119–130.

Freeman, E. et S. Fertig (1995). Lifestreams: Organizing your electronic life. In *AI Applications in Knowledge Navigation and Retrieval*.

Gemmell, J., G. Bell, R. Lueder, S. Drucker, et C. Wong (2002). Mylifebits: fulfilling the memex vision. In *ACM international conference on Multimedia*, pp. 235–238.

Hamdi, F., B. Safar, N. Niraula, et C. Reynaud (2009). Taxomap in the oaei 2009 alignment contest. In *OM*.

Schilit, B. et M. Theimer (1994). Disseminating active map information to mobile hosts. *Network, IEEE 8*, 22–32.

Wu, Z. et M. Palmer (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp. 133–138.

Summary