Actes des 26es journées francophones d'Ingénierie des Connaissances

Marie-Hélène Abel

▶ **To cite this version:**

Marie-Hélène Abel. Actes des 26es journées francophones d'Ingénierie des Connaissances. Plate-Forme Intelligence Artificielle, Association Française pour l'Intelligence Artificielle, 2015. hal-04596480

HAL Id: hal-04596480

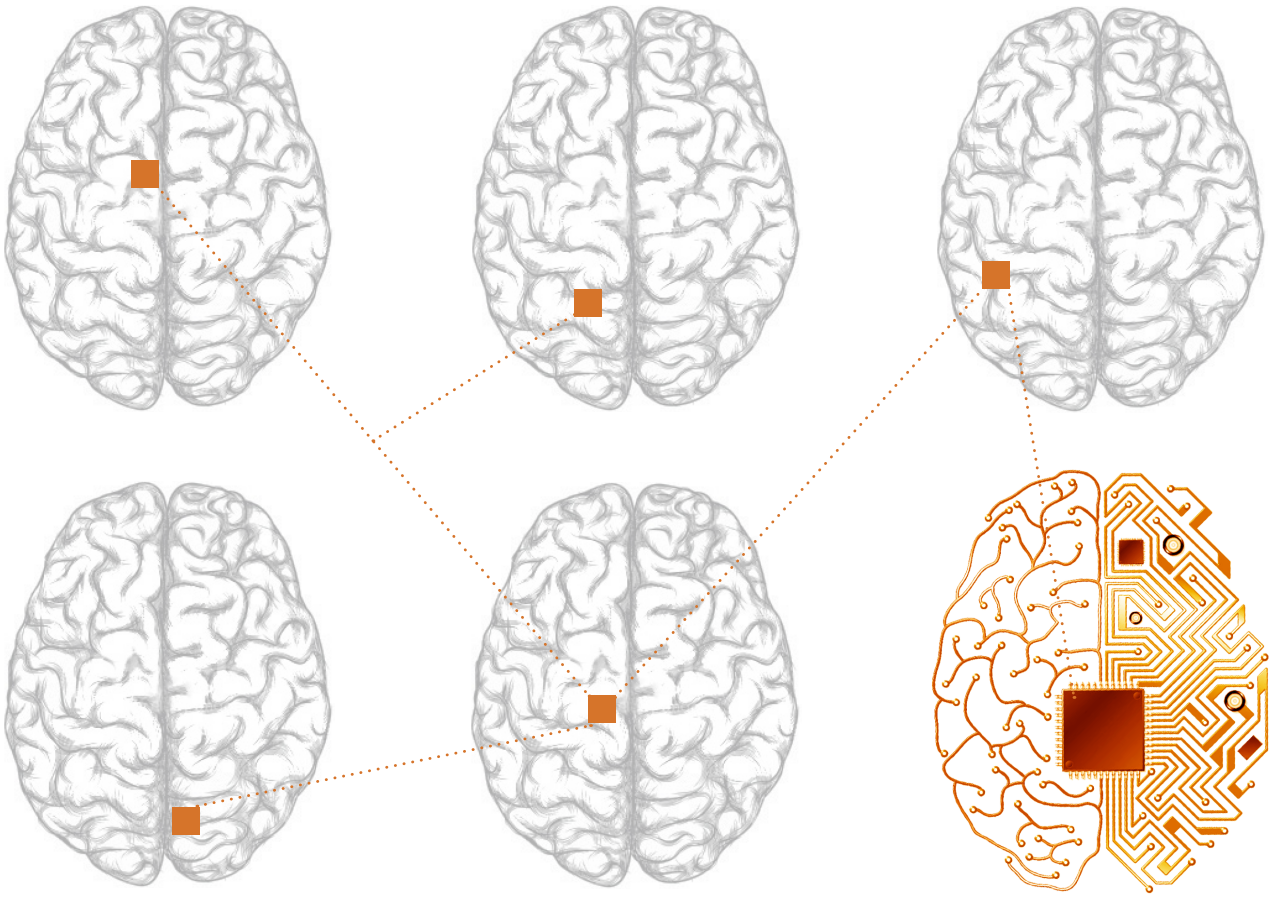
https://ut3-toulouseinp.hal.science/hal-04596480v1

Submitted on 31 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License



PFIA 2015

Plate-forme Intelligence Artificielle

Rennes

Actes IC

Président CP : Marie-Hélène Abel

Inria
INVENTEURS DU MONDE NUMÉRIQUE

AfIA
Association française
pour l'Intelligence Artificielle

26es Journées Francophones d'Ingénierie des Connaissances

Vers le traitement de la masse des données disponibles sur le Web

Présentation de la conférence

Organisée chaque année depuis 1997 sous l'égide du GRACQ (Groupe de Recherche en Acquisition des Connaissances) puis du collège IC de l'AFIA, les journées francophones d'Ingénierie des Connaissances (IC) constituent un lieu d'échanges et de réflexions de la communauté francophone académique et industrielle sur les concepts, méthodes et techniques permettant de modéliser et d'acquérir les connaissances dans des domaines peu ou pas formalisés.

Ces concepts, méthodes et techniques progressent au rythme de l'évolution des usages et des technologies. Les Technologies de l'Information et de la Communication, le web social, le web des données ont ainsi engendré des mutations des pratiques individuelles et collectives. L'arrivée de l'internet des objets a permis l'émergence d'un web des objets visant la communication entre objets connectés et leur lien avec des acteurs humains. Aujourd'hui, plus que jamais, de nombreuses données, informations, sources de connaissances sont produites. Il est donc naturel de s'interroger sur leur exploitation (représentation, interprétation, gestion, diffusion, partage, visualisation, etc.) et sur les outils, méthodes, modèles, standards permettant leurs traitements.

Les 26es journées francophones d'Ingénierie des Connaissances se déroulent dans le cadre de la plate-forme de l'Intelligence Artificielle PFIA 2015.

Le comité de programme de cette édition 2015 a permis d'accueillir des représentants de 11 pays francophones. Cette édition a recueilli cinquante intentions de soumission pour finalement 46 articles longs et 2 articles courts déposés provenant de 8 pays. Parmi ces soumissions, 19 articles longs ont été retenus, 6 ont été retenus en papiers courts. 6 démonstrations ont également été soumises et retenues pour être présentées dans une session partagée par les conférences hébergées par la plate-forme PFIA2015. Les thèmes abordés ont été variés et recouvrent bien l'ensemble des préoccupations de la communauté concernant l'exploitation de données, d'informations, de sources de connaissances et la définition des outils, des méthodes, et des modèles visant le traitement de connaissances. A l'issue de cette édition, une demande d'extension d'article sera proposée à une sélection des meilleures contributions de IC2014 et IC2015 pour soumission à un numéro spécial de la Revue d'Intelligence Artificielle.

Je remercie le bureau du collège de l'AFIA de m'avoir offert l'opportunité de présider ces 26èmes journées. Je remercie chaleureusement Jean-Luc Vuillaume (Altran) d'avoir accepté d'être notre conférencier invité et de partager avec la communauté son expérience de consultant sénior en knowledge management. Je remercie également les différents contributeurs qui font de ces journées un évènement majeur pour la communauté d'ingénierie des connaissances : auteurs, membres du comité de programme, participants. Je remercie enfin Nathalie Aussenac-Gilles, Jean Charlet et Catherine Faron Zucker pour leurs nombreux conseils qui m'ont permis de mener au mieux ma mission, ainsi que les membres du comité d'organisation de la plate-forme PFIA2015 pour leur disponibilité.

Marie-Hélène Abel
Présidente du comité de programme de IC2015

Conférencier Invité : Jean-Luc Vuillaume (Altran France)

Biographie

Jean-Luc Vuillaume est consultant et formateur chez Altran/CIS. En partant d'une double compétence « Organisation d'entreprise et Gestion de la connaissance », J.-L. Vuillaume a acquis une très bonne expérience en modélisation des connaissances et des compétences dans le cadre des stratégies de gestion des connaissances (Knowledge Management) et de conception pédagogique. J.-L. Vuillaume s'intéresse en particulier à la conception de dispositifs pédagogiques pour la formation professionnelle continue et à l'élaboration de modélisation de processus.

Pérennisation de la connaissance via un Electronic Performance Support System (EPSS)

La société MBDA (Groupe EADS) fait partie du secteur de l'armement et emploie 10.000 personnes. Il y a 6 ans les questions de la pérennisation et de la capitalisation de la connaissance liée aux applications informatique ont été posées aux Responsable des Système d'Informatique et de la formation. Le retour d'expérience que je vous propose de partager correspond à la mise en place d'une base de connaissances pendant les 6 années qui ont suivi.

Je commencerai par la présentation du contexte et l'explicitation des questions posées après 10 ans de déploiement informatique par les salariés après une étude de satisfaction. J'aborderai la réflexion que nous avons eue sur la notion de connaissance, d'apprentissage et de compétence. Celle-ci nous a amené à la mise en place d'un EPSS (Electronic Performance Support System) que je définirai ainsi que l'organisation informatique, organisationnelle et de production que nous avons mise en place. Je ferai ensuite un bilan sur la bibliothèque de connaissance applicative métier créée pour finir par les perspectives envisageables dans les prochaines années.

Comité de Programme

Présidente du comité de programme

Marie-Hélène Abel, Université de Technologie de Compiègne, UMR CNRS 7253 Heudiasyc

Membres du comité de programme

— Xavier Aimé, INSERM, France
— Yamine Ait Ameur, INPT-ENSEEIHT, Toulouse - IRIT, France
— Patrick Albert, IBM France
— Florence Amardeilh, Mondeca, France
— Nathalie Aussenac-Gilles, CNRS - IRIT, France
— Bruno Bachimont, Université de Technologie de Compiègne - Heudiasyc, France
— Jean-Paul Barthès, Université de Technologie de Compiègne - Heudiasyc, France
— Sadok Ben Yahia, Faculté des Sciences de Tunis, Tunisie
— Aurélien Bénel, Université de Technologie de Troyes - ICD/Tech-CICO, France
— Nacéra Bennacer, SUPELEC - E3S, France
— Bertrand Braunschweig, INRIA Rennes-Bretagne Atlantique, France
— Patrice Buche, INRA- IAT/LIRMM, France
— Elena Cabrio, INRIA Sophia Antipolis, France
— Jean-Pierre Cahier, Université de Technologie de Troyes - ICD/Tech-CICO, France
— Sylvie Calabretto, Institut National des Sciences Appliquées de Lyon - LIRIS, France
— Pierre-Antoine Champin, Université Claude Bernard Lyon 1 - LIRIS, France
— Jean Charlet, Université Pierre et Marie Curie - INSERM U1142 - LIMICS, France
— Olivier Corby, INRIA Sophia Antipolis, France
— Amélie Cordier, Université Claude Bernard Lyon 1 – LIRIS, France
— Michel Crampes, Ecole des Mines d'Ales - LGI2P, France
— Philippe Cudré-Mauroux, Université de Fribourg, Suisse
— Olivier Curé, Université Pierre et Marie Curie – LIP6, France
— Célia Da Costa Pereira, Université Nice Sophia Antipolis - I3S, France
— Mathieu D'Aquin, The Open University - Grande Bretagne
— Jérôme David, Université Pierre-Mendès-France - LIG, France
— Rim Djedidi, Université Paris 13 - LIMICS, France
— Jean-Pierre Evain, EBU, Suisse
— Gilles Falquet, Université de Genève, Suisse
— Catherine Faron Zucker, Université Nice Sophia Antipolis - I3S, France
— Cécile Favre, Université Lyon 2 - ERIC, France
— Béatrice Fuchs, Université Jean Moulin Lyon 3 - LIRIS, France
— Frédéric Fürst, Université de Picardie Jules Verne - MIS, France
— Jean-Gabriel Ganascia, Université Pierre et Marie Curie - LIP6, France
— Fabien Gandon, INRIA Sophia Antipolis, France
— Aldo Gangemi, Université Paris 13 - LIPN, France
— Catherine Garbay, CNRS - LIG, France
— Faïez Gargouri, Université de Sfax - ISIMS-Miracl, Tunisie
— Serge Garlatti, Telecom Bretagne - Labsticc, France
— Alain Giboin, INRIA Sophia Antipolis, France
— Monique Grandbastien, Université de Lorraine, LORIA, France
— Christophe Guéret, Data Archiving and Networked Services - DANS/KNAW, La Haye, Pays-Bas
— Ollivier Haemmerlé, Université de Toulouse Jean Jaurès - IRIT, France

— Mounira Harzallah, IUT de Nantes - LINA, France
— Nathalie Hernandez, Université de Toulouse Le Mirail - IRIT, France
— Antoine Isaac, Europeana & Vrije Universiteit Amsterdam, Pays-Bas
— Marie-Christine Jaulent, INSERM U1142 - LIMICS, France
— Clément Jonquet, Université de Montpellier - LIRMM, France
— Nadjet Kamel, Université de Sétif, Algérie
— Gilles Kassel, Université de Picardie Jules Verne - MIS, France
— Khaled Khelif, Airbus Defence and Space, France
— Pascale Kuntz, Université de Nantes - LINA, France
— Philippe Laublet, Université Paris Sorbonne - STIH, France
— Florence Le Ber, Université de Strasbourg /ENGEES - ICube, France
— Michel Leclère, Université Montpellier 2 - LIRMM, France
— Alain Léger, Orange Labs, France
— Dominique Lenne, Université de Technologie de Compiègne, Heudiasyc, France
— Moussa Lo, Université Gaston Berger, Sénégal
— Jean-Charles Marty, Université de Savoie - LIRIS, France
— Nada Matta, Université de Technologie de Troyes – ICD/Tech-CICO, France
— Alain Mille, Université Claude Bernard Lyon 1 - LIRIS, France
— Pascal Molli, Université de Nantes - LINA, France
— Alexandre Monnin, INRIA – Sophia Antipolis, France
— Claude Moulin, Université de Technologie de Compiègne, Heudiasyc, France
— Amedeo Napoli, CNRS - LORIA, France
— Emmanuel Nauer, Université de Lorraine - LORIA, France
— Jérôme Nobécourt, Université Paris 13 - LIMICS, France
— Gilbert Paquette, Télé-université - LICEF, Canada
— Alexandre Passant, mdg.io, Irelande
— Nathalie Pernelle, Université Paris 11 - LRI, France
— Yannick Prié, Université de Nantes - LINA, France
— Cédric Priski, LIST, Luxembourg
— Sylvie Ranwez, Ecole des Mines d'Ales - LGI2P, France
— Chantal Reynaud, Université Paris-Sud - LRI, France
— Roger Roberts, RTBF, Belgique
— Catherine Roussey, Irstea, France
— Pascal Salembier, Université de Technologie de Troyes - ICD/Tech-CICO, France
— Francois Scharffe, Université de Montpellier 2 - 3Top, France
— Karim Sehaba, Université Lumière Lyon 2 - LIRIS, France
— Hassina Seridi, Université Badji Mokhtar-Annaba, Algérie
— Nathalie Souf, Université Paul Sabatier Toulouse - IRIT, France
— Milan Stankovic, Université Paris-Sorbonne - Sépage & STIH, France
— Sylvie Szulman, Université Paris 13 - LIPN, France
— Eddie Soulier, Université de Technologie de Troyes – ICD/Tech-CICO, France
— Andrea Tettamanzi, Université Nice Sophia Antipolis - I3S, France
— Yannick Toussaint, INRIA Nancy Grand-Est - LORIA, France
— Francky Trichet, Université de Nantes - LINA, France
— Cassia Trojahn, Université de Toulouse Le Mirail - IRIT, France
— Raphaël Troncy, EURECOM, France
— Pierre-Yves Vandenbussche, Fujitsu Limited Ireland, Irelande
— Serena Villata, INRIA Sophia Antipolis, France
— Amel Yessad, Université Paris 6 - Laboratoire LIP6, France
— Haifa Zargayouna, Université Paris 13 - LIPN, France
— Antoine Zimmermann, École des Mines de Saint-Étienne, France
— Pierre Zweigenbaum, CNRS - LIMSI, France

Table des matières

Méthodes et outils d'acquisition des connaissances

Ontologie I

Web sémantique, web 2.0, web des données, web des objets et applications à base de connaissances

Langages, méthodes et outils pour la gestion des connaissances

Représentation des connaissances : manipulation

Ontologie II

Traitement et raisonnement sur les connaissances

Démonstrations

Construction d'un vocabulaire patient/médecin dédié au cancer du sein à partir des médias sociaux

Mike Donald Tapi Nzali[1,2], Sandra Bringay[2,3], Christian Lavergne[1,3], Thomas Opitz[4], Jérôme Azé[2], Caroline Mollevi[5]

[1] I3M, Université Montpellier, France
mike-donald.tapi-nzali@univ-montp2.fr, christian.lavergne@univ-montp2.fr

[2] LIRMM, Université Montpellier, France
sandra.bringay@lirmm.fr, jerome.aze@lirmm.fr

[3] Université Paul Valery Montpellier, France

[4] Biostatistique et Processus Spatiaux (BioSP), INRA Avignon, France
thomas.opitz@paca.inra.fr

[5] Unité de biostatistique, Institut de Cancérologie de Montpellier, France
Caroline.Mollevi@icm.unicancer.fr

Résumé : De nos jours, les médias sociaux sont de plus en plus utilisés par les patients et les professionnels de santé. Les patients, généralement profanes dans le domaine médical, utilisent de l'argot, des abréviations et un vocabulaire qui leur est propre lors de leurs échanges. Pour analyser automatiquement les textes des réseaux sociaux, l'acquisition de ce vocabulaire spécifique est nécessaire. En nous appuyant sur un corpus de documents issus de messages de médias sociaux de type forums et Facebook, nous décrivons la construction d'une ressource lexicale qui aligne le vocabulaire des patients à celui des professionnels de santé. Ce travail permettra, d'une part d'améliorer la recherche d'informations dans les forums de santé et d'autre part, de faciliter l'élaboration d'études statistiques basées sur les informations extraites de ces forums.

Mots-clés : Extraction d'information, Médias sociaux, Vocabulaire patient

1 Introduction

Les vocabulaires contrôlés (e.g. SNOMED, MeSH, UMLS, etc.) jouent un rôle clé dans les applications biomédicales de fouille de textes. Ces vocabulaires contiennent seulement les termes utilisés par les professionnels de santé. Depuis 10 ans, des vocabulaires dédiés aux consommateurs de soins de santé (Consumer Health Vocabularies - CHV), ont également été crées (Zeng & Tse, 2006). Ces CHV lient des mots de tous les jours se rapportant au domaine de la santé à des mots d'argot technique utilisés par les professionnels de santé.

Dans cet article, nous proposons une méthode semi-automatique pour construire un tel CHV pour la langue française. Par exemple, nous cherchons à relier le mot "onco" utilisé par les patients à "oncologue" utilisé par les professionnels de santé. L'originalité de notre approche est d'utiliser les textes rédigés par les patients (PAT Patient-Authored Text), provenant des messages issus des médias sociaux de type forums ou Facebook, ainsi que la structure de l'encyclopédie universelle collaborative Wikipédia. Notre méthode a été expérimentée avec succès sur un jeu de données réelles dans le domaine du cancer du sein. Elle a été validée automatiquement en utilisant la ressource collaborative du site JeuxDeMots.org. Une validation manuelle a été également réalisée par 4 personnes, dont un expert du domaine du cancer du sein.

Cet article est organisé comme suit. Dans la section 2, nous motivons notre travail et donnons un état de l'art rapide. Dans la section 3, nous décrivons chaque étape de la méthode. Dans la

section 4, nous présentons le cadre expérimental utilisé pour évaluer les performances de cette méthode. Dans la section 5, nous discutons des premiers résultats. Finalement, dans la section 6, nous concluons et donnons quelques perspectives à ces travaux.

2 Motivations et état de l'art

Selon une enquête réalisée en 2011 par la fondation HON [1], Internet est devenu la deuxième source d'information des patients après les consultations chez les médecins. 24% de la population utilise Internet pour trouver des informations sur leur santé au moins une fois par jour (et jusqu'à 6 fois par jour) et 25% au moins plusieurs fois par semaine. Ces « patients 2.0 » sont motivés par un accès facile à Internet à domicile, le manque général de temps pour des consultations plus classiques, un soutien humain (surtout pour les maladies chroniques), la nécessité de connaître les expériences des autres, ainsi que le désir d'obtenir plus d'informations avant ou après une consultation (Hancock *et al.*, 2007; Merolli *et al.*, 2013). En maintenant l'anonymat, ces médias sociaux (forums, groupes Facebook) leur permettent de discuter librement avec d'autres utilisateurs, usagers, personnes, et aussi avec des professionnels de santé. Ils parlent de leurs résultats médicaux et de leurs options de traitement, mais ils reçoivent également un soutien moral.

Dans des travaux précédents (Opitz *et al.*, 2014), nous nous sommes intéressés à l'étude de la qualité de vie des patientes atteintes d'un cancer du sein à partir des médias sociaux. Nous avons cherché à capturer et quantifier ce que les patientes expriment dans les forums à propos de leur qualité de vie. Une importante limitation à ces travaux vient du type de textes traités. En effet, la plupart des patients sont des profanes dans le domaine médical. Lors de leurs échanges, ils utilisent des mots d'argot, des abréviations et un vocabulaire spécifique construit par la communauté en ligne, à la place des termes médicaux que l'on retrouve dans les ressources terminologiques utilisées par les professionnels de santé comme la SNOMED (Nomenclature systématisée de médecine) [2], le MeSH (Medical Subject Headings) [3], l'UMLS (Unified Medical Language System) [4]. Les méthodes de fouille de textes mises en œuvre ont montré leurs limites à cause de ce vocabulaire particulier. Nous nous proposons donc dans cet article de construire un vocabulaire dédié aux « consommateurs de soins de santé » (Consumer Health Vocabularies - CHV).

Initialement, la création de ces CHV a été motivée par la réduction des écarts de connaissances entre les patients et les professionnels de santé (Zeng *et al.*, 2007). En effet, la littérature montre que la compréhension par les patients de la terminologie médicale est essentielle pour appréhender leur maladie et pour participer au processus de décision médicale. En outre, les communications réussies patient-médecin sont intrinsèquement liées à la confiance que le patient a envers son médecin (Fiscella *et al.*, 2004). S'il ne comprend pas de quoi le médecin lui parle, le patient est moins enclin à lui faire confiance. Certains chercheurs ont ainsi utilisé des CHV pour améliorer la lisibilité des documents médicaux (Wu *et al.*, 2013) ou du dossier patient électronique (Ramesh *et al.*, 2013) par les non-experts. (Doing-Harris & Zeng-Treitler,

1. HON (Health On the Net) How Do General Public Search Online Health Information ? Avril 2011
2. http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html
3. http://mesh.inserm.fr/mesh/
4. http://www.nlm.nih.gov/research/umls/

2011) ont proposé une méthode pour générer automatiquement des termes candidats à traiter par des humains pour inclusion dans un CHV. Ils n'apparient pas automatiquement les termes des patients à ceux des médecins comme nous allons le proposer dans cet article. Actuellement, seuls deux CHV sont disponibles : 1) MedlinePlus [5], librement disponible, est produit par la National Library of Medicine ; 2) Open and collaborative Consumer Health Vocabulary (CAO CHV) [6] est inclus dans l'UMLS. À notre connaissance, en français, il n'existe pas de CHV.

Dans les médias sociaux (forums, Facebook), le volume des textes rédigés par les patients (PAT Patient-Authored Text) est de plus en plus important (MacLean & Heer, 2013). Si de tels PAT ne sont pas suffisamment précis pour des objectifs scientifiques, ils donnent accès à de très nombreuses descriptions de l'expérience des patients, sur un large éventail de sujets, en temps réel. Au cours des cinq dernières années, il y a eu un intérêt croissant dans l'exploitation de ces PAT comme outil pour la santé publique, par exemple pour des analyses de la propagation de la grippe (Sadilek *et al.*, 2012) ou la découverte d'effets secondaires grâce à des sites comme CureTogether [7] et PatientsLikeMe [8]. Dans cet article, notre objectif est d'utiliser les PAT issus des médias sociaux en entrée d'une méthode semi-automatique permettant de construire un CHV français pour le domaine du cancer du sein, en recueillant différents types d'expressions de patients, comme des abréviations, des fautes d'orthographe fréquentes ou des mots de tous les jours détournés par les non experts pour parler de leurs maladies.

L'originalité de notre approche est d'utiliser l'architecture de l'encyclopédie universelle collaborative Wikipédia [9] pour rapprocher des termes utilisés par les patients et des termes utilisés par des professionnels de la santé. Wikipédia est une encyclopédie sur le Web multilingue qui couvre de très nombreux domaines. La version française, en date du 25 février 2015 contient plus d'un million et demi d'articles. Les articles étant finement structurés, Wikipedia a été utilisée avec succès dans des applications de questions/réponses (Buscaldi & Rosso, 2006), de catégorisation de textes (Wang *et al.*, 2009). Plus particulièrement, on trouve des approches permettant de calculer la parenté sémantique entre des termes (Ponzetto & Strube, 2006; Gabrilovich & Markovitch, 2007). Ces derniers ont développé une technique permettant de représenter le sens des mots dans un espace de dimension élevée de concepts issus de Wikipedia. (Chernov *et al.*, 2006) ont utilisé les liens entre les catégories présentes sur Wikipédia pour extraire de l'information sémantique. (Witten & Milne, 2008) utilisent plutôt les liens entre les articles de Wikipedia pour déterminer la proximité sémantique entre les mots. Dans ce travail, nous allons comme (Witten & Milne, 2008), utiliser la structure de liens entre les termes Wikipédia pour rapprocher le vocabulaire des patients, de celui des médecins.

3 Méthodes

La figure 1 illustre la méthode proposée, structurée en 5 étapes. Cette méthode prend en entrée une ressource médicale à laquelle nous allons apparier les termes des patients. Nous avons

5. http://www.nlm.nih.gov/medlineplus/
6. http://www.consumerhealthvocab.org/
7. http://curetogether.com/
8. http://www.patientslikeme.com/
9. http://fr.wikipedia.org/wiki/Wikipédia:Accueil_principal

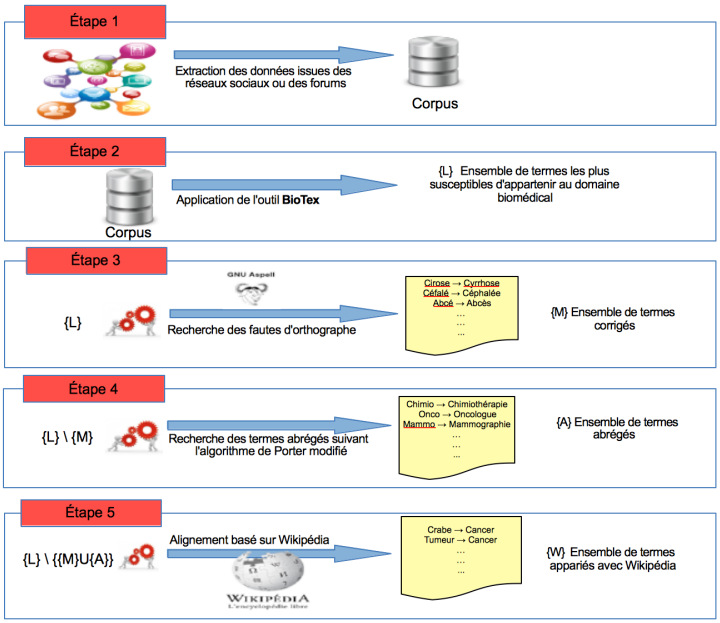


FIGURE 1 – Diagramme d'activité utilisé dans notre algorithme.

choisi comme ressource de référence le vocabulaire donné sur le site de l'INCa [10] composé de 1 227 termes, tous présents dans le MeSH en version française, que nous noterons INCa.

Étape 1 : Développement du corpus de messages. Nous utilisons des messages issus du réseau social Facebook et de forums échangeant sur le cancer du sein. Les groupes de paroles Facebook facilitent la connexion avec d'autres patientes ou associations de patientes. Les groupes permettent de publier des mises à jour, des photos ou des documents et d'envoyer des messages à tous les membres du groupe. La figure 2 correspond à un post commenté par 5 membres. Dans le post initial, apparaît l'abréviation *chimio*. Dans la première réponse, apparaît la faute d'orthographe *catheter* pour *cathéter*. Dans la troisième réponse, on trouve le terme *rdv* pour *rendez-vous*. Nous avons récolté ainsi 96 792 messages publiés par 1 389 membres entre 2010 et 2014 des groupes Facebook publics tels que *Cancer du sein, Octobre rose 2014, Cancer du sein - breast cancer, brustkrebs*. Nous avons aussi travaillé avec les données provenant du forum *lesimpatientes.com*, nous avons récolté 134 334 messages provenant de 4 627 utilisateurs. Chaque document du corpus contient l'ensemble des messages d'un utilisateur d'un forum de santé ou d'un groupe Facebook. Dans ce travail, nous travaillons uniquement sur les textes et n'utilisons aucune autre métadonnée. Un avantage de notre approche est qu'aucun traitement spécifique n'a pas été effectué sur ces messages (pas de correction automatique, ni de lemmatisation).

Étape 2 : Extraction des termes candidats à partir du corpus. À partir du corpus, nous cherchons les termes ayant une grande probabilité d'appartenir au domaine médical. Pour cela, nous utilisons l'outil BioTex (Lossio-Ventura *et al.*, 2014a). BioTex est une application d'extraction automatique de termes biomédicaux qui met à disposition un ensemble de mesures sta-

10. http://www.e-cancer.fr/cancerinfo/ressources-utiles/dictionnaire/

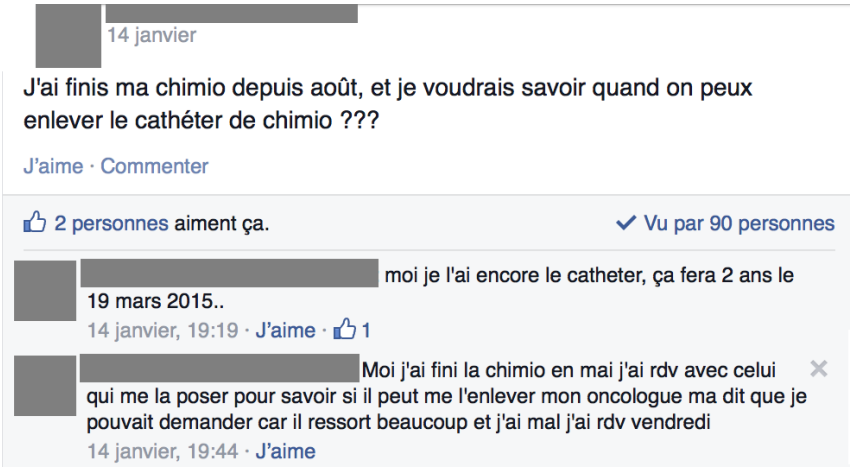


FIGURE 2 – Posts anonymisés et commentés par des utilisateurs d'un groupe Facebook.

tistiques pour la sélection de ces termes. La sélection est essentiellement basée sur la fréquence d'apparition et la construction linguistique qui doit être similaire à celle des termes présents dans les ressources médicales de type MeSH. Pour cela, 200 motifs linguistiques ont été utilisés (voir table 1). La mesure choisie est *LIDF-value (Linguisitic patterns, IDF, and C-value information)* Lossio-Ventura *et al.* (2014b) car (Lossio-Ventura *et al.*, 2014c) ont démontré que cette mesure donne de meilleurs résultats comparés à d'autres comme *TF-IDF, Okapi, C-value*. À l'issue de cette étape, nous obtenons en sortie un ensemble $T = t_1, ..., t_N$ de N **n**-grammes (n $\in [1..4]$), dont certains ne sont pas répertoriés dans l'INCa, que nous allons utiliser dans les étapes 2, 3 et 4 décrites ci-dessous. Il est important de noter que nous obtenons ici des candidats composés de plusieurs mots. Ces candidats sont spécifiques aux textes des patients traitant des sujets médicaux.

Motif	**Texte instantiant le motif**
Nom Adj	Echographie mammaire
Nom Prep :det Nom	Cancer du sein
Nom Prep NomPropre	Maladie d'Alzheimer

TABLE 1 – Exemples de motifs linguistiques utilisés dans BioTex

Étape 3 : Correction orthographique des termes candidats mal orthographiés. À partir des mots identifiés à l'étape 2, fréquemment utilisés par les patients, on recherche ceux qui correspondent à des fautes d'orthographes courantes. Nous cherchons à apparier tous les termes $t_i \in T$, avec un mot bien orthographié présent dans l'INCa. Pour cela nous utilisons le logiciel Aspell [11] pour obtenir un ensemble $M = \{m_1, m_2, ..., m_m\}$ de m propositions de corrections du mot t_i et ne conservons que les propositions présentes dans l'INCa. Nous utilisons ensuite la

11. `http://aspell.net/`

mesure de Levenshtein pour calculer la distance entre le terme t_i et chaque terme m_j. La mesure de Levenshtein entre deux termes est le nombre minimum de modifications à caractère unique nécessaires pour changer t_i en m_j. Seul les termes dont la distance est inférieure ou égale à 2 ne sont conservés comme appariement. Trois autres conditions sont également nécessaires : 1) les mots appariés doivent commencer par la même lettre ; 2) la longueur des mots appariés est de plus de trois caractères ; 3) la comparaison est insensible à la casse. Si toutes les conditions sont vérifiées, le terme t_i est associé au terme m_j avec un $poids(m_j, t_i) = 1/|M|$. La table 2 présente quelques fautes d'orthographe fréquemment rencontrées.

Termes biomédicaux	**Termes patients**
cirrhose	Cyrose
abcès	abcé
métastase	metastase

TABLE 2 – Équivalent entre termes biomédicaux et termes patients (contenant des fautes d'orthographe)

Étape 4 : Recherche des termes abrégés. La plupart des expressions biomédicales sont longues (composées de 2, 3 mots voir plus). Très souvent, ces expressions sont tronquées par les patients. À partir des mots identifiés à l'étape 2, fréquemment utilisés par les patients, on recherche ceux qui correspondent à des abréviations. Pour cela, nous avons adapté l'algorithme de (Paternostre *et al.*, 2002) en utilisant la liste des suffixes les plus utilisés dans le domaine biomédical (e.g : logie, logue, thérapie, thérapeute...). Pour un terme $t_i \in T$, on obtient un ensemble $A = \{a_1, a_2, ..., a_k\}$ de k propositions d'abréviations incluses dans l'INCa. Le terme t_i est associé à une abréviation a_j avec un $poids(a_j, t_i) = 1/|A|$. Des exemples de termes appariés avec cette méthode sont listés dans la table 3.

Termes biomédicaux	**Termes patients**
Oncologue	Onco
Chimiothérapie	Chimio
mammographie	mammo

TABLE 3 – Équivalent entre termes biomédicaux et termes patients (abréviations)

Étape 5 : Alignements basés sur Wikipédia. Nous nous intéressons ici à tous les termes produits à l'étape 2 qui ne contiennent ni des mots comportant des fautes d'orthographe fréquentes (repérées à l'étape 3), ni des abréviations (repérées à l'étape 4). Pour cela nous travaillons sur l'architecture de la ressource encyclopédique Wikipédia que nous interrogeons grâce à son API [12]. Dans cette encyclopédie un terme (*mot*) référencé est décrit par une page (*http ://fr.wikipedia.org/wiki/**mot***) et est lié à d'autres termes eux mêmes décrits par d'autres pages. Les pages (mots) liées à un terme se retrouvent dans une page dédiée

12. `http://fr.wikipedia.org/w/api.php?`

(*http ://fr.wikipedia.org/wiki/Spécial :Pages_liées/**mot***). Certaines relations entre termes Wikipédia sont typées (e.g. synonymie). Sur la partie droite de la figure 3, on retrouve la page du terme *Tumeur* et sur la partie gauche, les termes liés. Soit W l'ensemble des termes liés par Wikipédia à un terme t_i et appartenant à la ressource INCa. Un terme t_i est associé à un terme w_i selon un poids calculé avec la formule 2. Ce poids est important pour éliminer des associations comme *tumeur* et *jules cesar* dans l'exemple de la figure 3 car la page tumeur ne contient pas de référence à Jules César alors que cette dernière en contient (référence à son état de santé).

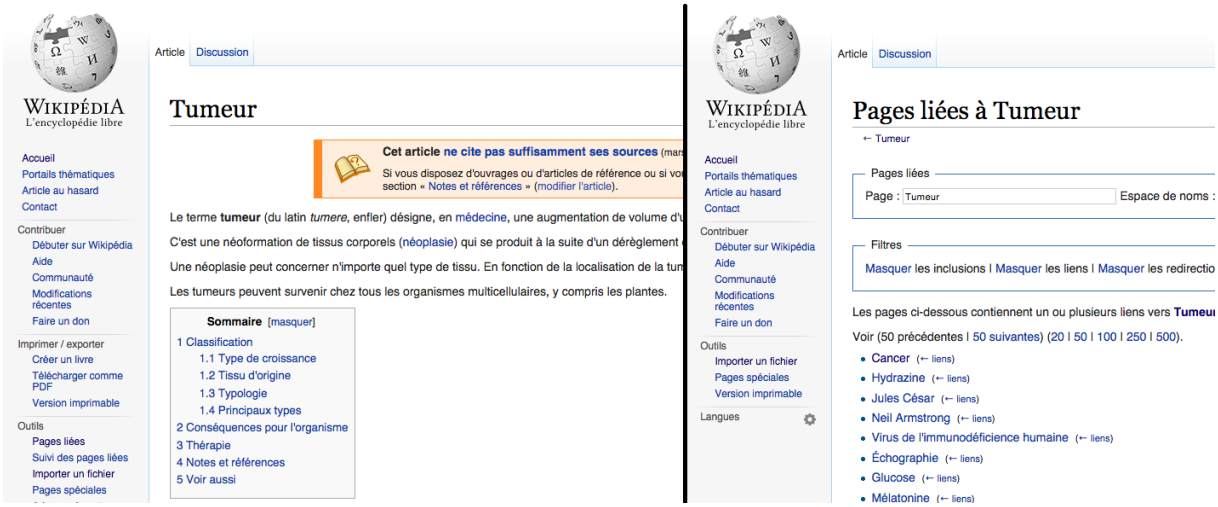


FIGURE 3 – Page Wikipédia et page liée.

$$MoyNb(w_k, t_n) = \frac{Nb(w_k, PageW(t_n)) + Nb(t_n, PageW(w_k))}{2} \quad (1)$$

$$Poids(w_j, t_i) = \frac{MoyNb(w_j, t_i)}{\sum_{k=1}^{|W|} MoyNb(w_k, t_i)} \quad (2)$$

où $Nb(a, PageW(b))$ est la fréquence d'apparition du terme a dans la page wikipédia du terme b.

4 Résultats

À l'issu du processus précédent, nous avons obtenu l'ensemble de K relations r_i avec $i \in [1, K]$. Chaque relation r_i relie un mot patient pat_i [13], avec un mot médecin bio_i [14]. Chaque relation est associée à une méthode d'obtention $meth \in$ *{orthographe,abréviation,association}* et à un poids $poids \in [0, 1]$. Dans cette section, nous présentons les deux méthodes de validation utilisées (automatique et manuelle) et les différents résultats obtenus. Comme les associations détectées automatiquement dépendent beaucoup du contenu de Wikipédia, la validation finale

13. Les pat_i sont les termes issus du corpus
14. Les bio_i sont les termes du dictionnaire fourni par l'INCa

manuelle est importante pour présentir les faiblesses des associations obtenues avec les méthodes quantitatives.

4.1 Validation automatique

Nous validons automatiquement des relations r_i, si l'un des deux critères suivants est vérifié :

— Le poids de la relation est égale à 1. Par exemple, pour une faute d'orthographe avec une seule possibilité de correction, nous considérons la correction validée.

— La paire pat_i - bio_i existe dans le dictionnaire de relations fournis par le jeux contributif www.JeuxDeMots.org, dont le but est de construire un vaste réseau lexical-sémantique (Lafourcade & Joubert, 2012). Cette ressource, construite par les internautes, rassemble 112 types de relations dont 179 578 occurences de la relation synonymie. L'avantage de cette validation est que nous obtenons une étiquette supplémentaire pour typer les relations.

Des exemples de relations validées automatiquement sont présentées dans le tableau 4. Sur les 432 relations obtenues après exécution de notre programme, 211 relations ont été validées automatiquement, soit 49% des relations. Nous avons donc 169 relations de type « association », 32 relations de type « erreur », et 10 relations de type « abréviation ».

Terme Patient	**Terme biomédical**	**Relation**
Chir	Chirurgie	Abréviation
Chimio	Chimiothérapie	Abréviation
mammo	mammographie	Abréviation
hopital	hôpital	Erreur Orthographique
cheuveux	cheveux	Erreur orthographique
radiotherapie	radiothérapie	Erreur orthographique
♡	Cœur	Association
tumeur	cancer	Association
chute des cheveux	Alopécie	Association

TABLE 4 – Exemples de termes validés automatiquement en utilisant JeuxDeMots

4.2 Validation manuelle

Toutes les relations r_i n'ayant pas pu être validées automatiquement sont présentées à l'expert pour validation manuelle [15]. Le tableau 5 présente des exemples de résultats validés manuellement. Les annotations ont été faites par 4 personnes, dont un expert du domaine du cancer du sein.

Lors de la validation manuelle, trois choix sont proposés à l'utilisateur : 1) **Yes** : pour valider la relation ; 2) **No** : pour invalider la relation ; 3) **Neutre** : l'annotateur ne sait pas. Nous considérons qu'une relation r_i est validée si le nombre de « Yes » est supérieur au nombre de « No » et de « Neutre ».

15. Un image de l'interface de valisation est présente à cette url : `http://www.lirmm.fr/~tapinzali/Validation/Validation.php`

Suite à la validation automatique de 214 relations, il nous restait 218 relations à valider manuellement. Après consensus entre les différents annotateurs, 93 relations sur les 218 ont été validées. Un coefficient de kappa de Fleiss a été calculé pour mesurer l'accord inter-annotateur, nous obtenons un ***k*** de **0,21** (accord faible, dû à la variabilité individuelle du jugement des annotateurs sur l'intérêt médical des termes).

Terme Patient	**Terme biomédical**	**Relation**
Psy	Psychologue	Abréviation
Onco	Oncologue	Abréviation
gynéco	gynécologue	Abréviation
constipation	Laxatif	Association
libido	Sexologie	Association
morphine	Douleur	Association

TABLE 5 – Exemples de termes validés manuellement

5 Discussions

Finalement, sur les deux corpus (10 Mo chacun), nous avons extrait plus de 432 relations. Ce nombre de relations augmente lorsque nous diminuons les seuils sur LIDF-value à l'étape 2 décrite dans la section 3 par exemple. Toutefois, le temps d'exécution devient alors très grand (45 minutes pour un corpus de 1Mo), car diminuer le seuil implique de prendre en compte de nouveaux termes à traiter. Par ailleurs, le nombre limité de relations obtenues s'explique également par le nombre de termes médecin cibles auxquels nous cherchons à apparier les termes des patients (ceux de l'INCa qui contient uniquement 1 227 termes). En effet, nous cherchons à créer une ressource pour le cancer du sein et non une ressource généraliste.

Comme (Doing-Harris & Zeng-Treitler, 2011), nous avons cherché à évaluer la précision globale de notre méthode. Pour cela nous avons utilisé la formule 3.

$$P = \frac{|R_a| + |R_m|}{|R|} \tag{3}$$

où R_a est l'ensemble contenant la liste des relations validées automatiquement, R_m est l'ensemble contenant la liste des relations validées manuellement et R est l'ensemble des relations ayant été fournies en sortie par notre outil. Nous avons obtenu une précision ***P*** de 71% et validé 307 relations sur les 432 obtenues. La figure 4 présente les résultats obtenus selon le type des relations, sur ce premier jeu de données nous montrent que sur des données textuelles bruitées biomédicales extraites extraites des forums de santé et des réseaux Facebook, nous obtenons de meilleurs résultats que ceux de (Doing-Harris & Zeng-Treitler, 2011). Ces auteurs ont créé un CHV en langue anglaise. Sur 88 994 n-grammes, ils ne trouvent que 774 relations et n'en valident que 237, soit 31% de précision.

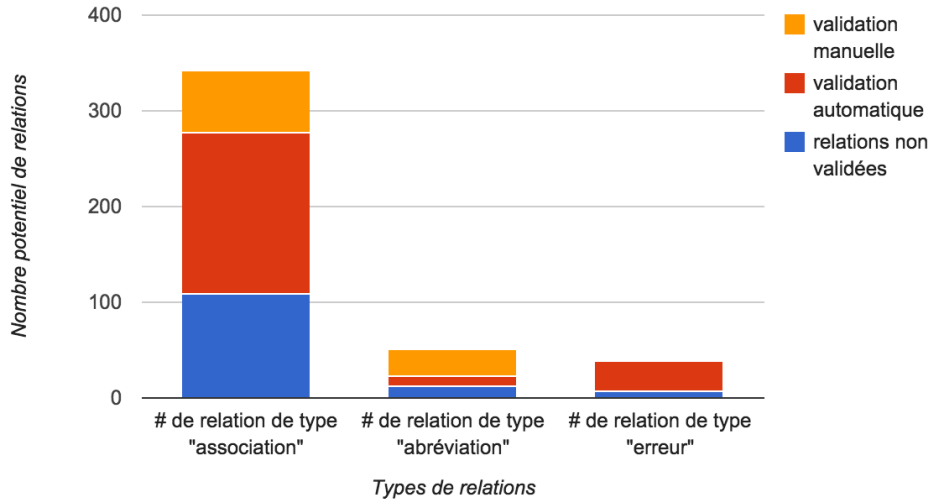


FIGURE 4 – Histogramme du nombre de relations validées automatiquement, manuellement et celles non validées.

6 Conclusion et perspectives

Dans cet article, nous avons présenté une méthode permettant de relier les termes utilisés par les patients et constituant un CHV à ceux utilisés par les professionnels de santé et présents dans les vocabulaires contrôlés. Un avantage de cette méthode est qu'elle permet d'aligner des expressions pouvant être composées de plusieurs mots et de solliciter l'expert uniquement pour les termes pour lesquels il reste un doute (n'ayant pas été éliminés par le filtre automatique). Contrairement à la plupart des CHV existant uniquement en langue anglaise et construits manuellement, nous proposons une méthode semi-automatique originale pour construire un tel CHV pour le français. Nous avons appliqué cette méthode au domaine de la cancérologie mais elle peut être appliquée à de nombreux autres domaines. Une telle ressource sera une brique essentielle à l'exploitation automatiquement du contenu des médias sociaux dans le domaine médical.

La ressource est actuellement téléchargeable librement pour la communauté `http://www.lirmm.fr/~tapinzali/Ressources/VocPatMed`. Nous sommes actuellement en train de transformer cette ressource dans un format lisible par l'être humain et par l'ordinateur. Pour ce faire, nous allons donc créer une ontologie au format SKOS pour l'intégrer sur la plateforme BioPortal (Noy *et al.*, 2009). SKOS fournit le vocabulaire nécessaire pour définir les attributs d'un concept et les relations entre les concepts qui nous permettra de garder des informations sur la méthode d'obtention du terme patient (orthographe, abbréviation et association), le type de validation et éventuellement le type de relation identifiée automatiquement dans Wikipédia ou dans la ressource jeux de mots (e.g. synonyme). Un mapping sera fait entre notre ontologie au format SKOS et celles déjà existantes sur BioPortal puisque nous apparierions des termes patients à des termes de l'INCa tous présents dans le MeSH. Grâce à l'*Annotator* de BioPortal,

nous pourrons alors annoter n'importe quels types des textes issus des médias sociaux échangeant dans le domaine biomédical.

Les expérimentations réalisées vont également être complétées. Nous allons notamment étudier la relation entre le poids et les validations expertes, afin de définir un seuil à partir duquel il n'est plus intéressant de proposer une relation à l'expert. Actuellement, nous avons mesuré à quel point nos associations étaient justes en calculant la précision mais nous devons également nous intéresser au rappel pour mesurer combien de relations nous ne sommes pas capables d'identifier en repartant des mots non appariés de l'INCa par exemple. Nous souhaitons également appliquer cette méthode sur d'autres jeux de données réels pour enrichir notre ressource. Entre chaque mise à jour de la ressource, nous conservons une *Liste noire* de tous les relations rejetées. Nous comparerons le vocabulaire acquis dans les forums et celui acquis dans les réseaux de type Facebook. Les forums seraient ils plus intéressants car plus techniques ?

Le choix de la ressource Wikipédia est également discutable car n'entrant pas dans la catégorie des PAT en se situant entre le discours des praticiens et des médecins, ni scientifique, ni grand public. D'autres ressources doivent être envisagées pour le filtrage réalisé lors de l'étape 5 par exemple en traduisant un CHV anglais.

À plus long terme, nous envisageons de ré-exploiter les données utilisées pour étudier la qualité de vie des patientes atteintes d'un cancer du sein, et ainsi améliorer nos processus comme celui présenté dans (Opitz *et al.*, 2014). Nous pourrons mesurer l'impact de la ressource. De même, que donnerait notre méthode sur des médias sociaux en anglais pour étendre les CHV existants ? Nous allons également étudier l'évolution du vocabulaire des patients au cours du temps en utilisant un modèle de type LDA (Lattent Dirichlet Allocation). Il s'agit d'un modèle bayésien hiérarchique fondé sur une catégorie de modèles « topic models » et qui cherchent à découvrir des structures thématiques cachées dans des vastes archives de documents.

7 Remerciements

Ces travaux ont été financés par l'ANR SFIR (Semantic Indexing of French Biomedical Data Resources) et par le projet « Comparison of longitudinal analysis models of the health-related quality of life in oncology » (financement IRESP).

Références

Buscaldi D. & Rosso P. (2006). Mining knowledge from wikipedia for the question answering task. In *Proceedings of the International Conference on Language Resources and Evaluation*, p. 727–730.

Chernov S., Iofciu T., Nejdl W. & Zhou X. (2006). Extracting semantics relationships between wikipedia categories. volume 206 : Citeseer.

Doing-Harris K. M. & Zeng-Treitler Q. (2011). Computer-assisted update of a consumer health vocabulary through mining of social network data. volume 13 : JMIR Publications Inc.

Fiscella K., Meldrum S., Franks P., Shields C. G., Duberstein P., McDaniel S. H. & Epstein R. M. (2004). Patient trust : is it related to patient-centered behavior of primary care physicians ? volume 42, p. 1049–1055 : LWW.

Gabrilovich E. & Markovitch S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, p. 1606–1611.

Hancock J. T., Toma C. & Ellison N. (2007). The truth about lying in online dating profiles. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 449–452 : ACM.

LAFOURCADE M. & JOUBERT A. (2012). Increasing long tail in weighted lexical networks. In *Cognitive Aspects of the Lexicon (CogAlex-III), COLING*, p.1̃6.

LOSSIO-VENTURA J. A., JONQUET C., ROCHE M. & TEISSEIRE M. (2014a). Biotex : A system for biomedical terminology extraction, ranking, and validation. In *Proceedings of the 13th International Semantic Web Conference (ISWC'14). Trento, Italy*.

LOSSIO-VENTURA J. A., JONQUET C., ROCHE M. & TEISSEIRE M. (2014b). Integration of linguistic and web information to improve biomedical terminology extraction. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, p. 265–269 : ACM.

LOSSIO-VENTURA J. A., JONQUET C., ROCHE M. & TEISSEIRE M. (2014c). Yet another ranking function for automatic multiword term extraction. In *Advances in Natural Language Processing*, p. 52–64 : Springer.

MACLEAN D. L. & HEER J. (2013). Identifying medical terms in patient-authored text : a crowdsourcing-based approach. p. amiajnl–2012 : BMJ Publishing Group Ltd.

MEROLLI M., GRAY K. & MARTIN-SANCHEZ F. (2013). Health outcomes and related effects of using social media in chronic disease management : A literature review and analysis of affordances. volume 46, p. 957–969 : Elsevier.

NOY N. F., SHAH N. H., WHETZEL P. L., DAI B., DORF M., GRIFFITH N., JONQUET C., RUBIN D. L., STOREY M.-A., CHUTE C. G. *et al.* (2009). Bioportal : ontologies and integrated data resources at the click of a mouse. p. gkp440 : Oxford Univ Press.

OPITZ T., AZÉ J., BRINGAY S., JOUTARD C., LAVERGNE C. & MOLLEVI C. (2014). Breast cancer and quality of life : medical information extraction from health forums. In *Medical Informatics Europe*, p. 1070–1074.

PATERNOSTRE M., FRANCQ P., LAMORAL J., WARTEL D. & SAERENS M. (2002). Carry, un algorithme de désuffixation pour le français.

PONZETTO S. P. & STRUBE M. (2006). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, p. 192–199 : Association for Computational Linguistics.

RAMESH B. P., HOUSTON T. K., BRANDT C., FANG H. & YU H. (2013). Improving patients' electronic health record comprehension with noteaid. In *MedInfo*, p. 714–718.

SADILEK A., KAUTZ H. A. & SILENZIO V. (2012). Modeling spread of disease from social interactions. In *ICWSM*.

WANG P., HU J., ZENG H.-J. & CHEN Z. (2009). Using wikipedia knowledge to improve text classification. volume 19, p. 265–281 : Springer.

WITTEN I. & MILNE D. (2008). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence : an Evolving Synergy, AAAI Press, Chicago, USA*, p. 25–30.

WU D. T., HANAUER D. A., MEI Q., CLARK P. M., AN L. C., LEI J., PROULX J., ZENG-TREITLER Q. & ZHENG K. (2013). Applying multiple methods to assess the readability of a large corpus of medical documents. In *MedInfo*, p. 647–651.

ZENG Q. T. & TSE T. (2006). Exploring and developing consumer health vocabularies. volume 13, p. 24–29 : BMJ Publishing Group Ltd.

ZENG Q. T., TSE T., DIVITA G., KESELMAN A., CROWELL J., BROWNE A. C., GORYACHEV S. & NGO L. (2007). Term identification methods for consumer health vocabulary development. volume 9 : Internet Healthcare Coalition.

Acquisition de connaissances pour importer des traces existantes dans un système de gestion de bases de traces

Mohamed Besnaci[1], Nathalie Guin[2] & Pierre-Antoine Champin[2]

[1] Université BADJI MOKHTAR - ANNABA
besnacimohamed@yahoo.fr

[2] Université de Lyon, CNRS Université Lyon1, LIRIS, UMR5205, F-69622, France
nathalie.guin@liris.univ-lyon1.fr, pierre-antoine.champin@liris.cnrs.fr

Résumé : Les systèmes de gestion de bases de traces (SGBT) proposent des fonctionnalités de transformation et de requêtage sur les traces, pouvant intéresser de nombreux utilisateurs des systèmes tracés. Notre objectif est d'assurer l'importation de traces variées à kTBS, un SGBT développé au laboratoire LIRIS. Pour pallier le problème de l'hétérogénéité des traces, nous proposons de définir un collecteur pour chaque système tracé. Pour définir un tel collecteur, un utilisateur ayant une bonne connaissance du système tracé est invité à définir son modèle de trace dans kTBS ainsi que les correspondances entre les éléments de ce modèle et des éléments des traces à importer. Le système généralise les mises en correspondances ainsi explicitées, en interaction avec l'utilisateur, pour créer des règles de Mapping. Après cette phase, la collecte va consister à générer des traces modélisées à partir des traces existantes et du Mapping ainsi défini.

Mots-clés : Traces, Acquisition interactive de connaissances, Système de Gestion de Bases de Traces, Collecte, Mapping, xPath

1 Introduction

Les traces numériques sont des inscriptions temporalisées liées à un système informatique donné et ce pour diverses exploitations. Bien que la naissance des traces date de plusieurs dizaines d'années, elles ont subi une mutation importante cette dernière décennie. En effet, de nombreux systèmes informatiques ont commencé à produire et/ou exploiter les traces, mais chacun avec son propre format, son propre contenu et pour ses propres objectifs. Le concept de trace s'est bien développé, par rapport aux formats possibles adoptés, aux contenus considérés et aux exploitations visées. Cependant, des défis majeurs se posent : celui de leur compréhension et celui de leur interopérabilité. Le problème de compréhension se pose généralement durant les processus d'exploitation des traces ayant pour but la visualisation, l'analyse et la déduction de connaissances. Notre travail s'intègre dans le domaine de l'intégration de données, pour justement proposer une solution au problème de l'interopérabilité. Nous visons à intégrer des traces variées dans un formalisme unique pour permettre ensuite leur exploitation.

Un système tracé est un système capable de générer des traces numériques d'interaction liées aux activités qu'il propose à l'utilisateur. Les systèmes tracés actuels utilisent des traces numériques très variées, à des fins aussi variées. Après consultation et analyse des traces d'un ensemble de systèmes tracés, nous avons abouti à quelques constatations :

— Plusieurs représentations existent, essentiellement des traces en XML, fichiers textes et sous forme de BDDR (Bases De Données Relationnelles).
— Peu de traces sont associées à des modèles explicites (un document DTD est un exemple de modèle explicite pour des traces XML).

— Les traces peuvent être stockées à des endroits uniques ou encore éparpillées dans plusieurs objets qui n'ont pas forcément la même structure. Une trace peut par exemple être stockée dans un seul document XML, ou dans plusieurs tables d'une BDDR ou encore constituée d'un document texte et d'un morceau multimédia.
— Abstraitement, une trace doit englober l'inscription du temps avec d'autres données liées à l'activité exercée et au système tracé. À part la donnée temporelle (bien qu'elle puisse aussi avoir plusieurs variantes), les données inscrites dans les traces existantes sont très variées vu le grand nombre de possibilités d'activités dans les systèmes tracés.
— L'exploitant et l'exploitation de la trace sont d'autres critères différenciant les traces existantes. Selon leur niveau d'abstraction, les traces peuvent être destinées à des utilisateurs finaux à des fins d'orientation, d'assistance et d'auto-formation. Elles peuvent être destinées à des concepteurs, des enseignants, des techniciens ou des cogniticiens à des fins de diagnostic, d'évaluation, d'analyse, etc. Les traces peuvent être aussi exploitées par des programmes pour les mêmes buts, mais aussi à des fins d'inférence et de visualisation.

Devant ces constatations, la question est : comment permettre à un utilisateur d'un système tracé, qui possède des traces d'activités issues de son système, de les importer en tant que traces modélisées (traces associées à leurs modèles explicites) dans un système adoptant un unique formalisme de représentation des traces, afin de pouvoir profiter des fonctionnalités de visualisation et d'analyse d'un tel système ?

Le problème de la collecte des traces a été largement abordé dans les travaux sur les traces d'interaction. En effet, la collecte est une étape critique dans le processus logique de traitement des traces. Son cas le plus usuel consiste à instrumenter les systèmes tracés afin de produire des traces de format et structure spécifiques (May *et al.*, 2008) – ou une combinaison de types de trace dans certains cas comme celui de Ji *et al.* (2013). À la différence de ces situations où la collecte et l'exploitation des traces sont toutes deux assurées par le même système, il existe d'autres situations où ces deux traitements sont séparés. Dans un travail comme celui de Chaachoua *et al.* (2007) ou Sanchez (2006), le rôle du système tracé se résume à la collecte. Les traces collectées sont destinées à des plateformes dédiées au stockage et à l'analyse (Bouhineau *et al.*, 2013; Champin *et al.*, 2013). Notre travail s'intègre dans ce dernier processus et essaye de résoudre le problème de l'importation de traces variées dans l'une des plateformes de gestion des traces appelée kTBS (kernel for Trace Based Systems). Ainsi, cette plateforme est le système de gestion de traces dans lequel nous souhaitons importer des traces venant de systèmes divers. Par ailleurs, nous souhaitons offrir cette fonctionnalité à n'importe quel utilisateur du système tracé qui souhaiterait en exploiter les traces, y compris si il/elle n'a pas de compétences en programmation. En ce sens, nous nous démarquons de travaux similaires (Choquet *et al.*, 2007) ayant une approche plus technique. Ainsi, nous présenterons dans cet article notre processus d'importation de traces variées dans kTBS et l'outil xCollector qui le concrétise. Ce processus a l'avantage de traiter des formats variés de traces (texte, XML et BDDR). Son principe est basé sur l'acquisition de l'expérience de l'utilisateur en matière de collecte de données éparpillées dans les traces existantes. xCollector n'est pas lié à un système tracé spécifique, nous pouvons le considérer comme étant un générateur de collecteurs pour kTBS.

Dans la section suivante, nous présenterons la définition de quelques concepts liés aux traces. La section 3 sera réservée au détail de notre processus d'importation de traces en expliquant chacune de ses étapes : réécriture, création du modèle de trace et Mapping. En section 4 nous présentons xCollector, l'outil matérialisant ce processus, nous décrivons ses fonctionnalités es-

sentielles et nous présentons un cas réel de son utilisation. La section 5 sera la conclusion de cet article.

2 Concepts autour des traces

Dans cette section, nous présentons les définitions proposées par le laboratoire LIRIS sur le concept de trace, ainsi que les concepts afférents.

Selon notre approche (Champin *et al.*, 2013), une *trace* est composée d'éléments temporellement situés appelés *obsels* (pour observed element / élément observé). Ces derniers sont des éléments considérés comme potentiellement porteurs de sens dans l'activité tracée. Un obsel est constitué essentiellement : d'un type rattachant cet obsel à une catégorie explicite d'éléments observés, et d'un ensemble d'*attributs* de la forme <attribut : valeur>.

Une *m-trace* (pour modeled trace / trace modélisée) est une trace associée à un modèle qui en fournit un guide de construction et de manipulation. Un modèle de trace doit définir :

— la manière de représenter le temps,
— les types d'obsels permettant de décrire l'activité,
— pour chaque type d'obsel, les types d'attributs possibles,
— les types de *relations* binaires que peuvent entretenir les obsels entre eux.

Un Système de Gestion de Bases de Traces (*SGBT*) est un outil informatique permettant manipulations et transformations de traces modélisées. En effet, un SGBT joue le même rôle qu'un SGBD (Système de Gestion de Bases de données) dans les applications standard, mais gère plutôt des traces modélisées (m-traces). Le SGBT est alimenté par un ensemble de collecteurs, dont le rôle est de récolter les informations nécessaires à la constitution des traces.

kTBS est un système informatique mettant en pratique la notion de SGBT, développé au sein du laboratoire LIRIS. kTBS propose plusieurs fonctionnalités de gestion des traces, à savoir la création, l'interrogation, la transformation et la visualisation. kTBS utilise le formalisme RDF (Schreiber & Raimond, 2014) pour décrire les traces et les modèles, et expose ses données dans différents formats (XML, JSON, Turtle).

3 Importation des traces

A l'instar du LIRIS, l'exploitation des traces intéresse de plus en plus d'utilisateurs dans des disciplines variées. En effet, les traces sont largement considérées comme des sources de connaissances et de raisonnement. Ainsi, dans une perspective de capitaliser les traces, d'en faire des déductions et des interprétations, les SGBT sont proposés. Si les potentialités des SGBT sont suffisantes pour motiver leur utilisation, leur accessibilité et leur utilisabilité pourraient rebuter certains utilisateurs. Ce travail est donc une manière de faciliter la tâche aux utilisateurs pour importer leurs traces dans le SGBT kTBS. Comme mentionné dans l'introduction, nous entendons par utilisateur toute personne ne voulant et/ou ne pouvant pas programmer directement son collecteur. Il n'est donc pas demandé à cet utilisateur d'avoir des compétences en programmation pour profiter de ce que nous proposons. Il doit cependant avoir certaines connaissances de base notamment sur : la structure XML, les caractères séparateurs dans les fichiers texte et la manière d'accéder aux BDDRs, si besoin est.

Pour chaque système tracé, le processus d'importation de traces que nous proposons est composé de deux étapes : une étape de conception et de création (génération) de collecteur et une étape d'utilisation de ce dernier (l'importation en elle-même). Trois types de traces sont gérés par ce processus : texte, XML et BDDR.

3.1 Génération du collecteur

Le rôle essentiel des collecteurs que le générateur doit produire est de construire des traces importables à kTBS à partir des traces existantes. Les traces importables sont des traces conformes et associées à des modèles kTBS de traces. Le problème revient donc à créer un modèle de trace pour chaque collecteur et à construire des traces qui lui sont conformes à partir de celles fournies en entrée. La question qui se pose est : comment former des traces kTBS à partir d'exemples de traces externes ? Pour répondre à cette question, on propose de définir des règles de transformation qui vont permettre de chercher et calculer les éléments de la trace kTBS à partir de la trace fournie en entrée. L'utilisateur intervient dans ce processus pour montrer, à travers des exemples de traces, où sont les données à transformer et en quoi on devrait les transformer. La définition de règles de Mapping entre les éléments de la trace externe et les éléments du modèle de trace est à la base de ce processus. Le Mapping sert ainsi à définir les correspondances entre tous les éléments du modèle kTBS de trace et leurs instances dans les traces en entrée. Il doit être effectué de la façon la plus générale possible pour qu'il reste pertinent quelles que soient les traces introduites. Afin de pouvoir créer des modèles kTBS de traces accompagnés des règles de Mapping capables de convertir des traces externes en traces importables respectant ce modèle, un processus en trois étapes est proposé : la réécriture, la création du modèle et le Mapping (Figure 1).

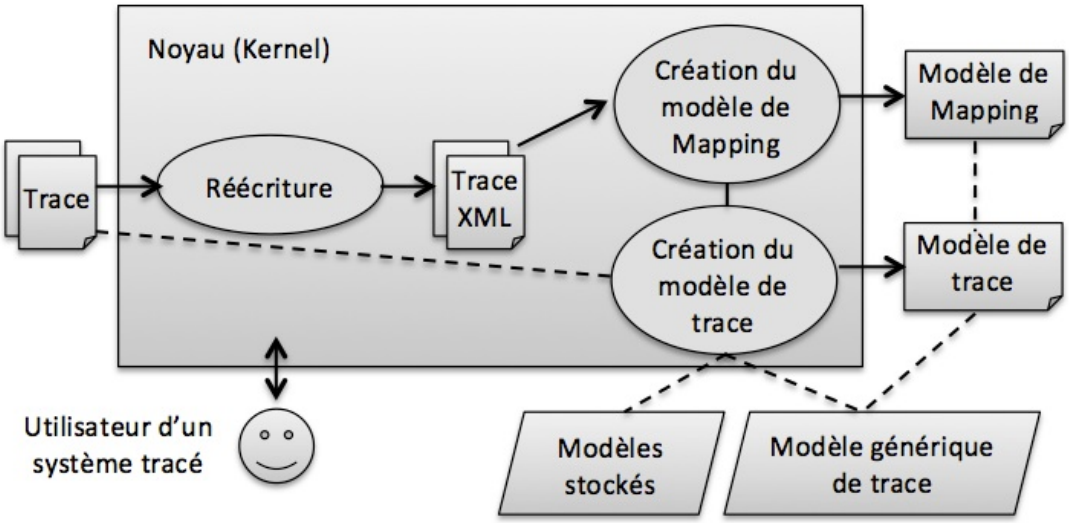


FIGURE 1 – Processus de génération des collecteurs

3.1.1 Réécriture

Compte-tenu des constatations effectuées en introduction, trois formats de traces sont essentiellement distingués. Pour simplifier le processus d'importation des traces à kTBS et éviter un traitement spécialisé pour chaque format, nous avons choisi de considérer XML comme format pivot. Autrement dit, les traces dans d'autres formats seront réécrites en XML, ce qui va ainsi unifier le processus d'importation. La réécriture est un processus automatique dépendant du format de la trace. Elle consiste à transformer les traces textuelles en XML en respectant des contraintes de séparation du texte définies par l'utilisateur. Pour les traces en BDDR, elle consiste à les transformer en XML en prenant en compte leur structure dans la BDDR. XML (W3C, 2008) est un standard largement utilisé pour sérialiser des contenus structurés (processus de transformation des objets en flux de données). Ce qui motive aussi le choix de XML, ce sont les possibilités qu'il offre en termes d'outils d'interrogation, de transformation et de repérage. La structure arborescente d'XML rend (à notre avis) son contenu facilement accessible et même plus lisible dans certains cas de traces.

La réécriture est une étape préparatoire par rapport au Mapping. Son objectif peut être, selon le format d'entrée, la structuration ou la restructuration des traces avant leur chargement. Ainsi, la réécriture va consister à :

— Structurer la trace dans un format XML s'il s'agit d'un texte : les lignes du texte deviennent des balises <ligne> XML dont les fils sont des balises <cellule> contenant chacune un des items de la ligne texte en question.

— Restructurer la trace dans un format XML s'il s'agit d'une source de BDDR (table ou requête) : les lignes de la source de BDDR deviennent des balises <ligne> XML dont les attributs et leurs valeurs sont les noms des colonnes avec les valeurs de la ligne de la source BDDR en question.

3.1.2 Création du modèle kTBS de trace

D'après la définition donnée dans la section 2, un modèle de trace dans kTBS est une description de trace exprimée dans un langage fondé sur RDF, définissant des types d'obsels, des types d'attributs d'obsels et des types de relations pouvant exister entre les obsels. Pour faciliter la rédaction du code RDF décrivant ces éléments du modèle par l'utilisateur, nous proposons un moyen graphique pour le faire. Dans cette même perspective d'aider l'utilisateur durant le processus de création du modèle, un exemple de trace externe peut être visualisé. La réutilisation de modèles kTBS existants ainsi que leur modification pour en créer d'autres est un moyen proposé pour construire facilement des modèles kTBS de traces.

3.1.3 Mapping

Dès que la trace chargée par l'utilisateur ou sa copie réécrite est visualisée, et dès que l'utilisateur commence à construire son modèle de trace (éventuellement guidé par le contenu de la trace) en définissant ses éléments, il peut parallèlement commencer à dresser des liens sémantiques entre les éléments de sa trace et des éléments de son modèle en construction. La sémantique de ces liens est de la forme "est de type". Si on suppose qu'un lien <eltXML - eltModel> est dressé, ceci s'interprète par "eltXML est de type eltModel".

Pour mieux justifier l'utilité de tels liens, il est utile de rappeler le but recherché, i.e. un ensemble de règles de conversion des traces externes en traces importables au kTBS. Ainsi, ces règles doivent être capables de repérer des instances des éléments du modèle dans la trace externe et aussi de les convertir en instances dans la trace résultat. Les liens que l'utilisateur est invité à faire ont donc pour objectif la définition des éléments pertinents dans la trace externe. Il reste au générateur de collecteurs à généraliser ces définitions pour prendre en compte les variations éventuelles entre les exemples de traces externes.

À ce propos, plusieurs travaux ont été menés dans le contexte de l'annotation dans le web, dont l'objectif était de générer des repères (expressions xPath) dans des pages web correspondant aux clics de l'utilisateur (Abe & Hori, 2003; Kitano *et al.*, 2010; Kowalkiewicz *et al.*, 2006; Paz & Díaz, 2010), qui soient robustes aux changements les plus fréquents dans la structure de ces pages web. Effectivement, xPath (W3C, 2010) - le premier outil XML de pointage - n'a pas seulement prouvé ses capacités de repérage mais aussi il en donne des choix multiples.

Dans notre contexte le problème devient : comment générer des expressions xPath correspondant au choix de l'utilisateur (éléments XML sélectionnés) qui restent valides pour les futurs exemples de traces XML ? Une expression <Exp> créée à partir d'un élément sélectionné <Elt> qui correspond au type <Type> dans le modèle de trace est valide si et seulement si elle repère tous les éléments de type <Type>. Les liens que l'utilisateur doit dresser et que nous avons représentés par des couples <eltXML - eltModel> auront la forme <ExpressionXPath - Type> où Type est un type d'obsel ou d'attribut dans le modèle.

Dans les travaux de Abe & Hori (2001, 2003) et Hori *et al.* (2003b,a, 2004), plusieurs modèles d'expressions xPath sont discutés :

— Les expressions "absolues" qui effectuent le repérage en suivant la hiérarchie du document XML de la racine jusqu'au nœud cible, avec la forme : */Racine/fils/ ... /eltSélectionné*.
— Les expressions "relatives" qui repèrent un nœud par rapport à des positions d'ancres stables, avec la forme : *ExpressionAncre//eltSélectionné [Distance]*.
— Les expressions "conditionnelles" repérant les nœuds par le biais d'une condition à satisfaire, avec la forme : *//eltSélectionné [Condition]*.

Quant à nous, nous introduisons la notion de "contexte" qui est à la base de nos modèles d'expressions xPath. Un contexte est une expression xPath repérant un endroit voisin d'un élément instance (obsel ou attribut) du modèle de trace ou d'un autre contexte. Pour donner plus de robustesse à nos expressions, nous avons aussi proposé la combinaison d'expressions relatives et conditionnelles. Globalement, nos expressions ont la forme suivante : *ExpressionContexte //eltSélectionné [Condition]*. Nous avons utilisé ce modèle d'expression pour exprimer des repères, d'instances de types d'obsel, d'instances de types d'attributs et aussi pour définir des contextes.

Concrètement, après que l'utilisateur ait défini son Mapping entre un élément XML et un élément du modèle, notre système propose un nombre de suggestions xPath avec des conditions variées selon le contenu de la trace, pour désigner l'élément XML sélectionné. Ces conditions peuvent porter sur :

— La valeur de l'élément sélectionné, par exemple : *//N [text() = "Exercice"]* pour choisir tous les nœuds N dont le texte est égal à "Exercice".
— Les valeurs des attributs de l'élément sélectionné, par exemple : *//N [@action = "Envoyer*

Msg"] pour choisir tous les nœuds N dont l'attribut "action" égale "Envoyer Msg".
- La valeur du texte ou des attributs des fils de l'élément sélectionné, par exemple : *//N [./time/@begin = "10 :00"]* pour choisir tous les nœuds N dont l'attribut "begin" du fils "time" est égal à "10 :00".
- Le type de valeur de l'élément sélectionné, par exemple : *//N [matches(@option, Numérique)]* pour choisir tous les nœuds N dont le type de l'attribut "option" est numérique.
- La position de l'élément sélectionné, par exemple : *//P/N [2]* pour choisir tous les nœuds N dont la position est égale à 2 par rapport au nœud parent P.

Les propositions xPath repérant un élément sélectionné par l'utilisateur sont générées par l'algorithme présenté ci-dessous. Son principe est de générer des expressions xPath prenant en compte le nom de l'élément sélectionné, sa valeur, le type de sa valeur, sa position, les valeurs de ses attributs et le contenu de ses fils.

Algorithm 1 Génération des expressions xPath

Input : $SelectedElt$, trace element selected by user
Output : $Expressions$, the generated xPath expressions
$ExpName \Leftarrow \phi$
$Conditions \Leftarrow \phi$
$ExpName.Create(SelectedElt.name)$
if $(SelectedElt.attributes \neq \phi)$ **then**
  for (each attribute in $SelectedElt.Attributes$) **do**
    $Conditions.addValue(attribute.value)$
  end for
else
  $Conditions.addValue(SelectedElt.value)$
end if
for (each child in $SelectedElt.Childs$) **do**
  if $(child.Attributes \neq \phi)$ **then**
    for (each attribute in $child.Attributes$) **do**
      $Conditions.addValue(attribute.value)$
    end for
  else
    $Conditions.addValue(child.value)$
  end if
end for
$Conditions.addType(SelectedElt.valueType)$
$Conditions.addPosition(SelectedElt.position)$
$Expressions.add(ExpName)$
for (each condition in $Conditions$) **do**
  $Expressions.add(ExpName + condition)$
end for

Cet algorithme produit des expressions de la forme : *//eltSélectionné [Condition]*. L'utilisateur aura le choix d'utiliser un contexte pour repérer sa sélection ou pas. Si un contexte est

utilisé, le repérage de l'élément sélectionné devient relatif à ce contexte et l'expression devient : *ExpressionContexte//eltSélectionné [Condition]*. À son tour, ExpressionContexte a la forme : *[ExpressionContexte'] //eltSélectionné'*, etc. Dans certains cas de traces, la combinaison de plusieurs types de conditions peut avoir de la valeur afin de concevoir des expressions pertinentes. Selon le contenu des traces traitées, on peut imaginer des expressions xPath : (1) avec une seule condition (avec/sans contexte) ou (2) avec plusieurs conditions combinées (avec/sans contexte).

En effet, les expressions xPath sont proposées à l'utilisateur après une transcription en langage naturel. Pour choisir une proposition ou un contexte donné, on propose aussi à l'utilisateur de se fonder sur l'effet de son choix d'expression. Autrement dit, quand l'utilisateur choisit une (ou plusieurs) propositions, le système lui indique les éléments repérés par cette (ou ces) proposition(s), ce qui permet à l'utilisateur de juger de la pertinence de son choix. Ce choix est donc fondé d'une part sur la formulation en langue naturelle des expressions xPath et d'autre part sur l'effet de ces expressions pour repérer les éléments de la trace que l'utilisateur souhaite récupérer.

À travers un ou plusieurs exemples, l'utilisateur va pouvoir créer son modèle kTBS de trace, définir tous les Mapping nécessaires et donc choisir toutes les expressions efficaces qui vont servir à la conversion des traces externes en traces importables. Les couples <ExpressionXPath - Type> (le modèle de Mapping) ainsi produits et le modèle kTBS de trace forment une partie importante du collecteur cherché.

3.2 Utilisation du collecteur

Les modèles de Mapping et de traces sont des unités statiques incapables à elles seules de convertir ou de collecter des traces. Le Module Kernel intégré dans notre modèle sert à compléter ces deux modèles pour former de véritables collecteurs. Le rôle du Kernel est d'assurer deux fonctionnalités : (1) la réécriture des traces externes en traces XML et (2) l'utilisation du modèle de Mapping pour créer des traces kTBS puis les associer à leur modèle pour qu'elles soient importables. Les collecteurs résultants sont ainsi définis par le triplet : <ModelMapping - ModelTrace - Kernel> (Figure 2).

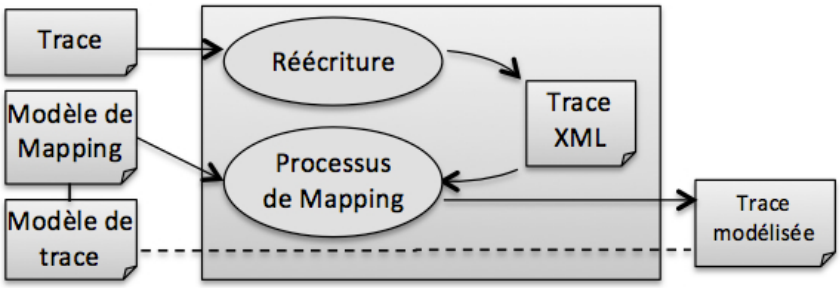


FIGURE 2 – Utilisation du collecteur

À la présentation d'une nouvelle trace externe, le kernel identifie le type de la trace puis procède éventuellement à sa réécriture en exploitant les informations de réécriture (le type de la trace, les séparateurs, le type de codage du texte, le chemin de la BDDR, etc.) déjà définis lors de la création du collecteur. Le kernel commence ensuite à utiliser le modèle de Mapping

et à chercher des occurrences de chaque type appartenant au modèle de Mapping. Il parcourt les types du modèle et en cherche toutes les occurrences présentes dans la trace. La recherche est simplement faite en lançant les expressions xPath correspondantes. Chaque occurrence trouvée est affectée à la valeur de l'obsel ou de l'attribut correspondant au type en question. La trace kTBS se construit ainsi progressivement en détectant tous les obsels et attributs existants dans la trace et en en déduisant les valeurs correspondantes. De cette façon le collecteur construit des traces kTBS conformes à leurs modèles et prêtes à être importées au kTBS.

4 L'outil xCollector

4.1 Présentation

xCollector est un outil java destiné aux utilisateurs voulant importer à kTBS des traces texte, XML ou BDDR. Via son interface graphique, xCollector permet de créer des modèles kTBS de trace et de générer des repères (expressions) xPath pour identifier, dans les traces externes, les éléments nécessaires pour créer de nouvelles traces importables conformes à leurs modèles. xCollector propose diverses fonctionnalités :

— Le chargement des traces externes : les traces de l'utilisateur sont chargées et visualisées sous forme d'arbre XML avec un habillage proche de l'original. xCollector affiche les traces texte sous forme de lignes de texte et les traces BDDR sous forme de lignes de champs identiques. Lors du chargement, des données doivent éventuellement être renseignées : le format de la trace, les séparateurs pour les traces texte, la requête/la table et la BDDR pour les traces stockées dans des BDDR, etc. En effet, ces données serviront à choisir et à lancer le processus automatique de réécriture lors de l'utilisation des collecteurs.

— La création du modèle kTBS de trace : sans se soucier de coder en RDF ni de connaître la syntaxe kTBS pour définir des modèles de trace, l'utilisateur peut facilement créer son modèle de trace en créant des types d'obsels, d'attributs et de relations. xCollector affiche sous forme arborescente le modèle de trace et permet la sélection de ses éléments. A tout moment, il est possible de générer et d'accéder au modèle créé au format RDF-turtle.

— La création du modèle de Mapping : ce processus doit être initié par l'utilisateur en établissant des liens entre des éléments de la trace et des éléments du modèle. Un lien est défini si l'utilisateur sélectionne, d'un côté un élément XML de la trace et de l'autre coté un élément de l'arbre du modèle. Selon le type, le contenu et le voisinage de l'élément de trace sélectionné, le système propose un ensemble de références pour cet élément. Afin de connaitre les éléments XML repérés par une référence donnée et pour faciliter ainsi le choix de celle-ci, xCollector utilise une métaphore de drapeau levé sur les éléments correspondant à cette référence. L'utilisateur choisit alors la proposition si et seulement si un drapeau est levé sur toutes les instances du type (d'obsel ou d'attribut) en question. En effet, chaque proposition est une expression xPath avec une condition différente. Le nombre et le type de ces conditions dépend alors du type, du contenu et du voisinage de l'élément sélectionné. Si, par exemple, le nœud sélectionné contient un attribut, une expression formée d'une condition sur la valeur de cet attribut est générée. Si cette valeur est numérique, une autre expression considérant ce type de valeur est générée. S'il y a plusieurs nœuds frères du nœud sélectionné ayant le même nom, une expression pre-

nant en compte l'ordre de l'élément sélectionné est générée, etc. L'utilisateur a aussi le choix de repérer un élément sélectionné par rapport à un contexte précédemment défini. Une expression correspondant à un type d'obsel ou d'attribut donné peut être utilisée en tant que contexte. En dehors des types du modèle, l'utilisateur peut définir des contextes propres pour améliorer le repérage. Leur création suit les mêmes étapes : sélection, utilisation d'un contexte, choix de la proposition. Après sa génération, l'expression xPath, couplée avec le type sélectionné du modèle, est stockée dans un fichier XML représentant le modèle de Mapping.

4.2 Exemple d'utilisation

Le système GeoNotes (Sanchez, 2006) produit des traces textuelles composées de lignes contenant chacune une liste d'informations séparées par des virgules. Ces informations correspondent aux traitement effectués sur des images géographiques et ce à des fins pédagogiques. Le nombre et le type des informations est variable dans chaque ligne. Après le chargement d'une trace GeoNotes, xCollector l'affiche dans son interface sous forme d'arbre XML mais avec une apparence proche du format texte (cf. Figure 3). Pour pouvoir faire des Mapping, au moins un élément doit être créé dans le modèle de trace. Supposons la création d'un type d'obsel appelé "zoom" : pour le faire il suffit d'introduire l'identificateur "zoom" ; et éventuellement un super type (s'il y a héritage). Ce type d'obsel nécessite un attribut "chemin" pour stocker le chemin de l'image à zoomer. En sélectionnant l'obsel "zoom", l'insertion d'un attribut "chemin" se fait en introduisant son identifiant, label et type (Figure 3). Il y a alors deux éléments (zoom et chemin) pour lesquels on peut définir des Mapping. Afin de repérer le type "zoom", il suffit qu'on choisisse un des éléments XML de la trace correspondant à ce type, et le système génère les propositions d'expressions xPath. On choisit dans ce cas la proposition qui prend tous les éléments dont l'attribut @txt contient l'information "zoom" (drapeau jaune dans Figure 4). L'utilisateur confirme l'enregistrement de la proposition (expression xPath) choisie en tant que règle de Mapping pour le type "zoom" (apparaissant dans Figure 3). Pour utiliser le type "zoom" comme contexte pour le type d'attribut "chemin", il doit être chargé afin de former une expression de contexte pour le type "chemin". Après sélection d'un élément XML exemple pour le type "chemin", des propositions sont encore générées, parmi lesquelles l'utilisateur choisit celle qui prend tous les nœuds fils du contexte "zoom" nommés Cell et se trouvant en quatrième position (drapeaux rouges dans Figure 4). Ainsi, les drapeaux jaunes indiquent les contextes (obsels "zoom"), les drapeaux rouges indiquent les attributs "chemins". Les règles de Mapping et le modèle kTBS de trace sont visualisés dans la partie droite de l'interface.

5 Conclusion

Les SGBT sont des plateformes susceptibles d'intéresser beaucoup d'acteurs intervenant sur des systèmes tracés, étant données les fonctionnalités qu'ils proposent. Pour pouvoir profiter de ces fonctionnalités de traitement et d'analyse des traces, les utilisateurs doivent actuellement faire l'effort de programmer des collecteurs pour importer leurs traces dans un SGBT. xCollector, le fruit de notre travail, propose une manière interactive de construire des collecteurs propres aux systèmes tracés, sans recours à la programmation. xCollector a été conçu et réalisé suite à une étude d'un corpus de traces de systèmes variés. Nous avons ensuite exploité ce

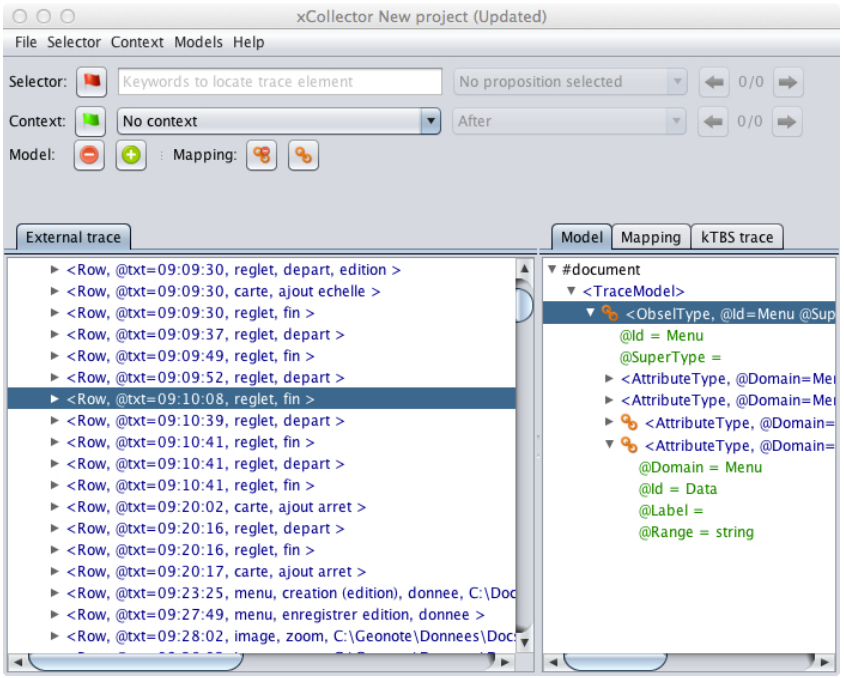


FIGURE 3 – xCollector, visualisation et création de modèle

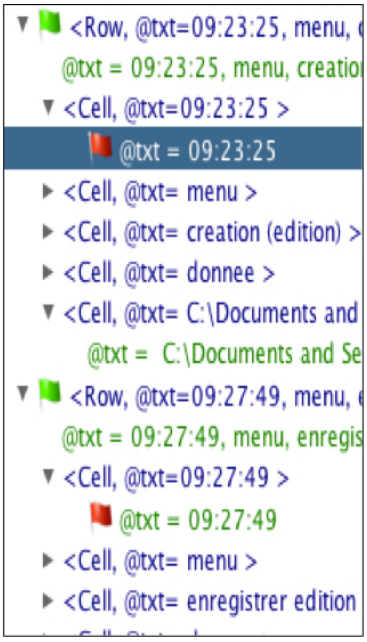


FIGURE 4 – xCollector, Mapping

corpus pour effectuer des tests montrant la validité et la généricité de l'approche choisie. Nous avons évalué la généricité de xCollector en l'utilisant pour collecter les traces de systèmes qui ne figuraient pas dans notre corpus initial, et nous avons également évalué son utilisabilité auprès d'un ensemble d'utilisateurs souhaitant importer leurs traces dans kTBS. Ces tests ont été conduits auprès de six utilisateurs ne pouvant et/ou ne voulant pas programmer des collecteurs pour leurs logiciels. Nous avons testé l'importation des traces de quatre logiciels pour lesquels les traces étaient au format texte, deux au format XML et un au format BDDR. Les utilisateurs ont pu essayer les fonctionnalités essentielles de xCollector, comme le repérage des données dans les traces chargées et leur Mapping avec les éléments du modèle de traces. xCollector a montré la validité de l'approche que nous avons choisie pour l'importation de traces, et a répondu aux principaux besoins des utilisateurs. Cependant d'autres améliorations notamment en ergonomie sont à faire, pour faciliter encore la prise en main de xCollector par les utilisateurs visés. D'autres fonctionalités restent aussi à développer pour augmenter le nombre de logiciels pour lesquels il pourra être utilisé.

Références

ABE M. & HORI M. (2001). A visual approach to authoring xpath expressions. *Markup languages theory and practice*, **3**(2), 191–212.

ABE M. & HORI M. (2003). Robust pointing by xpath language : Authoring support and empirical evaluation. In *Applications and the Internet, 2003. Proceedings. 2003 Symposium*, p. 156–165 : IEEE.

BOUHINEAU D., LUENGO V., MANDRAN N., ORTEGA M., WAJEMAN C. *et al.* (2013). Open platform to model and capture experimental data in technology enhanced learning systems. In *Workshop Data Analysis and Interpretation for Learning Environments*.

CHAACHOUA H., CROSET M.-C., BOUHINEAU D., BITTAR M., NICAUD J.-F. *et al.* (2007). Description et exploitations des traces du logiciel d'algèbre aplusix. *Revue STICEF*, **14**.

CHAMPIN P.-A., MILLE A. & PRIÉ Y. (2013). Vers des traces numériques comme objets informatiques de premier niveau : une approche par les traces modélisées. *Intellectica*, **1**, 59.

CHOQUET C., IKSAL S. *et al.* (2007). Modélisation et construction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un eiah. *Revue STICEF*, **14**.

HORI M., ABE M. & ONO K. (2003a). Extensible framework of authoring tools for web document annotation. In *Proceedings of International Workshop on Semantic Web Foundations and Application Technologies (SWFAT)*, p. 1–8 : Citeseer.

HORI M., ABE M. & ONO K. (2003b). Robustness of external annotation for web-page clipping : Empirical evaluation with evolving real-life web documents. In *Semannot 2003 : Knowledge Markup and Semantic Annotation*, Sanibel, Florida, USA.

HORI M., ONO K., ABE M. & KOYANAGI T. (2004). Generating transformational annotation for web document adaptation : tool support and empirical evaluation. *Web Semantics : Science, Services and Agents on the World Wide Web*, **2**(1), 1–18.

JI M., MICHEL C., LAVOUÉ E. & GEORGE S. (2013). An architecture to combine activity traces and reporting traces to support self-regulation processes. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, p. 87–91 : IEEE.

KITANO T., IGUCHI K. & KOYAMA K. (2010). Generating robust xpaths for service customization. In *6th World Congress on Services (SERVICES-1)*, p. 166–167 : IEEE.

KOWALKIEWICZ M., KACZMAREK T. & ABRAMOWICZ W. (2006). Myportal : robust extraction and aggregation of web content. In *Proceedings of the 32nd international conference on Very large data bases*, p. 1219–1222 : VLDB Endowment.

MAY M., GEORGE S. & PRÉVÔT P. (2008). A closer look at tracking human and computer interactions in web-based communications. *Interactive Technology and Smart Education*, **5**(3), 170–188.

PAZ I. & DÍAZ O. (2010). Providing resilient xpaths for external adaptation engines. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, p. 67–76 : ACM.

SANCHEZ, OLIVIER LEFEVRE E. (2006). Geonote : un environnement informatique d'aide au travail sur le terrain pour l'enseignement des sciences de la terre. In *Biennale de l'éducation*, Lyon.

SCHREIBER G. & RAIMOND Y. (2014). Rdf 1.1 primer, w3c working group note 24 june 2014. http ://www.w3.org/TR/rdf11-primer/.

W3C (2008). Extensible markup language (xml). `http://www.w3.org/TR/xml/`.

W3C (2010). Xml path language (xpath) version 2.0. `http://www.w3.org/TR/xpath20/`.

Assistance à la découverte de connaissances contextuelles à partir de l'analyse des traces

Assitan Traoré[1, 2], Alain Mille[1] et Hélène Tattegrain[2]

[1] IFSTTAR, LESCOT
{assitan.traore, helene.tattegrain}@ifsttar.fr
http://www.ifsttar.fr
[2] Université Lyon 1, LIRIS, CNRS UMR 5205
{assitan.traore, alain.mille}@univ-lyon1.fr
http://liris.cnrs.fr

Résumé : L'observation d'une activité pour comprendre des comportements particuliers nécessite de discerner ce qui relève ou non du contexte. En intelligence artificielle la notion de contexte est une approche modale du raisonnement et plusieurs études ont été faites pour proposer des modèles génériques de représentation des connaissances *contextuelles*. La question de la découverte interactive des connaissances contextuelles et de leur inscription dans un modèle générique n'est toutefois que rarement examinée dans la littérature. Cette communication propose une approche exploitant le potentiel de la représentation de l'activité par ses traces modélisées d'interaction. Elle consiste à assister la découverte du contexte explicatif par une approche interactive associant l'analyse des données, les connaissances de l'analyste et leur mise en situation dans les traces modélisés. Cette approche interactive facilite l'élicitation de ce qui est contexte ou non. De plus, nous montrons qu'il est alors possible de représenter le contexte découvert sous une forme générique telle que proposée dans la littérature spécialisée. La méthode a été implémentée et a permis la découverte de contextes explicatifs pour la consommation en carburant lors de l'activité de conduite automobile. Les expérimentations, les données et les analyses ont été menées à l'IFSTTAR dans des conditions de conduite réelles.

Mots-clés : Découverte interactive de connaissances, contexte, traces modélisées, transport, consommation en carburant.

1 Introduction

La question de l'ingénierie des connaissances *contextuelles* se pose de manière aigüe dans le cadre de l'analyse d'activités complexes comme les comportements de conduite automobile, se déroulant dans des environnements variés, avec de fortes dynamiques temporelles et spatiales. C'est alors le *contexte* qui permet d'expliquer les différences entre divers comportements pour réaliser une activité similaire. Ce cadre d'analyse de l'activité de conduite centré sur l'explication de la consommation de carburant a été utilisé pour étudier et proposer une méthode de découverte interactive de contexte à partir d'une analyse de l'activité tracée selon l'approche des *traces modélisées*. L'article précise tout d'abord le cadre applicatif du travail avant de réaliser l'état de l'art sur la notion de contexte, de sa modélisation et de sa découverte. L'approche est ensuite décrite avec ses différentes étapes de la préparation des données à la découverte des éléments de contexte « candidats ». Cette approche a été appliquée sur un jeu de données dans le domaine de la conduite automobile, et la première analyse menée avec l'aide de cette approche est décrite. Cette analyse montre des résultats très encourageants, démontrant le potentiel de cette approche de découverte interactive d'éléments de contexte.

2 Cadre du travail de recherche

Ce travail est effectué en collaboration entre le LIRIS et l'IFSTTAR dans le but d'étudier et d'expérimenter une méthodologie de découverte de connaissances contextuelles en l'appliquant à l'explication d'un critère de classification de segments temporels d'une activité.

2.1 Enjeux sociaux pour la recherche dans les transports

Le phénomène du réchauffement climatique est attribué, entre autres, aux gaz à effet de serre produits par les activités humaines l'industrie, le transport, le résidentiel-tertiaire et l'agriculture. L'un des objectifs principaux de l'IFSTTAR est de réduire l'impact environnemental du transport. Les études environnementales montrent que le transport est l'un des principaux responsables du réchauffement climatique (RAC-F, 2007). Plusieurs études ont été menées pour remédier à ce problème environnemental. Elles ont vulgarisé en 1999 le terme d'Eco-conduite avec des systèmes d'assistance de conduite comme: ISA, LAVIA (Rakotonirainy et al., 2011). Des études récentes sur les impacts des systèmes d'Eco-conduite et d'assistance à la conduite sur la consommation montrent que ces systèmes sont moins efficaces dès que la vitesse limite est supérieure à 80km/h (par exemple sur l'autoroute) et que les cours d'Eco-conduite ne réduisent la consommation que pendant une durée courte après l'apprentissage du conducteur (Beusen et al., 2009). D'autre part, cet impact environnemental est amplifié par l'augmentation constante du parc automobile, avec en France, 5% d'augmentation sur les 10 dernières années (MEDDE, 2014). D'autres pistes sont ouvertes pour proposer des systèmes d'assistance capables de réduire la consommation de carburant de façon fiable et durable dans des contextes variés.

2.2 Présentation de la question de recherche

« Le contexte est toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité peut être une personne, un endroit ou un objet que l'on considère pertinent dans l'interaction entre un utilisateur et une application, y compris l'utilisateur et l'application eux-mêmes » (Dey, 2001). D'après cette définition du contexte de Dey, pour caractériser une activité il faut tenir compte des informations relatives aux entités (les éléments en contexte) qui sont impliquées dans la réalisation de cette activité. Mais ces informations sont inconnues au début de l'activité et ne sont définies que pendant la réalisation de cette dernière. Par exemple, pour modéliser les comportements de conduite réduisant la consommation en carburant, il faut tenir compte des entités impliquées dans l'activité de conduite comme le véhicule, le conducteur et l'environnement de conduite et déterminer les valeurs prises par ces entités lorsque la consommation en carburant est faible ou forte pendant la conduite. Les informations prises par les entités de l'activité correspondent aux facteurs dont dépend la consommation en carburant comme le type du véhicule, l'état du véhicule, le mode de conduite du conducteur, l'infrastructure, le trafic. Ces facteurs décrivent les entités de l'activité de conduite et ils permettent d'avoir une définition précise de l'activité à un moment précis ou sur un segment temporel donné. Ils pourront être utilisés pour expliquer des comportements précis ou des phénomènes observés pendant l'activité. Les éléments contexte sont difficiles à définir en amont car ils sont dynamiques. Pour les définir nous proposons un processus de découverte à partir de l'analyse des traces (Georgeon et al., 2012 ; Mathern, 2012) associée à des méthodes de fouille de données et à l'expertise de l'analyste.

2.3 Présentation des données utilisées

Les données utilisées dans ce travail ont été collectées lors de l'activité de conduite de plusieurs participants pendant plusieurs mois. Ces données ont été recueillies en conduite *naturelle* avec des véhicules équipés de capteurs (bus CAN, radar, GPS et cameras).

3 Etat de l'art de la notion de contexte et de l'ingénierie des connaissances contextuelles

3.1 La notion de contexte en intelligence artificielle : une histoire ancienne qui pose la question d'une ingénierie spécifique des connaissances contextuelles

La notion de contexte a été étudiée très tôt en intelligence artificielle par John McCarthy, (McCarthy, 1993). Dans cet article, la relation de base *ist(c,p)* établit que la proposition *p* est vraie dans le *contexte c.* Ceci permet d'étendre ou de restreindre un raisonnement en fonction d'un contexte, qu'il faut alors établir. Le contexte porte un nom *« c »* et est défini en extension par les variables qui sont contenues dans les propositions qui permettent de produire une inférence valide dans ce contexte. La liste des variables à mobiliser dans ce contexte constituent, d'une certaine façon, le contexte en extension. Comment identifier ces variables ? Ces travaux ont été rapidement approfondis et comparés et le lecteur intéressé trouvera une étude comparative très bien documentée dans (Akman et al., 1996). Comment savoir à quelles hypothèses répondent tels ou tels contextes ? En pratique, il est ressorti rapidement que la définition d'un contexte était problématique et soulevait aussi bien des questions de complexité de calcul que de difficultés à les définir. La complexité de l'ingénierie de cette modélisation pour la représentation des connaissances dans le système CYC (Lenat, 1998) a imposé une méthode de réduction de complexité. En effet, une grande variété de combinaisons de valorisations de variables permet de définir un contexte, et chaque contexte doit être « affirmé » en examinant la validité des hypothèses de validité des règles que ses variables peuvent (ou non) déclencher. Pour résoudre cette question, Lenat, dès 1998, propose de décrire « presque » n'importe quel contexte selon « seulement » 12 dimensions qui permettraient de décrire 99 % des contextes qu'il a imaginés. Ces dimensions sont (nous reprenons la liste en anglais telle qu'elle a été publiée):

- *Absolute Time: a particular time interval in which events occur*
- *Type Of Time: a non-absolute type of time period, such as "just after eating"*
- *Absolute Place: a particular location where events occur, such as "Paris"*
- *Type Of Place: a non-absolute type of place, such as "in bed"*
- *Culture: linguistic, religious, ethnic, age-group, wealth, etc. of typical actors*
- *Sophistication/Security: who already knows this, who could learn it, etc.*
- *Topic/Usage: drilling down into aspects and applications – not subsets*
- *Granularity: phenomena and details which are (and are not) ignored*
- *Modality/Disposition/Epistemology: who wants/believes this content to be true?*
- *Argument-Preference: local rules for how to resolve pro-con argument disputes*
- *Justification: are things in this context generally proven, observed, on faith…*
- *Let's: local bindings of variables etc. that hold true in that context.*

Ces dimensions sont surtout descriptives et chaque définition possède des raffinements importants, en particulier pour la gestion du temps. En pratique, avant de lancer une inférence, il faut définir un « motif contextuel » (un intervalle, une zone géographique, une contrainte sur telle ou telle autre dimension) et sélectionner les règles d'inférence qui sont valides pour ce motif de contexte, ce qui laisse supposer que toute règle est annotée par le motif contextuel pour lequel elle est valide.

(Brézillon, 2000) fait le point des connaissances sur la notion de contexte en y associant les notions de révision, de raisonnement hypothétique, de raisonnement analogique également très étudiés en Intelligence Artificielle. Avec Fausto Giunchiglia (Giunchiglia et al., 1996 ; Bouquet et al., 2001) Patrick Brezillon anime la série des conférences CONTEXT[1] depuis 2007. Cette conférence a permis d'élargir considérablement l'étude de la notion de contexte à d'autres champs de recherche que l'intelligence artificielle. La section suivante se fonde principalement sur ces derniers travaux pour en faire un rapide état de l'art, en particulier sous

1 http://cyprusconferences.org/context2015/

l'angle de la modélisation, en l'analysant avec le prisme de l'ingénierie des connaissances pour l'appliquer au cas particulier de l'observation d'une activité.

3.2 La modélisation des connaissances contextuelles lors de l'observation d'une activité

La définition la plus reprise du contexte est celle de (Dey, 2001). Pour Dey une information est considérée comme du contexte si cette information permet de caractériser la *situation* d'une entité engagée dans une activité. Les entités engagées dans une activité sont les éléments qui expliquent spécifiquement une séquence particulière dans l'activité en question.

Par exemple dans un accident de la route le fait que le conducteur ait consommé récemment de l'alcool ou pas est une information relevant du contexte. En effet, elle permet de savoir si le conducteur était dans un état normal pour conduire ou non. Cette information caractérise la ***situation***[2] *de l'entité « conducteur » engagée dans l'activité de conduite au moment de l'accident.*

Cette définition est générale car, quel que soit le domaine, elle permet de déterminer les informations relevant du contexte. Si cette définition est retenue, la question de la modélisation reste entière. (Dey et al., 1999) proposent une plateforme de gestion des connaissances contextuelles en suivant cette approche méthodologique. (Gandon et al., 2004) proposent une variante de cette approche garantissant une plus grande interopérabilité. Ces travaux montrent que l'on saurait gérer les connaissances de contexte, y compris pour les exploiter pour des applications tierces. La question de leur découverte, posée comme difficile par tous les auteurs, n'apparaît toutefois pas dans les plateformes proposées.

En 2002, (Rey et al., 2002) définissent le contexte comme un espace infini d'informations évolutives, qui ne sont pas connues en avance. Ce caractère dynamique du contexte rend sa modélisation très difficile. Pour réduire cette complexité de représentation du contexte d'une activité, en 2005, (Bazire et al., 2005) proposent une décomposition générique d'une activité selon les principales entités qu'elle mobilise. Ce modèle représente les entités ou composants d'une activité et les relations qui peuvent exister entre eux pour représenter le contexte de l'activité. Le modèle générique de (Bazire et al., 2005) permet de catégoriser les informations candidates au contexte des entités d'une activité mais elle ne dit rien sur la manière de découvrir ces informations. Il s'agit d'une question d'ingénierie de la connaissance qui associe l'acquisition des connaissances auprès des experts et les méthodes cherchant à découvrir des connaissances dans les données caractéristiques de l'activité mobilisant ces connaissances.

Dans le cadre de ce travail, nous retenons cette représentation générique telle qu'elle est aujourd'hui reprise par la communauté CONTEXT, sans étudier sa validité, ce qui est, hors de notre question de recherche. Nous nous intéressons par contre à l'étude d'une ingénierie des connaissances qui soit adaptée à ce type de connaissance *contextuelle*. Dans un travail se référant à la question de l'ingénierie des connaissances contextuelles, (Castelli et al., 2008), les auteurs considèrent la manière de « produire » les informations contextuelles à partir de capteurs RFID, sans aborder la question de la construction du sens qui devra ensuite avoir lieu pour exploiter cette information estampillée « contexte ». Un travail plus orienté sur la découverte de connaissances contextuelles, (Ngoc et al., 2005) dans le cadre d'un middleware robotique, repose sur l'association de plusieurs méthodes d'apprentissage automatique à partir d'un jeu de capteurs et actionneurs avec une mise en correspondance avec une ontologie contextuelle pour l'interprétation. Les auteurs ne s'intéressent pas à la construction de l'ontologie. Un travail de même nature (Milea et al., 2008), mais plus général encore, propose une manière d'intégrer la médiation contextuelle dans un processus de découverte de connaissance. La notion de contexte y est bien définie et les connaissances contextuelles permettent de multiplier les points de vue sur les patterns découverts. La connaissance découverte est en relation avec le contexte qui lui est attribué, mais cette connaissance

2 Travaux argumentant sur la nécessité de formaliser les contextes revendiquent une approche « située » de la cognition, telle que Clancey l'a théorisé par exemple (McCarthy, 1993).

contextuelle doit venir de l'expert médiateur de la découverte de connaissance. Plusieurs travaux s'intéressent à la représentation ontologique des contextes avec par exemple le langage CoOL (Strang et al., 2003), tandis que des travaux européens (Mullins et al., 2008) se sont efforcés d'exploiter ces possibilités dans le cadre de l'interopérabilité ubiquitaire.

Nous n'avons pas repéré de travaux spécifiquement dédiés à la question de la découverte de connaissances contextuelles. Il est difficile pour un expert de nommer et qualifier les contextes explicatifs pour chaque situation observable dans une activité comme le montre le champ de recherche de l'IFSTTAR qui cherche à élucider les contextes explicatifs de comportement de consommation de carburant. L'observation instrumentée de l'activité produit des données suffisamment riches et nombreuses pour potentiellement expliquer les situations à étudier. La découverte de connaissances à partir de données, utilise des méthodes numériques (statistiques surtout) ou symboliques (algorithmes de fouille surtout) pour l'identification de motifs, avec une approche supervisée ou non, à partir de données collectées dans le monde « réel » observé. Cette approche mobilise différemment les connaissances de l'expert analyste qui est sollicité aussi bien pour la préparation des données que pour l'interprétation des motifs découverts. (Fayyad et al., 1996a, 1996b) proposent la définition suivante : « La découverte de connaissances dans des bases de données est le processus non trivial d'identification, dans les données, de motifs valides, nouveaux potentiellement utiles et ultimement compréhensibles.» et ils proposent également le cycle de ce processus de découverte de connaissance. Lorsque les données ne sont pas « données », mais collectées à la suite de l'observation d'une activité produisant des traces de cette activité, nous parlerons alors d'ingénierie des connaissances *tracées*, en considérant que le choix de ce qui est observé est déjà le fruit d'une connaissance explicite qu'il convient de modéliser dès la collecte. Dans le cadre de cet article, nous nous appuierons sur cette approche pour étudier les possibilités offertes pour assister l'ingénierie des connaissances contextuelles. Nous rappelons les travaux de notre équipe de recherche et l'environnement mobilisé pour cette étude.

3.3 Ingénierie des connaissances à partir de traces d'observation

L'ingénierie des connaissances à partir de traces d'observation est une forme d'ingénierie de découverte *dynamique* des connaissances, dans le sens où la connaissance s'établit dynamiquement et que cette dynamique est explicitement gardée comme explicative de telle ou telle interprétation. Les connaissances manipulées réfèrent à des observations situées dans le temps (obsels : observed elements). Le concept de trace modélisée a été développé par l'équipe SILEX du laboratoire LIRIS dans le but de construire des connaissances à partir de l'observation des interactions observables dans une activité (Laflaquiere et al., 2008 ; Settouti et al., 2011). Le modèle d'une trace en fournit une certaine sémantique, le « vocabulaire » de la trace (les types d'obsels observables et les types de relations observables) et les types des attributs de chaque type d'obsel. Une M-Trace est donc déjà une certaine interprétation explicite de l'observation. Chaque Obsel est situé temporellement selon le modèle de représentation du temps de la trace (représentation temporelle ou séquentielle, temps absolu ou relatif, unités de temps utilisées, etc.). Ce modèle explicite est utilisé aussi bien pour documenter la trace auprès des utilisateurs que pour permettre des calculs inférentiels appelés transformations de trace. La trace *première* issue de la collecte peut en effet être transformée par des opérateurs de transformation construisant une trace transformée si un motif d'obsels est reconnu dans cette trace première, selon une interprétation particulière et explicite. Le même processus de transformation s'applique à une trace transformée pour raffiner un raisonnement. Des raisonnements différents (différents points de vue) permettent de construire des interprétations différentes à partir de la même observation. D'un point de vue épistémologique, cela revient à interpréter différemment le même jeu de données. Une M-Trace contient ainsi, non seulement des informations relatives à ce qui a été observé, mais aussi sur la manière d'interpréter ces éléments observés. C'est cette capacité à exprimer des points de vue différents que nous souhaitons exploiter pour mettre en évidence des contextes différents à définir pour expliquer convenablement tel ou tel motif comportemental. Nous présentons ci-dessous l'état actuel de cette étude et les premiers résultats que nous obtenons.

4 Proposition d'un processus d'assistance à la découverte de connaissances contextuelles à partir de l'expérience tracée

L'objectif de cette méthode est d'identifier les facteurs contextuels pouvant expliquer un critère sur des données d'activité temporelle. Le contexte étant spécifique à la tâche en cours (sous partie de l'activité globale), il est important de travailler sur des segments homogènes de l'activité pour identifier quels sont les composants contexte dont les variations vont expliquer celles du critère à explorer. Nous cherchons à construire une méthode de découverte de connaissance interactive et itérative basée sur l'observation et sur des connaissances du domaine de l'analyste. Elle doit être *interactive* car c'est l'analyste qui pilote chaque étape et doit être *itérative* car l'analyste peut revenir en arrière à n'importe quelle étape. Dans cette approche, l'analyste a un rôle très important car c'est lui qui guide l'analyse en fonction de ses connaissances du domaine et de l'activité. C'est lui qui propose les transformations sur une M-Trace (source) à faire pour « voir » l'activité selon « son » interprétation (M-Trace transformée). Ces transformations sont conservées dans la base de traces modélisées et la trace transformée obtenue peut alors être explicitée facilement et confrontée à d'autres interprétations également représentées par des transformations. Les observations collectées constituent les M-Traces premières et les M-Traces transformées constituent les différentes façons d'interpréter les choses selon telle ou telle expertise. La découverte est *supervisée* et les séquences à caractériser sont *étiquetées* par l'analyste qui y « reconnaît » quelque chose qui fait sens. Cinq étapes sont proposées pour ce processus dans les paragraphes suivants.

4.1 Définition des composants génériques du contexte

Cette étape nous permet à partir du modèle des composants contexte de (Bazire et al., 2005) d'identifier « toutes » les variables possibles « candidates » au contexte lors de l'analyse d'une activité en fonction des objectifs ou des besoins d'analyse. Le modèle des composants contexte est constitué des contextes des entités qui interviennent lors de la réalisation d'une activité. Par suite, le contexte d'une activité est défini par l'ensemble des contextes de chaque entité de cette activité qui sont : le contexte de l'utilisateur « Cu », le contexte de l'item (système, objet, application, etc.) « Ci », et le contexte environnemental « Ce ». Toutes les activités possèdent ces entités car une activité se traduit par l'interaction entre un utilisateur et un objet ou une application dans l'environnement de la tâche.

Le contexte de l'observateur « Co » permet de définir le contexte de l'analyse comme les objectifs et le périmètre de l'analyse d'une activité. Les informations relatives aux entités impliquées dans une activité peuvent être utilisées pour caractériser cette activité ou un segment particulier de cette dernière. A ce niveau de l'analyse il s'agit d'informations *potentielles* de contexte de l'activité analysée. Pour identifier celles qui sont réellement explicatives, nous utiliserons donc les techniques de l'ingénierie des connaissances tracées.

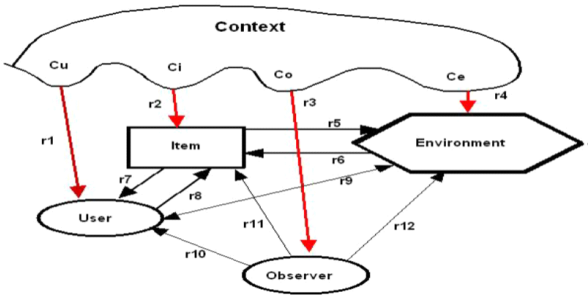


FIGURE 1 – Modèle des composants contexte (Bazire et al., 2005).

4.2 Collecte et préparation des données

Cette étape a pour but de faire la sélection et la contextualisation des données de l'activité en fonction des objectifs d'analyse. Elle mobilise les techniques de préparation de données telles que la gestion des valeurs manquantes, la création de nouvelles variables, la transformation des variables. Puis elles seront contextualisées en ajoutant des informations

relatives aux différents composants contexte définis dans l'étape 4.1. Il s'agit donc d'associer pour chaque segment d'activité sélectionné les variables potentielles explicatives du contexte. Ces variables peuvent être utilisées pour caractériser les contextes spécifiques aux différents comportements observés de l'activité analysée. Par exemple, pour le composant « Ce » de l'étape 4.1, il faut collecter toutes les informations relatives à la localisation de l'activité et les contraintes associées. La contextualisation des données consiste donc à ajouter les informations permettant de définir, de caractériser de façon précise le contexte de chacune des entités de l'activité. Ces informations peuvent être collectées automatiquement à partir de capteurs (ex. le type de route) ou par codage manuel de l'activité (ex. le trafic, la météo).

4.3 Construction de la base de connaissances (les modèles des M-Traces)

La base de connaissances est composée de règles de reformulations pour décrire une activité ou une situation de l'activité comme une succession d'événements pertinents. Cette étape peut être très rapide si l'analyste maîtrise suffisamment le domaine pour créer directement ces règles lors de l'analyse tracée de l'activité. L'élaboration même de ces reformulations est très utile à l'analyste pour préciser et formaliser ses connaissances. Par exemple, dans cet article, cette étape a été utilisée pour segmenter l'activité de conduite en fonction des situations homogènes de conduite comme AllerToutDroit, TournerADroite etc.

4.4 Processus d'assistance à l'analyse à la découverte de connaissances contextuelles

Le processus d'assistance à la découverte du contexte lors de l'analyse d'une activité est effectué à partir de l'analyse des traces. Tracer une activité consiste à la décrire avec les aspects pertinents (*obsels*) à partir d'observations et des connaissances de l'analyse. Dans l'étape 4.3, les données ont été préparées. Avec cette base de données cible, la M-Trace première collectée T_0 est créée. A partir de T_0, l'objectif est de décrire une situation comportementale d'une activité et d'en expliciter le contexte. Dans un premier temps, les premières transformations τ_n vont permettre de créer les différents segments temporels de l'activité correspondant aux différentes tâches qui constituent l'activité analysée. Puis la transformation τ_{sh} permet de sélectionner dans ces segments, ceux qui correspondent à la tâche de l'activité que l'on souhaite analyser (ex. : segments S1 dans la FIGURE 2).

Ces transformations sont faites grâce aux règles élaborées lors de l'étape 4.3 et complétées par l'expertise de l'analyste. La transformation τ_{crit} permet alors d'étiqueter les segments en fonction des valeurs du critère à expliquer (S1-- et S1++ dans la FIGURE 2). Pour identifier parmi les variables contextuelles candidates, celles qui font vraiment partie du contexte, il est tenté, à partir de l'ensemble des segments homogènes de l'activité, des transformations $\tau_{contexte1}$, $\tau_{contexte2}$ en fonction des valeurs du critère à expliquer. Ces transformations sont effectuées en utilisant uniquement les variables candidates du contexte définies dans l'étape 4.1.

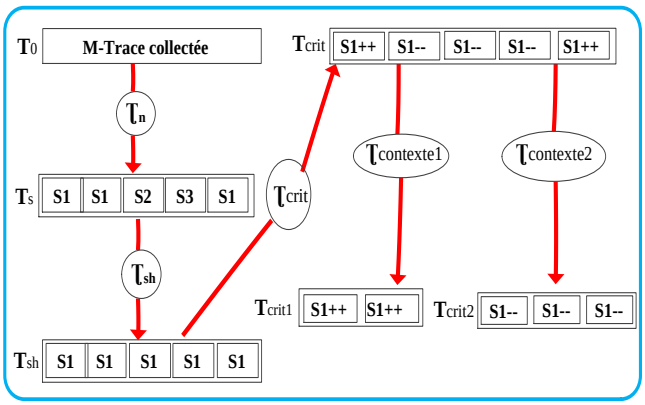


FIGURE 2 – Processus d'assistance à la découverte de connaissances contextuelles.

Les transformations $\tau_{contexte1}$, $\tau_{contexte2}$ permettent de construire les M-traces T_{crit1} et T_{crit2}. Le contexte explicatif des critères crit1 et crit2 est défini par les variables candidates du contexte de l'activité qui ont été **retenues** respectivement dans les transformations $\tau_{contexte1}$, $\tau_{contexte2}$.

4.5 Evaluation et validation des variables de contexte à retenir

L'étape d'évaluation consiste à appliquer les connaissances découvertes sur de nouveaux jeux de données afin d'en déterminer la stabilité et la validité. Cela consiste à voir si les connaissances découvertes sont suffisamment stables sur de nouveaux jeux de données pour permettre une modélisation d'un modèle d'assistance contextuel à partir de ces connaissances. Cette validation est faite à l'aide des transformations considérées comme satisfaisantes, en les appliquant sur les M-Traces (observation des transformations issues de l'étape 4.4 sur de nouveaux jeux de données) du corpus d'observation complet. Si le taux de réussite des transformations est supérieur à un certain taux, les variables retenues dans ces transformations sont intégrées dans la représentation du contexte à prendre en compte.

5 Application à la découverte du contexte dans le domaine des transports

Dans notre cas d'analyse, le processus d'assistance de découverte du contexte est appliqué au domaine du transport dans le but d'expliquer la consommation en carburant selon des comportements de conduite automobile. Pour cela, nous cherchons à identifier le contexte explicatif du critère consommation en carburant. Ce critère est défini par le gain de consommation en carburant réalisable sur un segment donné de l'activité de conduite. Plus le gain potentiel est important, plus le conducteur a surconsommé. Ce gain est calculé par la formule suivante :

$$Gain = 1 - (ConsoOpt/ConsoReelle) \quad (1)$$

Avec : *ConsoOpt* la consommation optimisée du segment issue d'un logiciel d'optimisation de la consommation du LTE (Laboratoire Transport Environnement de l'IFSTTAR (Mensing et al., 2013) et *ConsoRelle* la consommation réelle de carburant effectué sur le segment.

L'objectif de cette analyse étant d'expliquer ce gain en tenant compte du contexte en utilisant le processus d'assistance à la découverte de connaissances contextuelles, nous allons suivre les cinq étapes de ce processus.

5.1 Définition des composants contexte de l'activité de conduite

L'activité de conduite automobile consiste à se déplacer d'un point A à un point B au moyen d'un véhicule. Pour cela, les entités Conducteur et Véhicule interagissent ensemble dans un environnement donné qui est l'environnement de conduite (la route). Non seulement ces entités interagissent ensemble, mais elles ont une influence sur l'activité de conduite.

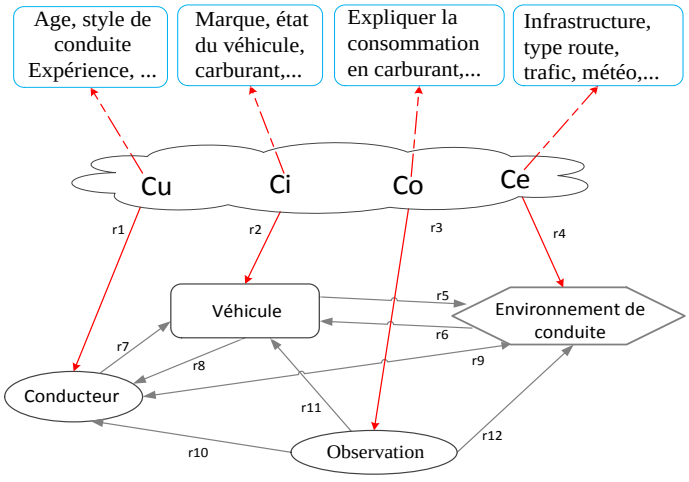


FIGURE 3 – Représentation générique du contexte de l'activité de conduite à partir de (Bazire et al., 2005).

Par exemple, en cas de trafic dense, la conduite est lente car l'environnement extérieur contraint le conducteur à tenir compte des autres véhicules. Lors de la conduite automobile, le conducteur adapte en permanence sa conduite en fonction de ce qui se passe autour de lui, donc de son environnement de conduite (trafic, météo, type de route, …). Cette adaptation aux conditions de circulation influence les comportements ou actions de conduite. Ces changements de comportements de conduite influencent à leur tour la consommation en carburant.

Ces informations pourraient potentiellement expliquer des différences de consommation. Elles seront donc des éléments candidats au contexte de l'activité de conduite dans le cadre de cette analyse d'après la définition du contexte de (Dey, 2001). A partir du modèle de composant contexte de (Bazire et al., 2005), nous définissons les informations candidates aux contextes des entités de l'activité de conduite automobile. Les informations relatives aux entités de l'activité de conduite (Conducteur, Véhicule, Environnement de conduite, Observateur) permettront de collecter le maximum d'informations pour mieux expliquer le critère qui est dans notre cas la consommation en carburant. Par exemple dans nos données, la conduite change en fonction du contexte de l'environnement « Ce » : type de la route (autoroute, urbain, etc.), trafic (fluide, dense, bouchon etc.), météo (neige, pluie, soleil), mais aussi du contexte conducteur « Cu » (expérience de conduite, âge, style de conduite etc.) et du contexte du véhicule « Ci » (marque, carburant, etc.). Les informations relatives à ce critère à expliquer sont définies par l'entité Observateur. Ici, on cherche à expliquer la consommation en carburant de la situation de conduite « AllerToutDroit ».

5.2 Collecte et préparation des données K/Consommation en carburant

Les données brutes collectées lors de l'activité de conduite sont des données numériques et des vidéos. Avec l'outil d'analyse de l'activité de conduite, il nous est donc possible de coder manuellement les paramètres ou variables candidates relatives au contexte de l'activité conduite grâce aux vidéos telle que : la météo, le trafic, l'infrastructure, les incidents etc. A ce niveau, il est préférable de renseigner le maximum d'informations sur les variables candidates au contexte de l'activité de l'étape 5.1 et sur les autres paramètres issus des capteurs.
Pour préparer les données, plusieurs opérations de préparation ont été effectuées, par exemple la création de nouveaux paramètres (calculés et contextuels). Les paramètres calculés sont définis par l'expert du domaine à partir des paramètres bruts issus de la collecte. Par exemple la moyenne de la vitesse, les fréquences d'utilisation des rapports de la boîte de vitesse sur les différents segments, le gain potentiel de consommation etc.

5.3 Construction de la base de connaissances K/Consommation en carburant

Cette étape a permis de définir des règles de reformulations « sûres » (requêtes basées sur des éléments objectifs, non contextuels), afin identifier des segments temporels homogènes en termes d'action de conduite : AllerToutDroit, TournerAGauche etc. Elle a permis également d'étiqueter ces segments de conduite en fonction de leur gain de consommation en carburant (gain calculé à partir des consommations en carburant réelle et optimale) sur l'ensemble des segments. Les segments issus de cette étape seront utilisés dans l'étape suivante pour étudier le contexte explicatif de la surconsommation dans des séquences comparables.

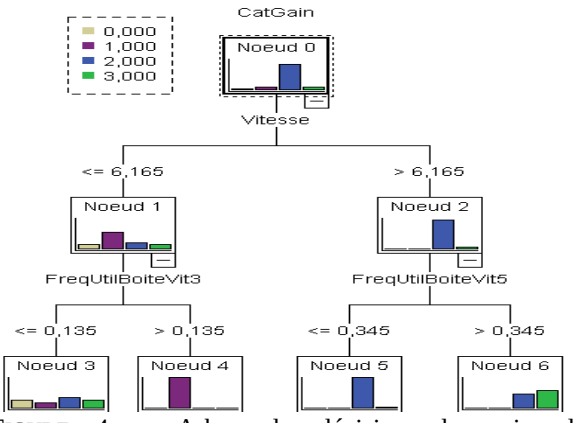


FIGURE 4 – Arbre de décision du gain de consommation.

Nous avons utilisé une méthode de classification supervisée (Arbre de décision CRT) pour identifier les paramètres non contextuels prédisant le gain potentiel de carburant sur les segments «AllerToutDroit» de l'activité de conduite. Cette classification a permis de bien classer 90% de l'ensemble des segments utilisés pour l'apprentissage. Ce type de méthode permet d'identifier les paramètres pertinents pour séparer des segments. La méthode d'arbre de décision été choisie car elle fournit des règles de décisions mais toute autre méthode efficace pourrait être utilisée.

5.4 Processus d'assistance à l'analyse K/Consommation en carburant

Notre objectif est de montrer qu'en découvrant les variables utiles parmi les variables candidates au contexte (définies en 5.1) sur les nœuds contenant des éléments hétérogènes (10% de segments mal classés le sont dans les *nœuds 3 et 6* de l'arbre de décision FIGURE 4), on arrive à améliorer le taux de classification. Nous étudions les M-Traces T_3 et T_6 issues des règles de reformulations des nœuds 3 et 6 de l'étape 5.3 avec les transformations suivantes (la ***vitesse*** est en ***m/s***):

τ_3= « *SELECT * FROM* T_{sh}
WHERE ***Vitesse<=6*** *AND* ***FreqUtilBoiteVit3 <=0.14*** *PRODUCT* ***CatGain =2***».

τ_6 = « *SELECT* ******* *FROM* T_{sh}
WHERE ***Vitesse>6 AND FreqUtilBoiteVit5 >0.35*** *PRODUCT* ***CatGain =3***».

Pour identifier parmi les variables contextuelles candidates, celles qui font vraiment partie du contexte, il est nécessaire à partir de l'ensemble de ces segments de créer des transformations cherchant à mieux discriminer les segments selon leur étiquette de gain.
La comparaison des requêtes tentées a permis d'identifier la variable contextuelle « ***TypeInfra*** » qui définit le type d'infrastructure de la route (rond-point, intersection, ligne droite, tunnel etc.) permettait de mieux classer les obsels des M-Traces T_3 et T_6 avec les règles de transformations suivantes :

$\tau_{Contexte1}$= « *UPDATE* T_3 *SET* ***CatGain =0***
WHERE ***TypeInfra*** = ***'LigneDroite'*** *OR* ***TypeInfra*** = ***'RondPoint'***».

$\tau_{Contexte2}$= « *UPDATE* T_6 *SET* ***CatGain =2***
WHERE ***TypeInfra*** =***' Intersection'*** *OR* ***TypeInfra*** = ***' Tunnel'***».

Sur les obsels des deux M-Traces T_3 et T_6, ces transformations ont permis d'en classer correctement 67% au lieu des 43% classés avec uniquement les paramètres non contextuels. En répétant ce processus sur l'ensemble des variables candidates au contexte, nous avons identifié que les variables contextuelles qui expliquent le mieux le gain de consommation sont le type d'infrastructure et le trafic.
En fin d'analyse, nous avons les résultats suivants. Avec uniquement les variables non contextuelles nous avions un taux de 90% de bonne prédiction de la classe de gain. Avec l'ajout de deux variables contextuelles ce taux est augmenté de 7% soit un taux global de bonne prédiction de 97% sur les données initiales ayant servies au processus de découverte des connaissances.

5.5 Evaluation et validation des variables contextuelles découvertes

Cette étape est importante pour savoir si les connaissances découvertes sur les données utilisées dans l'étape d'analyse des traces sont avérées sur de nouvelles données. Elle doit être effectuée avant chaque mise à jour de la base de connaissances. Pour évaluer la cohérence des connaissances produites, nous les avons utilisées sur de nouvelles données (données Test). Le taux de bonne prédiction global sur les données de test est de 84% est certes inférieur à celui de l'échantillon initial (97%), cela peut expliquer par le volume pas très important de données codées manuellement.

TABLE 1 - *Test de validation des connaissances.*

Classification

Echantillon	Observations	Prévisions				
		0	1	2	3	% correct
Initial	0	3	0	0	0	100%
	1	1	9	1	0	82%
	2	0	0	113	0	100%
	3	1	0	1	7	78%
	% global	4%	7%	85%	5%	97%
Test	0	1	0	0	0	100%
	1	0	2	2	0	50%
	2	0	2	25	0	93%
	3	0	0	2	3	60%
	% global	3%	11%	78%	8%	84%

Méthode de développement : CRT
Variable dépendante : CatGainFuel10

Pour vérifier la stabilité des connaissances découvertes, d'autres tests de validation sont prévus sur un plus grand volume de données.

5.6 Conclusion

Le travail de recherche présenté dans cet article articule deux contributions principales dans le domaine de l'ingénierie des connaissances : **la clarification d'une démarche de découverte d'éléments contextuels par l'observation d'une activité tracée**, débouchant sur des principes, une méthode et un environnement de découverte interactive de connaissances contextuelles ; **une démarche d'analyse originale dans le domaine de la conduite automobile** avec de premiers résultats très encourageants mettant en évidence des éléments de contexte qu'il serait difficile d'établir sans l'assistance à l'analyse mise en place. Les premiers résultats obtenus permettent d'une part, de déterminer les facteurs qui favorisent la surconsommation et constituent donc le contexte explicatif de la consommation en carburant et d'autre part, de déterminer la classe à laquelle appartient le gain de consommation potentiel sur un segment de l'activité de conduite selon les valeurs prises par les variables contexte. En appliquant le même processus sur les différents segments de l'activité de conduite, on peut déterminer les variables dont dépendent la consommation en carburant et la classe de gain à laquelle ils appartiennent. Une base de connaissances contextuelles sur la consommation en carburant est ainsi progressivement construite. Une utilisation prometteuse de cette base est la mise en place d'un système d'assistance contextuel à la conduite automobile. L'un de nos objectifs futurs est d'évaluer l'efficacité de l'approche proposée dans cet article par rapport à d'autres possibles approches de découvertes de connaissances contextuelles.

Remerciements : Nous tenons à remercier le CEESAR pour la mise à disposition des données naturelles de conduite.

Références

AKMAN, V., & SURAV, M. (1996). Steps toward formalizing context. *AI magazine*, *17*(3), 55.

BAZIRE, M., & BRÉZILLON, P. (2005). Understanding Context Before Using It. Dans *Modeling and using context* (pp. 29‑40). Berlin Heidelberg : Springer.

BEUSEN, B., BROEKX, S., DENYS, T., BECKX, C., DEGRAEUWE, B., GIJSBERS, M., PANIS, L. I. (2009). Using on-board logging devices to study the longer-term impact of an eco-driving course. *Transportation Research Part D: Transport and Environment*, *14*(7), 514‑520.

BOUQUET, P., GHIDINI, C., GIUNCHIGLIA, F., & BLANZIERO, E. (2001). *Theories and uses of context in knowledge representation and reasoning* (Technical Report No. DIT-02-010).

BRÉZILLON, P. (2000). Modeling and Using Context: Past, Present and Future. *Decision Support throught Knowledge Management*, 301–320.

CASTELLI, G., MAMEI, M., & ZAMBONELLI, F. (2008). Engineering executable agents using multi-context systems. Dans *Engineering Environment-Mediated Multi-Agent Systems*, *5049* (pp. 223‑239). Journal of Logic and Computation.

DEY, A. K. (2001). Understanding and Using Context. *Personal and ubiquitous computing*, *vol. 5*, p. 4‑7.

DEY, A. K., SALBER, D., FUTAKAWA, M., & ABOWD, G. (1999). *An architecture to support context-aware applications*. GVU Technical Report GIT-GVU-99-23.

FAYYAD, U., PIATETSKY-SHAPIRO, G., & SMYTH, P. (1996a). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37.

FAYYAD, U., PIATETSKY-SHAPIRO, G., & SMYTH, P. (1996b). The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM*, *39*(11).

GANDON, F., & SADEH, N. (2004). Gestion de connaissances personnelles et contextuelles, et respect de la vie privée. Dans *15èmes Journées francophones d'Ingénierie des Connaissances* (pp. 5–16). Presses universitaires de Grenoble.

GEORGEON, O. L., MILLE, A., BELLET, T., MATHERN, B., & RITTER, F. E. (2012). Supporting activity modelling from activity traces. *Expert Systems*, *29*(3), 261–275.

GIUNCHIGLIA, F., & BOUQUET, P. (1996). Introduction to contextual reasoning: an artificial intelligence perspective. Dans *Perspectives on Cognitive Science, 3* (pp. 138‑159). Sofia : B Konikof.

GUHA, R. V. (1991). *Contexts: A Formalization and Some Applications*. Stanford University Computer Science Department.

LAFLAQUIERE, J., PRIE, Y., & MILLE, A. (2008). Ingénierie des traces numériques d'interaction comme inscriptions de connaissances. Dans *19es Journées Francophones d'Ingénierie des Connaissances (IC 2008)* (pp. 183–195).

LENAT, D. (1998). The dimensions of context-space. *available online at URL http://www. casbah. org/resources/cycContextSpace. shtml*.

MATHERN, B. (2012). *Découverte interactive de connaissances à partir de traces d'activité : Synthèse d'automates pour l'analyse et la modélisation de l'activité* (thèse de doctorat en informatique). Université Claude Bernard Lyon 1.

MCCARTHY, J. (1993). Notes on formalizing context.

MEDDE. (2014). Observation et statistiques. Repéré à http://www.statistiques.developpement-durable.gouv.fr/transports/r/parcs.html

MENSING, F., BIDEAUX, E., TRIGUI, R., & TATTEGRAIN, H. (2013). Trajectory optimization for eco-driving taking into account traffic constraints. *Transportation Research Part D: Transport and Environment*, *18*, 55‑61. doi :10.1016/j.trd.2012.10.003

MILEA, V., FRASINCAR, F., & KAYMAK, U. (2008). Knowledge Engineering in a Temporal Semantic Web Context (pp. 65‑74). Eighth International Conference on : IEEE.

MULLINS, R., CARSTEN PILS, T., ROUSSAKI, D. I., & NTUA, D. (2008). Context and Knowledge Management. *Mobile Service Platforms Cluster, White paper, June*, 1–47.

NGOC, K. A. P., LEE, Y.-K., & LEE, S.-Y. (2005). Context knowledge discovery in ubiquitous computing. Dans *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops* (pp. 33–34). Springer.

RAC-F. (2007, février). Changement Climatique et Transports par Réseau Action Climat-France (RAC-F). Repéré à http://www.rac-f.org/IMG/pdf/Changement_Climatique_et_Transports.pdf

RAKOTONIRAINY, A., HAWORTH, N., SAINT-PIERRE, G., & DELHOMME, P. (2011). Research issues in Eco-driving. *Queensland University of Technology and French Institute in science and technology of transport*.

REY, G., COUTAZ, J., & CROWLEY, J. L. (2002). The contextor: a computational model for contextural information. *Wokshop Building Bridges Interdisciplinary Context-Sensitive Computing*.

SETTOUTI, L. S. (2011). *Systèmes à Base de Traces Modélisées : Modèles et Langages pour l'exploitation des traces d'Interactions.* (Thèse de doctorat Informatique). Université Claude Bernard Lyon 1.

STRANG, T., LINNHOFF-POPIEN, C., & FRANK, K. (2003). CoOL: A context ontology language to enable contextual interoperability. Dans *Distributed applications and interoperable systems* (pp. 236–247). Springer.

Gestion de connaissances pour l'acquisition, le traitement et la valorisation des connaissances du patrimoine technique

Benjamin Hervy[1], Matthieu Quantin[1,2], Florent Laroche[1], Alain Bernard[1], Jean-Louis Kerouanton[2]

[1] IRCCyN UMR CNRS 6597, École Centrale de Nantes, France
prenom.nom@irccyn.ec-nantes.fr

[2] Centre François Viète EA 1161, Université de Nantes, France
Jean-Louis.Kerouanton@univ-nantes.fr

Résumé : L'histoire des techniques traite des données hétérogènes et produit des connaissances à l'issue d'un processus de rédaction. Ces connaissances sont nécessairement liées à une épistémologie (approche de l'historien) ainsi qu'à un objet. Cette double spécificité empêche la création a priori de classes de données uniformisées pour stocker ces connaissances. Parallèlement, leur forme rédigée (libre) les rend difficilement exploitables pour la mise en œuvre de moyens d'accès, de capitalisation, et d'analyse.

L'approche proposée consiste à fournir des moyens d'acquisition et de gestion des connaissances pour l'historien. L'objectif principal est de proposer une lecture multi points de vue en complément du récit linéaire. Cette méthode s'appuie sur un modèle conceptuel de représentation pour la formalisation des connaissances historiques associées à un objet d'étude. Notre contribution permet la gestion de la diversité des approches et des objets pour le traitement des données. Nous illustrons également notre méthodologie au travers de cas d'étude pour la valider et identifier des perspectives d'amélioration.

Mots-clés : Gestion des connaissances, Aide à la décision, Collaboration, Épistémologie, Histoire des techniques

1 Introduction

Présentation du contexte de recherche

Les outils des sciences pour l'ingénieur peuvent être réappropriés à des fins de reconception, de conservation et de valorisation d'objets anciens. L'évolution des technologies numériques permet d'ailleurs d'élargir les modalités d'interaction avec le patrimoine historique. Par exemple, les méthodes de numérisation 3D et de reconception constituent un des champs d'investigation afin de pouvoir capitaliser l'objet dans son contexte. En effet, considérer un objet patrimonial seul n'a pas de sens. Il est indispensable de restituer son contexte d'origine, dans ses différentes phases de vies afin d'en cerner au mieux les dimensions et les transformations. Ainsi, compte-tenu de la masse d'informations hétérogènes à traiter, il est intéressant de s'appuyer sur les méthodes et outils de gestion des connaissances pour structurer cette capitalisation.

En effet, (Ermine *et al.*, 2004) montre au travers d'un cas d'étude l'utilité et la complémentarité d'une modélisation des connaissances avec un travail d'historien, afin de concevoir des interfaces graphiques de navigation dans un portail métier dédié. Le travail d'historien peut alors être réutilisé pour aboutir à « un gain certain en termes de temps de recueil et de structuration, en termes de qualité et richesse du contenu, et en termes de diversité des livrables produits ».

Notre objectif est donc de proposer une approche de gestion des connaissances couvrant le processus de patrimonialisation dans son intégralité :

1. depuis la numérisation de l'objet et la capitalisation des connaissances associées ;
2. en passant par la modélisation numérique de l'objet physique et de sa dynamique ;
3. jusqu'à sa conservation et sa valorisation dans un cadre muséal par exemple.

Enjeux

L'enjeu de ce processus réside dans la complémentarité des 3 étapes. Une des difficultés réside notamment dans la complexité de vouloir généraliser la méthode. En effet chaque objet patrimonial est unique et fait appel à des connaissances, des savoir-faire ancestraux et des expertises contemporaines. De plus, lorsqu'un objet est capitalisé/numérisé, la dernière étape de valorisation doit pouvoir évoluer. Aussi, il convient de capitaliser un maximum d'informations, de les organiser et les lier afin de pouvoir envisager un large spectre de possibilités de valorisation. Nos objets contenus dans les lieux de mémoire ne sont pas uniquement destinés à « prendre la poussière sur une étagère ». Ils doivent pouvoir devenir support de savoirs et savoir-faire, potentiellement réappropriables au présent. L'ensemble des connaissances associées à ces objets doit pouvoir être diffusé à une vaste diversité d'utilisateurs. Nous faisons l'hypothèse que la valorisation d'un objet patrimonial peut servir de support à l'élaboration d'une méthodologie de gestion de connaissances historiques. Pour cela, nous nous appuyons sur une approche interdisciplinaire croisant les méthodes de gestion des connaissances en génie industriel et les méthodes d'étude du patrimoine technique.

Complexité des données historiques

La diversité des objets étudiés nous apprend que les connaissances historiques sont nécessairement fonction d'une épistémologie (approche de l'historien) ainsi que d'un objet. Par exemple, l'étude des Halles Alstom [1] admet deux approches (analyse via les textes, analyse via l'objet [2]) appliquées à certaines dimensions de l'objet : mécanique, architecturale, humaine, et en occulte d'autres : culturelle, artistique, industrielle…

Chacune de ces dimensions peut donner lieu à un récit historique différent. Par ailleurs, ces dimensions sont dépendantes de l'objet étudié. Le croisement des sources de différents auteurs est au cœur du mécanisme de production des connaissances historiques. En ce sens, la méthodologie proposée doit permettre la collaboration de différents acteurs pour le chaînage des connaissances, leur critique et l'évolution des modèles (Srinivasan & Huang, 2004). Le problème se pose alors du croisement des récits pour la valeur historique de l'information et l'exploitation des connaissances à des fins de création et de valorisation du patrimoine.

À partir de cas d'étude du patrimoine technique, nous proposons une approche originale pour la résolution des difficultés rencontrées dans la gestion des connaissances historiques.

Cette approche combine l'intégration de données historiques dans des systèmes d'information spécifiques et des formes d'interaction au moyen d'interfaces d'accès. Elle vise à apporter des outils d'aide à la décision en complément des méthodes classiques d'histoire des techniques (exemple : monographies). L'objectif principal est de proposer une lecture multi points de vue qui réponde à un besoin méthodologique exprimé par les historiens (Cotte, 2010). Ces outils

1. Les Halles Alstom sont des halles industrielles de la fin du XIXe siècle et implantées sur le port de Nantes.
2. Voir figure 4

s'appuient sur les technologies numériques devant être pilotées par les connaissances disponibles et les intentions des utilisateurs (ici les historiens et/ou conservateurs de musée).

2 Proposition d'une méthode de gestion de connaissances historiques

Depuis une dizaine d'années, avec l'essor des technologies de numérisation, conservation et valorisation, le patrimoine culturel fait l'objet de nombreux travaux de recherche visant à définir de nouvelles modalités d'interaction, de compréhension et de diffusion.

Les technologies numériques permettent autant de répondre à certains besoins actuels des sciences humaines que de créer de nouvelles modalités d'interaction et d'exploration.

Pourtant, il n'existe toujours pas de méthodologie outillée formalisée pour la gestion du processus global de patrimonialisation allant de la conservation à la valorisation et la diffusion des connaissances. Il existe certes de nombreux travaux proposant des méthodes détaillées pour la numérisation (3D ou 2D) d'artefacts, notamment en archéologie, ou pour la structuration des données à destination du web de données, mais à notre connaissance, il n'existe pas de méthodologie couvrant l'intégralité du processus de patrimonialisation.

Le patrimoine étant une construction culturelle collective, il est le résultat d'un choix fait par « nous, qui depuis le présent, avons reconnu à cet objet une valeur et considérons que ceux qui l'ont créé feraient, pour nous, de "bons" ancêtres culturels » (Davallon, 2002). Par conséquent, il nous semble important d'intégrer une dimension contributive (Stiegler, 2009) dans la construction d'une méthode de gestion des connaissances historiques. Dans cette partie, nous décrivons la méthodologie proposée selon 3 axes :

1. l'acquisition des données pour l'analyse, la compréhension et la modélisation d'un objet ;
2. le traitement et l'exploitation à des fins de production et de chaînage des connaissances par des contributeurs de divers profils ;
3. la finalité vis-à-vis du patrimoine : visualisation et exploitation des données à des fins de transmission et de médiation.

2.1 Acquisition

Classiquement, l'acquisition se déroule de la façon suivante :

1. **Construction** d'un espace de sujets et d'hypothèses de recherche ;
2. **Identification** des sources nécessaires à l'étude de ces hypothèses. Il s'agit d'établir un corpus initial de sources historiques primaires et secondaires [3] pertinent au vu du sujet d'étude. Les experts concernés pour la réalisation de cette étape sont entre autres les archivistes, les historiens, et (pour les objets ayant déjà atteint la phase de patrimonialisation de leur cycle de vie) les conservateurs ;
3. **Numérisation** des sources en vue de leur traitement. C'est une étape optionnelle pouvant être utile pour l'analyse des sources (exemple de la rétro-conception 3D d'un objet technique en ruine) et leur conservation.

3. Par source primaire, nous entendons toute source historique contemporaine à l'objet. Une source secondaire est un document provenant d'une synthèse préalable de sources primaires.

4. **Analyse** du corpus. Cette étape implique tous les acteurs utiles au décryptage de ces sources (dessins techniques, plans, fichiers CAO, langues étrangères). Cette interdisciplinarité est nécessaire pour comprendre et restituer à la fois la vue interne des objets (structure, analyse systémique) mais également leur vue externe (comment l'objet s'insère et évolue dans son environnement).

Dans ce contexte, le verrou scientifique majeur réside dans la constitution d'un espace de travail commun à ces différents métiers. Il s'agit d'interfacer des points de vue complémentaires dont la nature et le nombre ne sont pas connus *a priori*. Cela implique *a minima* une modélisation flexible et évolutive des connaissances ainsi que la prise en compte des profils de contributeurs. Sur ce point, nous pensons que l'approche du web socio-sémantique est une source d'inspiration intéressante. Le web socio-sémantique propose une approche collaborative pour la modélisation et la recherche d'information (Zacklad *et al.*, 2007). Ainsi, un même item, caractérisé par des attributs peut-être défini selon différents points de vue, et différents thèmes.

Il n'y a donc pas ici de présupposition de l'organisation globale des données. L'objectif est de comprendre l'objet d'étude par une mise en réseau de toutes les connaissances potentiellement disponibles. Le système n'est pas prégnant sur l'organisation des informations mais laisse la possibilité d'enrichir et de faire évoluer les modèles.

Cette flexibilité permet de gérer la complexité multidimensionnelle de l'histoire de l'objet. Intégrer de telles données et les connaissances associées implique de proposer un modèle générique pouvant être spécialisé par l'usage.

L'amorce consiste à choisir un modèle conceptuel adapté à l'étude (patrimoine industriel, artistique, religieux, etc.) en fonction des hypothèses de recherche définies par les experts au début de l'acquisition. Les descripteurs ou attributs devant servir à la formalisation des connaissances sont définis, mais le modèle doit pouvoir évoluer avec l'enrichissement du système et la multiplicité des acteurs.

Pour structurer les connaissances manipulées dans le contexte de l'étude, nous choisissons une représentation atomique sous la forme d'une fiche à l'instar de (Ardans, 2011). En effet, plus le choix de modélisation est compatible avec du récit linéaire, plus la méthodologie sera adaptée aux pratiques de l'histoire. Ainsi, chaque élément (bâti, événement, personne) se voit attribuer une fiche, vecteur de la connaissance qui s'y rattache, dépendant d'un point de vue, pouvant être mise en relation et annotée.

2.2 Traitement

Une fois la fiche produite par un contributeur, celle-ci doit être formalisée dans le système et se voit affecter des métadonnées (auteur et statut entre autres) ainsi que des liens avec d'autres fiches. L'intégration de la fiche dans la base de données peut apporter des éléments d'enrichissement à la fois du contenu mais également du modèle de données lui-même. La structure de données peut ainsi évoluer au fur et à mesure de la diversification des contributions.

L'affectation des métadonnées, sous la forme de descripteurs, permet à la fois de représenter le contenu et sa provenance. Par exemple, les mots-clés représentatifs du contenu d'une fiche permettent de créer des associations (à la manière d'un lien hypertexte) et d'éventuellement saisir une part de la subjectivité du contributeur (celui-ci ayant un jargon et un style d'écriture) ou d'une communauté de pratiques (voir la notion de "cogniton" (Serrafero, 2000)).

La méthodologie proposée permet donc de créer un réseau d'informations sous la forme de fiches liées entre elles. Les connaissances formalisées dans ces fiches sont associées à un corpus de documents historiques et à différents profils de contributeurs.

Pour supporter cette méthodologie, nous proposons un méta-modèle conceptuel représenté sous la forme d'un diagramme de classes UML par la figure 1.

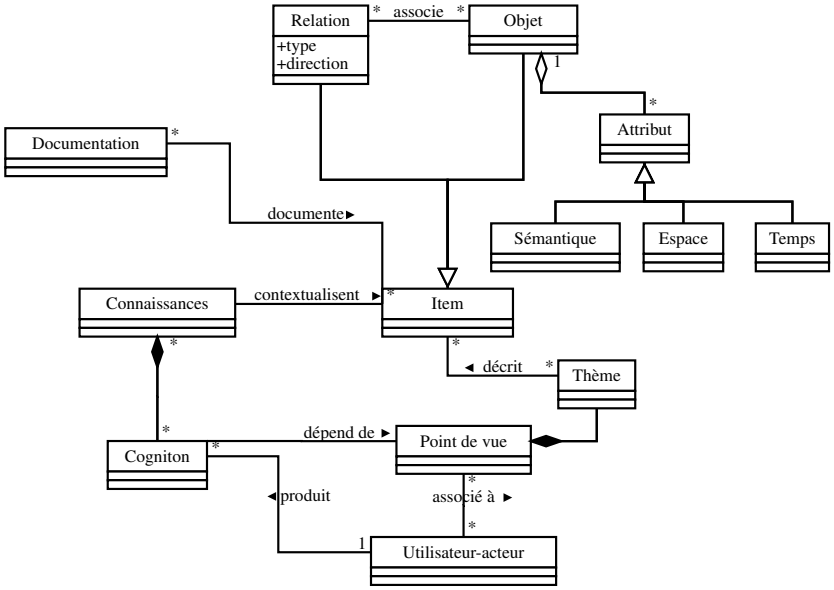


FIGURE 1 – Représentation UML du système d'organisation des connaissances historiques

Ce modèle s'appuie sur un certain nombre de classes élémentaires :

— La classe **item** (en suivant la typologie du modèle proposé par (Zacklad *et al.*, 2007)) est ici une classe abstraite, permettant d'englober la diversité des données manipulées dans le contexte du patrimoine culturel. Ainsi, un item peut être :

1. un **objet** au sens large, c'est-à-dire aussi bien un objet physique [4] qu'un sujet ou une thématique. Dans ce deuxième cas, l'objectif est de pouvoir considérer de manière égale les différentes projections permettant de saisir le contexte historique de l'objet [5]. Un objet peut-être décrit par différents attributs (position dans l'espace et le temps, caractérisation sémantique). ;
2. une **relation**. La particularité ici est l'attribution d'une importance égale à la relation et à l'objet. En effet, la compréhension du contexte d'un objet implique de pouvoir aussi bien expliciter l'objet en lui-même que ses relations avec son environnement [6].

4. Exemple d'un pont roulant : un instrument de manutention consistant à déplacer une charge sur une superficie.

5. Dans le cas des halles Alstom, l'histoire du pont roulant peut être créée par la projection selon différentes dimensions ou temporalités : l'analyse par empreintes sur l'architecture, l'analyse mécanique interne de l'objet, l'analyse de son intégration dans un processus industriel, etc.

6. Par exemple, dans le cas des halles Alstom, relier l'objet « Pont roulant » à l'objet « Établissements Maurice Ménard » (bureau d'étude) est une donnée historique à part entière.

— La classe **connaissances** fait directement référence aux connaissances formalisées sous formes de fiches et liées à un **item**.
— La classe **cogniton** est une adaptation du concept proposé par (Serrafero, 2000) au sens de "connaissance métier". Il s'agit ici de différencier les **connaissances** accumulées par les experts relatives à un item et les connaissances élémentaires produites par un expert d'après son point de vue et ses connaissances personnelles. La traçabilité de l'information, nécessaire au travail historique, est conservée mais seule la partie métier de celle-ci est modélisée (par la classe cogniton).
— La classe **utilisateur-acteur** caractérise à la fois les utilisateurs du système mais aussi les utilisateurs générant potentiellement de nouvelles connaissances. Ces utilisateurs-acteurs sont associés à un **point de vue** qui aura un impact tant sur la phase de visualisation et d'interaction que sur la traçabilité des informations du système.

2.3 Exploitation/Visualisation

La question se pose désormais de la finalité vis-à-vis des pratiques en histoire et dans le domaine de la conservation et la valorisation du patrimoine. En quoi ces méthodes et les outils numériques apportent un complément utile aux approches classiques (récit linéaire) ?

Michel Cotte explique en effet que l'usage du numérique « complète et fait évoluer les méthodes de l'histoire, de l'archéologie, du patrimoine ou de la muséographie, en offrant des possibilités nouvelles de compilation et de mise en scène des connaissances » (Cotte, 2009).

De nombreux programmes de recherche s'attachent à la conservation et la valorisation du patrimoine culturel via le numérique (Fleury, 1997; Guidi *et al.*, 2008; Prévôt, 2013). De la simple conservation numérique à la « visite immersive » en passant par la simulation numérique ou la vérification d'hypothèses, l'utilisation du numérique permet d'explorer de vastes champs de recherche par les interfaces de visualisation disponibles.

Tout d'abord, le processus d'acquisition et de traitement décrit précédemment permet dans certains cas une conservation des traces historiques par le numérique [7].

Un apport majeur de la méthodologie proposée réside dans la manipulation des connaissances produites par l'historien. En effet, l'intégration du corpus de documents historiques et la formalisation de ces connaissances permettent la mise en place de modes d'interaction et de visualisation. Le système d'information va permettre de confronter différents points de vue et/ou approches sur un même objet. Le récit linéaire classique ne permet pas de les obtenir aussi facilement que les outils informatiques.

La question de la représentation d'informations pour les données historiques fait déjà l'objet de nombreux travaux. Par exemple, le projet SyMoGIH (Beretta & Vernus, 2012) s'oriente plutôt vers la cartographie géographique, tandis que la visualisation sous forme de graphe est utilisée pour la représentation de réseaux historiques [8]. Ce mode est particulièrement intéressant pour l'analyse visuelle d'informations sémantiques (Greffard *et al.*, 2013).

7. Si l'on reprend le cas des ponts roulants dans les halles Alstom, ceux-ci sont voués à disparaître. L'acquisition décrite en section 2.1 des différentes sources relatives à ces ponts roulants, incluant l'artefact virtuel de l'objet (par numérisation), permet une certaine forme de conservation de ce patrimoine.

8. Voir par exemple : `http://www.fas.harvard.edu/~histecon/visualizing/`

À ce mode de visualisation, on peut ajouter l'intérêt des représentations 3D par exemple pour la reconstitution virtuelle d'un objet, la vérification d'hypothèses ou la simulation numérique. Cependant, ces représentations sont le plus souvent vouées à la recherche d'un consensus fidèle à la réalité physique. Certains travaux tentent néanmoins de conjuguer ces représentations 3D avec différents points de vue, laissant la place à l'incomplétude des informations (De Luca *et al.*, 2011). Cette incomplétude est une particularité inhérente à la pratique de l'histoire.

La dernière phase de la méthodologie proposée consiste donc à piloter les interfaces de visualisation avec les connaissances capitalisées lors de la phase d'acquisition. Cela permet de combiner des représentations abstraites (graphes d'information multi-niveaux) et des représentations physiques des objets étudiés. Le contributeur peut alors manipuler les différentes informations numérisées et identifier des pistes d'enrichissement du système. L'objectif ici est de permettre l'analyse des données selon différentes approches, en fonction de besoins et d'hypothèses de recherche. Par le biais de l'interaction, l'utilisateur peut adopter :

1. une posture passive consistant à « consommer » un certain volume d'informations. Le système va alors permettre la transmission de connaissances par la restitution du contexte de l'objet à l'utilisateur.
2. une posture active menant à un rôle de contributeur. En fonction de son profil et de ses intentions de recherche d'information, le système lui propose différents accès aux informations qu'il contient. Le contributeur peut alors explorer de nouvelles hypothèses, identifier des manques dans le système et les intégrer. Le système devient un outil d'aide à la décision pour assister l'ajout d'informations de manière cohérente avec l'existant tout en restant piloté par l'utilisateur.

Parmi les exemples de fonctionnalités possibles pour l'exploitation des connaissances historiques et l'aide à la décision, on peut citer :

— le traitement automatique du langage (extraction et normalisation de mots-clés) ;
— des opérations d'analyse automatique des propriétés du graphe d'information (entités isolées, parcours, suggestion de champs d'investigation faisant défaut, de liens manquants ou redondants, sérendipité (Deuschel *et al.*, 2014)) ;
— la visualisation et la mise en contexte dans un environnement physique (par le biais d'interfaces en réalité virtuelle ou augmentée par exemple).

Cette phase d'exploitation permet donc de compléter la méthodologie afin d'atteindre plusieurs objectifs :

— la capitalisation pour faciliter d'autres formes d'exploitation des connaissances ;
— la valorisation de l'objet et des connaissances associées ;
— la vulgarisation et la transmission auprès des différents publics.

3 Illustration de la méthode autour de deux cas d'étude

Deux cas d'études ont permis d'avoir une approche empirique et inductive pour la constitution de la méthodologie présentée dans la partie précédente :

— le projet « Nantes1900 »[9] visant à concevoir un système d'information muséologique

9. `http://nantes1900.chateau-nantes.fr`

capitalisant des données historiques à des fins de recherche et de médiation culturelle. La démarche de production des connaissances a été adaptée pour ce projet.

— une monographie portant sur l'histoire d'un lieu industriel et contenant des informations complémentaires au projet « Nantes1900 ». En revanche, la démarche de production des connaissances n'a ici pas pris en compte le projet « Nantes1900 ».

Nous verrons dans cette partie les caractéristiques de ces deux cas d'étude et comment ils peuvent converger.

3.1 Résultats liés à la maquette du port de Nantes en 1900

Dans le cadre d'un partenariat entre l'IRCCyN, le Centre François Viète et le musée d'histoire de Nantes, un projet de recherche a été initié en 2008 pour l'étude et la valorisation d'un objet de collection du musée : la maquette du port de Nantes en 1900, un plan-relief réalisé en 1899 de $9,20m$ par $1,80m$ représentant près de $9km^2$ du port de Nantes à l'échelle $1/500$.

Un ensemble de documents historiques a ainsi été étudié, aboutissant à la création d'une base de données de près de 500 points d'intérêts (entreprises, personnes clés, thématiques). Cette base a été conçue conformément au modèle présenté figure 1.

Afin d'illustrer la structure de données choisie pour modéliser les informations historiques, nous proposons un exemple d'instanciation comme le montre la figure 2. Les mots-clés identifiés par les auteurs d'une fiche permettent de générer deux types d'information :

1. des liens entre la fiche rédigée et les fiches correspondantes aux mots-clés et déjà existantes au sein du corpus (nœuds orange) ;
2. des fiches "vides" (nœuds bleus) relatives aux mots-clés et ne correspondant pas à des fiches existantes. Cela permet alors non seulement de générer des "catégories" ("pont", "construction navale") mais également d'indiquer aux contributeurs les manques identifiés par le système (un sujet important mais n'ayant pas été traité).

Seuls deux types de relations ont été pris en compte (« direct » ou « indirect » lorsqu'il s'agit d'un lien vers une thématique) afin de permettre la multiplicité des points de vue. Ces relations peuvent être spécialisées *a posteriori* par l'ajout d'annotations, commentaires ou attributs.

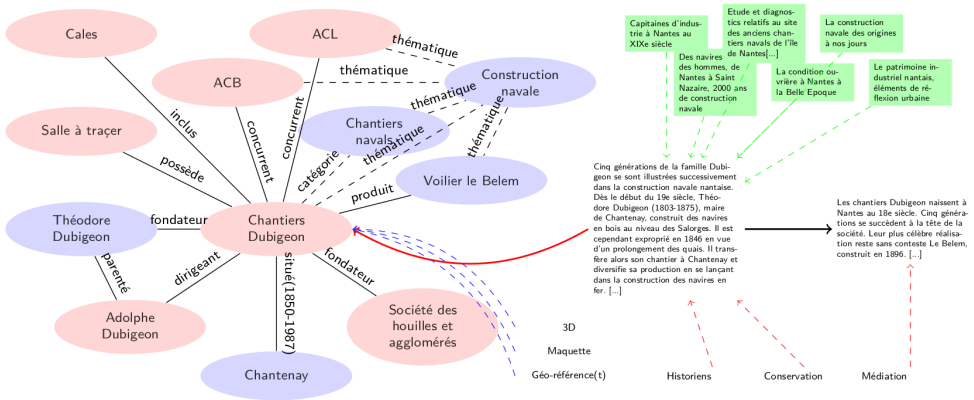


FIGURE 2 – Exemple d'instanciation du modèle avec le cas des chantiers Dubigeon à Nantes.

Les « fiches » descriptives associées à ces points d'intérêts sont liées entre elles par le biais de mots-clés mis en évidence par les chercheurs et conservateurs du musée.

123 références bibliographiques (cases vertes sur la figure 2) ont également été identifiées pour servir à la rédaction des fiches. En parallèle, 1071 sources iconographiques ont été identifiées. Parmi elles, des dessins, estampes, photographies, cartes postales, affiches, plans, etc.

Le choix de l'outil a reposé sur les compétences disponibles au moment de la mise en œuvre du projet, et sur des critères d'interopérabilité (avec les technologies web notamment) et de facilité de gestion. En effet, Postgres est un système très répandu pour la gestion de bases de données relationnelles, avec un module SIG également répandu et robuste. Par ailleurs, sa mise en œuvre est relativement simple, de même que le langage de requêtage SQL. Cela permet de développer rapidement des modules d'extraction des données pour les différentes interfaces.

Chaque entité (fiche connaissance « objet », sources, relation, données géographiques) est représentée dans la base de données. Chacune intègre les méta-données nécessaires à leur description. La plupart des méta-données, notamment pour les sources, sont basées sur celles du système de gestion des collections du musée. La figure 3a illustre la mise en relation des 500 fiches décrivant les points d'intérêt de la maquette.

(a) Graphe sémantique

(b) Prototype

FIGURE 3 – Résultats de l'acquisition (a) et de l'exploitation (b) des données historiques récoltées dans le cadre de ce projet autour de la maquette du port de Nantes en 1900.

Aujourd'hui, cette base de données est exploitée au sein d'un système d'information dédié pour le musée. Ce système permet aux professionnels du musée de s'appuyer sur cet outil de gestion des connaissances pour la valorisation des objets de collection [10]. Les visiteurs du musée quant à eux peuvent accéder à l'ensemble des informations récoltées par les historiens par le biais d'une interface en réalité augmentée et les visualiser par rapport à l'objet de musée qu'ils contemplent. Le projet a abouti à un prototype fonctionnel (cf. figure 3b) et le dispositif muséographique final verra le jour courant juin 2015. Le prototype muséographique a fait l'objet d'une évaluation en situation de visite du musée. Le retour des utilisateurs a permis de valider l'intérêt de ce dispositif pour les visiteurs. L'implication des équipes du musée (conservateurs

10. processus de validation des contenus, mise à jour automatique des dispositifs muséographiques, création de parcours de visite personnalisés, assistance pour la préparation des visites guidées pour les médiateurs, etc.

et service du public principalement) autour de ce système démontre l'intérêt des avancées apportées. Le système d'information prévoit la possibilité d'étendre le corpus grâce à de futures campagnes de recherche ou par le biais de contributions d'utilisateurs.

3.2 Le cas des halles Alstom : limites et perspectives de la méthodologie

Les halles Alstom constituent un cas représentatif de l'histoire du port de Nantes : ce fut un des premiers lieux à être industrialisé (1850) et un des derniers en activité (2001). Ce lieu est un point d'intérêt déjà identifié dans la base de données du projet « Nantes1900 » présenté précédemment. Il s'agit ici d'une étude dont le contenu plus détaillé permettrait de créer de nouveaux liens dans le réseau d'informations.

Ce travail historique typique présente des informations relatives à un objet avec une approche difficilement transposable dans une base de données. C'est donc un cas d'étude intéressant pour confronter la méthodologie à la pratique de l'histoire. Il s'agit de mesurer la plus-value et les éventuelles pertes de notre approche vis-à-vis d'un travail historique :

— la réaction du système à l'intégration de nouveaux contenus hétérogènes et présentant de nouvelles caractéristiques (niveaux de détail différents, nature des informations, etc.) ;
— l'apport de nouvelles pistes de recherche par rapport au travail historique classique.

Un exemple de difficulté consiste à différencier les deux approches historiques des halles Alstom décrites par la figure 4. Le système proposé doit permettre à l'utilisateur de saisir ces lectures complémentaires. De nouvelles lectures doivent également pouvoir être proposées par l'un (système) ou l'autre (utilisateur) et validées conjointement. Cette mise en réseau des informations proposera de multiples portes d'entrée pour un accès multi-niveaux à ces connaissances historiques permettant la compréhension du lieu industriel étudié et de son contexte. Parmi

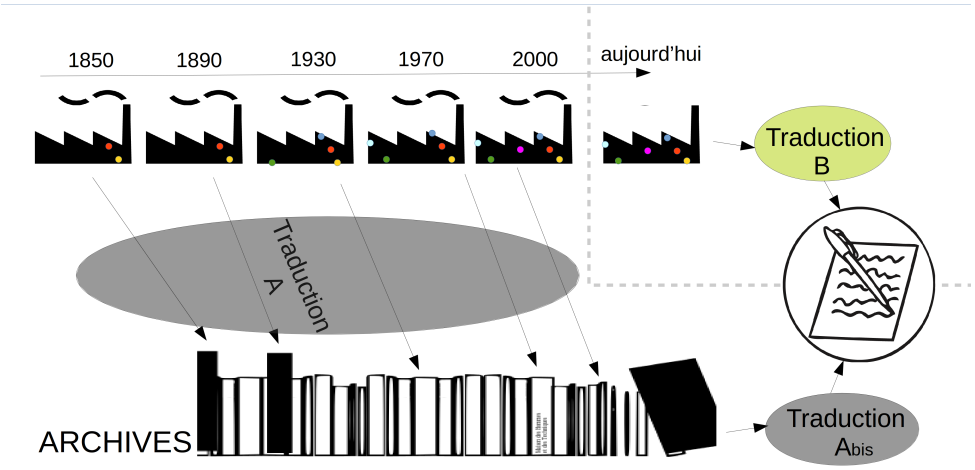


FIGURE 4 – Plusieurs approches complémentaires peuvent contribuer à l'écriture d'un récit historique. Ici, l'archéologie industrielle (traduction B) c'est-à-dire l'étude *in situ* des traces (symbolisées par des points de couleurs sur l'usine) est complétée par l'étude des archives (traductions A+Abis).

ces portes d'entrée, on peut citer des approches : archéologie industrielle, témoignages, étude d'archives, sources secondaires… ; ou bien des dimensions : économie, culture, mécanique,

social... ; mais également des sous-systèmes : ponts roulants, chaînes de montage, processus industriels comme l'usinage des engrenages...

Ce cas d'étude illustre le besoin de mécanismes supplémentaires pour l'intégration des données sans reformuler l'intégralité du récit.

Nous travaillons à l'implémentation d'outils de traitement automatique du langage comme ANA (Enguehard & Pantera, 1995). Ces outils permettent une *indexation supervisée* de productions écrites d'historiens pour faciliter l'intégration de nouveaux contenus, proposer un accès multi points de vue et apporter des moyens complémentaires d'analyse. Cette indexation supervisée automatise le processus de création de liens entre les fiches descriptives (nœuds orange et bleus sur la figure 2) à partir d'une monographie. Ces nœuds automatiquement créés et indexés vont pouvoir être enrichis par de nouvelles études plus spécifiques comme celle sur les halles Alstom. Les liens permettent d'organiser les connaissances autour d'un même objet avec différents points de vue (figure 4) et de les organiser au sein d'un réseau d'informations plus vaste (comme celui créé pour le projet "Nantes1900"). Cela permet finalement un accès multi échelles : du port et des thématiques génériques associées ("techniques de construction") jusqu'à l'étude d'un bâtiment en particulier et de ses thématiques spécifiques ("maçonneries de moellons hourdés de chaux"). Plusieurs études complémentaires provenant de différents auteurs pourraient également être prises en compte et confrontées les unes aux autres. Le modèle conceptuel proposé figure 1 va permettre cette saisie.

4 Discussion et conclusion

D'autres cas études en cours nous permettent de compléter la méthodologie proposée et illustrée précédemment. Parmi ces exemples, on peut citer le cas de la reconstitution des forges de Paimpont [11] ou celle de la poudrerie royale de Saint-Chamas [12]. Tous ces projets présentent des caractéristiques différentes. Cependant, la méthodologie proposée dans ce papier permet d'avoir une approche unifiée pour la gestion des connaissances et leur valorisation.

À partir de ces cas d'étude, nous identifions des limites, conceptuelles ou technologiques :

— la prise en compte des profils contributeurs nécessite une analyse et une modélisation fine qui méritent un approfondissement spécifique ;

— pour garantir la cohérence du système, il est nécessaire de mettre en place des traitements de texte spécifiques comme l'extraction (semi-)automatique et la normalisation des descripteurs pour permettre une indexation contrôlée. C'est une perspective d'amélioration forte sur laquelle nous travaillons ;

— la mise en place de procédures de *crowd-sourcing* pour la validation d'éventuelles propositions automatiques du système. Ces mécanismes permettraient d'augmenter la fiabilité et la maturité des informations.

L'objectif est de permettre une meilleure appropriation des informations existantes pour l'identification de « zones d'ombre », de litiges ou de nouvelles hypothèses à explorer. Il s'agira notamment de garantir la cohérence du système en identifiant les liens possibles entre les enregistrements de la base de données, mais également de mettre en évidence de nouveaux liens entre les fiches.

11. Voir http://forgesdepaimpont.fr/La-realite-augmentee-aux-Forges-de-Paimpont

12. Voir https://projetpoudrerie.wordpress.com

La méthode actuelle, en grande partie manuelle et donc fastidieuse implique une normalisation du vocabulaire par les utilisateurs et un nombre croissant d'opérations pour la gestion des informations à mesure que celles-ci sont ajoutées dans le système.

Références

ARDANS (2011). *Ardans Knowledge Maker : Introduction , principes et philosophie implantés dans cet environnement de gestion des connaissances*. Rapport interne 0.

BERETTA F. & VERNUS P. (2012). Le projet SyMoGIH et la modélisation de l'information : une opération scientifique au service de l'histoire. *Les Carnets du LARHRA*, (1), 81–107.

COTTE M. (2009). Les techniques numériques et l'histoire des techniques. Le cas des maquettes virtuelles animées. *Documents pour l'histoire des techniques*, (18), 7–21.

COTTE M. (2010). La génétique technique a t-elle un avenir comme méthode de l'histoire des techniques ? In A.-L. REY, Ed., *Méthode et Histoire, journées d'études de la SFHST, Lille, 2007*, p. 187–201, Paris : publications de la SFHST.

DAVALLON J. (2002). Comment se fabrique le patrimoine ? *Sciences humaines. Hors série 36*, (36), 74–77.

DE LUCA L., BUSAYARAT C., STEFANI C., VÉRON P. & FLORENZANO M. (2011). A semantic-based platform for the digital analysis of architectural heritage. *Computers & Graphics*, **35**(2), 227–241.

DEUSCHEL T., HEUSS T., HUMM B. & FRÖHLICH T. (2014). Finding without Searching - A Serendipity-based Approach for Digital Cultural Heritage. In *Digital Intelligence*, Nantes.

ENGUEHARD C. & PANTERA L. (1995). Automatic natural acquisition of a terminology*. *Journal of quantitative linguistics*, **2**(1).

ERMINE J.-L., PAUGET B., BERETTI A. & TORTORICI G. (2004). Histoire et Ingénierie des Connaissances. In *Sources et ressources pour les sciences sociales*, Paris, France.

FLEURY P. (1997). La Rome Antique sur l'Internet. *Revue Informatique et Statistique dans les Sciences Humaines*, **33**, 146–162.

GREFFARD N., PICAROUGNE F. & KUNTZ P. (2013). TempoSpring : A new immersive hands-free prototype for visualizing social networks : Demonstration paper. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, p. 1–2 : IEEE.

GUIDI G., FRISCHER B., LUCENTI I. & DONNO J. (2008). Virtualising ancient imperial Rome : from Gismondi's physical model to a new virtual reality application. *International Journal of Digital Heritage*.

PRÉVÔT N. (2013). The Digital Puzzle of the talatat from Karnak. A Tool for the Three-Dimensional Reconstruction of Theban Buildings from the Reign of Amenhotep IV. In S. POLIS & J. WINAND, Eds., *Texts, Languages & Information Technology in Egyptology.*, p. 129–138, Liège.

SERRAFERO P. (2000). Cycle de vie, maturité et dynamique de la connaissance : des informations aux cognitons de l'Entreprise Apprenante. *Revue Annuelle UE des Arts et Métiers sur le Knowledge Management*, p. 158–169.

SRINIVASAN R. & HUANG J. (2004). Fluid ontologies for digital museums. *International journal on digital libraries*, **5**(3), 193–204.

STIEGLER B. (2009). Technologies culturelles et économie de la contribution. *Culture et recherche*, (121), 31.

ZACKLAD M., BÉNEL A., ZAHER L., CAHIER J.-P. & ZHOU C. (2007). Hypertopic : une métasémiotique et un protocole pour le Web socio-sémantique. In FRANCKY TRICHET, Ed., *18eme journées francophones d'ingénierie des connaissances*, p. 217–228 : Cépaduès.

Une ontologie documentaire pour l'accès aux contenus juridiques

Nada Mimouni, Adeline Nazarenko et Sylvie Salotti

LIPN, CNRS (UMR 7030), Université Paris 13
Sorbonne Paris Cité, F-93430 Villetaneuse
Nada.Mimouni, Adeline.Nazarenko, Sylvie.Salotti@lipn.univ-paris13.fr

Résumé : Hormis le vocabulaire juridique souvent complexe, une des difficultés de l'accès à l'information juridique réside dans le fait qu'un document contient généralement de nombreuses références vers d'autres textes qui sont nécessaires à son interprétation. Ces références sont de différentes natures (références à la législation, décisions de jurisprudence, modification, abrogation, codification, etc.). L'usager se trouve donc face à un réseau documentaire complexe pour lequel il est nécessaire de proposer des fonctionnalités d'accès avancées. Les systèmes mis en place dans différents pays (comme Legifrance et Normattiva) ne répondent que partiellement aux besoins des professionnels du domaine, car ils ne permettent pas d'interroger directement les différentes relations entre les documents. Dans cet article, nous proposons une modélisation ontologique d'une collection documentaire juridique permettant de représenter et d'interroger différentes dimensions : le contenu sémantique des documents, les relations intertextuelles et l'évolution de la collection, et nous présentons sa mise en oeuvre dans le cadre du projet Légilocal.

Mots-clés : Accès à l'information juridique, collection documentaire, relations intertextuelles, réseau sémantique, modèle ontologique.

1 Introduction

Dans le domaine juridique, des documents de différents types sont créés tous les jours : directives européennes, textes législatifs, jurisprudence, décisions administratives. Ils sont souvent créés pour transposer, modifier, codifier, appliquer ou annuler des documents antérieurs générant éventuellement de nouvelles versions de ces documents. De ce fait, ces documents sont reliés entre eux par différents types de liens et forment ainsi un large réseau de documents. La diversité des liens entre les documents a été reconnue comme étant la principale source de complexité dans un réseau de documents juridique (Bourcier, 2011) et elle doit être prise en compte dans les systèmes d'accès à l'information juridique pour satisfaire au mieux les utilisateurs du domaine.

L'ensemble des documents reliés à un document juridique par des références forment un contexte nécessaire à son interprétation. Dans les systèmes existants, une recherche par mots clés permet de retrouver les documents sur la base de leurs contenu. Les relations vers d'autres documents sont parfois traduites par des liens hypertextes, et l'utilisateur doit naviguer de proche en proche dans les liens pour construire le contexte.

Dans le cadre du projet Légilocal (Amardeilh *et al.*, 2013) [1], une analyse des besoins en collaboration avec des experts juridiques partenaires du projet a montré que les utilisateurs expriment leurs besoins sous forme de requêtes complexes qui portent aussi bien sur le contenu que sur les liens intertextuels entre les documents. Par exemple : « *Quels arrêtés municipaux concernant les chemins ruraux ont fait l'objet d'un recours et ont été annulés par décision de tribunal ?* ». Les systèmes existants ne permettent pas de traiter ce genre de requêtes.

Nous considérons une collection juridique comme étant l'ensemble des documents reliés par des liens intertextuels. Nous proposons un modèle documentaire basé sur une approche sémantique dans le but de modéliser la collection puis de l'interroger. L'essentiel de notre effort a consisté à intégrer la dimension intertextuelle dans une ontologie documentaire adaptée aux documents juridiques. Une telle ontologie permet de représenter le contenu sémantique du document (ce dont parle le document), sa structure logique, ses différentes versions, ainsi que la structure de la collection documentaire qui organise différents types de documents dans un vaste réseau de liens intertextuels.

La section 2 présente les principaux modèles documentaires existants. La section 3 décrit les besoins auxquels la recherche d'information juridique se trouve confrontée. La section 4 présente l'ontologie documentaire que nous proposons avec les différents modules la composant et leurs dépendances. La section 5 présente des exemples d'utilisation de cette ontologie pour la mise à jour d'une collection documentaire et pour répondre à des requêtes relationnelles.

2 Modélisation de collections documentaires

Les modèles documentaires classiques décrivent le contenu des documents par des métadonnées. Des modèles standards ont été créés, comme le Dublin Core dont s'inspirent plusieurs modèles actuels pour la description et l'interrogation des documents. Les approches généralistes de recherche d'information (RI) modélisent les documents comme un sac de mots sur lesquels porte la recherche. Ils ont été améliorées avec des fonctionnalités sémantiques pour faire face à la richesse du contenu (Baziz *et al.*, 2005) mais ces modèles considèrent généralement le document comme "un texte plat".

Dans le domaine juridique, des efforts importants ont été consacrés à la structuration et la publication de l'information juridique. Plusieurs standards juridiques XML ont été développés (Sartor *et al.*, 2011), comme CEN-Metalex [2] et AkomaNtoso [3], dans le but d'améliorer l'interopérabilité et l'échange de sources entre les instances parlementaires et

1. Ce travail a été partiellement financé par le projet LEGILOCAL (FUI-9, 2010-2013) et par le Labex EFL (ANR-10-LABX-0083).

2. http ://www.metalex.eu/

3. http ://www.akomantoso.org/

les gouvernements. D'autres fonctionnalités avancées sont prévues telles que la gestion de références et de modifications (Palmirani & Cervone, 2009) , la mise à jour automatique (Brighi & Palmirani, 2009) ou l'accès à une version d'un texte législatif à une date donnée [4]. Cependant, ces fonctionnalités sont gérées au niveau document sans tenir compte de la structure globale de la collection et dans la plupart des cas pour un seul type de documents (la législation).

Les collections de documents sont souvent représentées comme un ensemble de documents isolés mais aussi comme un graphe hypertextuel (le web) où les relations intertextuelles correspondent à des liens hypertextes non typés. Le modèle FRBR (Tillett, 2004) [5] a été défini pour tenir compte des différentes classes d'objets informationnels permettant de rattacher les différentes versions d'un document à une source commune, l'œuvre.

Au-delà de la modélisation du contenu, des ontologies ont été produites pour modéliser les propriétés documentaires. Elles s'inspirent naturellement des langages de métadonnées définis dans la tradition des documentalistes, comme le Dublin Core. Ces ontologies sont souvent conçues pour des usages particuliers et mettent l'accent sur différents types de propriétés documentaires (par ex. les ontologies SDO [6], SAO [7]) ou sur le cycle de vie d'un document de travail (par ex. l'ontologie PDO [8]). Des ontologies comme LKIF core (Hoekstra *et al.*, 2009) ou l'ontologie qui décrit le vocabulaire du standard MetaLex (Hoekstra, 2011) ont été proposées dans le domaine juridique.

Cette analyse montre qu'il n'existe pas un modèle qui rend compte de toute la richesse d'une collection juridique. Nous estimons que l'exploitation de la structure de graphe de documents et la sémantique des nœuds et des liens dans la modélisation des collections documentaires permettra d'offrir de nouvelles fonctionnalités d'accès à l'information. Dans la suite nous présentons une modélisation qui rend compte de ces aspects dans le but de faciliter l'accès à l'information dans les collections juridiques.

3 Besoins d'accès à l'information juridique

Traditionnellement, la recherche se fait en utilisant des termes qui peuvent être des mots clés dans le texte, des métadonnées attachées aux documents ou des concepts qui décrivent leur contenu sémantique. Les systèmes juridiques actuels reposent sur des méthodes logiques pour répondre à la contrainte de l'exhaustivité des résultats d'une recherche dans

4. Fonctionnalité proposée par UK Legislation : http ://www.legislation.gov.uk/ ou Normattiva : http ://www.normattiva.it/ricerca/avanzata/aggiornamenti

5. FRBR introduit la distinction entre l'œuvre (*work*), ses différentes expressions (*expression*), les manifestations (*manifestation*) de ces dernières et les différents exemplaires (*item*) qui en résultent.

6. SALT Document Ontology, http ://salt.semanticauthoring.org/ontologies/sdo

7. SALT Annotation Ontology, http ://salt.semanticauthoring.org/ontologies/sao

8. Project Documents Ontology, http ://vocab.deri.ie/pdo-Document

ce domaine : les utilisateurs ont besoin d'avoir tous les documents correspondants à leurs critères de recherche. Sur un critère sémantique, une requête peut être : « *Quels sont les arrêtés concernant les **chemins ruraux** et les **véhicules à moteurs** ?* ».

Certains systèmes offrent en plus des moyens pour satisfaire des besoins juridiques spécifiques tel que l'accès à une version consolidée d'un document (il s'agit de retrouver la version en vigueur d'un document ou accéder à une version antérieure en vigueur à une date donnée). Ceci permet de répondre à des requêtes comme : « *Quelle est la **dernière version** (ou la **version en vigueur**) de l'article L362-1 du code de l'environnement ?* ».

D'autres besoins spécifiques ne sont pas satisfaits par les systèmes actuels. Nous avons identifié ces besoins en analysant les requêtes des experts juristes partenaires du projet Légilocal :

- *"J'aimerais voir les arrêtés municipaux concernant les chemins ruraux quiont **fait l'objet d'un appel** et ont été **annulés par** décision de jurisprudence"*
- *"Quels sont les articles de code **cités par** les arrêtés municipaux qui concernent les chemins ruraux et qui ont été **confirmés** ?"*
- *"Je me demande si les textes **visés par** les arrêtés municipaux portant sur les chemins ruraux sont également **cités par** ceux concernant les véhicules à moteur"*
- *"Quelle est la jurisprudence qui **applique** l'article sur la responsabilité pour faute du Code Civil ?"*

Ces requêtes montrent que l'on a besoin d'interroger les collections juridiques en fonction des types de documents (décrets, lois, etc.), des descripteurs sémantiques qui leur sont associés (*e.g.* chemins ruraux, véhicules à moteur), de leur structure interne (articles de code, etc.) mais aussi en fonction des relations que les documents entretiennent (annule, cite, applique, confirme, etc.), ce qui n'est pas possible avec les systèmes existants.

4 Modèle ontologique d'une collection juridique

Nous proposons de définir une ontologie qui permette de représenter de manière homogène toutes les caractéristiques d'une collection juridique : i) le contenu sémantique des documents, ii) la typologie et structure des documents, iii) la gestion des versions et des liens intertextuels. L'ontologie est décrite en OWL-DL et elle réutilise des vocabulaires définis dans le web de données extraits du schéma Dublin Core, des ontologies DCMI terms, WGS84 Geo Positioning, Metalex, FRBR et Event [9].

9. http ://purl.org/dc/elements/1.1/, http ://purl.org/dc/terms/, http ://www.w3.org/2003/01/geo/wgs84_pos, http ://justinian.leibnizcenter.org/MetaLex/metalex-cen.owl, http ://vocab.org/frbr/core.html, http ://-purl.org/NET/c4dm/event.owl

4.1 Contenu sémantique des documents

Pour répondre au besoin d'accéder au contenu sémantique des documents, nous proposons de modéliser les annotations sémantiques, généralement représentées dans des ressources sémantiques, comme un module de l'ontologie décrit en SKOS. Ce choix est fait pour deux raisons : avoir un modèle plus léger qu'avec des classes OWL et leurs individus (Reymonet *et al.*, 2007), faciliter la réutilisation de ressources existantes qui sont généralement décrites en SKOS. Utiliser conjointement OWL et SKOS dans une même conceptualisation est décrit par une recommandation de W3C [10]. Nous faisons correspondre à une ressource sémantique un concept terminologique qui représente la classe de termes de cette ressource. Par exemple, `GeoConcept` est un concept terminologique, en relation d'héritage (`rdfs:subClassOf`) avec `skos:Concept`, qui représente la classe de tous les termes géographiques (parc régional, parc naturel marin, etc.) comme décrit sur la figure 1. Les termes du domaine correspondent à des instances de cette classe.

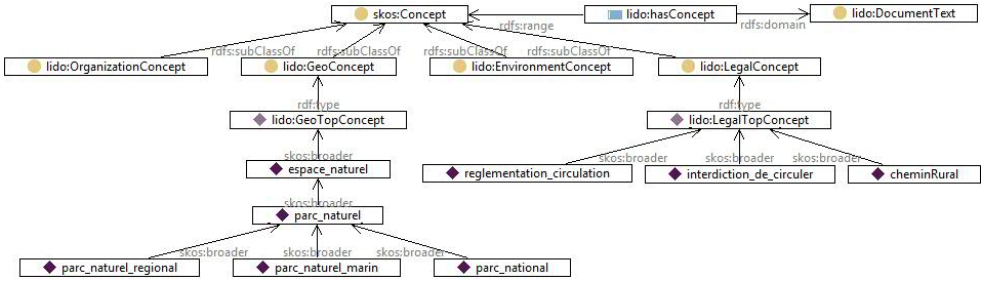


FIGURE 1 – Concepts terminologiques représentant les ressources sémantiques (utilisation des propriétés `skos:broader`, `skos:narrower`).

Les descripteurs sémantiques de contenu (les instances) sont organisés en hiérarchie, ce qui permet d'exploiter les liens de spécialisation/généralisation pour ajuster la précision des requêtes. Par exemple, la requête « *Quels sont les textes législatifs qui parlent de* ***véhicules à moteur*** *?* » est une généralisation de la requête « *Quels sont les textes législatifs qui concernent* ***les 4x4*** *?* » et permet potentiellement de retrouver plus de documents.

Le module sémantique est relié aux documents par la propriété `hasConcept` définie entre un texte juridique `DocumentText` et un concept du module sémantique.

Dans le projet Légilocal, des ressources terminologiques ont été développées pour définir les termes utilisés pour l'annotation sémantique et sont organisés en différents modules (figure 1) : géographique (`GeoConcept`), organisation (`OrganizationConcept`) et juridique (`LegalConcept`).

10. http ://www.w3.org/2006/07/SWD/SKOS/skos-and-owl/master.html#Hybrids

4.2 Typologie et structure des documents

4.2.1 Hiérarchie des types des documents

Pour préparer un acte municipal sur un sujet particulier, les agents des administrations locales (agents de mairie) doivent examiner la législation nationale et la jurisprudence sur le même sujet. Il faut pouvoir accéder à tous ces types de documents. Dans le domaine juridique, on distingue différents types de documents que nous avons modélisés dans le cadre du projet Légilocal : législation (lois, codes), jurisprudence (décisions de justice, jugements), documents administratifs ou actes locaux (décisions, arrêtés), documents éditoriaux. La hiérarchie des types de documents est fortement structurée. Nous modélisons cette hiérarchie par les classes de l'ontologie, une classe par type de document, à laquelle sont attachées des attributs et des propriétés spécifiques à ce type de document. Le haut niveau de cette hiérarchie est présenté dans la figure 2. Les trois catégories principales permettent de distinguer :

- les documents des collectivités territoriales (dont ceux des mairies) : classe `LocalAuthorityAct` ;
- les documents éditoriaux (revues, guides, modèles) : classe `EditorialDocument`. Ces documents aident les administrateurs locaux à créer leurs propres actes et font généralement référence à la législation et à la jurisprudence ;
- les documents correspondant aux sources du droit (classe `SourceOfLaw`) parmi lesquels on distingue la législation (`Legislation`) et la jurisprudence (`CaseLaw`).

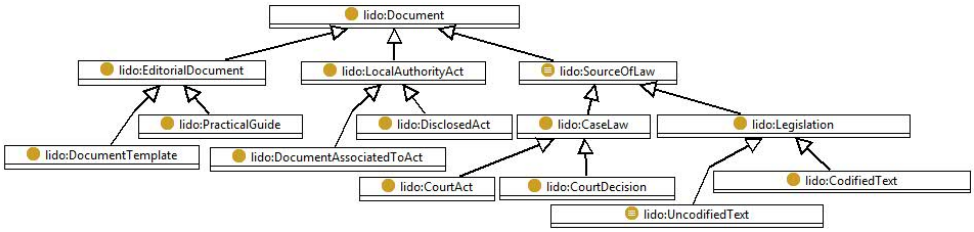


FIGURE 2 – Hiérarchie des types de documents.

4.2.2 Structure : fragments et unités documentaires

Selon son type (par ex. acte local ou document législatif) un document possède une structure particulière qui est importante à préciser. L'intérêt de l'utilisateur (citoyen, personnel administratif ou juriste dans le cas de Légilocal) porte souvent sur une partie du texte (fragment) plutôt que sur le texte dans son ensemble. Cela suppose que les métadonnées d'identification et les annotations sémantiques soient attachées non pas au texte

globalement mais à ses sous-parties (Hoekstra, 2011). Les mêmes besoins sont valables pour les références entre les textes permettant une analyse fine de leurs interdépendances.

Une unité documentaire est un fragment qui peut être cité et le réseau documentaire est construit entre ces unités. Dans certains textes juridiques, l'article constitue l'unité documentaire de base. Un article peut être directement cité et possède en général un cycle de vie propre : il peut être modifié, codifié, etc. indépendamment du document auquel il appartient. Pour d'autres types (par ex. les actes locaux), l'unité documentaire de base est le document tout entier.

Nous représentons une collection documentaire comme un ensemble d'unités documentaires (classe `DocumentaryUnitWork`) et de fragments de documents (classe `DocumentText`) plutôt que comme un ensemble de documents. La figure 3 montre les classes modélisant la structure d'un document.

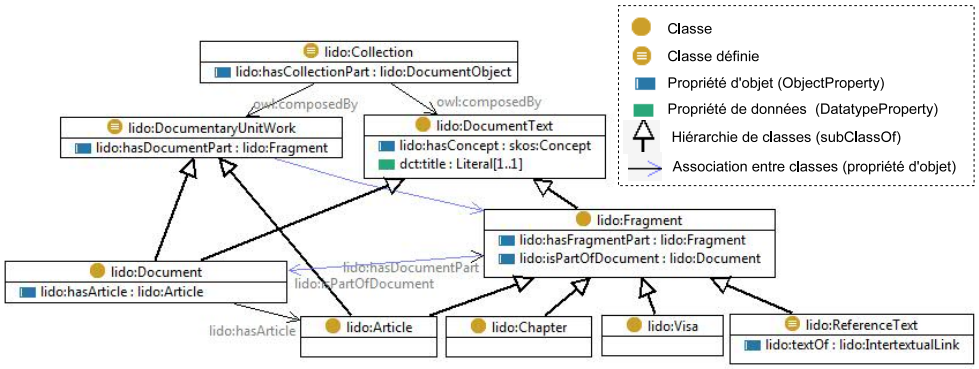


FIGURE 3 – Les classes modélisant les unités documentaires et la structure d'un document.

4.2.3 Niveaux du document : œuvre et expression

La modélisation du cycle de vie des collections documentaires amène naturellement à distinguer plusieurs niveaux de représentation des objets documentaires. En effet, il faut prendre en compte l'historique des différentes versions d'un document juridique : même si celles-ci se remplacent les unes les autres, elles coexistent au sein du système juridique puisque chacune est la version de référence (version en vigueur) pour une période donnée. Nous introduisons donc dans le modèle la distinction entre le document maître, l'œuvre, et les différentes versions (*expressions*) qui en sont données. Nous suivons en cela l'approche proposée par Metalex qui repose sur la distinction classique introduite par le modèle FRBR. Cela permet de factoriser une partie des propriétés documentaires sur l'œuvre sans les dupliquer sur chacune de ses expressions.

Dans l'ontologie, nous distinguons les classes `DocumentaryUnitWork` et `DocumentaryUnitExpression`. La première classe correspond à l'unité documentaire en tant qu'œuvre comme par exemple l'*article L2213 – 2 du code général des collectivités territoriales*, alors que la seconde classe correspond à chaque version de cet article (des expressions différentes). Les documents de la collection sont représentés comme des instances de ces deux classes.

4.3 Structure relationnelle d'une collection

4.3.1 Liens œuvres-expressions

Une œuvre est un objet documentaire réalisé par une ou plusieurs expressions et une expression est un objet documentaire qui réalise une œuvre. Dans l'ontologie nous utilisons les propriétés `metalex:realizes` et `metalex:realizedBy` pour décrire cette relation (figure 4).

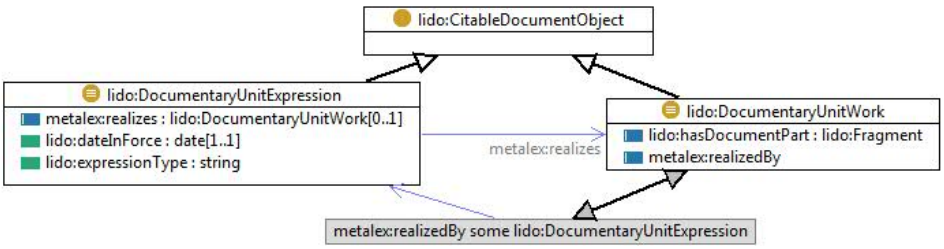


FIGURE 4 – Relation structurelle entre les niveaux de document : œuvre et expressions.

Les expressions représentent les versions d'une même œuvre. Toute modification dans une expression produit une nouvelle expression. Des relations de version (version suivante, version précédente) lient les expressions entre elles. Bien qu'elles soient accessibles via des liens hypertextes (par ex. dans Legifrance), ces relations sont implicites : elles ne sont pas décrites directement dans les textes des documents. On peut toujours retrouver une version à une date données à partir de l'œuvre à laquelle elle se rattache et de sa date.

4.3.2 Liens structurels

Les collections sont composées d'unités documentaires : des documents ou des articles (figure 3). Certains documents, comme les codes ou les lois, sont composés d'articles. La relation de composition est décrite par la propriété `hasArticle` entre les deux classes `Document` et `Article`.

4.3.3 Liens sémantiques entre documents

Les liens intertextuels qui apparaissent dans les textes des documents (*e.g.* citation, visas, application) sont modélisés par la classe `Citation` et ses propriétés `citationSource` et `citationTarget` vers des unités documentaires. Dans notre modèle, chaque type de relation est associée à une source et une cible spécifiques, ce qui permet de spécifier non seulement à quels types et parties de textes il réfère, mais aussi dans quels types de textes et parties de textes le lien modélisé peut apparaître.

La distinction que nous définissons entre œuvre et expression implique de spécifier à quel niveau se situent les relations d'intertextualité introduites. Ces relations partent en général des expressions du fait qu'une nouvelle expression (version) peut faire référence à un ensemble de documents différent de celui référencé par la version précédente. Selon les cas, la cible de la relation est soit une expression (dans le cas d'un article ayant par nature plusieurs versions) soit une œuvre (dans le cas où une nouvelle version implique la création d'un nouveau document).

4.3.4 Opérations documentaires

Certaines opérations documentaires comme la modification font intervenir plus de deux documents : les documents source et cible de la modification mais aussi le document résultat (voir figure 5).

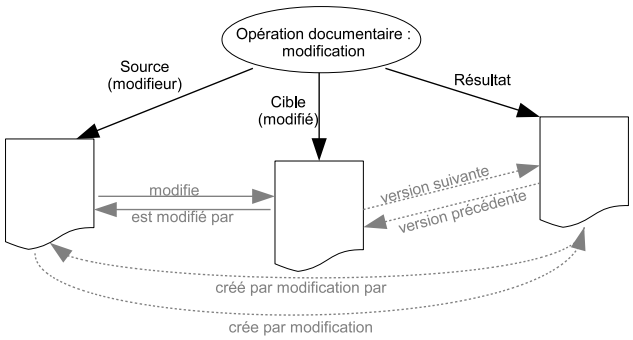


FIGURE 5 – Opération documentaire de modification : participants et liens de référence et citation résultants.

En effet, l'analyse de collections documentaires montre que les relations intertextuelles décrivent l'activité juridique où les actions (décisions, jugements, recours, régulation, etc.) se réalisent au travers de la publication de nouveaux documents qui font référence aux précédents et à ceux qui les modifient ou dont ils s'inspirent. Cela nous incite à modéliser explicitement les opérations documentaires sous-jacentes aux relations intertextuelles (pour

prendre en compte le cas de relations qui prennent une, deux ou trois unités documentaires comme arguments).

Dans l'ontologie, l'intertextualité est modélisée par la classe `IntertextualLink` et ses sous-classes `Citation` (représente les liens qui possèdent une trace dans le texte) et `DocumantaryOperation` (représente les opérations documentaires et leurs propriétés (l'agent, la date, le lieu) même si on peut aussi les attacher au document source de l'opération) comme décrit dans la figure 6. Les objets de la classe `IntertextualLink` sont attachés par la propriété `textOf` aux fragments de texte qui contiennent le lien, représentés par la classe `ReferenceText` (figure 3).

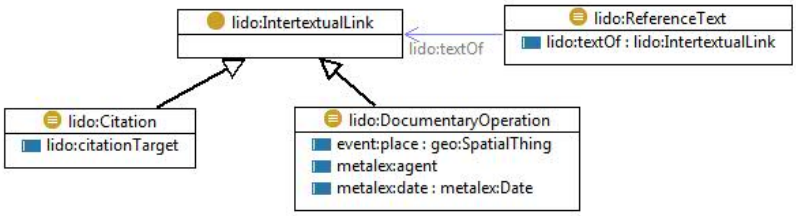


FIGURE 6 – Modélisation des liens intertextuels.

5 Mise en œuvre

5.1 La collection Légilocal

La collection sur laquelle nous avons travaillé dans le cadre du projet Légilocal contient des documents de différents types, les documents peuvent être composés de plusieurs articles et possèdent plusieurs types de relations entre eux. Les documents sont collectés à partir de plusieurs sources : il s'agit de décisions publiées par des collectivités locales, de décisions de jurisprudence et de textes législatifs (lois, décrets, etc) issus de portails juridiques, principalement Legifrance. Nous avons construit un jeu de requêtes relationnelles avec leurs réponses pertinentes sur cette collection avec l'aide des experts juristes du projet (Mimouni *et al.*, 2014).

Le codage de cette collection se fait de manière naturelle sous la forme de graphes RDF. Des contraintes ont été créées dans l'ontologie afin d'assurer la cohérence des données au moment de l'insertion de nouveaux objets dans la base. La figure 7 montre un exemple d'instantiation avec un micro corpus pour décrire l'opération de codification de l'article $L362-1$ du code de l'environnement par l'ordonnance n°$2000-914$.

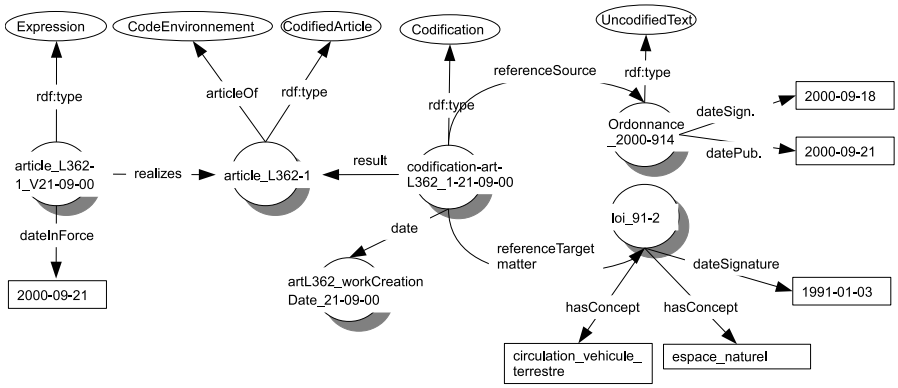


FIGURE 7 – Codification de l'article $L362-1$ du code de l'environnement par l'Ordonnance n°$2000-914$.

5.2 Interrogation

Une fois une collection documentaire modélisée comme une instantiation de cette ontologie, les requêtes relationnelles peuvent se traduire sous la forme de requêtes SPARQL. Par exemple : « *Quels sont les textes législatifs sur lesquels s'appuient les décisions de jurisprudence qui ont annulé des arrêtés municipaux parlant d'interdiction de circuler ?*». La relation "s'appuie sur" correspond à la relation applique modélisée par la classe `Application` sous classe de `Citation` (relation binaire). La relation "annule" est modélisée par une opération documentaire `Annulation` sous-classe de `Decision` (relation ternaire).

```
SELECT ?text ?decision
WHERE {
   ?decision rdf:type lido:CaseLaw .
   ?decree rdf:type lido:LocalDecree .
   ?application rdf:type lido:Application .
   ?annulation rdf:type :Annulation .
   ?application lido:citationSource ?decision .
   ?application lido:citationTarget ?text .
   ?annulation lido:referenceSource ?decision .
   ?annulation lido:referenceTarget ?decree .
   ?decree lido:hasConcept :interdiction_de_circuler .
}
```

6 Conclusion

Nous avons proposé un modèle de collection documentaire qui permet d'articuler le niveau local du contenu du document avec le niveau global de la collection. Dans ce mo-

dèle, une collection documentaire est un réseau sémantique dont les noeuds sont des unités documentaires qui peuvent être soit des documents complets soit des articles qui le composent, il s'ensuit que la relation de composition est une relation intertextuelle au même titre que les relations de citation ou de modification. Ceci montre la complexité intertextuelle d'une collection juridique, critère important à prendre en compte pour la définition de fonctionnalités pour l'interrogation relationnelle dans un système opérationnel. Pour aboutir à un tel système, un travail de préparation de la collection doit être fait avec notamment l'annotation des documents (par des métadonnées et des descripteurs sémantiques), l'analyse des liens (repérer les liens, les typer sémantiquement et identifier leurs cibles) et la définition de méthodes de traduction automatique des requêtes en SPARQL.

Références

AMARDEILH F., BOURCIER D., CHERFI H., DUBAIL C., GARNIER A., GUILLEMIN-LANNE S., MIMOUNI N., NAZARENKO A., ÈVE PAUL, SALOTTI S., SEIZOU M., SZULMAN S. & ZARGAYOUNA H. (2013). The légilocal project : the local law simply shared. In I. PRESS, Ed., *Legal Knowledge and Information Systems - JURIX, Italy*, p. 11–14.

BAZIZ M., BOUGHANEM M., AUSSENAC-GILLES N. & CHRISMENT C. (2005). Semantic cores for representing documents in ir. In *Proc. of the ACM symposium on Applied computing*, p. 1011–1017, NY, USA.

BOURCIER D. (2011). Sciences juridiques et complexité. un nouveau modèle d'analyse. *Droit et Cultures*, **61**(1), 37–53.

BRIGHI R. & PALMIRANI M. (2009). Legal text analysis of the modification provisions : a pattern oriented approach. In *Proceedings ICAIL '09*, p. 238–239, New York, NY, USA : ACM.

HOEKSTRA R. (2011). The metalex document server : legal documents as versioned linked data. In *Proceedings of the 10th International Conference on the Semantic Web*, ISWC'11, p. 128–143.

HOEKSTRA R., BREUKER J., BELLO M. D. & BOER A. (2009). Lkif core : Principled ontology development for the legal domain. In *Proceedings of the 2009 conference on Law, Ontologies and the Semantic Web*, p. 21–52, Amsterdam : IOS Press.

MIMOUNI N., NAZARENKO A., PAUL È. & SALOTTI S. (2014). Towards graph-based and semantic search in legal information access systems. In *JURIX 2014*, p. 163–168 : IOS Press.

PALMIRANI M. & CERVONE L. (2009). Legal change management with a native xml repository. In *JURIX 2009 : The Twenty-Second Annual Conference on Legal Knowledge and Information Systems, Rotterdam, The Netherlands, 16-18 December 2009*, p. 146–155.

REYMONET A., THOMAS J. & AUSSENAC-GILLES N. (2007). Modelling ontological and terminological resources in owl dl. In *Proceedings of ISWC*, volume 7.

SARTOR G., PALMIRANI M., FRANCESCONI E. & BIASIOTTI M. A. (2011). *Law, Governance and Technology : Legislative Xml for the Semantic Web : Principles, Models, Standards for Document Management*. Law, Governance and Technology Series, 4. Springer London, Limited.

TILLETT B. (2004). *What is FRBR ? A Conceptual Model for the Bibliographic Universe*. Library of Congress Cataloging Distribution Service.

Conception interactive d'ontologies par élimination de mondes possibles

Sébastien Ferré

IRISA/Université de Rennes 1
Campus de Beaulieu, 35042 Rennes cedex
ferre@irisa.fr

Résumé : La conception d'ontologies constitue souvent un frein à l'adoption des techniques de l'ingénierie des connaissances et du Web sémantique. Une raison est bien sûr l'emploi de formalismes et des concepts logiques qui y sont associés. Une autre raison qui nous semble plus profonde est le fossé entre syntaxe et sémantique, c'est-à-dire entre la forme de surface de l'ontologie (axiomes) et ce qu'elle rend nécessaire/possible/impossible (modèles). Ce fossé entraîne des divergences entre l'intention du concepteur et sa modélisation qui se manifestent par des inférences inattendues, voire des incohérences. Nous proposons une nouvelle approche de conception d'ontologies fondée sur l'exploration et l'élimination interactive de "mondes possibles" (modèles). Elle réduit le fossé syntaxe/sémantique en interdisant par construction la production d'incohérence, et en montrant en permanence au concepteur ce qui peut être inféré ou non. Un prototype, PEW (Possible World Explorer), permet d'expérimenter cette approche et de la comparer à d'autres éditeurs d'ontologies.

Mots-clés : Web sémantique, ontologies, OWL, conception, fossé syntaxe/sémantique, approche interactive.

1 Introduction

La conception d'ontologies est généralement une étape essentielle dans l'application des techniques d'ingénierie des connaissances et du Web sémantique. Les méthodologies existantes distinguent en général les phases de *conceptualization* et de *formalisation* dans un langage formel. On s'intéresse ici à la phase de formalisation qui pose un certain nombre de difficultés, en particulier pour les débutants mais pas seulement. Certaines difficultés sont liées à la manipulation d'un langage formel tel que OWL, et des outils tels que Protégé visent à réduire ce type de difficultés. D'autres difficultés sont liées aux différences qui peuvent survenir entre l'intension du concepteur et ce qu'exprime vraiment l'ontologie (Rector *et al.*, 2004; Corman, 2013). Par exemple, "ne mange que des légumes" n'implique pas "mange des légumes". Ou encore, savoir que "X est une femme" ne permet pas de déduire que "X n'est pas un homme" si on n'a pas explicitement dit que "hommes et femmes forment des classes séparées". Les contraintes négatives, telles que les séparations de classes ou les inégalités entre individus, sont souvent omises tellement elles semblent aller de soi. Leur omission est difficile à détecter car elle ne se manifeste pas par des inférences erronées, mais par des absences d'inférence.

Les éditeurs d'ontologie tels que Protégé (Noy *et al.*, 2001) favorisent l'expression de contraintes positives, c'est-à-dire d'axiomes permettant l'inférence de faits positifs : ex., hiérarchie de classes, domaines et co-domaines des propriétés. L'utilisateur a donc avant tout une vision syntaxique de son ontologie et se trouve très peu en contact avec sa sémantique, c'est-à-dire ce qu'elle rend possible ou non comme situation. Il est possible de faire appel à un raisonneur pour tester la cohérence de l'ontologie ou la satisfiabilité d'une classe. Un certain nombre d'outils existent aussi pour détecter des erreurs courantes et compléter les ontologies de façon systématique (Meilicke *et al.*, 2008; Poveda-Villalón *et al.*, 2012; Corman, 2013). Cependant,

ces approches ne sont pas constructives, mais correctives. De plus, elles sont généralement limitées aux axiomes de séparation, la forme la plus simple de contrainte négative. Dans un papier précédent (Ferré & Rudolph, 2012), nous avons montré des erreurs et omissions importantes dans l'ontologie des pizzas [1], laquelle sert pourtant de modèle et de support pédagogique depuis longtemps. Par exemple, les classes `Food` et `Country` ne sont pas séparées, et il s'avère qu'une pizza végétarienne peut bien contenir de la viande ou du poisson comme ingrédient.

Nous proposons une nouvelle approche de la formalisation d'ontologies qui est centrée sur la sémantique plutôt que sur la syntaxe. Plutôt que de voir une ontologie comme un ensemble d'axiomes, nous proposons de la voir comme un ensemble de modèles, c'est-à-dire comme l'ensemble des interprétations autorisées par l'ontologie. Et plutôt que de voir la construction d'une ontologie comme l'ajout d'axiomes, nous proposons de la voir comme l'élimination de modèles. Chaque élimination d'un sous-ensemble de modèles produit un axiome et on obtient bien au final un ensemble d'axiomes, mais celui-ci n'est que le résultat du processus de conception, pas le moyen. Le principal avantage de cette approche est de permettre au concepteur de travailler au niveau des instances – les mondes possibles – comme pour le peuplement d'une ontologie (connaissances particulières), mais en définissant néanmoins le niveau terminologique de l'ontologie (connaissances générales).

La structure de l'article suit le cycle "exploration-élimination" de notre approche. La section 2 présente tout d'abord l'exploration des mondes possibles d'une ontologie. La section 3 explique ensuite la conception d'ontologie par l'élimination de mondes possibles trouvés lors de l'exploration. Nous illustrons ces deux phases avec le prototype PEW [2] et l'ontologie des pizzas, et rapportons de premières expériences en Section 4. Cet article s'appuie sur les résultats théoriques d'un travail précédent (Ferré & Rudolph, 2012) et en étend l'application à la conception d'ontologies.

2 Exploration de mondes possibles

Nous commençons par décrire ici une méthode d'exploration sûre et complète des mondes possibles d'une ontologie OWL. Aucune hypothèse n'est faite sur cette ontologie. Elle peut contenir des instances ou non. Elle peut se limiter à des taxonomies ou bien contenir des axiomes complexes. Pour des raison de concision, nous adoptons la notation des logiques de descriptions (Baader *et al.*, 2003) pour les axiomes et expressions de classe. PEW offre le choix entre cette notation et la notation de Manchester.

À tout moment, une vue partielle sur les mondes possibles est présentée au concepteur d'ontologie. Cette vue comprend trois parties interdépendantes (voir la figure 1 montrant une capture d'écran du prototype PEW) : une expression de classe satisfiable avec focus, une liste d'instances connues et une liste d'incréments possibles (*adjuncts*). L'expression de classe combine des classes, des propriétés et des individus et décrit une situation rendue possible par l'ontologie. Dans la figure 1, l'expression de classe décrit la situation où une pizza n'a pas de *topping*. Le focus désigne une sous-expression de cette expression de classe (surlignée en vert dans PEW) et sert à "mettre le focus" sur une entité de la situation décrite. Dans l'exemple, le focus est mis sur la pizza. Les instances connues sont tous les individus de l'ontologie qui sont

1. `http://protege.stanford.edu/ontologies/pizza/pizza.owl`
2. `http://www.irisa.fr/LIS/softwares/pew`

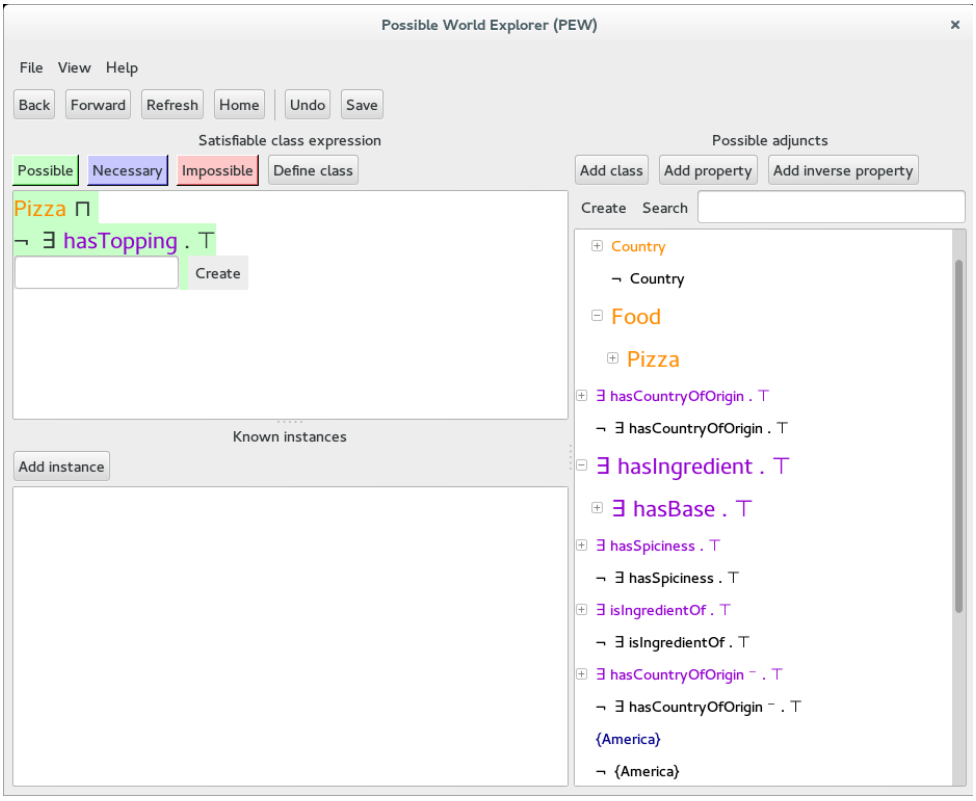


FIGURE 1 – Capture d'écran de PEW explorant les pizzas sans *topping*.

nécessairement des instances de l'expression de classe. Dans l'exemple de la figure 1, il n'y en a pas. Enfin, les incréments possibles sont toutes les classes simples qui sont "compatibles" avec le focus dans l'expression de classe. Une classe simple est "compatible" si son insertion au focus préserve la satisfiabilité de l'expression de classe. Une classe simple a une forme parmi : A, $\exists r.\top$, $\exists r^{-}.\top$, $\{o\}$, et leurs compléments. Si une classe simple est possible mais pas son complément, elle est dite *nécessaire*. Dans l'exemple de la figure 1, on voit qu'une pizza sans *topping* a nécessairement une base ($\exists hasBase.\top$), et que, de façon surprenante, elle peut être un pays ($Country$).

Un grand nombre de situations peuvent être décrites par une expression de classe. Le langage couvert comprend les classes atomiques (A), les restrictions existentielles qualifiées ($\exists r.C$), les propriétés inverses (r^{-}), la classe *top* ($\top$), les intersections ($C \sqcap D$), les unions ($C \sqcup D$), les compléments ($\neg C$) et les classes nominales ($\{o\}$). Les restrictions universelles sont indirectement couvertes par combinaison de restrictions existentielles et de compléments ($\forall r.C \equiv \neg\exists r.\neg C$). Les classes atomiques et nominales, ainsi que les restrictions existentielles, sont insérées dans l'expression de classe par sélection d'incréments. Les sélections successives forment des conjonctions. Les restrictions existentielles $\exists r.\top$ peuvent être qualifiées en mettant le focus sur le *top* $\top$, puis en sélectionnant des incréments. Les unions et les compléments sont insérés au focus via le menu contextuel. Si le focus est une sous-expression de classe C, l'insertion d'une union la remplace par l'expression $C \sqcup \top$, où $\top$ initialise une situation alternative à C. Celle-ci peut alors être affinée par insertion d'incréments. Si le focus est une sous-expression C,

axiome	ensemble équivalent d'axiomes de la forme $C \sqsubseteq \bot$
$C \sqsubseteq D$	$C \sqcap \neg D \sqsubseteq \bot$
$C(a)$	$\{a\} \sqcap \neg C \sqsubseteq \bot$
$r(a,b)$	$\{a\} \sqcap \neg\exists r.\{b\} \sqsubseteq \bot$, $\{b\} \sqcap \neg\exists r^{-}.\{a\} \sqsubseteq \bot$
$a = b$	$\{a\} \sqcap \neg\{b\} \sqsubseteq \bot$, $\{b\} \sqcap \neg\{a\} \sqsubseteq \bot$
$a \neq b$	$\{a\} \sqcap \{b\} \sqsubseteq \bot$

TABLE 1 – Équivalences pour les principaux axiomes de logiques de description.

l'insertion d'un complément la remplace par l'expression $\neg C$.

Nous avons démontré dans un papier précédent (Ferré & Rudolph, 2012) la *sûreté* et la *complétude* du processus de construction d'expressions de classe, sauf pour l'insertion de compléments que nous discutons ci-après. La sûreté implique que toute expression de classe pouvant être construite est satisfiable, et la complétude implique que toute expression de classe satisfiable peut être construite en un nombre fini d'étapes. Pour être sûre, l'insertion d'un complément dans une expression de classe $C \sqcap \underline{D}$ (focus sur D) n'est permise que si l'expression $C \sqcap \neg D$ est satisfiable. Dans le cas contraire, D est nécessairement vrai quand C est vrai (C est subsumé par D), et cette information est reflétée dans PEW par un focus de couleur bleu (signifiant "nécessaire") plutôt que vert (signifiant "possible").

La construction interactive d'expressions de classes satisfiables permet une exploration des mondes possibles d'une ontologie. Le simple fait de pouvoir ou non construire une expression de classe renseigne déjà sur les situations rendues possibles ou non par l'ontologie. Dans l'exemple de la figure 1, on découvre par exemple que l'ontologie des pizzas autorise une pizza à ne pas avoir de *topping*. Ensuite, la liste des instances connues indique directement au concepteur si la classe courante est habitée. Enfin, la liste des incréments indique pour toutes les classes simples si elles sont nécessaires, possibles ou impossibles. Dans l'exemple, on découvre qu'une pizza doit avoir une base, ce qui est correct, mais peut également être un pays, ce qui est absurde. Ces incréments correspondent à autant de réponses à des questions que le concepteur n'a même pas eu à poser. Ils offrent un *feedback* précieux sur la sémantique de l'ontologie explorée.

3 Conception par élimination de mondes possibles

Dans l'approche proposée dans cet article, la conception d'une ontologie fonctionne essentiellement par l'élimination de mondes possibles, mais également par l'augmentation de la signature de l'ontologie, c'est-à-dire par la création de nouvelles entités OWL : classes, propriétés et individus. La création de nouvelles entités OWL peut se faire soit à la volée, lors de la construction d'une expression de classe (bouton `Create` dans la figure 1), soit en les ajoutant à la liste des incréments possibles (voir boutons `Add class`, `Add property` et `Add inverse property`). Ces créations ne produisent aucun axiome, mais permettent le démarrage à froid dans la conception d'ontologie. Par exemple, pour l'ontologie de pizzas, on peut commencer par créer les classes *Pizza* et *Country*, et la propriété *ingredient*. À ce stade, tout est possible : une pizza peut être un pays, un pays peut être un ingrédient de pizza, etc.

L'élimination de mondes possibles s'effectue en déclarant que la situation décrite par l'expression de classe courante C est impossible (voir bouton `Impossible` dans la figure 1). Vi-

suellement, dans PEW, la couleur du focus passe alors du vert au rouge. Rendre une classe impossible revient à la rendre vide de tout individu, et donc à en faire une sous-classe de la classe *bottom* $\bot$. Cette élimination produit donc l'axiome $C \sqsubseteq \bot$. Dans l'ontologie de pizza, on peut par exemple rendre impossible les expressions de classes suivantes : $Pizza \sqcap Country$ ("les pizzas ne sont pas des pays et réciproquement"), $Country \sqcap (\exists ingredient.\top \sqcup \exists ingredient^{-}.\top)$ ("les pays ne sont pas des ingrédients et n'ont pas d'ingrédients"), $Pizza \sqcap \neg\exists ingredient.\top$ ("les pizzas ont nécessairement au moins un ingrédient"). Grâce aux compléments de classes et aux classes nominales, la plupart des axiomes des logiques de description, et donc de OWL, peuvent être reformulés sous cette forme par un ou deux axiomes. La table 1 donne ces équivalences. Les axiomes qui ne peuvent pas être reformulés ainsi sont les axiomes portant sur les propriétés : subsomption, réflexivité, transitivité, etc.

Le prototype PEW offre un certain nombre de raccourcis pour exprimer des impossibilités. Les motivations de ces raccourcis sont : (a) une plus grande efficacité dans l'interaction et la production d'axiomes, et (b) l'évitement de l'emploi de compléments pour les axiomes positifs simples de type $A \sqsubseteq B$ et $C(a)$. Chaque raccourci est associé à une des trois zones de l'interface. Au-dessus de l'expression de classe, le bouton `Necessary` permet de rendre la sous-expression du focus nécessaire par rapport au reste de l'expression. La couleur du focus passe alors du vert au bleu. Ce raccourci produit des axiomes de la forme $C \sqcap \neg D \sqsubseteq \bot$, équivalente à $C \sqsubseteq D$, où D est la sous-expression au focus. Par exemple, partant de l'expression $AmericanPizza \sqcap \underline{\exists origin.\{America\}}$, on produit l'axiome $AmericanPizza \sqsubseteq \exists origin.\{America\}$ ("Toute pizza American est originaire d'Amérique"). En partant d'un autre focus, $AmericanPizza \sqcap \exists origin.\underline{\{America\}}$, on produit l'axiome $\exists origin^{-}.AmericanPizza \sqsubseteq \{America\}$ ("La seule origine des pizza American est l'Amérique"). Le raccourci `Define class` permet de créer une nouvelle classe A comme équivalente à l'expression de classe courante C. Il produit donc l'axiome $A \equiv C$, équivalent aux deux axiomes $A \sqsubseteq C$ et $C \sqsubseteq A$. On peut ainsi définir la classe $VegetarianPizza$ comme équivalente à l'expression $Pizza \sqcap \neg\exists ingredient.(Meat \sqcup Fish)$.

Au-dessus de la liste d'instance, le bouton `Add instance` permet de créer un nouvel individu a qui soit une instance de l'expression de classe courante C. Cela produit l'axiome $\{a\} \sqcap \neg C \sqsubseteq \bot$, équivalent à $C(a)$. Dans l'arbre d'incréments possible, le menu contextuel offre plusieurs raccourcis s'appliquant à une sélection de un ou plusieurs incréments. Tout d'abord, la commande `Add subclass` appliquée à une classe A permet de créer une nouvelle classe B et d'en faire une sous-classe de A en produisant l'axiome $B \sqsubseteq A$. Ce type d'axiome est en effet très commun. La commande `All impossible` (resp. `All necessary`) permet de rendre chaque incrément sélectionné impossible (resp. nécessaire) par rapport à l'expression de classe courante. Elle a le même effet que d'insérer successivement chaque incrément puis de presser le bouton `Impossible` (resp. `Necessary`). Ces commandes permettent de définir très rapidement le domaine et le co-domaine des propriétés, ainsi que les propriétés pouvant ou devant s'appliquer aux classes : par exemple, "les pays ne sont pas des ingrédients et n'ont pas d'ingrédients", et "les pizzas doivent avoir un ingrédient". Enfin, la commande `All disjoint` permet d'établir la séparation deux-à-deux d'un ensemble de classes simples. Elle permet donc aussi bien de générer les axiomes de la forme $A \sqcap B \sqsubseteq \bot$ (ex., $Pizza$ et $Country$) que de la forme $a \neq b$ (ex., $America$ et $France$). Un ensemble de n classes simples génère $n(n-1)/2$ axiomes et cette commande offre donc un raccourci précieux. À cause de leur combinatoire, ces axiomes sont souvent omis alors qu'ils sont cruciaux pour l'inférence.

4 Premières expériences

Nous avons réalisé de premières expériences qui se sont révélées encourageantes. Nous avons recréé le schéma de l'ontologie de pizzas dans l'ordre suivant : création de la hiérarchie de classes et séparation des classes sœurs entre elles ; pour chaque classe, création des instances connues (par ex. pays) et séparation entre ces instances ; création des propriétés et définition de leur domaines et co-domaines ; nécessité et impossibilité de certaines propriétés pour chaque classe. À ce stade, nous avons créé en quelques clics toutes les contraintes négatives pertinentes, contrairement à l'ontologie originale. Par contre, les axiomes de propriétés ne sont pas couverts (par ex. transitivité de $hasTopping$). Puis nous avons décrit différentes sortes de pizzas, telle que l'*American*. Il est possible de spécifier la liste de *toppings* d'une pizza sans utiliser de restriction universelle, ni de complément, mais l'utilisation d'une union est nécessaire pour exclure tout autre *topping*. En effet, un axiome de la forme $X \sqsubseteq \forall r.(A \sqcup B)$ peut être obtenu en rendant nécessaire le focus dans l'expression de class $X \sqcap \exists r.(\underline{A \sqcup B})$.

5 Conclusion et perspectives

La conception d'ontologie par élimination de mondes possibles présente l'avantage de montrer la réalité sémantique d'une ontologie et ainsi d'assister le concepteur dans la définition la plus complète possible de son ontologie. Cependant, l'affichage de tous les possibles peut être déroutant pour le concepteur car ils sont souvent bien plus nombreux que ce qui est attendu intuitivement et car il peut être fastidieux de les éliminer tous. Il sera nécessaire de faire des évaluations utilisateurs pour valider notre approche. Notons qu'une implémentation sous forme de plugin Protégé permettrait d'entrelacer librement cette approche avec l'approche classique.

Références

F. BAADER, D. CALVANESE, D. L. MCGUINNESS, D. NARDI & P. F. PATEL-SCHNEIDER, Eds. (2003). *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press.

CORMAN J. (2013). Explorer les théorèmes d'une TBox. In *Journées francophones d'Ingénierie des Connaissances*.

FERRÉ S. & RUDOLPH S. (2012). Advocatus diaboli - exploratory enrichment of ontologies with negative constraints. In A. TEN TEIJE ET AL., Ed., *Int. Conf. Knowledge Engineering and Knowledge Management (EKAW)*, LNAI 7603, p. 42–56 : Springer.

MEILICKE C., VÖLKER J. & STUCKENSCHMIDT H. (2008). Learning disjointness for debugging mappings between lightweight ontologies. In A. GANGEMI & J. EUZENAT, Eds., *EKAW*, volume 5268 of *LNCS*, p. 93–108 : Springer.

NOY N., SINTEK M., DECKER S., CRUBEZY M., FERGERSON R. & MUSEN M. (2001). Creating semantic web contents with Protege-2000. *Intelligent Systems, IEEE*, **16**(2), 60–71.

POVEDA-VILLALÓN M., SUÁREZ-FIGUEROA M. & GÓMEZ-PÉREZ A. (2012). Validating ontologies with OOPS ! In *Knowledge Engineering and Knowledge Management (EKAW)*, p. 267–281. Springer.

RECTOR A., DRUMMOND N., HORRIDGE M., ROGERS J., KNUBLAUCH H., STEVENS R., WANG H. & WROE C. (2004). OWL pizzas : Practical experience of teaching OWL-DL : Common errors & common patterns. In *Engineering Knowledge in the Age of the Semantic Web*, p. 63–81. Springer.

HuTO: une Ontologie Temporelle Narrative pour les Applications du Web Sémantique

Papa Fary Diallo[1,2,3] , Olivier Corby[1,2] , Isabelle Mirbel[2]

Moussa Lo[3] and Seydina M. Ndiaye[3]

[1] INRIA Sophia Antipolis, FRANCE,
{papa-fary.diallo, olivier.corby}@inria.fr
[2] Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, FRANCE,
isabelle.mirbel@unice.fr
[3] Université Gaston Berger - UFR SAT - LANI, SENEGAL,
{moussa.lo, seydina.ndiaye}@ugb.edu.sn

Abstract : Un défi majeur en informatique est la modélisation et le raisonnement sur les données temporelles. Ce travail est devenu encore plus important avec l'émergence du Web sémantique où de grandes quantités données hétérogènes sont manipulées. Ces données comportent souvent des informations temporelles informelles, semi-formelles ou formelles qui doivent être interprétées par les agents logiciels. Dans cet article nous présentons notre ontologie, Humain Time Ontologie (HuTO), une ontologie en RDFS pour annoter des ressources en RDF et représenter les expressions narratives temporelles. Une des contributions majeures de HuTO est la modélisation des intervalles non-convexes c'est-à-dire les intervalles répétitifs comme `tous les mercredi` mais également la possibilité d'écrire des requêtes sur ce type d'intervalle. HuTO intègre aussi des règles de normalisation et de raisonnement pour expliciter certaines informations temporelles. HuTO propose aussi une approche qui permet de garder distincte la dimension temporelle et les annotations du domaine métier. Cela facilite la recherche d'informations qu'elles soient temporelles ou non.

Mots-clés :Ontologies Temporelles, Web Sémantique, RDFS, SPARQL, Règles.

1 Introduction

Les phénomènes temporels ont de nombreuses facettes qui sont étudiées par différentes communautés. Ainsi, la dimension temporelle des données est aussi étudiée dans le domaine de l'informatique où il y a un besoin croissant de modéliser des systèmes calendaires, des événements répétitifs et des faits qui sont vrais pour un certain temps et faux par ailleurs. C'est le cas des Systèmes d'Information qui doivent faire face au problème des données obsolètes. En Intelligence Artificielle, des modèles abstraits ont été proposés pour pouvoir raisonner sur des concepts temporels. Dans ce domaine, Allen (Allen, 1984, 1981) a présenté un modèle de calcul entre les intervalles de temps qui a influencé les travaux sur la modélisation du temps. Ces travaux de Allen ont été étendus aux intervalles non-convexes (intervalles répétitifs) par Ladkin (Ladkin, 1987). Dans le Traitement Automatique des Langages Naturels (TALN) les modèles développés cherchent à extraire les expressions temporelles mais aussi leur sémantique en langue naturelle. Ainsi, un défi important dans le domaine de l'informatique est la représentation et le raisonnement sur des informations temporelles. L'intérêt de ce travail est de plus en plus important maintenant avec l'émergence du Web sémantique où de gros volumes de données hétérogènes sont manipulés.

Dans le domaine du Web sémantique sont présentes à la fois des notions temporelles informelles, semi-formelles et formelles qui doivent être comprises par les agents logiciels. Nous distinguons deux axes de travail: la modélisation d'expression temporelle et l'annotation temporelle des données. La modélisation d'expression temporelle permet de modéliser une date, un intervalle, des notions temporelles répétitives, relatives ou absolues, etc. L'annotation temporelle des données permet la représentation de notions temporelles de façon à annoter des connaissances (exprimées sous forme de triplet en RDF) et cela en conservant l'évolution des données (changement de valeur) dans le temps. Pour cela, le Web sémantique repose sur des ontologies qui sont une *spécification explicite et formelle d'une conceptualisation partagée* (Studer et al., 1998). Ainsi, l'objectif principal de ce travail est de proposer une ontologie pour représenter des notions temporelles et annoter temporellement des données.

Dans (Diallo et al., 2011, 2014) nous avons développé une ontologie socioculturelle et une plateforme de partage et de co-construction de connaissances sur les communautés sénégalaises. La manipulation de ces données socioculturelles fait intervenir beaucoup de notions temporelles. Ainsi dans cet article, nous présentons notre ontologie temporelle, Human Time Ontology (HuTO), et nous illustrons son utilisation sur des données extraites de cette plateforme.

Ce document continue par un état de l'art dans lequel nous présenterons les travaux sur la modélisation des notions temporelles et l'annotation temporelle des données dans le Web sémantique. Ensuite la troisième partie détaillera notre proposition d'ontologie: HuTO. En premier lieu nous présenterons les concepts de l'ontologie qui servent à modéliser la représentation d'énoncé de temps complexe. En deuxième lieu nous présenterons notre approche pour l'annotation temporelle des données. Dans la partie quatre, nous présenterons les raisonnements et les règles proposés dans HuTO. Dans la cinquième partie, nous montrerons des exemples de requêtes en SPARQL sur des connaissances temporellement annotées à l'aide de HuTO. Nous terminerons par une conclusion et des perspectives pour ce travail.

2 État de l'art

Plusieurs spécifications ont été proposées pour modéliser des expressions (énoncés) temporelles parmi lesquelles nous pouvons citer TimeML (Sauri, 2006), OWL-Time (Pan et Hobbs, 2005) (Pan, 2007) et CNTRO (Tao et al., 2010, 2011). TimeML est un langage d'annotation pour les informations temporelles dans des documents textuels utilisé dans le TALN. Il est basé sur un système de balises XML standard. L'inconvénient principal de ce langage est qu'il annote les événements et les expressions temporelles dans des segments textuels isolés, ce qui rend la recherche d'information plus difficile. TimeML ne permet pas non plus d'exprimer des expressions comme `every 3rd Monday`. OWL-Time est une ontologie temporelle qui permet de fournir une description temporelle de documents du Web et de Web services. CNTRO est une ontologie en OWL pour la modélisation des informations temporelles dans les récits et rapports cliniques. Ces deux ontologies permettent entre autres la modélisation d'intervalles non-convexes et la représentation des relations comme celles définies par Allen (Allen, 1984, 1981). Cependant, Pour modéliser des intervalles non-convexes CNTRO modélise la périodicité dans des chaînes de caractères d'où la perte de la sémantique. OWL-Time ne permet pas de modéliser les expressions humaines du temps comme le temps déictique.

Dans les langages du Web sémantique comme RDF, un énoncé (statement) est une relation binaire qui est utilisée pour relier deux individus (instances) ou un individu et une valeur. Or, pour introduire une dimension temporelle, il devient nécessaire de manipuler des relations ternaires. La modélisation des relations ternaires est un cas particulier d'une problématique plus générale qui est la modélisation et l'interrogation des relations n-aires dans le Web sémantique. Ainsi dans la littérature il existe des approches générales qui essaient de répondre à cette problématique comme l'approche des N-ary relations (W3C Working Group, 2006) qui

propose l'introduction d'un blank node entre l'objet et le sujet du triplet. Ainsi, le blank node peut par exemple être temporellement annoté. Il existe aussi l'approche des graphes nommés qui permettent de contextualiser un ensemble de triplets en les regroupant dans un même graphe (URI) qui peut par exemple être temporellement annoté. Une autre approche est la Réification en RDF qui permet grâce à `rdf:Statement` d'ajouter d'autres informations sur un triplet comme des informations temporelles.

Il existe aussi des approches spécifiques à la modélisation temporelle comme 4D-Fluents (Welty et Fikes, 2006), une ontologie en OWL qui propose une approche basée sur les occurrents et les perdurants pour modéliser l'évolution temporelle des données. Dans cette approche les auteurs considèrent que tout objet a une partie temporelle et que ce sont ces parties temporelles qui sont en interaction. Il y a aussi l'approche de SOWL (Batsakis et Petrakis, 2011) qui étend le 4D-Fluents en y ajoutant les relations d'Allen. Il existe aussi l'approche Temporal RDF (Gutierrez et al., 2005) qui étend la Réification en RDF en ajoutant une dimension temporelle sur les données. Ainsi le graphe peut être accédé selon deux vues, selon qu'on s'intéresse à la temporalité ou aux connaissances du domaine modélisé. Dans (Rula et al., 2014), les auteurs proposent une approche générique pour extraire des informations temporelles du Web et de leur durée de validité. (Scheuermann et al., 2013) propose une approche empirique centrée sur les perspectives des utilisateurs ce qui a permis de définir différents modèles temporels.

Excepté Temporal RDF, le principal inconvénient pour les autres approches est la perte des relations directes entre les ressources pour l'ajout de l'information temporelle. Ainsi pour ajouter des informations temporelles les triplets sont cassés on ajoutant des ressources intermédiares (N-ary relations, la réification en RDF) ou déplacés vers des `timeSlices` (4D-Fluents).

3 HuTO

HuTO[1] est une ontologie formalisée en RDFS permettant l'annotation temporelle de ressources en RDF à l'aide d'expressions temporelles du langage courant. Cette ontologie permet également de définir des ancrages temporels liés au contexte et de capturer les changements temporels associés aux ressources annotées. Elle rend possible l'interrogation temporelle de la base de connaissances à l'aide de requêtes SPARQL. Plus précisément, HuTO permet de:

- Modéliser des expressions temporelles:
  1. **Explicites**: elles sont immédiatement ancrées; par exemple: 30 Août 2014, été 2014;
  2. **Déictiques**: elles forment une relation spécifique avec le temps du discours; par exemple: aujourd'hui, demain;
  3. **De durée**: elles indiquent un intervalle de temps; par exemple, 2 heures, 20 minutes;
  4. **Cycliques**: elles permettent de modéliser des dates répétitives; par exemple: chaque lundi, tous les deux mois;
  5. **Mixtes**: elles combinent les expressions pré-cités; par exemple deux mois l'année dernière.
- Normaliser les expressions temporelles afin de pouvoir leur appliquer des raisonnements et de pouvoir les interroger.

3.1 Date, Temps calendaire et granularité

Dans HuTO, les concepts principaux pour la datation sont `Datation` et `TemporalUnit` (Fig. 1). `Datation` est un concept abstrait (qui n'a pas d'instances directes) dont dérivent les

[1] http://ns.inria.fr/huto/

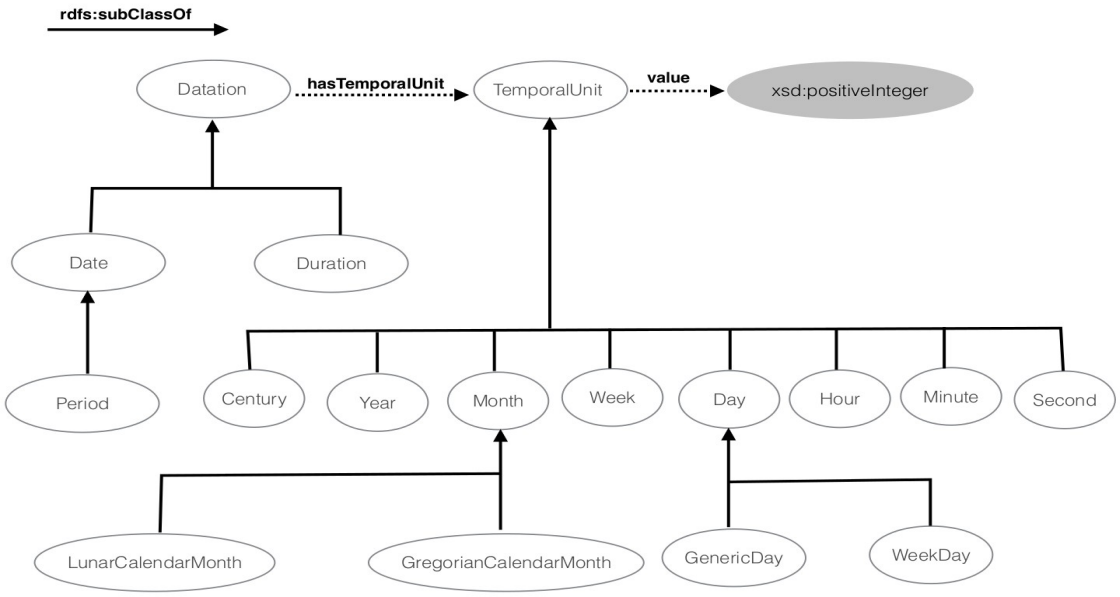


FIGURE 1 – Modélisation des types `Datation` et `TemporalUnit`.

concepts `Date` et `Duration`. Le concept `Date` permet de modéliser des dates comme celles du type `xsd:dateTime` excepté la partie fuseau horaire (exemples 1a et 1b). Le concept `Duration` permet de définir des durées comme celles du type `xsd:duration` (exemple 1d). Les granularités définies dans `TemporalUnit` vont de `Century` à `Second`.

Notons aussi que le concept `WeekDay` rassemble les jours de la semaine comme sous-concepts. Le concept `GenericDay` rassemble des sous-concepts comme `Today`, `Yesterday`, etc. Notons également la relation `hasContext` (exemple 1c), qui est utilisée pour contextualiser le concept `GenericDay`.

a. Date(Mardi 17 Février 2015 à 10H)

```
[a :Date;
   :hasHour  [a :Hour;
               :hour   10];
   :hasDay [a :Tuesday;
              :day 17];
   :hasMonth [a :February];
   :hasYear [a :Year;
              :year  2015]].
```

b. Date(15 04) “15 Avril”

```
[a :Date;
   :hasDay [a :Day;
              :day 15];
   :hasMonth
    [a :Month;
       :month 4]].
```

c. Date(Aujourd’hui)- Vendredi 29 Août 2014

```
[a :Date;
   :hasDay [a :Today;
            :hasContext
             [a :Date
                :hasDay
                 [a :Friday;
                    :day 29];
                :hasMonth
                 [a :August];
                :hasYear
                 [a :Year;
                    :year
                    2014]]]].
```

d. 2 heures 30 minutes

```
[a :Duration;
   :hasHour [a :Hour
               :value 2];
   :hasMinute [a :minute;
                 :value 30].
```

EXEMPLE 1 – *Modélisation de notions de dates simples.*

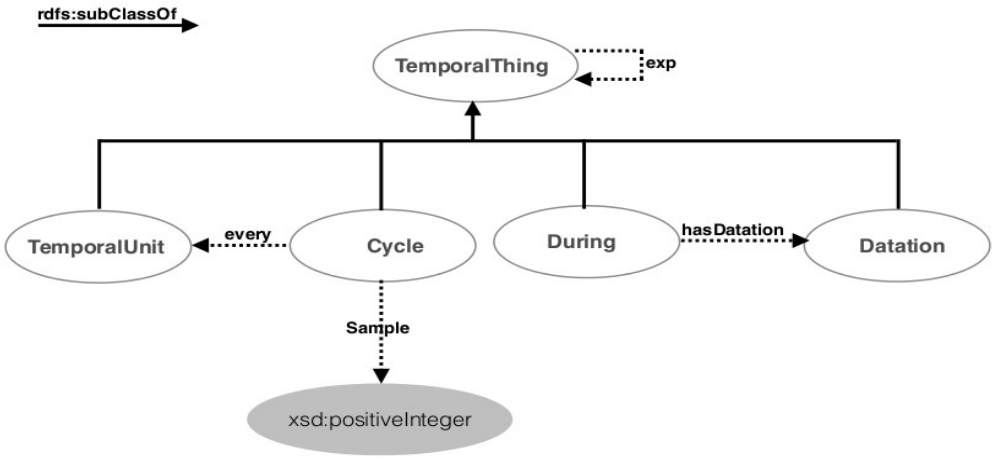


FIGURE 2 – Les concepts temporels de HuTO.

Notons que la propriété `hasTemporalUnit` (Fig. 1) est spécialisée par les propriétés `hasCentury`, `hasYear`, etc et la propriété `value` par `century`, `year` etc.

3.2 Instant, Intervalle et Durée

Un élément temporel peut être considéré comme un instant, un intervalle ou une durée. Nous avons fait le choix de représenter tous les éléments temporels comme des intervalles modélisés à l'aide du concept `During` (Fig. 2). De ce fait, si pour un intervalle, la date de fin ou la durée n'est pas spécifiée alors l'intervalle considéré est celui de l'unité de la date. Par exemple, la date `Vendredi 15 Août 2014` est considérée comme un intervalle de `24H`. Pour préciser le début et/ou la fin d'un intervalle, il faut utiliser le concept `During` avec les propriétés `hasBegin` et/ou `hasEnd`. Pour modéliser une durée, le concept `During` est aussi utilisé avec les propriétés `hasBegin` pour spécifier le début et `hasDuration` pour la durée.

Le concept `Cycle` sert à modéliser les intervalles non-convexes (répétitifs). Un intervalle non-convexe est caractérisé par deux entités: la fréquence de répétition et l'occurrence de l'intervalle convexe à répéter. Ainsi, le concept `Cycle` est relié à sa fréquence par la relation `every`. Cette fréquence est un sous-concept de `TemporalUnit` qui représente l'unité de temps à laquelle le cycle se répété. L'intervalle convexe est relié au concept `Cycle` par la relation `exp`. La propriété `sample` permet de modéliser pour les `Cycle` des échantillons de date comme `tous les 8 heures` (cf. Exemple 2b).

Notons que les propriétés `hasDate`, `hasDuration`, hasBegin et `hasEnd` sont des spécialisations de la propriété `hasDatation`.

Avec la modélisation proposée, nous faisons la distinction entre les intervalles infinis et les intervalles fermés. De ce fait, si les deux propriétés `hasBegin` et `hasEnd` sont spécifiées ou la propriété `hasDate` est utilisée, nous avons un intervalle fermé. Si l'une des propriétés `hasBegin` ou `hasEnd` est omise, nous avons un intervalle infini.

a. The first Sunday of every April.

```
[a :Cycle;
   :every [a :Year];
   :exp  [a :During;
            :hasDate
             [a :Date;
                :hasDay
                 [a :Sunday;
                    :week 1];
                :hasMonth
                 [a :April]]]].
```

b. Every 8H for 10 days starting from today

```
[a :During;

   :hasBegin    [a :Today];
   :hasDuration [a :Duration;
                   :hasDay
                    [a :Day;
                       :value 10]];
    :exp [a :Cycle;
            :every [a :Hour];
            :sample 8]].
```

EXEMPLE 2 – *Modélisation d'intervalles non-convexes*[2].

3.3 HuTO et Intervalles d'Allen

Allen (Allen, 1984, 1981) définit une algèbre de 13 relations pour permettre de positionner des intervalles convexes les uns par rapport aux autres et d'en déduire des relations. Dans le sens d'Allen, un intervalle convexe est fermé et ordonné. Ainsi, il définit six paires de relations inverses: `before/after`, `during/contains`, `meet/metBy`, `start/startedBy`, `finishes/finishedBy` et `overlaps/overlappedBy`. Ainsi, à chaque fois que l'une des relations est vraie son inverse l'est aussi. La treizième relation, `equal`, est son propre inverse.

Dans HuTO, nous n'avons représenté pour l'instant que les relations `before` et `after`. Elles nous permettent de modéliser la représentation du temps en datation relative c'est-à-dire que la dimension temporelle d'une ressource est exprimée par rapport à la dimension temporelle d'une autre ressource (Exemple 3b). Notons que cette utilisation nous permet d'avoir deux informations implicites (cf. section 4.2): la date de la ressource référencée et les deux relations d'Allen entre les ressources. HuTO permet de spécifier les relations `before` et `after` entre intervalles, entre ressources et entre une ressource et un intervalle.

4 Annotation Temporelle des Données

L'annotation temporelle des données consiste à lier une donnée (une ressource, un triplet ou un graphe nommé) à sa dimension temporelle. Grâce à notre modélisation deux dimensions peuvent co-exister: une temporelle et une non temporelle. La dimension temporelle est spécifiée à l'aide des concepts de HuTO et la dimension non temporelle, celle du domaine de connaissance, est spécifiée au travers des triplets décrivant des aspects autres que ceux temporels.

L'annotation temporelle peut être associée à une ressource, un triplet ou un graphe nommé. Ces derniers sont temporellement annotés à l'aide de la propriété `exp` qui relie un intervalle convexe (`During`) ou non-convexe (`Cycle`) à un `TemporalThing`, elle même associée aux connaissances à annoter comme suit:

- S'il s'agit d'une ressource, le `TemporalThing` a pour valeur (`rdf:value`) la ressource concernée (cf. l'exemple 3b pour un intervalle convexe et l'exemple 3a pour un intervalle non-convexe);
- S'il s'agit d'un triplet, nous utilisons une réification RDF sur le triplet pour le `TemporalThing` (cf. exemple 3d);

[2]Dans l'exemple 2b, le contexte du Today a été omis volontairement pour ne pas surcharger l'exemple

- S'il s'agit d'un graphe nommé, nous utilisons `Graph`, sous concept de `TemporalThing`, dont la propriété `uri` pointe sur l'URI du graphe nommé (cf. exemple 3c.).

a. Le premier Samedi de chaque mois de Décembre, le Fanal de Ndar est organisé

```
[a :Cycle;
   :every [a :Year];
   :exp [a :During;
           :hasDate
            [a :Date;
           :hasDay
             [a :Saturday;
                :week 1];
           :hasMonth
            [a:December]];
        :exp
         [a :TemporalThing
            rdf:value
             <FanalOfNdar>]]].
```

b. la Bataille de Dekheulé a eu lieu après la Bataille de Mékhé.

```
[a :During;
   :after [a :Period;
             rdf:value
              <BattleOfMekhe>];
   :exp
    [a :TemporalThing;
       rdf:value
        <BattleOfDerkheule>]].
```

c. En 2011 la Commune de Dakar compte 1056009 d'habitants, c'est la plus peuplée et son maire est M. Sall.

```
[a :During;
   :hasDate [a :Date;
               :hasYear
                 [a :Year;
                    :year 2011]];
   :exp[a :Graph;
          :uri
           <http://example.org/g/>]].

 <http://example.org/g/> {
    <Dakar> <population> 1056009;
            <rang>       1;
            <mayor>      <Sall>}.
```

d. Senghor a été le Président du Sénégal de septembre 1960 à décembre 1980

```
[a :During;
   :hasBegin [a :Date;
                :hasMonth
                    [a :September];
                 :hasYear
                    [a :Year;
                       :year 1960]];
   :hasEnd [a :Date;
              :hasMonth
                 [a :December];
              :hasYear
                 [a :Year;
                    :year 1980]];
    :exp
        [rdf:subject   <Senghor>;
         rdf:predicate <presidentOf>;
         rdf:object    <Senegal>]].
```

EXMPLE 3 – *Annotation temporelle des données.*

Notons que HuTO permet aussi d'utiliser une ressource comme une référence temporelle grâce au concept `Period`. Ainsi une fois datée, une ressource peut être utilisée comme un marqueur temporel (exemple 3b).

L'utilisation de HuTO présente certains avantages comparée aux approches présentées dans la deuxième section. Dans la modélisation des expressions temporelles, HuTO permet de représenter des énoncés complexes comme dans l'exemple 2b. HuTO intègre aussi la modélisation des intervalles fermés et infinis et permet aussi d'utiliser une ressource comme référence temporelle. Ces aspects ne sont pas considérés par les autres approches présentées. HuTO permet également de modéliser le temps déictique ce que ne font pas les autres approches excepté CNTRO qui le modélise dans une chaîne de caractère.

Pour l'annotation temporelle des données, HuTO propose une représentation qui permet d'annoter une ressource, un triplet ou plusieurs triplets dans un graphe nommé. Ceci nous permet de séparer la partie temporelle des données de celles du domaine contrairement aux autres approches, excepté Temporal RDF, où la sémantique des triplets est perdue par l'introduction d'un blank node (n-ary relations, la réification RDF) ou par le déplacement des relations sur des `timeSlice` (4D-Fluents). La principale différence de HuTO avec Temporal RDF est que ce dernier nécessite une extension légère du vocabulaire de RDF (Hurtado et Vaisman, 2006).

5 Raisonnement Temporel et Règles

HuTO fournit un modèle conceptuel en RDFS pour modéliser des expressions temporelles et pour annoter des ressources en RDF. Cependant beaucoup de relations temporelles sont exprimées implicitement dans les occurrences d'événements. Les réponses à de nombreuses questions axées sur le temps ne sont pas nécessairement représentées explicitement mais peuvent être déduites. Pour cela, nous avons proposé un ensemble de règles permettant de normaliser la représentation des données temporelles mais également des règles d'inférences et d'implications.

5.1 Normalisation de la Représentation Temporelle

Puisque HuTO est une ontologie en RDFS, nous avons proposé des règles, exprimées sous forme de requêtes CONSTRUCT en SPARQL et ayant pour objectif de déduire et d'expliciter le maximum d'information temporelle afin de permettre le raisonnement sur les données.

Les informations temporelles peuvent être exprimées de différentes façons. Par exemple, une date peut être représentée soit en utilisant la représentation calendaire (Exemple 1a), soit à l'aide de chiffres (exemple 1b). Aussi, nous avons créé des règles pour normaliser ces deux types d'écritures. De ce fait, quelque soit le mode d'écriture utilisé, toutes les représentations possibles seront ajoutées dans le graphe des données. Nous avons également proposé deux règles pour déterminer les années bissextiles. Ces règles nous permettent, entre autre, de connaître le nombre de jours dans l'année ce qui est utile pour répondre à certaines requêtes. Nous avons aussi proposé une règle pour normaliser l'utilisation du concept `Period` en ajoutant la date correspondant à la période aux concepts utilisant `Period` comme date. Nous avons également normalisé les intervalles définis par leur durée en ajoutant explicitement la date de fin (`hasEnd`) de l'intervalle. L'exemple 4 montre une règle pour expliciter la date de fin d'un intervalle défini par sa durée. Notons qu'il existe sept règles pour normaliser les intervalles définis par leur durée (car la règle dépend du type de la durée qui peut être en siècle, en année, en mois, en semaine, en jour, en minute ou en seconde).

```
PREFIX dt: <http://ns.inria.fr/huto/>
CONSTRUCT  { ?x dt:hasEnd [  ?z ?t;
                             dt:hasYear [a dt:Year;
                                         dt:year ?o]}
WHERE {?x  dt:hasBegin ?y;
           dt:hasDuration/dt:hasYear/rdf:type dt:Year;
           dt:hasDuration/dt:hasYear/dt:value ?l
       ?y  dt:hasYear/dt:year ?e;
           ?z ?t
       FILTER(?z != dt:hasYear)
       BIND(?e + ?l - 1 as ?o)}
```

EXMPLE 4 – Règle de normalisation d'une durée exprimée en année.

Dans cette règle nous récupérons toutes les propriétés liées à la date de début (`?y`) excepté l'année qu'on incrément de `?l-1` où `?l` est la durée de la ressource.

Nous avons aussi proposé une requête de vérification de la consistance entre les concepts `Cycle` et `During`. En effet, la granularité de la fréquence du concept `Cycle` doit être supérieure à celle de la date de l'occurrence de l'intervalle convexe. De même, si un `During` englobe un `Cycle` alors la granularité de la date du `During` doit être supérieure à celle de la fréquence du concept `Cycle`. Par exemple, dans l'exemple 2b nous ne devons pas inter-changer les positions de `During` et Cycle puisque la granularité de `Today` est supérieure à celle de `Hour`.

Par manque d'espace, nous ne pouvons pas détailler toutes les règles de normalisations utilisées. Cependant il reste un travail de normalisation à faire par rapport aux intervalles non-convexes puisque dans ces intervalles certaines informations sur les dates sont omises. L'idée serait de proposer un moyen de normaliser ces intervalles pour ajouter plus d'information sur le graphe des données.

5.2 Implications et Inférences

Puisque RDFS ne permet pas de modéliser certaines inférences de base comme la transitivité ou la réflexivité, nous avons créé des règles d'inférence à cet effet. Ainsi, nous avons défini par exemple des règles d'inférence pour la transitivité des propriétés `before+/after`. De même si une relation (`after` ou `before`) est exprimée entre deux événements (respectivement intervalles), il est nécessaire de propager cette relation entre les intervalles (respectivement ressources) concernés. Pour cela, nous avons proposé des règles de propagation.

Pour vérifier l'ordre d'englobement des concepts `Cycle` et `During`, nous avons défini une propriété `included` qui permet de hiérarchiser la granularité des dates. Ainsi nous avons explicité dans HuTO sept relations entre dates (`included(Year,Century`, `included(Month,Year)`, ...). De fait nous avons défini deux règles de propagation: une règle pour la transitivité et une autre pour la transitivité par la subordination; c'est-à-dire si `included(d2,d1)+` et `rdfs:subClassOf(d3,d2)` alors `included(d3,d1)`.

Notons que toutes ces règles d'inférences sont exprimées sous forme de requêtes CONSTRUCT en SPARQL interprétées comme des règles. Elles nous permettent d'ajouter plus d'information dans le graphe RDF soit en explicitant certaines informations (règles de normalisation) soit d'ajouter des informations implicites (règles de raisonnement). L'utilisation de ces règles dans des raisonneurs n'affectera que les implications de RDFS (RDFS entailment).

6 Requêtes en HuTO

L'ontologie HuTO (ontologie et règles) a été testée avec Corese (Corby et al., 2012), un moteur sémantique qui permet le traitement de ressources en RDF/S, SPARQL et un langage de règles adapté à RDF. Notre jeu de données compte 1014 triplets. L'exécution des implications de RDFS nous amène à 1660 triplets et celle de nos règles nous amène à 2378 triplets.

Nous distinguons deux types de requêtes: 1) celles qui déterminent les ressources relatives à une période ou relatives à une ressource temporellement annotée donnée et 2) les requêtes qui déterminent la période d'une ressource donnée.

6.1 Requêtes Temporelles sur les Ressources

La requête SPARQL suivante permet par exemple de déterminer la temporalité de la ressource `data:Gamou`.

```
PREFIX dt: <http://ns.inria.fr/huto/>
PREFIX data: <http://example.org/data/>
DESCRIBE ?x
WHERE{ {?x dt:exp+/(rdf:value|rdf:subject|rdf:object) data:Gamou}
       UNION
        {?x dt:exp+/dt:uri ?g
            graph ?g{ { data:Gamou ?p ?o} UNION { ?s ?p data:Gamou}}}
       FILTER NOT EXISTS {?x a dt:Period}
       FILTER NOT EXISTS {?j ?k ?x}}
```

EXEMPLE 5 – Requête déterminant la temporalité de la ressource `data:Gamou`.

Cette requête prend en compte les trois représentations qui peuvent être utilisées pour l'annotation temporelle d'une ressource. Ainsi quelque soit la représentation utilisée pour la ressource, cette requête nous permet de retrouver la temporalité de la ressource.

6.2 Requêtes sur les Éléments Temporels

Dans l'état actuel, nous pouvons par exemple déterminer les ressources récurrentes sur une période donnée. Dans l'exemple 6, la requête détermine les ressources mensuelles:

```
PREFIX dt: <http://ns.inria.fr/huto/>
DESCRIBE ?x
WHERE{ ?x a dt:Cycle;
          dt:every/rdf:type dt:Month
       FILTER NOT EXISTS {?x dt:sample ?t}}
```

EXEMPLE 6 – Requête déterminant les données qui se produisent mensuellement.

Par manque d'espace nous ne pouvons pas donner le détail de toutes nos requêtes types. Cependant nous pouvons déterminer:

- Les ressources répétitives par rapport à une fréquence donnée. Cette fréquence peut être un jour de la semaine, annuelle, mensuelle (Exemple 6), etc;
- Les ressources qui se produisent relativement (avant ou après) à une ressource donnée. Notons que cette requête ne concerne que les intervalles convexes puisque les propriétés `before` et `after` ne sont définies que pour ces types d'intervalles;
- Les ressources qui se produisent à une date donnée. Notons que pour ces requêtes nous avons besoin d'avoir le jour de la semaine comme argument de la requête pour avoir tous les résultats possibles. Par exemple, quelles sont les ressources qui se produisent le `Jeudi 1 Janvier 2015`;
- La date d'occurrence d'une ressource spécifique (Exemple 5).

Notons aussi que pour toutes ces requêtes types, les résultats sont des ressources annotées à l'aide d'intervalles convexes, non-convexes et ou du marqueur temporel `Period`.

Les requêtes types qui restent à traiter sont celles qui déterminent les ressources pour une période (intervalle de temps) donnée. Pour ces requêtes, la prise en compte des intervalles non-convexes est plus complexe.

7 Conclusions et Perspectives

Dans cet article, nous avons présenté HuTO qui est une ontologie en RDFS pour annoter temporellement des données en RDF à l'aide d'expressions du langage courant. Ce travail repose sur deux domaine de recherches dans la modélisation temporelle en Web sémantique.

Dans le domaine de la modélisation des expressions temporelles, HuTO permet la modélisation d'énoncés de temps complexe (exemple 2b). Notre ontologie comprend également un ensemble de règles afin de normaliser et de renforcer la cohérence des données temporelles. Dans notre approche nous considérons toute entité temporelle comme un intervalle pouvant être défini à l'aide de différentes granularités calendaires. Une correspondance existe entre le type `xsd:duration` et HuTO et entre le type `xsd:dateTime` et HuTO excepté la partie fuseau horaire. Une distinction est faite entre les intervalles fermés et infinis de même entre les intervalles convexes et non-convexes. Notre ontologie intègre aussi les relations temporelles `after` et `before` telles que définies dans (Allen, 1983 et 1984). Ces relations sont définies soit entre deux intervalles soit entre deux ressources soit entre un intervalle et une ressource. HuTO permet aussi d'utiliser une ressource comme un marqueur temporel pour dater une autre ressource. Une des contributions majeures de HuTO est la modélisation des intervalles non-convexes de façon à permettre l'écriture de requêtes SPARQL qui permettent de considérer tout type d'intervalle.

Pour l'annotation temporelle des données, HuTO propose une approche qui permet d'associer une dimension temporelle aux connaissances du domaine. HuTO permet également de garder la traçabilité des changements temporels sur les données que ce soit une ressource, un triplet ou un ensemble de triplets.

Plusieurs directions de recherche restent à explorer. À très court terme, nous souhaitons pouvoir traiter tous les types de requêtes concernant les intervalles (cf. section 5.2). Nous comptons traiter aussi les exceptions dans les intervalles non-convexes grâce aux graphes nommés. Nous comptons aussi étudier les autres relations définies dans (Allen, 1983 et 1984) et intégrer celles qui sont pertinentes pour le domaine socioculturel. À moyen terme, nous souhaitons proposer également des patrons de requêtes et des formats de réponses plus lisibles aux utilisateurs qui ne sont pas des experts RDF/SPARQL. À long terme, nous souhaitons enfin intégrer dans HuTO la modélisation de l'incertitude dans les notions temporelles comme dans l'expression `à la fin des années 1980`.

Références

ALLEN J. F (1984). Maintaining knowledge about temporal intervals. In CACM, 26(11):832–843.

ALLEN J. F. (1981). An Interval-Based Representation of Temporal Knowledge.. In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI'81), pages 221–226, Canada.

BATSAKIS S. & PETRAKIS G. M. (2011). SOWL: A Framework for Handling Spatio-temporal Information in OWL 2.0. RuleML Europe 2011: 242-249.

CORBY O., GAIGNARD A., FARON-ZUCKER C. & MOTAGNAT J. (2012). KGRAM Versatile Data Graphs Querying and Inference Engine. In Proc. IEEE/WIC/ACM International Conference on Web Intelligence, Macau.

DIALLO P. F., CORBY O., LO M., MIRBEL I. & NDIAYE S. M. (2014). Sociocultural Ontology: Upperlevel and Domain Ontologies. In Journées Francophone sur les Ontologies, Tunisie, pp 15-27.

DIALLO P. F., NDIAYE S. M & LO M. (2011). Study of Sociocultural Ontology. In the First International Conference on Social EcoInformatics., 1(5):69-74, Spain.

GUTIERREZ C., HURTADO C., & VAISMAN A. (2005). Temporal RDF. In European Conference on the Semantic Web (ECSW'05) (Best paper award), pp 93–107.

HURTADO C., & VAISMAN A. (2006). Reasoning About Temporal Constraints in RDF. Principles and Practice of Semantic Web Reasoning. Lecture Notes in Computer Science Volume 4187, pp 164-178

LADKIN P. B. (1987). The Logic of Time Representation. PhD Thesis at the University of California at Berkeley, November.

PAN F. (2007). Representing complex temporal phenomena for the semantic web and natural language. Phd Thesis at the University of Southern California, December.

PAN F. & HOBBS J. R. (2005). Temporal Aggregates in OWL-Time. FLAIRS Conference 560-565.

RULA A., PALMONARI M., NGOMO A. C. N., G ERBER D., L EHMANN J. & B ÜHMANN L. (2014). Hybrid Acquisition of Temporal Scopes for RDF Data. In The Semantic Web: Trends and Challenges, pp 488-503.

SAURI R., LITTMAN J., KNIPPEN B., GAIZAUSKAS R., SETZER A. & PUSTEJOVSY J (2006). TimeML Annotation GuidelinesVersion 1.2.1. In TimeML Specification.

SCHEUERMANN A., MOTTA E., MULHOLLAND P., GANGEMI A. & PRESUTTI V. (2013). 7th International Conference on Knowledge Capture, K-CAP, pp 89-96.

TAO C., SOLBRIGH H. R. & CHUTE C. G (2011). CNTRO 2.0: A Harmonized Semantic Web Ontology for Temporal Relation Inferencing in Clinical Narratives. In AMIA Summits on Translational Science Proceedings: 64–68. Available at http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3248753/

TAO C., WEI-QI WEI , SOLBRIGH H., SAVOVA G. & CHUTE C. G (2010). CNTRO: A Semantic Web Ontology for Temporal Relation Inferencing in Clinical Narratives. In AMIA Annual Symposium Proceedings: 787–791. Available at http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041418/

WELTY C. & WIKES R. (2006). A Reusable Ontology for Fluents in OWL. In Frontiers in Artificial Intelligence and Applications, 150:226–236.

W3C WORKING GROUP NOTE 12 APRIL (2006). Defining N-ary Relations on the Semantic Web. Available at http://www.w3.org/TR/swbp-n-aryRelations/ .

Construction semi-automatique d'une ontologie sur des manuscrits ouest sahariens

Mohamed Lamine DIAKITÉ[1], Béatrice BOUCHOU MARKHOFF[2]

[1]DMI, Université des Sciences de Technologie et de Médecine, Nouakchott, Mauritanie,
diakite@ustm.mr
[2] LI, Université François Rabelais de Tours, France,
beatrice.bouchou@univ-tours.fr

Résumé : Dans le cadre de la sauvegarde et de la valorisation des documents patrimoniaux, des campagnes de numérisation des manuscrits anciens ont été entreprises dans différents endroits notamment dans une partie de l'ouest africain. Ces campagnes de numérisation ont généré un nombre important des ressources numériques potentiellement riches en informations que les chercheurs en sciences humaines et sociales et le grand public désireraient exploiter. Dans cet article, nous proposons un moyen d'accès à toutes les informations sur les manuscrits qui soit plus riche que ceux disponibles dans les catalogues. Pour cela, nous avons construit de façon semi-automatique une ontologie regroupant les connaissances sur les manuscrits. Les différentes étapes suivies dans la construction de l'ontologie allant de l'acquisition des connaissances à partir d'un certain nombre de ressources jusqu'à son enrichissement semi-automatique à partir d'un thésaurus sont présentées. Nous avons par la suite procédé à son alignement avec certaines ontologies de référence.

Mots-clés : manuscrits arabes anciens, ontologie, thésaurus, CIDOC-CRM, FRBRoo.

1 Introduction

Dans le cadre de la sauvegarde et de la valorisation du patrimoine culturel, on a assisté ces dernières années à des campagnes de numérisation des documents manuscrits anciens qui constituent l'héritage culturel des nations. Cette campagne de numérisation a eu comme corollaire la génération d'un nombre important des ressources numérisées potentiellement riches en informations que les chercheurs en sciences humaines et sociales et le grand public désireraient exploiter. En effet, certaines institutions détentrices de ces manuscrits étaient caractérisées par une absence de moyens et de bonnes conditions de conservation ce qui avait comme conséquences la mise en danger de ces manuscrits à cause de leur exposition à la poussière, aux insectes, aux pillages, etc. La conservation pérenne de ces fonds patrimoniaux passait donc par leur numérisation.

C'est dans ce contexte que le projet BIBLIMOS (BIBLIothèque digitale Multilingues des sources inédites de l'Ouest Saharien) a vu le jour avec pour ambition la mise à la disposition des chercheurs mais aussi du grand public des corpus thématiques d'archives privées et publiques relatifs à l'ouest saharien.

Au delà de la seule conservation, l'idée est donc, une fois les manuscrits numérisés, de faciliter leur accès aux chercheurs et autres utilisateurs à travers une plateforme et selon une organisation thématique. Le problème qui se pose à ce niveau est qu'on se retrouve confronté à une grande masse d'informations caractérisées par une grande hétérogénéité dans les documents (mélange de textes, de graphiques, etc.). Ceci rend donc difficile toute exploitation automatique des manuscrits par le contenu. L'objectif que nous nous sommes fixé est d'offrir une description qui permettra aux utilisateurs d'exprimer plus précisément leurs recherches et ainsi permettre un meilleur accès au contenu. Pour cela nous avons proposé une modélisation

des connaissances sur les manuscrits à travers une ontologie. L'activité de modélisation s'est appuyée sur l'aide des experts du domaine, le catalogue contenant les métadonnées sur les manuscrits et sur l'exploitation d'un thésaurus. Ainsi l'accès au contenu de ces manuscrits se fera par le biais de l'ontologie.

2 L'ontologie comme moyen d'une meilleure exploitation du contenu des manuscrits

Le format image dans lequel les manuscrits sont sauvegardés et leur nombre élevé font qu'il est difficile de pouvoir exploiter directement leur contenu de façon satisfaisante. La plupart des bibliothèques numériques en ligne ont opté principalement pour une recherche par mots clés associée à une présentation des résultats sous la forme d'images de pages numérisées. Parmi ces bibliothèques numériques, on peut citer la base de données OMAR (Oriental MAnuscripts Resource) développée à l'université de Freiburg en Allemagne[1]. Cette base de données contient 2500 manuscrits mauritaniens en arabe sur différentes thématiques. Ces manuscrits peuvent être visualisés en ligne et téléchargés en format PDF.

Les résultats donnés par de tels systèmes sont souvent imprécis et parfois l'utilisateur n'a aucune information sur les termes à utiliser pour formuler ses requêtes.

Nous pensons que l'approche ontologique, en permettant d'exprimer un ensemble de connaissances sur les manuscrits, améliorera la qualité de l'accès au contenu en permettant une formulation plus riche pour les requêtes utilisateur et par conséquent des réponses plus précises aux requêtes.

Nous retrouvons dans la littérature des propositions orientées ontologies pour l'accès par le contenu, qui consistent à associer des informations ou des annotations, extraites manuellement ou automatiquement, aux images des documents. Quelques exemples sont décrits dans (COÜASNON & CAMILLERAPP, 2003) et (COUSTATY, 2011).

Pourtant le potentiel important des ontologies, et plus généralement du web sémantique, pour l'exploitation de manuscrits anciens est bien illustré par exemple dans (JORDANOUS et al. 2012), à propos du projet SAWS (Sharing Ancient WisdomS), dans lequel des informations sémantiques sont extraites de documents au format TEI. Les documents sont issus de collections relatives aux anciennes sagesses grecques et arabes et les informations, récupérées sous forme de triplets RDF, sont des relations entre les contenus, formant un réseau conceptuel interrogeable par les chercheurs.

3 Démarche de construction de l'ontologie

Pour construire l'ontologie nous avons identifié deux niveaux de connaissances pouvant caractériser les manuscrits. Le premier niveau correspond aux connaissances descriptives qui se rapportent à l'exploitation des manuscrits indépendamment de leur contenu et le deuxième niveau est relatif aux connaissances issues du contenu même des manuscrits et sont plus difficiles à acquérir que le premier type de connaissances.

3.1 Connaissances descriptives

Les connaissances externes au contenu des manuscrits, que nous avons appelées aussi connaissances descriptives, sont des connaissances se rapportant à l'exploitation des manuscrits indépendamment de leur contenu. La seule référence au contenu se trouve éventuellement dans le sujet associé aux manuscrits. Ces connaissances sont issues de l'exploitation des métadonnées existantes dans le catalogue de bibliothèque de l'IMRS[2] et des

[1] `http://omar.ub.uni-freiburg.de/`

[2] Institut Mauritanien de la Recherche Scientifique.

discussions que nous avons eues avec les experts sur les manuscrits. La modélisation de ces connaissances nous a permis de construire une première version de l'ontologie (figure 1).

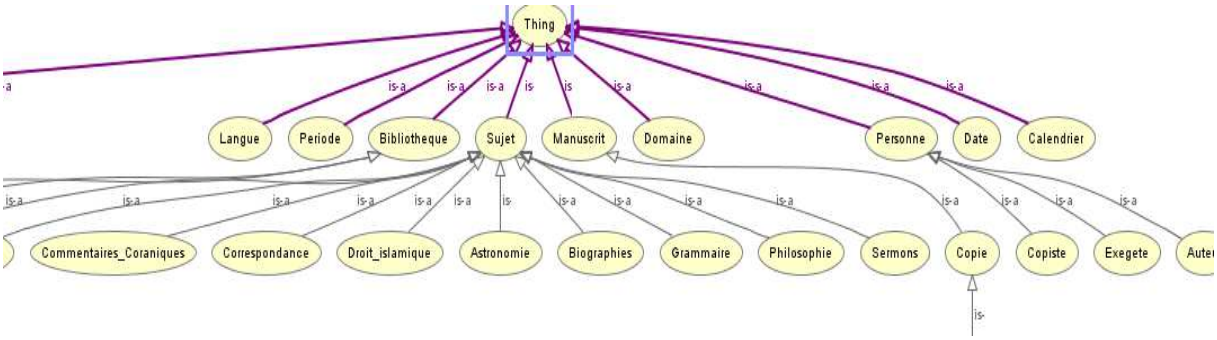


FIGURE 1 - Aperçu d'une partie de l'ontologie sur les manuscrits.

3.2 Connaissances sur le contenu des manuscrits

Ce type de connaissances associé au contenu des manuscrits est plus difficile à modéliser que le premier type de connaissances. Nous développons deux manières de les acquérir:

- soit avec des experts, proposer une modélisation du contenu des manuscrits,
- soit par l'exploitation des différents types de relations existantes dans un thésaurus.

3.2.1 Modélisation du contenu des manuscrits

Pour avoir accès aux connaissances contenues dans les manuscrits, l'apport d'un spécialiste maîtrisant le sujet de leur contenu est primordial. La difficulté d'acquisition de ces connaissances s'explique par le fait qu'elles sont de nature très diverse et sont associées à divers sujets et domaines. C'est donc une tâche qui demande l'intervention d'un grand nombre d'experts de domaines de compétence différents.

3.2.2 Enrichissement semi-automatique de l'ontologie

En complément de l'acquisition des connaissances issues des manuscrits avec l'aide des experts, nous avons développé une méthode d'enrichissement semi-automatique de l'ontologie par l'utilisation du thésaurus RAMEAU[3] mis à disposition en SKOS, donc au format RDF, par la BNF[4].

Il existe plusieurs relations entre les termes dans le thésaurus. Les relations que nous avons exploitées sont les relations hiérarchiques (terme générique et terme spécifique) et les relations d'équivalence (termes équivalents ou alternatifs). La récupération du thésaurus RAMEAU en format RDF s'est effectuée grâce à une requête SPARQL à partir du lien :

`http://data.bnf.fr/sparql`

Afin *d'enrichir la caractérisation des sujets* offerte par notre ontologie (figure 1), nous avons proposé un algorithme implémenté en java et les différentes relations (*skos:broader*, *skos:narrower* et *skos:altLabel*) ont pu être exploitées grâce à la librairie Jena.

L'idée de l'algorithme est de générer pour chaque concept correspondant à un sujet, tous les sous-concepts auxquels le concept est lié par la relation *skos:narrower*. S'il n'y a aucun sous-concept correspondant dans le thésaurus, les sous-concepts de tous les concepts ayant un libellé *skos:altLabel* en commun avec le concept initial sont générés et le choix est laissé à l'utilisateur (ou l'expert) de choisir parmi les sous-concepts générés, ceux qui lui semblent les mieux adaptés.

[3] Répertoire d'Autorité Matière Encyclopédique et Alphabétique Unifié.
[4] Bibliothèque Nationale de la France

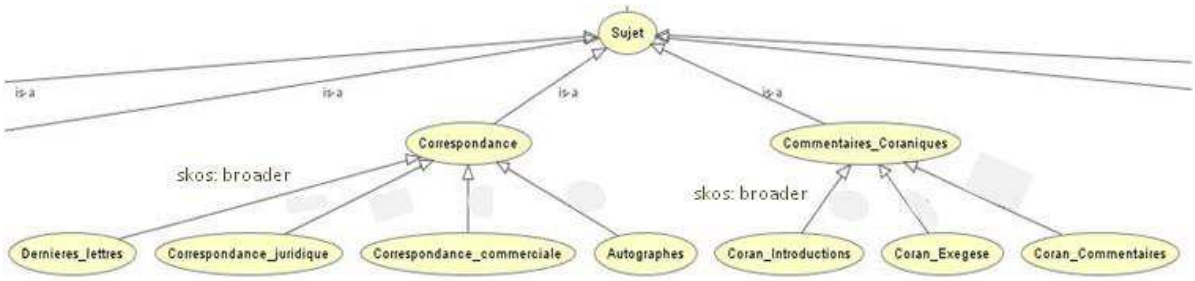


FIGURE 2 - Aperçu d'une partie de l'ontologie après son enrichissement.

La figure 2 illustre l'enrichissement de l'ontologie par des concepts SKOS décrivant les sujets des manuscrits.

4 Discussion sur les résultats obtenus par l'exploitation de RAMEAU

Lors de l'enrichissement de l'ontologie, la difficulté est de trouver pour chaque concept, une entrée correspondante qui est terme préférentiel dans RAMEAU. A défaut il faut chercher tous les termes préférentiels dans RAMEAU qui sont équivalents au label du concept.

Nous avons ainsi observé 4 cas de figure au moment de la comparaison du label d'un concept correspondant à un sujet avec les entrées RAMEAU :

- le cas où il existe une entrée qui concorde avec le label du concept correspondant à un sujet ;
- le cas où il n'existe aucune entrée qui concorde avec le label du concept correspondant à un sujet mais il existe au moins une entrée avec laquelle il existe une relation d'équivalence;
- le cas où il n'existe ni une entrée qui concorde avec le label du concept correspondant à un sujet, ni une entrée qui lui soit reliée par la relation d'équivalence, mais il existe une entrée qui a un sens très proche de celui porté par le label du concept correspondant à un sujet;
- le cas où il n'existe ni une entrée concordant avec le label du concept correspondant à un sujet, ni une entrée qui a un sens proche de celui porté par le label du concept correspondant à un sujet ni une entrée reliée au label du concept correspondant à un sujet par une relation d'équivalence.

Initialement nous avons constitué, à partir de l'exploitation du catalogue, une liste d'une vingtaine de termes correspondant aux labels des concepts correspondant à un sujet. A l'aide d'un programme que nous avons écrit, nous avons vérifié pour chacun des labels sa présence ou non dans le thésaurus. A l'issue de cette vérification, nous avons dû reprendre certains labels qui ne possèdent pas dans RAMEAU d'entrée correspondante, ni de termes qui leur sont reliés par la relation d'équivalence, mais pour lesquels il existe des termes dont la proximité sémantique était évidente pour nous.

Nous avons remplacé ces labels (40% du total des labels) par les termes présents dans RAMEAU qui leur étaient sémantiquement très proches. Par exemple, les concepts *Commentaires* et *Lettres* définis à partir du catalogue renvoient respectivement à *Commentaires coraniques* et *Correspondance* dans le contexte des manuscrits de l'ouest saharien, qui existent dans RAMEAU.

Finalement il y a 15% des labels des concepts correspondant à un sujet qui correspondent au deuxième cas et 70% correspondent au premier cas. Le fait de prendre en compte la proximité sémantique des termes a ainsi permis de ramener de 40 à 70% le nombre de labels qui possèdent une entrée correspondante dans RAMEAU. Dans les 3 premiers cas, l'algorithme permet de générer les sous-concepts du concept correspondant à un sujet. Dans le 4ième cas (qui concernait 15% des labels), aucune génération de sous-concepts n'est possible.

Nous avons constaté que parmi les sous-concepts générés à partir de RAMEAU, certains se prêtent peu au contexte des manuscrits ouest sahariens, du fait que RAMEAU est construit dans un contexte culturel différent. Par exemple, pour le concept *Sermons*, on trouve :

Jésus-Christ -- Passion -- Sermons
Marie, Sainte Vierge -- Sermons
Église catholique -- Sermons

Nous considérons que les sous-concepts générés doivent être en relation avec les manuscrits de l'ouest saharien traitant des sujets portant plus sur des questions en relation avec la religion musulmane.

5 Alignement aux ontologies de référence CIDOC-CRM et FRBRoo

Nous nous sommes fixés comme objectif de rendre l'ontologie interopérable avec d'autres modèles. Nous avons donc procédé à son alignement sur les ontologies de référence CIDOC-CRM[5] et FRBRoo (*Functional Requirements for Bibliographic Records* - Spécifications fonctionnelles des notices bibliographiques).

L'alignement de l'ontologie que nous avons construite avec les ontologies CIDOC CRM et FRBRoo se fait en vérifiant pour chaque concept de notre ontologie, le concept de l'ontologie de référence avec lequel il sera mis en correspondance par une relation hiérarchique (plus spécifique ou plus générique) ou une relation d'équivalence. Nous l'avons fait pour CIDOC CRM d'une part et pour FRBRoo d'autre part, en partant du principe que selon les applications l'un ou l'autre sera utile.

6 Conclusion et perspectives

A notre connaissance, l'ontologie que nous avons construite est la première du genre sur les manuscrits patrimoniaux arabes de l'ouest saharien. Elle permet de formaliser des connaissances explicites et implicites sur les manuscrits. L'acquisition des connaissances suit un processus incrémental, menant à une ontologie modulaire. Ce processus passe à la fois par une interaction avec les experts et par un enrichissement semi-automatique de l'ontologie à partir du thésaurus RAMEAU.

Afin de faciliter l'interopérabilité de notre ontologie avec d'autres modèles, nous utilisons les langages et les modèles du web sémantique et nous alignons l'ontologie sur les ontologies de référence CIDOC-CRM et FRBRoo dédiées au patrimoine culturel.

Dans sa version actuelle, notre ontologie contient les informations qui se trouvaient dans le catalogue, enrichies par les modélisations issues des experts. Nous avons fait vérifier par les experts que toutes les informations ainsi explicitées sont couvertes par les concepts définis dans l'ontologie. Elle doit bientôt être confrontée aux manuscrits au cours de campagnes d'annotation, ce qui constituera une autre forme de validation. Dans un premier temps nous avons aussi utilisé un outil en ligne appelé OOPS![6] (OntOlogy Pitfall Scanner) pour y détecter des erreurs courantes dans la construction d'ontologie.

L'ontologie que nous avons développée dans ce travail est donc une première version d'une ressource destinée à croître, laquelle est elle-même une première étape servant de base à une série d'actions à mener pour réaliser les objectifs du programme BIBLIMOS, à savoir la création d'un portail web dynamique support d'un réseau d'informations autour de l'histoire de l'Ouest-saharien. La prochaine étape, en cours, est la construction à partir de l'ontologie d'un outil d'annotation des manuscrits s'appuyant sur les technologies du web sémantique. En complément de l'outil, nous étudions la spécification de protocoles d'annotation, inspirés de

[5] `http://www.cidoc-crm.org/`
[6] `http://oops.linkeddata.es/`

ceux utilisés dans le domaine du traitement automatique du langage naturel pour l'annotation des corpus de textes.

Références

BANU A., FATIMA S. S. ET KHAN K.R. (2013). Building OWL Ontology: LMSO-Library Management System Ontology. N. Meghanathan et al. (Eds.): Advances in Computing & Inf. Technology. AISC 178. pp. 521–530.

BELAÏD A. & OUWAYED N. (2011). Segmentation of ancient Arabic documents. Volker Märgner and Haikal El Abed. Guide to OCR for Arabic Scripts. Springer.

BIBLIOTHEQUE NATIONALE DE FRANCE (2012). Fonctionnalités requises des notices bibliographiques. Traduction française de *Functional Requirements for Bibliographic Records* : final report. 2ième édition française.

BOUSSELLAA W., ZAHOUR A., TACONET B., BENABDELHAFID A., ALIMI A. (2006). Segmentation texte /graphique : Application aux manuscrits Arabes Anciens. Colloque International Francophone sur l'Écrit et le Document CIFED'06.

CHRISMENT C., HERNANDEZ N., GENOVA F., MOTHE J. (2006). D'un thesaurus vers une ontologie de domaine pour l'exploration d'un corpus. AMETIST, INIST, Vol. 0 : p. 59–92.

COÜASNON B. & CAMILLERAPP J. (2003). Accès par le contenu aux documents manuscrits d'archives numérisés. Document numérique. Volume 7 – n° 3-4/2003, pages 61 à 84.

COUSTATY M. (2011). Contribution à l'analyse complexe de documents anciens Application aux lettrines. Rapport de mémoire de thèse de l'université de la Rochelle.

COUSTATY M., RAVEAUX R. et OGIER J.M. (2012). Historical document analysis: A review of french projetcs and open issues. 19th European Signal Processing Conference (EUSIPCO 2011). Barcelona.

FARQU B., HALLACI S. & AL. (2009) Système Neuro-Markovien pour la Reconnaissance de l'Écriture Manuscrite Arabe à Vocabulaire Limité. Conférence Internationale sur l'Information et ses Applications (CIIA'09). Saida, Algerie. 3-4 mai.

GRUBER T.R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition Academic Press Inc. 5(2).

INTERNATIONAL WORKING GROUP ON FRBR AND CIDOC CRM HARMONISATION (2013). FRBR object-oriented definition and mapping from FRBR ER , FRAD and FRSAD (version 2.0).

ISAAC A.& BOUCHET T. (2009). RAMEAU et SKOS. in *Arabesques*. 54, pp. 13-14.

JORDANOUS A., HEDGES M., LAWRENCE K.F., TUPMAN C. (2012). Exploring Manuscripts : Sharing Ancient Wisdoms across the Semantic Web. International Conference on Web Intelligence, Mining and Semantics (WIMS'12). Craiova, Romania.

JOURNET N. (2006). Analyse d'images de documents anciens : une approche texture. Rapport de mémoire de thèse de l'université de la Rochelle.

KERGOSIEN E. (2011). Point de vue ontologique de fonds documentaires territorialisés indexés. Rapport de thèse de doctorat en informatique de l'université de Pau et des Pays de l'Adour.

NF ISO 21127, Information et documentation (2007). Une ontologie de référence pour l'échange d'informations du patrimoine culturel.

NIANG C., BOUCHOU B., SAM Y. and LO M. (2013).A Semi-Automatic Approach For Global-Schema Construction in Data Integration Systems. IJARAS,Vol. 4(2). pp. 35—53.

PRADEL C., HERNANDEZ N., KAMEL M., ROTHENBURGER B. (2012). Une ontologie du Cinéma pour évaluer les applications du Web Sémantique. in IC 2012.

SHVAIKO P.& EUZENAT J. (2013). Ontology Matching: State of the Art and Future Challenges. IEEE Transaction on Knowledge and Data Engineering. Vol. 25(1), pp. 158—176.

SOULAH M.O. & HASSOUN M. (2011). Which metadata for Ancient Arabic Manuscripts Cataloguing? Proc. International Conference on Dublin Core and Metadata Applications, The Hague, The Netherlands.

WACHE H., VÖGELE T., VISSER U., STUCKENSCHMIDT H., SCHUSTER G., NEUMANN H. and HU□BNER S. (2001).Ontology-Based Integration of Information – A Survey of Existing Approaches. IJCAI Workshop on Ontologies and Informations Sharing.pp 108–117.

WERNER L. (2003). Mauritania's Manuscripts. Saudi Aramco World. Vol. 54, No. 6. pp 2–16.

Un navigateur pour les données liées du Web

Olivier Corby[1], Catherine Faron Zucker[2]

[1] INRIA
olivier.corby@inria.fr

[2] UNIVERSITÉ NICE SOPHIA ANTIPOLIS
faron@unice.fr

Résumé : Nous présentons une plate-forme pour la conception de navigateurs permettant une navigation hypertexte dans des graphes RDF locaux ou sur le Web et présentant ces données RDF en HTML. Cette plate-forme, nommée ALIGATOR (A LInked data naviGATOR), est basée sur le langage de transformation de graphe RDF STTL (SPARQL Template Transformation Language), conçu comme une extension du langage SPARQL. ALIGATOR se présente sous la forme d'un serveur HTTP embarquant, outre un service SPARQL, un moteur de transformation et un service Web permettant d'exécuter des transformations. Nous montrons les capacités du système en présentant trois navigateurs : un premier pour exécuter des requêtes SPARQL sur un graphe RDF local ou sur le Web et présenter les résultat en HTML, un second pour naviguer sur le graphe RDF de DBpedia avec des formats de présentation dédiés à certains types de ressources et un troisième pour présenter une vue unifiée d'un graphe local lié au graphe de DBpedia.

1 Introduction

Le Web de documents structurés qui repose sur le standard XML a rapidement été muni du standard XSLT pour engendrer des formats de présentation tels que HTML ou bien pour écrire des transformations de XML vers XML. De la même manière, le Web de données sémantiques qui repose sur le standard RDF a maintenant besoin d'un langage de transformation pour présenter les données RDF dans des formats lisibles tels que HTML et pour écrire des transformations de RDF vers RDF.

La transformation et la présentation de données RDF est encore un problème ouvert. Nous avons proposé dans (Corby & Faron-Zucker, 2014) et (Corby & Faron-Zucker, 2015) un langage de règles de transformation pour RDF, *SPARQL Template Transformation Language* (STTL), dont le format de sortie est textuel — sans restriction a priori sur le format de texte. L'état de l'art sur les autres solutions existantes pour transformer des données RDF présenté dans les articles ci-dessus mentionnés montre qu'elles sont toutes liées à la syntaxe XML ou à un format spécifique de sortie, ou aux deux, sauf Fresnel (Bizer *et al.*, 2005). Mais Fresnel s'intéresse à la présentation des données RDF et ne traite pas du problème plus général de leur transformation. STTL permet une approche générique pour écrire des transformations RDF vers différents formats de sortie.

Nous nous intéressons dans cet article plus particulièrement à la transformation de données RDF en HTML et nous présentons une plate-forme basée sur le langage STTL, nommée ALIGATOR (acronyme de *A LInked data naviGATOR*), et permettant de concevoir des navigateurs pour le web des données liées et de présenter ces données en HTML. ALIGATOR est constitué d'un serveur HTTP, un service REST, un moteur de transformations STTL et une bibliothèque de transformations STTL. Le code HTML engendré par les transformations contient des liens hypertextes vers le serveur ce qui permet d'offrir une navigation hypertexte sur des graphes RDF.

Cet article est organisé comme suit : la section 2 présente brièvement le langage STTL, la section 3 présente le service REST appelant le moteur de transformations STTL. La section 4 présente différentes applications Web permettant de présenter et naviguer sur des données RDF.

2 Le langage de transformation STTL

Le langage STTL, que nous avons conçu, est une extension du langage *SPARQL 1.1 Query* avec une clause TEMPLATE permettant d'engendrer un résultat sous forme de texte à partir des solutions d'une clause WHERE. La clause TEMPLATE peut contenir du texte, des variables et des expressions. Les variables sont remplacées par les valeurs trouvées dans les solutions, les expressions sont évaluées et le tout est concaténé sous forme d'une chaîne de caractères qui est le résultat retourné par le template. L'exemple suivant montre la traduction d'un énoncé OWL exprimé dans la syntaxe RDF de OWL — une restriction de type "allValuesFrom" — dans la syntaxe fonctionnelle de OWL.

```
TEMPLATE {
  "allValuesFrom(" ?p " " ?c ")"
}
WHERE {
  ?in a owl:Restriction ;
    owl:onProperty ?p ;
    owl:allValuesFrom ?c
}
```

Une transformation STTL est un ensemble de templates dédiés à la transformation d'énoncés RDF dans un certain modèle (e.g. les données de DBpedia sur les rois de France) ou pour un certain langage (e.g. OWL/RDF) dans un format textuel (e.g. une présentation des rois de France en HTML ou une représentation d'énoncés OWL dans la syntaxe fonctionnelle du langage). L'exécution d'un template peut récursivement déclencher l'exécution d'autres templates de transformation ; l'appel à un template se fait au moyen d'une fonction d'extension `st:apply-templates`[1] appelée dans la clause TEMPLATE. Une variante de l'exemple précédent est le template suivant :

```
TEMPLATE {
   "allValuesFrom("
     st:apply-templates(?p) " "
     st:apply-templates(?c) ")"
}
WHERE {
  ?in a owl:Restriction ;
    owl:onProperty ?p ;
    owl:allValuesFrom ?c
}
```

1. Le préfixe `st:` correspond au namespace `http://ns.inria.fr/sparql-template/`

Dans ce template, l'appel de la fonction `st:apply-templates` avec la variable `?p` en paramètre déclenchera la sélection d'un template de transformation qui sera appliqué sur le résultat (binding) associé à la variable `?p` lors de l'appariement du graphe requête de la clause WHERE avec le graphe RDF à transformer. La même chose se produira pour la variable `?c`. Des templates nommés sont également prédéfinis (et d'autres peuvent être définis et nommés) qui peuvent être exécutés directement par un appel à la fonction d'extension `st:call-template`. Voici un template de transformation équivalent au précédent où les templates appelés dans la clause TEMPLATE sont indiqués explicitement : le template `st:property` dédié à la présentation d'une propriété dans la syntaxe fonctionnelle de OWL est appelé avec la variable `?p` en paramètre et le template `st:value` dédié à la présentation des valeurs de propriété est appelé avec la variable `?c` en paramètre.

```
TEMPLATE {
   "allValuesFrom("
     st:call-template(st:property, ?p) " "
     st:call-template(st:value, ?c) ")"
}
WHERE {
  ?in a owl:Restriction ;
    owl:onProperty ?p ;
    owl:allValuesFrom ?c
}
```

3 Le serveur ALIGATOR

Nous présentons dans cette partie la plate-forme ALIGATOR qui repose sur le langage STTL et permet de réaliser des navigateurs hypertextes pour le Web de données. Le code source est disponible dans la distribution de Corese-KGRAM [2] (Corby & Faron-Zucker, 2010) (Corby *et al.*, 2012). Un serveur de démonstration conçu avec ALIGATOR est disponible en ligne : `http://corese.inria.fr`.

Plus précisément, Corese intègre un serveur HTTP Jetty avec des services Web REST qui implémentent le protocole SPARQL 1.1 [3]. Un SPARQL endpoint Corese exécute des requêtes SPARQL convoyées par HTTP et retourne les résultats également par HTTP. Ces résultats sont exprimés dans les formats standards SPARQL Query Results XML Format, Turtle, RDF/XML ou JSON-LD.

Outre cette implémentation standard, ALIGATOR fournit un service Web qui permet d'exécuter des transformations STTL pour engendrer du code HTML à partir de données RDF. Nous appelons *RDF2HTML* ces transformations particulières. Ce service permet de réaliser des navigateurs hypertextes sur un graphe RDF local ou distant.

3.1 Le service STTL

Le service `STTL` répond à deux scénarios décrits sur la Figure 1. Dans le Scénario 1, une transformation RDF2HTML est exécutée sur un graphe RDF. Etant donné un URI, la transfor-

2. http://wimmics.inria.fr/corese
3. http://www.w3.org/TR/sparql11-protocol/

mation engendre une page HTML décrivant la ressource RDF correspondante. Le code HTML contient des liens hypertextes, vers le service STTL, pour décrire les ressources liées à cet URI. Ces liens hypertextes déclenchent un appel au service STTL qui applique la transformation et engendre en retour une nouvelle page HTML. On implémente ainsi une navigation hypertexte sur un graphe RDF. La transformation peut, par exemple, engendrer une table HTML avec les triplets dont l'URI est le sujet ou l'objet, comme le fait DBpedia [4].

Dans le Scénario 2, le service STTL exécute une requête SPARQL sur le graphe puis applique une transformation RDF2HTML sur le résultat de la requête. Pour les requêtes de type CONSTRUCT et DESCRIBE, la transformation est exécutée directement sur le graphe RDF résultat de la requête. Pour les requêtes SELECT et ASK, le résultat de la requête est traduit en graphe RDF en utilisant le vocabulaire publié par le W3C, RDF Data Access Working Group [5]. La transformation est ensuite appliquée sur le graphe produit.

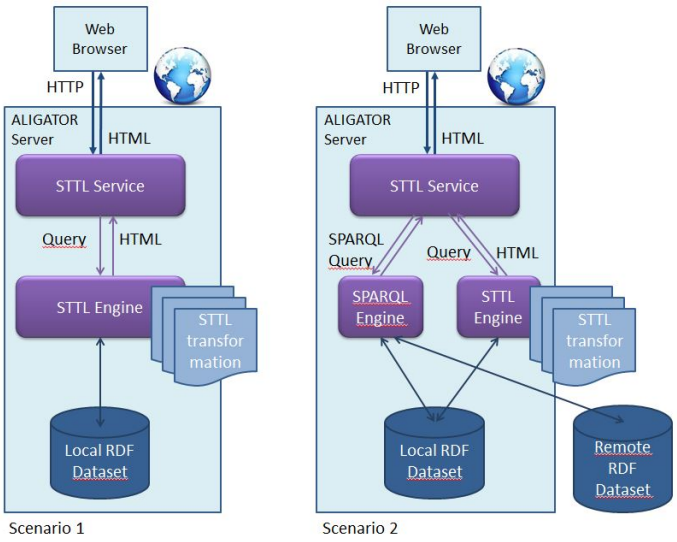


FIGURE 1 – Deux scénarios de transformation RDF2HTML. Scénario 1 : Transformation d'un graphe RDF. Scénario 2 : Transformation du résultat d'une requête SPARQL

Une requête pour une transformation STTL est envoyée à un service ALIGATOR dans un URL dont la partie hiérarchique se termine par `/template` et dont la partie requête contient des paires clé-valeur spécifiant la requête au serveur. La clé `query`, reprise du protocole SPARQL, permet d'indiquer une requête SPARQL à exécuter. La clé `transform` permet de spécifier l'URL de la transformation à appliquer sur le résultat de la requête ou sur le graphe du serveur. Par exemple, l'URL suivant demande l'exécution d'une transformation STTL spécifiée par `st:sparql` sur le résultat d'une requête SPARQL qui retourne tous les triplets d'un graphe RDF donné.

```
http://corese.inria.fr/template?
  query=SELECT * WHERE { ?x ?p ?y }&
  transform=st:sparql
```

4. e.g. http://dbpedia.org/resource/Berlin
5. http://www.w3.org/2001/sw/DataAccess/tests/result-set.n3

3.2 Profil de transformation

Un URL peut donc comporter une requête, une transformation ou les deux. Pour simplifier l'interaction avec le service STTL, nous introduisons la notion de *profil* de transformation RDF2HTML. Un *profil* permet de définir une chaîne de traitements simple, composée d'une requête SPARQL optionnelle et d'une transformation STTL, et de la nommer avec un URI. Un profil est décrit en RDF : la propriété `st:query` définit un chemin (un URL) vers une requête SPARQL et la propriété `st:transform` spécifie l'URL d'une transformation STTL. Voici un exemple de description de profil :

```
@prefix st: <http://ns.inria.fr/sparql-template/>
st:dbpedia a st:Profile ; st:query <q1.rq> ; st:transform st:navlab .
```

Dans cette description, `q1.rq` contient par exemple la requête SPARQL suivante qui interroge le serveur DBpedia :

```
CONSTRUCT { ?x ?p ?y }
WHERE { service <http://fr.dbpedia.org/sparql> { ?x ?p ?y } }
```

Dans l'URL transportant une requête pour une transformation STTL, l'URI d'un profil est indiqué par un argument `profile` dont la valeur est l'URI du profil :

```
http://corese.inria.fr/template?profile=st:dbpedia
```

Une requête peut préciser l'URI d'une ressource sur lequel focaliser l'exécution du service :

```
http://corese.inria.fr/template?profile=st:dbpedia
&uri=http://fr.dbpedia.org/resource/Antibes
```

La description de profil spécifie alors le nom d'une variable dont la valeur sera fixée dynamiquement dans la requête SPARQL comme étant la valeur de l'argument `uri` :

```
st:dbpedia a st:Profile ;
  st:query <q1.rq> ; st:variable "?x" ; st:transform st:navlab .
```

La requête SPARQL exécutée par le serveur est complétée par une clause `values` avec l'URI donné comme valeur de l'argument `uri`. Dans notre exemple, la requête que sera exécutée est donc la suivante :

```
CONSTRUCT { ?x ?p ?y }
WHERE { service <http://fr.dbpedia.org/sparql> {?x ?p ?y} }
VALUES ?x { <http://fr.dbpedia.org/resource/Antibes> }
```

La définition d'un profil de transformation permet d'interroger un serveur distant, DBpedia, sur une ressource particulière, la ville d'Antibes, puis d'appliquer au graphe résultat une transformation RDF2HTML qui engendre une page HTML décrivant la ressource. Le code HTML engendré peut contenir des liens hypertextes sur les autres ressources reliées à la ressource courante. Nous obtenons ainsi un navigateur hypertexte sur un graphe RDF distant (celui de DBpedia). Cette technologie permet donc d'engendrer un navigateur hypertexte sur un SPARQL endpoint. On peut aussi coupler l'interrogation d'un graphe local avec un SPARQL endpoint distant et réaliser ainsi un *mashup* de données liées.

3.3 Contexte de transformation

Nous avons défini la notion de contexte de transformation qui permet à un serveur STTL de transmettre au moteur de transformation STTL des informations relatives au contexte d'exécution de la transformation. Par exemple, pour engendrer des liens hypertextes, le moteur de transformation peut avoir besoin du nom du service. La spécification d'un contexte de transformation évite de "coder en dur" de telles informations dans la transformation, de manière à la rendre le plus générique possible.

Le contexte peut être consulté par le moteur de transformation au moyen d'une fonction d'extension SPARQL `st:get`. Les paramètres possibles du contexte sont les suivants : le nom du service, le nom du profil, le nom de la transformation, l'URI de la ressource courante. Voici un exemple de contexte :

```
st:get(st:service)   = template
st:get(st:profile)   = st:dbpedia
st:get(st:transform) = st:navlab
st:get(st:uri) = http://fr.dbpedia.org/resource/Antibes
```

3.4 Liens hypertextes dynamiques

Une des clés de notre approche et du système développé réside en la capacité à engendrer dynamiquement des liens hypertextes dans le code HTML produit. Lorsque l'un de ces liens est suivi, un nouvel appel au serveur est produit, pour engendrer de nouvelles pages HTML relatives à une nouvelle ressource (avec de nouveaux liens hypertextes), grâce à une nouvelle transformation RDF2HTML.

Voici un exemple d'un tel lien hypertexte ; l'attribut `href` de l'élément `a` a pour valeur un URL qui contient une requête au service STTL du serveur :

```
<a href='/template?profile=st:dbpedia
&uri=http://fr.dbpedia.org/resource/Antibes'>Antibes</a>
```

Voici un exemple de template qui engendre un lien hypertexte vers une ressource `?x` avec un titre `?t` :

```
TEMPLATE st:link(?x, ?t) {
  "<a href='/template?profile=st:dbpedia&uri=" str(?x) "'>"
  str(?t) "</a>"
}
WHERE { }
```

Pour éviter de coder en dur les information relatives au serveur, celles-ci peuvent être représentées dans le contexte :

```
TEMPLATE st:link(?x, ?t) {
   "<a href='/" st:get(st:service)
   "?profile="  st:get(st:profile)
   "&uri=" str(?x) "'>"
   str(?t) "</a>" }
WHERE { }
```

4 Trois navigateurs ALIGATOR

Plusieurs exemples de navigateurs construits avec ALIGATOR en réponse à différents besoins sont disponibles en ligne sur le serveur de démonstration `http://corese.inria.fr`. Nous présentons ici trois d'entre eux.

4.1 Navigateur pour un service SPARQL

Nous avons conçu un navigateur ALIGATOR permettant d'exécuter une requête SPARQL et d'en présenter le résultat en HTML. Le résultat d'une requête de la forme SELECT ou ASK est traduit en RDF en utilisant le vocabulaire W3C RDF Data Access Working Group [6]. Une transformation RDF2HTML est ensuite appliquée sur ce graphe. Nous avons défini le profil `st:sparql` pour identifier cette transformation. Pour les requêtes de la forme CONSTRUCT ou DESCRIBE, la transformation `st:sparql` est directement appliquée sur le graphe RDF résultat. Les règles de cette transformation sont disponibles en ligne [7].

La figure 2 est une capture d'écran de la transformation du profil `st:sparql` appliquée au résultat d'une requête SPARQL (cela est visible dans l'URI entrée dans le navigateur Chrome utilisé). Les liens hypertextes visibles sur la page HTML générée sont des liens vers le serveur ALIGATOR comme expliqué précédemment.

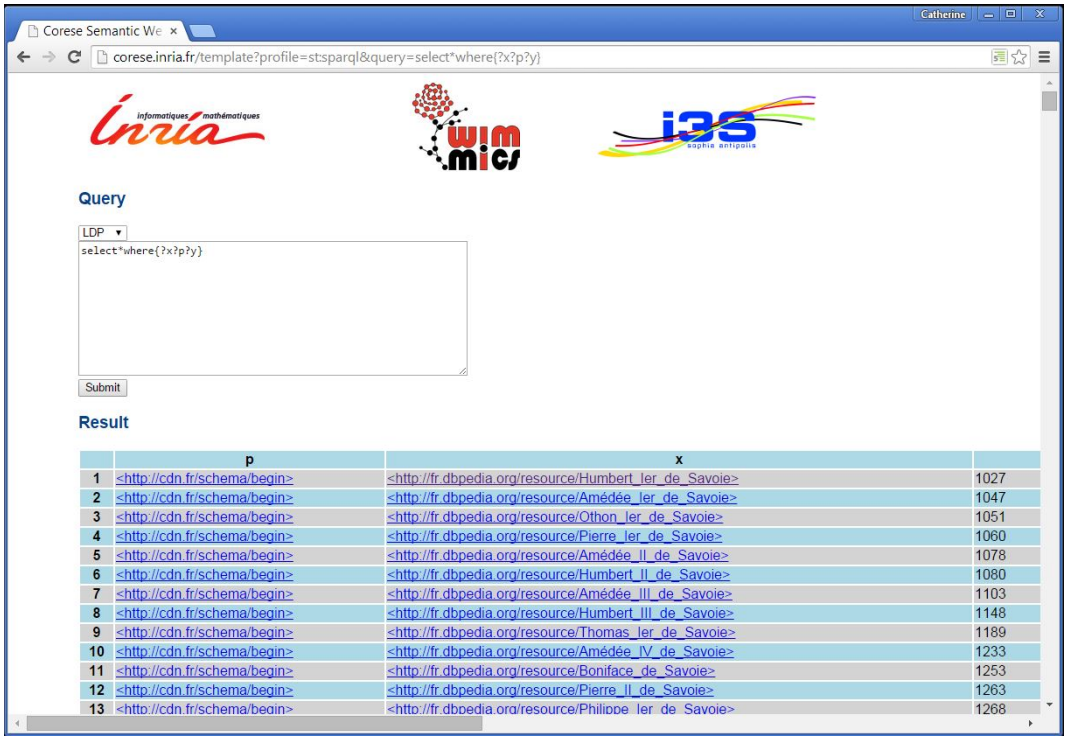


FIGURE 2 – Navigation dans les résultats d'une requête SPARQL

6. http://www.w3.org/2001/sw/DataAccess/tests/result-set.n3
7. http://ns.inria.fr/sparql-template/

Considérons par exemple la requête SPARQL : `SELECT ?x ?n WHERE { ?x rdfs:label ?n }` et l'exemple suivant de solution à cette requête, exprimée en RDF :

```
@prefix rs: <http://www.w3.org/2001/sw/DataAccess/tests/result-set#>
@prefix ex: <http://fr.dbpedia.org/resource/>

[] rs:resultVariable "x", "n" ;
   rs:solution
     [ rs:binding [ rs:variable "x" ; rs:value ex:Auguste ],
                  [ rs:variable "n" ; rs:value "Auguste"  ] ],
     [ rs:binding [ rs:variable "x" ; rs:value ex:Tibère  ],
                  [ rs:variable "n" ; rs:value "Tibère"   ] ].
```

Voici le template principal de la transformation `st:sparql` qui traite les résultats des requêtes de la forme SELECT :

```
prefix rs: <http://www.w3.org/2001/sw/DataAccess/tests/result-set#>
TEMPLATE {
  "<td>"
  coalesce(st:call-template(st:display, ?val), " ")
  "</td>" ; separator = " " }
WHERE {
  ?x rs:solution ?in
  ?x rs:resultVariable ?var
  OPTIONAL { ?in rs:binding [ rs:variable ?var ; rs:value ?val ] } }
ORDER BY ?var
```

La clause WHERE de ce template se focalise sur une solution `?in` qui est un ensemble de liaisons de variables. La clause OPTIONAL énumère ces liaisons de variables ; cette énumération est dans un sous-graphe optionnel car il se peut que certaines variables (`?val`) n'aient pas de valeur. La clause TEMPLATE engendre une cellule de table HTML pour chaque variable (`?val`) avec le résultat de la présentation de la valeur `st:call-template(st:display, ?val)`, ou bien un espace s'il n'y a pas de valeur disponible.

4.2 Navigateur DBpedia

Avec la même technologie, il est également possible de concevoir des navigateurs dédiés à des domaines ou des applications spécifiques. Nous avons ainsi développé un navigateur hypertexte dédié à certaines ressources de DBpedia : les personnes et les lieux. Il repose sur un serveur ALIGATOR offrant un service de transformation STTL avec une nouvelle transformation dédiée, dans le profil `st:navlab`. Son principe de fonctionnement est le suivant. Une requête SPARQL de la forme CONSTRUCT interroge le graphe distant DBpedia, avec une clause SERVICE, sur une personne ou un lieu et retourne un graphe RDF résultat. La transformation `st:navlab` est ensuite appliquée sur ce graphe résultat. Elle engendre une page HTML dédiée à la présentation des ressources retournées par la requête SPARQL. Les lieux sont géolocalisés sur une carte interactive. La figure 3 est une capture d'écran d'une page HTML générée par le navigateur ALIGATOR pour DBpedia. L'URI entrée dans le navigateur Chrome utilisé est

un exemple de requête HTTP envoyée au serveur ALIGATOR. Celui-ci produit en réponse une page HTML engendrée dynamiquement avec le profil `st:dbpedia` qui utilise la transformation `st:navlab`.

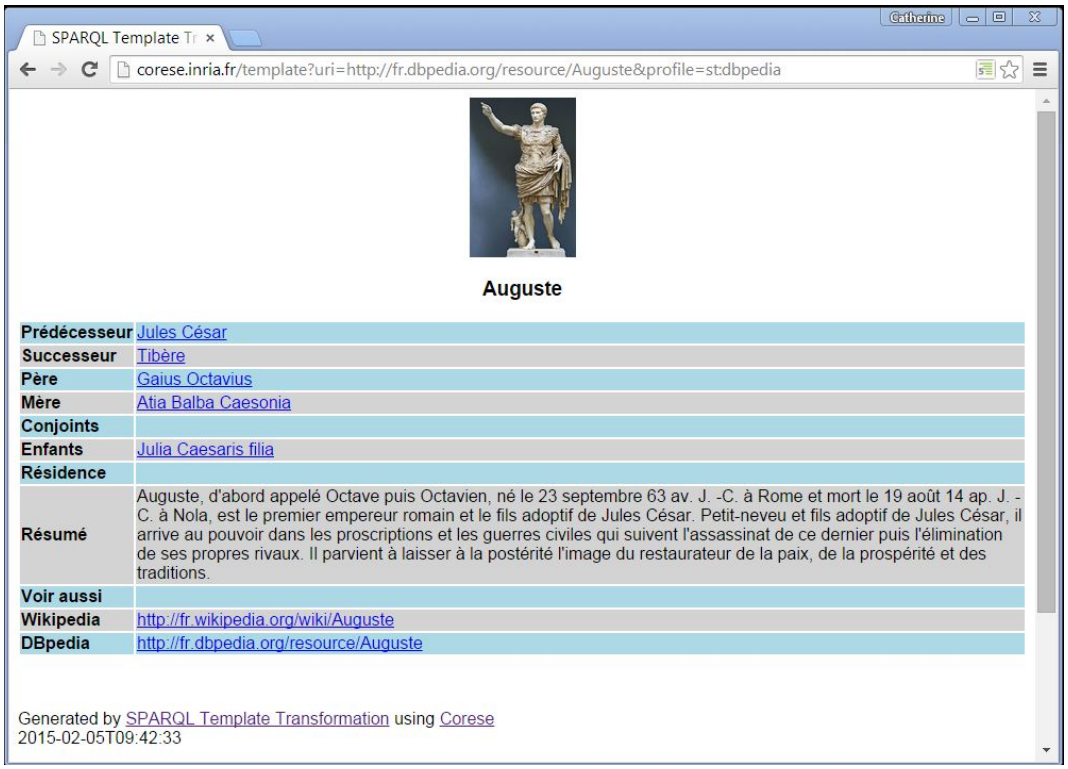


FIGURE 3 – Navigateur DBpedia

4.3 Navigateur Historique

Nous avons développé un troisième type de navigateur qui permet de présenter les données issues d'un graphe local lié à un graphe distant, selon le principe du Web de données liées. La source distante est encore une fois DBpedia ; le graphe RDF local contient un ensemble d'événements et de personnages historiques dont les URI sont ceux de DBpedia [8]. Les énoncés RDF locaux sont stockés dans des graphes nommés annotés avec des thèmes tels que la France, l'Empire, etc.

La transformation RDF2HTML que nous avons développée engendre une page HTML par siècle. Dans chaque siècle, les ressources sont classées par ordre chronologique et rangées dans les colonnes d'une table en fonction du thème de leur graphe nommé. Par exemple, une colonne "France" pour les descriptions dans le graphe annoté par le thème "France". Des liens hypertextes vers les ressources correspondantes de DBpedia sont engendrés selon le même principe que dans la section précédente (avec la transformation `st:navlab`). La figure 4 est une copie d'écran d'une page HTML engendrée par ce navigateur historique.

8. `http://fr.dbpedia.org/resource/`

FIGURE 4 – Navigateur historique

Le template suivant joue un rôle essentiel dans la transformation `st:navlab`. Sa clause WHERE retourne les dates (`?d`) comprises dans un intervalle (par exemple un siècle), triées par ordre chronologique. Sa clause TEMPLATE permet d'engendrer une ligne de table HTML pour chaque date et une cellule de table pour chaque thème dans laquelle sont affichées les éventuelles ressources correspondant à la date et au thème.

```
prefix cn: <http://cdn.fr/schema/>
TEMPLATE  cn:table(?min, ?max) {
  "<tr>"
  "<th class='date'>" st:call-template(cn:wikidate, ?d) "</th>"
  "<td>" st:call-template(cn:date, ?d, cn:fr)      "</td>"
  "<td>" st:call-template(cn:date, ?d, cn:emp)     "</td>"
  "<td>" st:call-template(cn:date, ?d, cn:mds)     "</td>"
  "<td>" st:call-template(cn:date, ?d, cn:context) "</td>"
  "<td>" st:call-template(cn:date, ?d, cn:cdn)     "</td>"
  "</tr>\n" }
WHERE {
  { SELECT DISTINCT ?d  WHERE { ?uri cn:date ?d }}
  FILTER(?min <= ?d && ?d <= ?max)
}
ORDER BY asc(?d)
```

5 Conclusion

Nous avons présenté la plate-forme ALIGATOR permettant de concevoir des navigateurs pour les données liées du Web sémantique. Elle repose sur le langage STTL qui est une extension de SPARQL permettant d'écrire des transformations déclaratives de RDF vers des formats textuels, en particulier ici vers HTML. Le moteur de transformation STTL et le serveur STTL qui constituent le navigateur sont disponibles (en open source) dans la plate-forme Corese. Plusieurs applications de cette technologie sont disponibles en ligne sur un serveur de démonstration à l'adresse `http://corese.inria.fr`.

Un avantage de cette approche, d'un point de vue technique, est qu'elle ne nécessite pas d'apprendre un nouveau framework Web. Il suffit de connaître SPARQL et HTML (et éventuellement JavaScript). STTL permet ainsi de passer "directement" de RDF à HTML avec SPARQL, sans langage de programmation.

Du point de vue de l'ingénierie des connaissances, les avantages de cette approche sont multiples. Tout d'abord, le fait de reposer sur le langage SPARQL est un atout de ce point de vue-là : avec une nouvelle forme de requête dont la clause WHERE est commune aux autres formes de requêtes SPARQL, STTL permet d'envisager de réutiliser des patrons de toutes formes de requêtes lorsque celles-ci sont capitalisées. Egalement, comme pour l'écriture de requêtes SPARQL en général, les patrons de conception mis en œuvre dans la construction des bases RDF sur lesquelles opèrera une transformation STTL peuvent être réutilisés pour écrire les templates qui composent celle-ci. Enfin, la déclarativité du langage STTL permet de capturer les connaissances expertes nécessaires pour opérer des transformations sur des données RDF. Les transformations STTL peuvent être vues comme des connaissances de *présentation* capitalisables, partageables, réutilisables dans des scénarios ou selon des points de vue sur les données similaires.

Dans la continuité de ces conclusions, nous envisageons d'une part d'exploiter les fonctions de HTML 5 couplées avec JavaScript ainsi que d'engendrer des vues graphiques avec des librairie dédiées (e.g. 3D.js). Nous projetons d'autre part d'explorer plus avant la capitalisation de patrons de requêtes SPARQL et patrons HTML et le couplage de ces deux types de patrons.

Remerciements

Nous remercions Eric Toguem (U. de Yaoundé, Cameroun) et Alban Gaignard (CNRS) pour la première version du serveur HTTP qui embarque Corese ainsi que Fuqi Song pour le déploiement du serveur de démonstration `corese.inria.fr` .

Références

BIZER C., LEE R. & PIETRIGA E. (2005). Fresnel - A Browser-Independent Presentation Vocabulary for RDF. In *Second International Workshop on Interaction Design and the Semantic Web @ ISWC'05*, Galway, Ireland.

CORBY O. & FARON-ZUCKER C. (2010). The KGRAM Abstract Machine for Knowledge Graph Querying. In *IEEE/WIC/ACM International Conference on Web Intelligence*, Toronto, Canada.

CORBY O. & FARON-ZUCKER C. (2014). SPARQL Template : un langage de Pretty-Printing pour RDF. In *Proc. 25e Journées francophones d'Ingénierie des Connaissances*, Clermont-Ferrand.

CORBY O. & FARON-ZUCKER C. (2015). STTL: A SPARQL-based Transformation Language for RDF. In *Proc. 11th International Conference on Web Information Systems and Technologies, WEBIST 2015*, Lisbon, Portugal.

CORBY O., GAIGNARD A., FARON-ZUCKER C. & MONTAGNAT J. (2012). KGRAM Versatile Data Graphs Querying and Inference Engine. In *Proc. IEEE/WIC/ACM International Conference on Web Intelligence*, Macau, China.

HARRIS S. & SEABORNE A. (2013). *SPARQL 1.1 Query Language*. Recommendation, W3C. http://www.w3.org/TR/sparql11-query/.

Sémantique et Internet des objets : d'un état de l'art à une ontologie modulaire

Nicolas Seydoux[1,2], Mahdi Ben Alaya[2,3], Nathalie Hernandez[1], Thierry Monteil[2,3], Ollivier Haemmerlé[1]

[1] IRIT Équipe MELODI, Toulouse, France
nicolas.seydoux@irit.fr
hernande@univ-tls2.fr
ollivier.haemmerle@irit.fr
[2] CNRS, LAAS , 7 avenue du Colonel Roche, F-31400 Toulouse, France
[3] Univ de Toulouse, INSA, LAAS , F-31400 Toulouse, France
ben.alaya@laas.fr
thierry.monteil@laas.fr

Résumé : La notion d'Internet des Objets désigne un réseau d'objets connectés entre eux et communiquant de manière automatique. Les notions de sémantiques y ont une place croissante, car plus que jamais elles apparaissent comme une solution aux problèmes d'interopérabilité et d'interprétation des données et des services par des machines. La diversité des applications possibles à l'intersection de l'internet des objets et du web sémantique a poussé de nombreuses équipes de recherche à travailler à l'interface entre ces deux disciplines. Nous souhaitons dans ce papier faire un inventaire de leurs propositions. Nous cherchons également à contribuer à l'évolution de ce domaine de recherche en proposant une ontologie pour décrire les réseaux d'objets connectés.
Mots-clés : Ontologies, Internet des Objets, Interopérabilité, Enrichissement de données

1 Introduction

Au cours des dernières années, l'Internet des objets (IoT en anglais [1]) a évolué à une vitesse exceptionnelle, connectant un nombre important d'objets hétérogènes (capteurs, actionneurs, smartphones, applications, etc.). L'IoT contribue largement à la production de données dans le cadre du Big Data, avec une estimation de 4,4 milliards de Go produits en 2013, certaines estimations projetant l'existence de plus de 26 milliards d'appareils IoT connectés à l'horizon 2020 [2]. L'un des principaux concepts de l'internet des objets est la communication machine à machine (Machine-to-Machine en anglais, abrégé M2M). Le M2M est l'association des technologies de l'information et de la communication, avec des machines communicantes dans le but de fournir à ces dernières les moyens d'interagir sans intervention humaine. Les domaines d'applications sont ainsi très larges : gestion de l'énergie, domotique, aide au diagnostic médical, confort de vie, etc. Le M2M a souffert d'une fragmentation verticale des approches adoptées pour couvrir les besoins des différents domaines d'application. Pour résoudre ce problème, l'Institut Européen de standardisation (ETSI) a développé en 2012 une spécification de standard d'une plate-forme de services horizontale M2M dont une première implementation a été fournie par le LAAS. Celle-ci permet de collecter des données et de piloter des objets variés de manière transparente. Le défi se situe maintenant au niveau de l'utilisation de ces données. La meilleure approche semble être d'associer de la sémantique à ces données par le biais d'ontolo-

1. Internet of Things
2. http://tinyurl.com/le-monde-informatique-IoT

gies dans le but de faciliter leur réutilisation et de permettre la mise en œuvre des mécanismes de raisonnement.

De récents travaux visent donc à proposer des ontologies permettant de représenter les données recueillies par des objets connectés ainsi que les interactions entre ces objets Barnaghi *et al.* (2012), L. Atzori & Morabito (2010). Ces approches sont, pour la plupart, spécifiques aux types d'objets pour lesquels elles sont conçues ainsi qu'aux données qu'ils recueillent. Un réel enjeu consiste à définir une approche générique pour la sémantisation des données dans un double objectif : favoriser l'interopérabilité entre objets, et rendre les données exploitables automatiquement. On pourrait alors les intégrer à des plates-formes mettant en œuvre des mécanismes de diagnostic et de supervision automatiques, ce qui est crucial pour le M2M.

L'objectif de ce papier est double. Nous dressons d'abord un panorama des besoins auxquels peuvent répondre les ontologies dans le domaine de l'IoT et nous présentons les ontologies actuellement disponibles dans le domaine. À partir de cette analyse, nous proposons ensuite une ontologie visant à capitaliser d'une part les travaux faits dans le domaine et d'autre part à prendre en compte les avancées matures dans le domaine de l'Ingéniérie des Connaissances.

Le papier est organisé de la façon suivante : la section 2 présente un état de l'art de l'utilisation des ontologies dans l'IoT, la section 3 décrit l'ontologie IoT-O que nous proposons et la section 4 illustre son intérêt à partir d'un cas réel.

2 État de l'art

Différents projets d'IoT s'appuient sur la sémantique : openiot [3], semsorgrid4env [4], sensei [5]... Les ontologies qu'ils utilisent répondent à des besoins spécifiques au domaine de l'IoT que nous décrivons dans la section 2.1. Ces ontologies se distinguent ensuite par les domaines qu'elles couvrent. Nous en dressons un panorama dans la section 2.2. Finalement, dans la section 2.3, nous listons les manques que nous avons identifiés dans les propositions existantes.

2.1 Les axes d'applications du web sémantique en IoT

2.1.1 Axe orienté données

Le premier apport du web sémantique en IoT se situe au niveau de la transformation des données collectées par les objets. Sémantisées, ces données peuvent prendre le statut d'informations porteuses de sens hors de l'application d'où elles sont collectées. Une fois remises dans un contexte global et interprétées, ces informations peuvent être transformées en connaissances.

La première transformation, de donnée vers information, consiste en l'enrichissement des données par des annotations faisant le lien avec des ontologies. Cet enrichissement peut se faire à différentes étapes du cycle de vie de la donnée : à la création, avant ou après stockage, comme décrit dans Corcho & Gray (2010). Dans certains cas, on peut même prendre le parti d'enrichir des données entreposées dans des bases relationnelles classiques, et dont la sémantisation n'avait pas été envisagée au moment de la collecte, à l'aide de langages tels que R2RML décrit dans Das *et al.* (2012).

3. http://openiot.eu/
4. http://tinyurl.com/semsorgrid4env
5. http://www.sensei-project.eu/

Par cet ajout de métadonnées, les données ne sont plus spécifiques à l'application les ayant collectées, et il y a alors un sens à l'intégration de ces informations au web de données liées. C'est d'ailleurs un autre aspect important de la place du web sémantique dans l'IoT Patni *et al.* (2010), Barnaghi & Presser (2010). Lier ses propres données à des entrepôts distants permet d'en augmenter la valeur, en faisant des données "5 étoiles" [6].

On est ici dans un paradigme proche du Big Data : les données sont certes collectées dans le but d'une application, mais une fois enrichies elles acquièrent une valeur intrinsèque qui les rend exploitables dans des contextes non-envisagés a priori, par exemple via des mashups comme dans SensorMasher [7]. Cette approche nécessite cependant une structuration sémantique des données, et une ouverture à des requêtes "libres" de la part d'utilisateurs distants, l'ouverture d'endpoint SPARQL sur une base de connaissance RDF étant la proposition classique.

Enfin, une fois sémantisées les informations générées par le réseau d'objet peuvent être enrichies par un moteur d'inférence s'appuyant sur les formalismes du W3C. Cette approche peut être utilisée par exemple pour détecter des événements caractérisés par un ensemble d'éléments observables par le système, à la manière de Taylor & Leidinger (2011). On peut ainsi actualiser une représentation du monde via les informations issues d'un réseau de capteurs Henson *et al.* (2012), mais on peut aussi envisager de détecter des pannes chez lesdits capteurs, afin de ne pas accumuler d'informations erronées issues d'un capteur dysfonctionnel.

2.1.2 Axe orienté services

La description d'objets comme des services lève encore une fois des problématiques d'interopérabilité : même si tous les fabriquants proposaient une API pour leurs objets, il est évident que même pour des objets comparables (deux ampoules intelligentes de marques distinctes par exemple), les API présentent des disparités. Pour contrer cette difficulté, une solution est d'abstraire l'interface du service en la sémantisant, ce qui permet un accès aux fonctions non plus uniquement par le nom mais par la description porteuse de sens que l'on en a. Les services seraient donc interprétables automatiquement, et on pourrait envisager de les découvrir "à chaud" sans les avoir spécifiés a priori, sans même avoir à tenir compte du modèle ou de la marque de l'objet les implémentant. Cette approche permet d'envisager la décorrélation entre les applications s'appuyant sur des objets connectés et les fabriquants desdits objets, et de permettre la création de programmes génériques capables de découvrir et d'exploiter des services sémantisés, commme dans Hachem *et al.* (2011).

De plus, approcher un réseau d'objets connectés comme un ensemble de services permet d'y intégrer des objets physiques (implémentant des services), mais aussi des entités virtuelles, des purs webservices. La notion d'objet virtuel peut alors faire son apparition. Illustrons-la par l'exemple d'un capteur de température ressentie : la température ressentie étant composée de la température, de la vitesse du vent et de l'humidité ambiante, tout réseau disposant de ces 3 types d'informations (fournies par des capteurs physiques) peut proposer l'accès à un capteur virtuel, composant ces 3 informations pour indiquer la température ressentie comme décrit dans Compton *et al.* (2009). On introduit en fait ici la possibilité de composer les services. Cette composition peut être faite manuellement, mais on peut aussi envisager qu'une descrip-

6. http://www.w3.org/DesignIssues/LinkedData.html
7. http://sensormasher.deri.org/

tion sémantique suffisamment riche de chaque service permette (dans une certaine mesure) la composition automatique Compton *et al.* (2009).

2.1.3 Axe orienté système

Dans ce cas, les ontologies ont pour rôle la description du réseau de capteurs lui-même plutôt que du phénomène observé. L'ontologie SSN est un bon exemple de cette utilisation, puisqu'elle fait le lien entre capteur et observations sans rien préciser sur la nature de celles-ci. Cette description des capteurs peut être associée à la description des données, pour adresser la problématique de la provenance en liant une donnée au capteur dont elle est issue, avec sa localisation, sa précision...

La nature changeante des réseaux d'objets connectés pose aussi problème : pour que la description du système reste correcte dans la durée, il est nécessaire qu'elle soit dynamique. En effet, les objets sont amenés à être en mouvement, et le réseau dans sa globalité n'est pas statique : ajouts ou retraits d'objets, déplacements au sein du réseau...

La question du dynamisme va de pair avec la configuration automatique. En effet, il est souhaitable que l'ajout ou le retrait d'objets dans un réseau existant ne nécessite que peu d'intervention de la part d'un opérateur humain, et surtout qu'elle ne demande pas une reconfiguration manuelle de l'ensemble du réseau. L'intégration par les fabriquants d'une "datasheet électronique" décrivant l'objet à l'intérieur de celui-ci peut permettre que la découverte du capteur par le réseau soit automatisée, comme proposé dans Kotis *et al.* (2012). Ce domaine particulier d'application est fortement lié aux moteurs d'alignement automatique, car il est difficilement envisageable que l'ensemble des constructeurs s'accordent sur un modèle de représentation unique. Cependant, des outils d'alignement assez fins pourraient permettre de faire le pont entre l'ontologie embarquée par le capteur et celle utilisée par le système auquel on le relie, facilitant l'interopérabilité entre des objets hétérogènes. À l'inverse, les objets consommateurs de services pourront aussi bénéficier de la description du système pour découvrir les autres objets et les éventuels fournisseurs des services qu'ils recherchent : la configuration automatique aurait alors deux aspects, et se ferait à la fois du côté système et du côté objet. On parle de "plug and play", approche visant à minimiser la configuration manuelle avant utilisation par l'utilisateur final, abordée dans Bröring *et al.* (2009).

2.1.4 Utilités fondamentales de la sémantique en IoT

Parmi les trois axes précédemment dérits, on repère des éléments communs, qui sont les apports fondamentaux de la sémantique à l'IoT.

1. **Interopérabilité :** La création d'information à partir de données a pour vocation première l'interopérabilité. Les formalismes de représentation des connaissances proposés par le W3C dans le cadre du web sémantique joue dans l'IoT un rôle de masque de l'hétérogénéité, en proposant une abstraction riche de sens de l'implémentation spécifique sous-jacente. La représentation non-ambiguë de ressources à travers différents formalismes permet aussi de faire fonctionner l'ensemble des systèmes reposant sur des informations comparables mais dans des formats différents, comme proposé dans Page *et al.* (2009). Les travaux sur l'alignement ont aussi pour vocation de permettre la compréhension mutuelle entre des systèmes sémantisés reposant sur des ontologies différentes.

2. **Intégration :** Sémantiser les données permet d'envisager l'intégration de données hétérogènes au sein d'une même structure, le web de données liées, et d'y permettre un accès transparent. Chaque application peut alors s'appuyer sur les données collectées dans son but spécifique, mais aussi sur des données collectées dans d'autres contextes. Cette approche peut mener par exemple à la création de "mashups", services dont le but est spécifiquement d'intégrer des données et services déjà existants afin de les synthétiser en un seul. Un exemple en est donné dans Phuoc & Hauswirth (2009). Ceci n'est possible qu'en sortant du modèle fermé ou seuls les constructeurs peuvent intégrer les données à l'application sans passer par un état où l'information est rendue disponible.
3. **Interprétation :** La création de connaissance peut avoir deux sources : soit la remise de l'information dans un contexte global, soit l'action de règles de déduction sur des connaissances déjà existantes. La première approche demande l'existance d'ontologies faisant référence, qui permet de mettre en regard des informations issues de diverses sources pour en faire des connaissances. C'est par ce biais que l'on peut créer de la connaissance grâce à SSN Barnaghi *et al.* (2011) depuis des réseaux de capteurs, SSN étant suffisamment réutilisée pour que l'information y faisant référence soit intégrée à un contexte plus général que celui local à sa collecte. C'est là aussi qu'interviennent les ontologies de haut niveau, qui situent les concepts les uns par rapport aux autres d'une manière indépendante de l'application. La seconde approche consiste pour sa part à déduire de la connaissance à partir de règles ainsi qu'en raisonnant sur d'autres connaissances. Cette approche permet de générer de la connaissance difficile à obtenir directement et automatiquement : un événement est souvent la corrélation d'un ensemble d'éléments, et ceux-ci doivent donc être considérés dans un contexte global pour prendre leur sens. Le raisonnement à base de règles permet aussi d'envisager la composition automatique de services, ou le diagnostic de pannes.

2.2 Les ontologies de l'IoT

2.2.1 Ontologies d'objets connectés

Le domaine le plus évident est celui de l'IoT lui-même. Plusieurs ontologies, comme l' iot-ontology proposée par Kotis & Katasonov (2013), SAREF [8], ou openiot-ontology [9], visent à décrire le domaine de l'IoT dans sa globalité.

Il est intéressant de mettre à part le domaine des capteurs connectés. Ceux-ci sont les objets connectés les plus "simples", et permettent des applications assez directes, par de la collecte et du traitement de données en masse (comme Semsorgrid4env [10]par exemple). Ces caractéristiques ont fait des réseaux de capteurs sémantisés un sujet d'étude privilégié pour la communauté du web sémantique, menant le W3C à créer un groupe de travail qui a proposé une ontologie des capteurs connectés, SSN, décrite en détail dans Barnaghi *et al.* (2011). Celle-ci a maintenant une valeur de standard, et elle est intégrée dans de nombreux projets reposant sur des réseaux de capteurs, mais aussi dans la plupart des ontologies d'objets connectés en général.

8. https://sites.google.com/site/smartappliancesproject/ontologies
9. http://openiot.eu/ontology
10. http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/activeprojects/56-semsorgrid

La volonté de proposer des ontologies généralistes de l'IoT a conduit à des travers que connaissent d'autres domaines pour lesquels un effort de sémantisation est mené. Aucune ontologie ne fait référence : de projet à projet, peu de réutilisation des concepts définis par des acteurs extérieurs sont effectives et de nombreux concepts sont redéfinis. Ceci pose un problème majeur dans le domaine de l'IoT où la volonté première de la sémantisation est de favoriser l'interopérabilité, l'intégration et l'interprétation. Par exemple, les ontologies iot-ontology et openiot définissent leur propre notion de service ad hoc. Dans l'ontologie SAREF, la notion de capteur est redéfinie alors qu'elle est présente dans SSN. Au lieu de la réutiliser, les auteurs conseillent dans un document séparé [11] de faire un mapping entre leur ontologie et SSN, ce qui ne nous paraît pas être une bonne pratique.

2.2.2 Ontologies de service

L'IoT est fortement lié aux architectures orientées services [12] (SOA en anglais), avec des travaux comme Abangar *et al.* (2010) ou De *et al.* (2011). Les objets connectés dans un réseau peuvent être vus comme autant de fournisseurs et/ou consommateurs de services, et leurs fonctionnalités peuvent donc être décrites comme des interfaces de webservices. Or l'ajout de sémantique dans la description des webservices est un domaine en soi, et des ontologies qui lui sont spécifiques ont été proposées comme WSMO [13] ou MSM [14]. Celles-ci visent à la description des méthodes accessibles via l'interface du service, des paramètres qu'elles attendent, et des réponses qu'elles renvoient. Encore une fois, le W3C a joué un rôle dans ce domaine en fédérant les recherches autour de OWL-S [15], puis de WSMO.

2.2.3 Ontologies de haut niveau

Les ontologies de haut niveau sont des ontologies très générales qui décrivent des concepts abstraits transversaux à de nombreuses applications. Leur intérêt dans le cadre de l'IoT est de faciliter l'interopérabilité des ontologies différentes qui les spécialisent. Par exemple, certains concepts de SSN spécialisent ceux de DUL, OntoSensor spécialise SUMO...

2.2.4 Les ontologies spécifiques

Le déploiement d'un réseau d'objets connectés n'est pas toujours une fin en soi ; il peut également viser une application qui peut elle aussi être modélisée sémantiquement. On retrouvera donc par exemple des ontologies de la maison dans le cadre de projets de domotique Ricquebourg *et al.* (2006), ou des ontologies de phénomènes météorologiques pour les des réseaux de capteurs météo [16]. À chaque fois, le choix de l'ontologie relève de besoins métier spécifiques.

11. https://docs.google.com/file/d/0B2nnxMhTMGh4UnVFMTh1S2R2cGc/edit
12. Architecture logicielle visant à faciliter l'intégrabilité et l'interopérabilité en proposant des services réutilisables, dont l'utilisateur ne connaît pas l'implémentation afin de garantir la souplesse de modification.
13. http://www.wsmo.org/
14. http://iserve.kmi.open.ac.uk/ns/msm/msm-2014-09-03.html
15. http://www.w3.org/Submission/OWL-S/
16. http://www.w3.org/2005/Incubator/ssn/wiki/Agriculture_Meteorology_Sensor_Network

De plus, les ontologies au domaine spécifique mais transversaux, comme la représentation du temps (OWL-Time dans SSN), de la géolocalisation (Geonames...) ou des unités de mesure (SWEET, QUDT...) jouent aussi un rôle important dans l'interopérabilité en permettant une représentation non-ambiguë de métadonnées essentielles à l'interprétation des données issues d'un réseau d'objets connectés.

2.3 Besoins non satisfaits par les ontologies d'IoT

Pour répondre aux enjeux de l'utilisation d'ontologies dans le domaine de l'IoT présenté dans la sous-section 2.1, nous retenons plusieurs manques au niveau des représentations existantes présentées en sous-section 2.2.

2.3.1 Une notion précise d'actionneur

Comme on l'a dit plus haut, SSN est une ontologie non pas d'objets connectés, mais de capteurs connectés, qui sont un sous-ensemble des objets connectés. Un actionneur est un objet qui peut avoir un effet sur le monde physique, soit par son mouvement (moteur, servomoteur), soit par un changement de propriété (température, luminosité, affichage). Les actionneurs sont essentiels à un réseau d'objet connectés dès qu'il a vocation à être actif, et pourtant leur très grande variété a été un frein à l'existence d'une ontologie aussi incontournable que SSN les décrivant. La notion d'actionneur (actuator en anglais) est présente de manière ad-hoc dans plusieurs ontologies, notamment iot-ontology [17], mais pas toujours détaillée, ni unifiée.

2.3.2 Une notion précise de service

IoT et SOA étant fortement liées, il nous paraissait impossible de ne pas intégrer la notion de service à une ontologie de l'IoT. Cependant, là encore, la grande diversité des services a compliqué leur représentation uniforme, et les ontologies ont tendance à s'appuyer sur des définitions ad-hoc, difficiles à accorder entre elles.

2.3.3 Minimiser le nombre de redéfinitions

Nous avons pu constater que certaines ontologies de l'IoT redéfinissaient des concepts déjà existants dans d'autres ontoloties plutôt que d'y faire référence, quitte à recommander des équivalences par la suite. L'IoT repose sur des domaines divers, et il nous paraît donc primordial de donner priorité à l'import d'ontologies conçues par des experts de chacun de ces domaines plutôt que d'effectuer des redéfinitions partielles propres à chaque application.

3 Proposition de IoT-O, une ontologie pour les objets connectés

3.1 Vue d'ensemble

L'IoT comprend un large panel de concepts, allant de "système" à "valeur observée", en passant par "unité de mesure" et "instant". Leur extrême diversité rend impossible leur descrip-

17. http://ai-group.ds.unipi.gr/kotis/ontologies/IoT-ontology

tion par une seule ontologie, et c'est pourquoi l'ontologie IoT-O que nous étendons, dont une première version a été proposée dans Ben Alaya *et al.* (2015), est constituée d'un ensemble de modules ontologiques couvrant correctement les domaines sous-jacents. Une autre volonté de notre conception est d'intégrer des relations entre les notions représentées dans les modules. Nous souhaitons ainsi mettre en œuvre les bonnes pratiques de conception arrivées à maturité dans le domaine de l'Ingéniérie des Connaissances d'Aquin (2012). Le schéma 1 propose une vue d'ensemble de l'ontologie mettant en évidence les modules et les relations les liant. IoT-O est constituée de 5 modules principaux, les modules de service, de capteur, d'observation, d'actionneur et d'action. L'ontologie IoT-O est disponible en ligne [18]

Nous nous appuyons sur DUL pour structurer la description de haut niveau de nos concepts, afin de faciliter la réutilisation d'une application à une autre. Nous réutilisons l'ontologie SSN déjà présentée comme module de capteur et d'observation. En l'absence d'ontologie spécialisée dans la description d'actionneurs, nous proposons dans la partie 3.2 l'ontologie Semantic Actuator Network, calquée sur SSN, décrivant les actionneurs connectés. Ces deux ontologies sont alignées avec DUL. Nous avons choisi de nous appuyer sur l'ontologie et le vocabulaire QUDV pour représenter les unités de mesure, les quantités, les dimensions et leurs valeurs. OWL-time nous sert à représenter les propriétés temporelles de nos événements (durées, instants, datation). Enfin, nous avons choisi de nous appuyer sur MSM comme ontologie décrivant les services ; elle est présentée plus en détail dans la section 3.3.

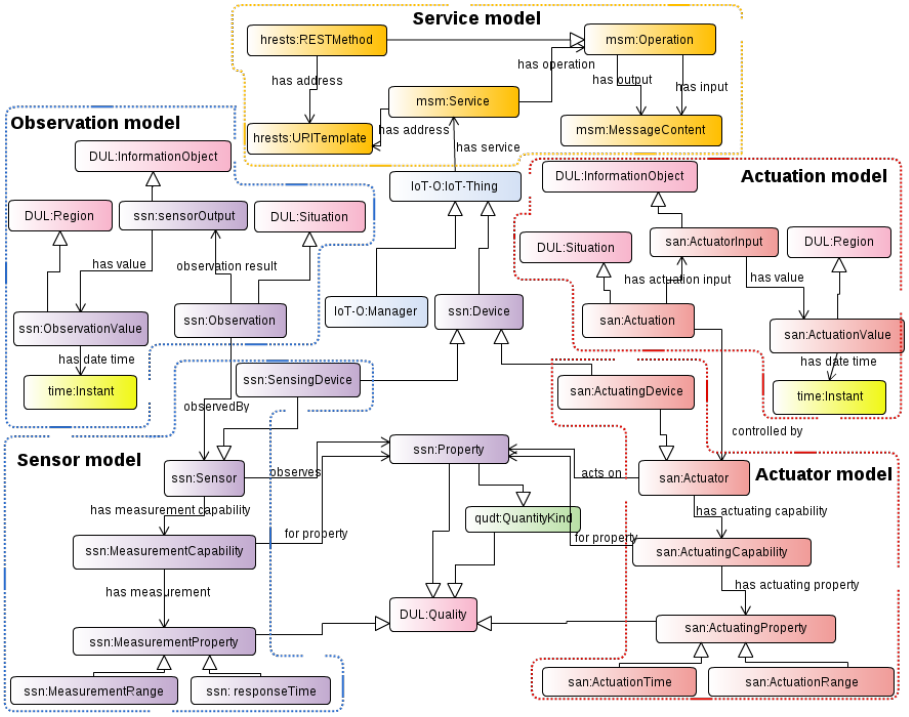


FIGURE 1 – Vue des différents modules consituant l'ontologie IoT-O

18. www.irit.fr/recherches/MELODI/ontologies/IoT-O.owl

3.2 Définition de la notion d'actionneur

Un des principaux apports de l'ontologie IoT-O que nous proposons est une définition riche de la notion d'actionneur. Nous sommes partis de deux constats : SSN est une ontologie très bien intégrée et faisant référence, et les parallèles entre les définitions de capteur et d'actionneur sont nombreux. Nous avons donc défini l'ontologie Semantic Actuator Network (SAN) en la mettant dès que possible en regard avec SSN. On remarquera d'ailleurs la symétrie dans le schéma 1. Nous avons adapté le patron de conception Stimulus-Sensor-Observation détaillé dans Janowicz (2010), structurant SSN, en Actuation-Actuator-Effect, résumé dans la figure 2. Dans le cas de SSN, c'est l'environnement qui est moteur, et c'est une modification de celui-ci (Stimulus) qui va conduire à une représentation par le système (Observation). Au contraire, dans le cas de SAN, c'est la représentation abstraite qui vient la première (Actuation) et qui conduit à avoir un impact sur le monde (Effect).

L'emploi de patrons de conception clairement identifiés est un élément important de l'interopérabilité, et est une bonne pratique aussi bien en ingénérie logicielle qu'en ingénérie des connaissances. Notre objectif est ici de définir une ontologie de capteurs connectés qui sera employée de la manière la plus large possible, et nous pensons que la forte intégration de SSN facilitera la réutilisation de notre ontologie de par leurs similitudes. De plus, on notera que SAN n'importe pas l'intégralité de SSN, mais seulement les concepts sur lesquels elle s'appuie, pour permettre à une application uniquement constituée d'actionneurs connectés de ne pas avoir un grand nombre de concepts non-utilisés dans leur ontologie. Ce choix est motivé par les capacités de raisonnement limitées de beaucoup d'objets de l'IoT.

Notre définition d'actionneur permet donc de représenter un objet ayant des effets sur des propriétés physiques du monde l'entourant suite à des commandes qu'il reçoit, d'une manière complètement indépendante de notre application particulière et pouvant être instanciée d'une manière très similaire à SSN.

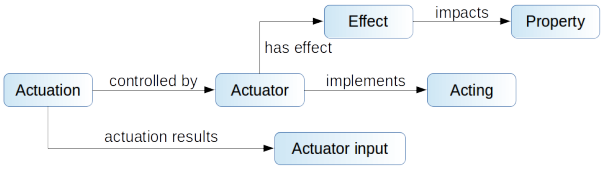


FIGURE 2 – Détail du patron de conception Actuation-Actuator-Effect

3.3 Intégration de la notion de service

La sémantisation de services est une problématique récurrente dans la communauté du web sémantique, et a fait l' objet de nombreux travaux. C'est pourquoi nous avons fait le choix de nous appuyer sur une ontologie existante permettant de représenter les services, proposée dans le cadre du projet SOA4ALL [19] par Jacek Kopecký (2008). Cependant, même si nous avons conservé globalement la forme des ontologies wsmo-lite, msm et hrefs, nous avons souhaité y apporter quelques enrichissements pour les rendre plus faciles à intégrer dans d'autres projets.

19. http://projects.kmi.open.ac.uk/soa4all/

Notre choix s'est porté sur MSM notamment pour la légèreté de sa définition, qui la rend facilement intégrable à tout projet et à tout type de webservices. En particulier, il nous a été difficile d'identifier une ontologie de service permettant à la fois la description des webservices de style REST (Representational State Transfer) et ceux de style RPC (Remote Procedure Call). Ici, nous ne décrivons que les webservices REST grâce à l'extension hRESTS, mais une intégration d'une ontologie de webservices RPC est tout à fait envisageable.

3.4 IoT-O, ontologie de synthèse

Comme on le voit sur le schéma 1, l'ontologie IoT-O comporte peu de définitions de concepts. Elle a plus pour but de faire la synthèse et les liens entre les diverses ontologies évoquées précédemment, et d'offrir une vision d'ensemble des concepts de l'IoT. On remarquera notamment le lien entre iot-o :IoT-Thing et msm :Service, qui indique que tout élément connecté peut fournir un service, qu'il soit un objet physique (ssn :Device) ou une application (iot-o :Manager).

4 Mise en application : le jeu de données ADREAM

Le projet ADREAM (Architectures for Dynamic Resilient Embedded Autonomous and Mobile systems) financé par l'Union Européenne et la région Midi-Pyrénées, porté par le LAAS, comprend un bâtiment intelligent servant aux expérimentations sur les réseaux de capteurs [20]. Il comprend en effet un très grand nombre de capteurs qui mesurent notamment la température, la luminosité, la consommation d'énergie. Dans sa version actuelle, la plate-forme de gestion n'arrive pas à détecter les capteurs dysfonctionnels. Les propositions faites dans cet article ont pour objectif la sémantisation des données de ce bâtiment. Nous illustrons dans cette section l'intérêt de notre ontologie sur un cas simplifié de diagnostic de panne.

4.1 Notion de rétrocontrôle

Le principe d'un actionneur pur est de convertir une commande reçue en un effet physique. Cependant, si on prend l'exemple d'un radiateur intelligent (notre actionneur), il est clair que l'absence de capteurs de température rend impossible son autonomie. Une relation primordiale apparaît donc entre capteurs et actionneurs, la relation de rétrocontrôle. Par celle-ci, un capteur mesure l'effet de l'action d'un actionneur pour en permettre une commande plus fine. Cette notion fait naître un nouveau type de système, un système asservi, dans lequel on mesure par des capteurs la mise en œuvre des commandes envoyées à un actionneur. Pour reprendre l'illustration précédente, asservir le radiateur à une sonde de température permet d'établir une loi de commande de la température du radiateur en fonction de la température de la pièce, assurant un contrôle de la température plus fin qu'en l'absence de la sonde.

4.2 Utilisation du rétrocontrôle dans le diagnostic de panne

Associer un capteur et un actionneur dans un même système asservi permet d'indiquer les actionneurs et les capteurs ayant effet sur/observant la même propriété du milieu. De ce fait,

20. https://www.laas.fr/public/fr/adream

si une action de l'actionneur n'est pas suivie de l'effet attendu mesuré par le capteur, on peut en déduire un dysfonctionnement du système global, sans pour autant pouvoir indiquer où se situe précisément la panne. Il est possible que le capteur ne donne pas une mesure correcte, que l'actionneur ne se comporte pas conformément aux ordres qu'il reçoit, qu'un phénomène extérieur perturbe le système. Le système dans sa globalité est perçu en dysfonctionnement, et une intervention humaine est nécessaire pour prendre les mesures qui s'imposent.

Dans notre implémentation, nous avons proposé un jeu de données simpliste décrivant deux systèmes bouclés minimaux, constitués par une ampoule intelligente et par un capteur de luminosité. Nous avons ensuite écrit une règle SWRL indiquant la déficience d'un système comme un dépassement de sa tolérance de consigne par la différence entre son action et l'observation qui en est faite : un système est déclaré défectueux si son action n'a pas de répercussion sur la propriété observée. Dans le cas de notre implémentation, nous avons donc simulé deux comportements : un comportement nominal pour le système 1, dans lequel la luminosité augmente quand on allume la lampe, et un comportement défectueux pour le système 2, dans lequel la luminosité reste très faible alors qu'on demande un allumage de la lampe. Face à cet état de fait, le raisonneur Pellet va inférer que le système 2 est un système défectueux, et pas le système 1. Il est important de noter qu'on s'est appuyé sur Pellet car il supporte les BuiltinAtoms de swrl, nécessaires à notre définition de système défectueux. La base de connaissance considérée ainsi que la règle SWRL sont disponibles en ligne [21].

5 Conclusion

Dans ce papier, nous avons exposé une synthèse des travaux de l'état de l'art visant à introduire des ontologies dans l'internet des objets. En s'appuyant sur nos observations, nous avons proposé une ontologie de l'IoT intégrant différents modules réutilisant et enrichissant les ontologies existantes. Enfin, nous avons illustré l'intéret de notre ontologie sur un cas réel pour montrer la validité et l'intérêt de notre proposition pour le diagnostic de panne. Actuellement, nous cherchons à comparer notre ontologie aux ontologies existantes à partir des méthodes de la littérature. Nous souhaitons étudier plus en détail les alignements entre les différents modules. Notre objectif à moyen terme est de réaliser un passage à l'échelle en traitant un grand volume de données issues du bâtiment intelligent ADREAM. Nos travaux s'inscrivent dans le cadre du projet OneM2M [22], nouvelle proposition faite conjointement par l'ETSI et d'autres organismes de standardisation internationaux [23], qui vise à intégrer la sémantique au niveau de la norme.

Références

ABANGAR H., BARNAGHI P., MOESSNER K., NNAEMEGO A., BALASKANDAN K. & TAFAZOLLI R. (2010). A service oriented middleware architecture for wireless sensor networks. In *Future Network & MobileSummit 2010 Conference Proceedings*.

21. www.irit.fr/recherches/MELODI/ontologies/IoT-O-InstSystemesBoucles.owl
22. https://www.brighttalk.com/webcast/11949/134201
23. http://onem2m.org/about-onem2m/partners

BARNAGHI P., COMPTON M., CORCHO O., CASTRO R. G., GRAYBEAL J., HERZOG A., JANOWICZ K., NEUHAUS H., NIKOLOV A. & PAGE K. (2011). *Semantic Sensor Network XG Final Report*. Rapport interne, W3C.

BARNAGHI P. & PRESSER M. (2010). Publishing linked sensor data. In *Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN10)*.

BARNAGHI P., WANG W., HENSON C. & TAYLOR K. (2012). Semantics for the internet of things : early progress and back to the future. *International Journal on Semantic Web and Information Systems, special issue on sensor networks, Internet of Things and smart devices*, **8**.

BEN ALAYA M., MEDJIAH S., MONTEIL T. & DRIRA K. (2015). Towards semantic data interoperability in onem2m standard. *IEEE Communications Magazine*.

BRÖRING A., JANOWICZ K., STASCH C. & KUHN W. (2009). Semantic challenges for sensor plug and play. In *Proc. Web & Wireless Geographical Information Systems 2009*.

COMPTON M., NEUHAUS H., TAYLOR K. & TRAN K. (2009). Reasoning about sensors and compositions. In *Proc. Semantic Sensor Networks*.

CORCHO J.-P. C. O. & GRAY A. J. G. (2010). Enabling ontology-based access to streaming data sources. In *ISWC 2010*.

DAS S., SUNDARA S. & CYGANIAK R. (2012). *R2RML : RDB to RDF Mapping Language*. Rapport interne, W3C.

DE S., BARNAGHI P., BAUER M. & MEISSNER S. (2011). Service modelling for the internet of things. In *FedCSIS 2011 proceedings*.

D'AQUIN M. (2012). Modularizing ontologies. In M. C. SUÁREZ-FIGUEROA, A. GÓMEZ-PÉREZ, E. MOTTA & A. GANGEMI, Eds., *Ontology Engineering in a Networked World*, p. 213–233. Springer Berlin Heidelberg.

HACHEM S., TEIXEIRA T. & ISSARNY V. (2011). Ontologies for the internet of things. In *International Middleware Conference*.

HENSON C., SHETH A. & THIRUNARAYAN K. (2012). Semantic perception : Converting sensory observations to abstractions. In *IEEE Internet Computing 16*.

JACEK KOPECKÝ, KARTHIK GOMADAM T. V. (2008). hrests : an html microformat for describing restful web services. In *International Conference on Web Intelligence*.

JANOWICZ, K. C. M. (2010). The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In *Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN10)*.

KOTIS K. & KATASONOV A. (2013). Semantic interoperability on the internet of things : The semantic smart gateway framework. In *International Journal of Distributed Systems and Technologies*.

KOTIS K., KATASONOV A. & LEINO J. (2012). Aligning smart and control entities in iot. In *The 5th Conference on Internet of Things and Smart Spaces 2012*, St. Petersburg, RU : Springer Springer.

L. ATZORI A. I. & MORABITO G. (2010). The internet of things : A survey. *Computer Networks*, **54**.

PAGE K., ROURE D. D., MARTINEZ K., SADLER J. & KIT O. (2009). Linked sensor data : Restfully serving rdf and gml. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks*.

PATNI H., HENSON C. & SHETH A. (2010). Linked sensor data. In *International Symposium on Collaborative Technologies and Systems, 2010*.

PHUOC D. & HAUSWIRTH M. (2009). Linked open data in sensor data mashups. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks*.

RICQUEBOURG V., MENGA D., DELAHOCHE L., MARHIC B., DURAND D. & LOGÉ C. (2006). Architecture de perception de contexte orientée service pour l'habitat communicant. In *Troisièmes Journées Francophones : Mobilité et Ubiquité*.

TAYLOR K. & LEIDINGER L. (2011). Ontology-driven complex event processing in heterogeneous sensor networks. In *The Semanic Web : Research and Applications (2011)*.

Un modèle de recommandation contextuel pour la prédiction des intérêts des consommateurs sur le Web

Mohamed Ramzi Haddad[1], Hajer Baazaoui[1], Djemel Ziou[2], Henda Ben Ghezala[3]

[1] Laboratoire Riadi-Gdl, École Nationale des Sciences de l'Informatique, Université de la Manouba, La Manouba 2010, Tunisie
haddad.medramzi@gmail.com, hajer.baazaouizghal@riadi.rnu.tn, henda.benghezala@riadi.rnu.tn

[2] Centre de recherche MoIVRe, Département d'informatique, Faculté des sciences, Université de Sherbrooke, 2500 Boul. Université, Sherbrooke, J1K 2R1, Canada
djemel.ziou@usherbrooke.ca

Résumé : Avec l'explosion du commerce sur le Web, il devient difficile de cibler les besoins des consommateurs avec les produits qui conviennent le plus à leurs préférences. Dans cet article, nous proposons et évaluons un modèle de recommandation probabiliste ayant pour objectif de prédire les intérêts et les achats des consommateurs. Le modèle de recommandation proposé tient compte des connaissances sur la psychologie du consommateur et intègre les différents facteurs qui influencent les comportements de consommation tels que la démographie, les caractéristiques des produits, les évaluations, le contexte et l'historique des achats. L'expérimentation comparative montre que notre modèle unifie les principales idées directrices des approches classiques et donne de meilleurs résultats sur un jeu de données réelles.

Mots-clés : recommandation comportementale, filtrage contextuel, modélisation du consommateur, prédiction des intérêts et des achats

1 Introduction

Les consommateurs sont souvent confrontés à un grand nombre d'alternatives et d'informations prêtant les moins expérimentés à la confusion. Dans les galeries marchandes classiques, les vendeurs avaient comme responsabilité de déterminer les besoins et les préférences des clients afin de mieux les conseiller et garantir leur satisfaction par le produit à acquérir. Dans le cas d'un achat en ligne, les consommateurs ne sont plus guidés ni conseillés en explorant les offres mises à leurs dispositions. Le consommateur peut faire face au problème de la paralysie par l'analyse (Iyengar & Lepper, 2000) et ne pas prendre une décision face à une surcharge d'information et de possibilités, surtout lorsque celui-ci n'a pas de connaissance sur le produit ou lorsque le coût d'acquisition est élevé. Par ailleurs, selon (Schwartz, 2005), plus les consommateurs ont de choix concernant un type donné de biens, plus ils seraient susceptibles de regretter les décisions et les choix qu'ils vont prendre.

Plusieurs études sur la psychologie du consommateur ont confirmé l'existence du problème de surcharge de choix et ses implications sur le bien-être des consommateurs (Schwartz, 2005; Jacoby *et al.*, 1974; Iyengar & Lepper, 2000). De plus, ils insistent sur la nécessité de réduire les choix mis à disposition des consommateurs afin de permettre et d'accélérer leur prise de décision (Schwartz, 2005). Cette solution reviendrait alors à filtrer les ressources proposées

au consommateur. Plusieurs propositions ayant pour objectif le filtrage de l'information ont été proposées comme les approches de personnalisation et de recommandation. Ces approches cherchent à cibler les consommateurs avec les produits ou le contenu les plus appropriés et limiter ainsi les effets de la complexité de l'information et de la surcharge de choix.

Le modèle de recommandation que nous proposons dans cet article s'inspire des travaux sur la psychologie du consommateur et a pour objectif de prédire les intérêts et les achats de celui-ci. Notre méthodologie consiste à prendre en considération tous les facteurs qui influenceraient les intérêts et les décisions d'achat du consommateur dans l'optique de généraliser les idées et les hypothèses directrices des approches de recommandation existantes. Pour cela, le modèle se base sur plusieurs facteurs, à savoir (1) la démographie, (2) les attitudes des consommateurs, (3) les propriétés des ressources (p. ex. produits, services, contenu informationnel, etc...), (4) le contexte de consommation et (5) les historiques des achats. Il s'agit donc de recenser et de formaliser ces facteurs afin d'étudier leurs causalités et mieux prédire les comportements de consommation moyennant un modèle statistique.

Cet article est organisé comme suit. Dans la section 2, nous présentons un état de l'art des approches recommandation. Ensuite, dans la section 3, nous définissons les facteurs retenus qui influenceraient les décisions des consommateurs afin de parvenir à la modélisation de l'approche recommandation. Par la suite, nous formalisons le modèle de recommandation proposé et détaillons la méthodologie de génération des recommandations. Enfin, dans la section 4, nous présentons une expérimentation comparative entre plusieurs variantes de notre modèle et des approches de recommandation classiques afin de mesurer et de discuter l'apport de notre proposition sur un jeu de données réelles.

2 État de l'art des approches de recommandation

Les approches de recommandation adoptent des idées directrices et des hypothèses différentes afin de déterminer les ressources les plus pertinentes pour un utilisateur donné. Par conséquence, selon leurs processus d'inférence et de génération des recommandations, les approches existantes peuvent être classées parmi les catégories suivantes.

2.1 Approches basées sur les attributs

Ces approches évaluent la pertinence des ressources sur la base de leurs attributs. En effet, plus les attributs d'une ressource concordent avec les préférences des utilisateurs, plus elle est susceptible d'être recommandée. Les principales approches basées sur les attributs sont les approches de filtrage par contenu (CBF) (Balabanović & Shoham, 1997; Mooney & Roy, 2000).

2.2 Approches basées sur les corrélations des utilisateurs

La génération des recommandations est basée sur les corrélations qui peuvent exister entre les utilisateurs. Dans ce cadre, les utilisateurs corrélés peuvent être ceux ayant des ressources préférées communes, des patrons de notation concordants (Resnick *et al.*, 1994; Goldberg *et al.*, 1992) ou des attributs démographiques similaires (Krulwich, 1997; Pazzani, 1999). Les recommandations proposées à un utilisateur regroupent les ressources ayant été jugées comme pertinentes par les utilisateurs qui lui sont corrélés. D'autres variantes tentent de prédire les notes

que donnerait l'utilisateur aux ressources qu'il ne connaît pas afin de ne retenir que celles qui seraient les plus intéressantes. Le filtrage collaboratif orienté utilisateur (UC-CF) et le filtrage démographique (DF) utilisent ces types de raisonnements afin de générer des recommandations.

2.3 Approches basées sur les corrélations des ressources

Les recommandations sont générées à partir des corrélations entre les ressources disponibles et celles déjà préférées ou adoptées par l'utilisateur. Par exemple, deux produits sont considérés comme corrélés s'ils sont fréquemment achetés ensemble ou jugés comme intéressants par les mêmes groupes d'utilisateurs (Linden *et al.*, 2003). Les recommandations issues sont souvent expliquées par le constat suivant : "les utilisateurs qui ont apprécié/acheté ce produit on aussi apprécié/acheté les produits A et B". D'autres approches utilisent les corrélations entre les notes obtenues par les ressources afin de déterminer les produits ayant les mêmes patrons de notations (Sarwar *et al.*, 2001; Deshpande & Karypis, 2004). Ensuite, la note qu'affecterait un utilisateur à un produit est estimée en se basant sur les notes qu'il aurait données aux ressources qui lui sont corrélées. Le filtrage collaboratif orienté item (IC-CF) et les algorithmes à base de règles d'association utilisent ces types de raisonnement pour générer les recommandations.

2.4 Approches basées sur les connaissances

Les recommandations sont issues d'un processus d'inférence encapsulant des connaissances du domaine et permettant de déduire les ressources pertinentes à partir des besoins et des préférences des utilisateurs (Burke, 2000; Lin *et al.*, 2002). Ces connaissances associent les besoins et les contraintes de l'utilisateur aux ressources pouvant les satisfaire. Ces connaissances peuvent être déterminées par des experts, exprimées explicitement par les consommateurs ou apprises à partir des données disponibles grâce aux techniques de fouille de données (Agrawal & Srikant, 1994). Cette classe d'approches se distingue par la représentation formelle et fonctionnelle des connaissances permettant le raisonnement et la déduction des recommandations grâce à des mécanismes d'inférence (p.ex. représentations à base de règles et à base de cas).

2.5 Approches contextuelles

Des recherches sur la personnalisation et la recommandation ont souligné l'influence du contexte sur les décisions et les intérêts des consommateurs. Ceci a conduit à des approches de recommandation sensible au contexte (Woerndl & Groh, 2007; Jones, 2005). De telles approches sont principalement dérivées des approches existantes augmentées par la dimension contextuelle afin d'identifier les cas où le contexte, tel que le cadre spatio-temporel ou le support d'accès à l'information, implique certains comportements de consommation communs et prévisibles (Boutemedjet & Ziou, 2008; Baazaoui *et al.*, 2014).

2.6 Approches hybrides et comportementales

Plusieurs recherches récentes s'intéressent à la composition et à l'utilisation simultanée de plusieurs approches de recommandation dans des approches hybrides. Ces travaux partent de l'hypothèse que l'hybridation des approches de recommandation pourrait améliorer leur efficacité et permettrait de surmonter leurs lacunes. En effet, les systèmes de recommandation peuvent

être vu comme des classifieurs dont le rôle est de séparer les produits pertinents des autres et peuvent donc bénéficier des techniques d'agrégation ou de composition de classifieurs. Ce type de systèmes hybrides repose donc sur un ensemble de sous-systèmes "naïfs", mais spécialisés, dont l'agrégation améliore la qualité de la recommandation. Dans (Burke, 2007), l'auteur justifie le recours aux systèmes de recommandation hybrides et définit plusieurs stratégies d'hybridation selon le contexte et les besoins du domaine. Enfin, plusieurs recherches s'intéressent à l'analyse des réseaux sociaux et des sentiments des consommateurs afin de mieux cibler leurs préférences et anticiper leurs comportements (hsien Liao *et al.*, 2012; Poussevin *et al.*, 2014).

2.7 Discussion et objectifs

Bien que les études sur les comportements de consommation ont recensé plusieurs facteurs qui influencent les décisions des consommateurs, les approches de recommandation existantes sont basées sur des hypothèses et des constatations simples et réductrices sur la façon avec laquelle les consommateurs font leurs choix (Jacoby *et al.*, 1974; Iyengar & Lepper, 2000; Hawkins *et al.*, 2003; Schwartz, 2005). En effet, les approches de recommandation hybrides ont démontré que la composition de plusieurs approches permettait d'améliorer la pertinence des recommandations en comblant des lacunes de chaque approche et en prenant en considération tous les facteurs qui influenceraient les intérêts du consommateur. Ainsi, même si les approches de recommandation classiques ont déjà prouvé leur utilité et leur efficacité, la majorité n'utilise pas toutes les connaissances sur les comportements des consommateurs et sont incapables d'expliquer leurs recommandations à l'utilisateur.

Dans ce travail, nous nous intéressons à la problématique d'unification des approches de recommandation afin d'intégrer et de profiter des connaissances théoriques et empiriques des recherches sur la psychologie du consommateur. La généralisation des idées directrices des approches existantes permettrait de proposer un modèle capable de prédire les choix des consommateurs qu'ils soient influencés par les caractéristiques des produits, par leurs démographies ou par le contexte. Notre objectif se présente donc, d'une part, du point de vue méthodologique par la modélisation des comportements des consommateurs et des facteurs desquels dépendent leurs intérêts et décisions. D'autre part, nous formalisons statistiquement le modèle proposé afin de prendre en considération la nature prédictive de la problématique de recommandation et mieux gérer les incertitudes concernant les comportements observés.

3 Formalisation du modèle de recommandation proposé

Les recherches sur la psychologie du consommateur identifient plusieurs facteurs qui influencent les comportements des consommateurs (Hawkins *et al.*, 2003). Cependant, seuls les descripteurs pouvant être recueillis sur les plateformes de commerce électronique avec une interaction minimale avec les utilisateurs, on été retenus dans ce travail. Ces facteurs sont modélisés par les variables suivantes :

– Les consommateurs : chaque utilisateur $u \in U$ est décrit par un ensemble de variables démographiques d_i comme l'âge, le sexe, le niveau d'études, les revenus annuels ou le pays tel que $u = (d_1, \ldots, d_{N_d})$. De plus, les utilisateurs sont aussi décrits par l'ensemble des objectifs qu'ils tentent de satisfaire moyennant les acquisitions à effectuer similairement au travail de Zhang *et al.* (2007).

– Les ressources : chaque ressource (p.ex. produit) $x \in X$ est décrite par un ensemble d'attributs f_i tel que $x = (f_1, \ldots, f_{N_f})$. Ces attributs permettent le calcul des similarités existantes entre les produits moyennant une mesure de similarité. Les produits peuvent aussi être décrits par les objectifs auxquels ils répondent à travers leurs propriétés (Zhang *et al.*, 2007).
– Les évaluations : les évaluations $e \in \{e_1, e_2, \ldots, e_{N_e}\}$ sont les notes que peuvent attribuer les consommateurs aux ressources. Une note e_{ij} reflète l'attitude et l'intérêt d'un utilisateur u_i vis-à-vis des attributs d'un produit x_j et leur concordance avec ses intention et ses objectifs.
– Le contexte : la prise en compte du contexte $q \in \{q_1, \ldots, q_{N_q}\}$ lors de l'analyse et la prédiction des comportements des consommateurs permet de capturer les intérêts et les achats périodiques ou occasionnels.
– L'historique des achats : l'achat est une variable binaire $a \in \{a^+, a^-\}$ et est la cible de notre modèle de recommandation. Contrairement à des approches recommandation existantes qui recommandent les items ayant la meilleure note prédite, nous estimons que la probabilité d'achat est la plus appropriée surtout dans les cas où l'acquisition a un certain coût. En effet, l'intérêt n'est pas le seul facteur qui influence les achats, mais aussi la concordance des items aux contexte et aux contraintes du consommateur.

Afin d'exploiter les similarités entre les consommateurs et les ressources, nous procédons à leur segmentation en un ensemble de classes homogènes et introduisons, deux variables représentant les catégories de produits et les groupes d'utilisateurs similaires. Le regroupement des utilisateurs similaires permet au modèle d'utiliser les opinions et les expériences du groupe pour mieux cibler un individu. De même, le recours à la segmentation des produits permet de prédire les intérêts d'un consommateur pour un produit en se basant sur ses intérêts exprimés à l'égard de ceux qui lui sont similaires. Ceci permettrait au modèle de recommander les nouveaux produits qui n'ont pas encore été vus, évalués ou achetés.

Pour simplifier le calcul des probabilités dans le modèle proposé, nous adoptons les hypothèses de dépendance et d'indépendance conditionnelles suivantes entre les variables :

– Les ressources ne dépendent que des catégories auxquelles ils appartiennent.
– Les utilisateurs ne dépendent que des groupes auxquels ils appartiennent.
– Les évaluations ne sont conditionnées que par les groupes d'utilisateurs, les catégories des ressources et du contexte. En effet, l'intérêt d'un consommateur u du groupe G par rapport à un produit x de la catégorie C pourrait être déduit à partir des évaluations attribuées par les individus du groupe G similaires à u aux produits de la catégorie C similaires à x.
– L'acte d'achat d'un produit x dépend de l'intérêt que sa catégorie C a suscité chez le groupe de consommateurs G auquel appartient l'individu considéré. Cependant, nous considérons que l'achat n'est pas une conséquence directe de l'intérêt puisqu'il dépend aussi du contexte du consommateur.

Pour pouvoir prédire les achats en utilisant le modèle proposé, nous nous intéressons à la probabilité qu'un consommateur donné achète un item donné dans un contexte précis. Cette probabilité s'exprime comme étant la probabilité d'achat sachant l'utilisateur, le produit et le contexte. En utilisant les hypothèses de dépendances et d'indépendances conditionnelles entre les variables du modèle, la probabilité recherchée s'écrit comme suit :

$$p(a^+|u,x,q) = \sum_{i=1}^{N_C} \sum_{j=1}^{N_G} \sum_{k=1}^{N_E} p(a^+|e_k,q)p(e_k|c_i,g_j,q)p(c_i|x)p(g_j|u) \quad (1)$$

$p(c_i|x)$ (resp. $p(g_j|u)$) représente le degré d'appartenance d'un item x (resp. un utilisateur u) à la catégorie c_i (resp. au groupe g_j). $p(c_i|x)$ et $p(g_j|u)$ sont reliés à l'étape de segmentation où un individu (utilisateur ou item) peut être affecté à un ou plusieurs "clusters" (groupes ou catégories). Ici, nous avons eu recours aux techniques de classification déterministe telles que *k-means* et probabiliste tel que *c-means*. Dans la classification déterministe, chaque individu est affecté exclusivement à une seule classe alors que dans une classification probabiliste, il pourrait être affecté à plusieurs classes en même temps avec des probabilités d'appartenance différentes. Cependant, nos premières expérimentations avec l'approche probabiliste ont montré de faibles résultats et une difficulté à séparer les classes achat et non-achat. Par contre, le recours à la classification déterministe permet de simplifier la probabilité d'achat puisque les probabilités $p(c_i|x)$ et $p(g_j|u)$ prennent seulement les valeurs un et zéro selon l'appartenance ou non de l'observation à la classe. Dans ce cas, la probabilité est la suivante :

$$p(a^+|u,x,q) = \sum_{k=1}^{N_E} p(a^+|e_k,q)p(e_k|c_n,g_m,q) \tag{2}$$

Le terme $p(e_k|c_i,g_j,q)$ dénote la probabilité d'observer un consommateur du groupe g_j attribuer une note e_k à un item de la catégorie c_i dans le contexte q. Ce terme fait le lien entre notre modèle et les approches de recommandation existantes. En effet, selon les données disponibles concernant l'utilisateur et le produit considérés, le degré d'intérêt prédit par ce terme pourrait être issu d'un filtrage collaboratif, par contenu ou contextuel. Dans les cas où les consommateurs n'ont pas accès à un système de notation, la valeur e_k pourrait être définie comme étant une attitude exprimée par le consommateur par rapport à un produit (p.ex. ajout aux favoris, nombre de consultations, durée de consultation). Lorsqu'elle est exprimée avant l'achat, l'attitude refléterait un certain degré d'intérêt que suscite le produit chez le consommateur. Cependant, lorsque l'attitude est exprimée suite à l'achat du produit, elle est interprétée comme étant une satisfaction ou un remords.

Le terme $p(a^+|e_k,q)$ représente la probabilité d'achat sachant l'intérêt ou la note e_k dans le contexte courant q. En effet, nous estimons que les items potentiels à la recommandation ne sont pas seulement ceux qui pourraient intéresser l'utilisateur (par rapport à la note prédite) mais aussi ceux qu'il pourrait acheter dans un contexte donné. Cette hypothèse est motivée le fait que l'achat ne dépend pas seulement de l'intérêt porté à l'item, mais aussi du contexte courant.

4 Évaluation comparative et résultats expérimentaux

Dans cette section, les détails de l'implantation du modèle de recommandation proposé sont d'abord présentés. Ensuite, les différents résultats expérimentaux sont discutés afin d'évaluer la capacité du modèle à prédire les intérêts et les achats des consommateurs et ainsi mesurer la qualité des recommandations générées.

4.1 Jeu de données

Afin d'évaluer les performances de notre proposition, nous utilisons le jeu de données de MovieLens [1] destiné à la recommandation de film puisqu'il permet d'intégrer la majorité des

1. http ://grouplens.org/datasets/movielens/

variables prévues par notre modèle. D'une part, les comportements des consommateurs sont décrits par les notes qu'ils donnent aux films qu'ils ont vus. D'autre part, le jeu de données comporte un ensemble de variables démographiques décrivant les utilisateurs ainsi qu'un ensemble de descripteurs énumérant les caractéristiques des films. Le jeu de données de MovieLens regroupe les informations suivantes :

- 1700 utilisateurs décrits par leurs âges, sexes et occupations.
- 950 films ayant chacun un identifiant, un titre, une date de sortie et un ensemble de genres parmi les 19 prédéfinis (Action, Aventure, Animation, etc...).
- 100000 notes, représentant chacune l'évaluation donnée par un utilisateur à un film ($1 \leq e_k \leq 5$). Chaque utilisateur du jeu de données a évalué au moins 20 films.

4.2 Algorithmes implémentés

Dans ce travail, quatre approches de recommandation ont été implémentées afin d'évaluer les prédictions de notre modèle. Dans ce cadre, l'objectif est de prédire la note r_{ac} que l'utilisateur actif u_a donnerait à un produit x_c, en se basant sur l'ensemble des utilisateurs $u \in U$ ainsi que sur leurs notes r_{ux} qu'ils auraient donné aux produits disponibles $x \in X$. Plusieurs variantes de chaque approche de recommandation ont été évaluées, chacune adoptant différentes implémentations de ses techniques sous-jacentes telles que les mesures de similarité (p. ex. le cosinus, le cosinus ajusté) et de distance (e.g. distance euclidienne et de Manhattan). Enfin, pour chaque approche, plusieurs techniques de segmentation (p. ex. déterministes et floues) et estimateurs de note ont été évalués afin de déterminer ceux qui sont les plus performants.

4.2.1 Filtrage collaboratif orienté utilisateur (UC-CF)

Afin d'établir une liste de produits recommandés pour un utilisateur actif u_a, cette approche commence par la sélection des utilisateurs $u_i \in S_{u_a}$ ayant les mêmes avis que celui-ci en comparant les notes qu'ils avaient donné au mêmes produits. Ensuite, la note r_{ac} que pourrait donner l'utilisateur courant u_a à un produit candidat x_c, est déduite par l'agrégation des notes r_{ic} affectées à x_c par le voisinage d'utilisateurs similaires S_{u_a} (Resnick *et al.*, 1994).

Les estimateurs suivants ont été implémentées et intégrées aux variantes évaluées de UC-CF pour l'estimation de la note r_{ac} que pourrait données un utilisateur u_a à un produit x_c :

- La moyenne : pour un utilisateur donné, l'estimation de la note qu'il pourrait donner à un produit x_c est la moyenne de celles qui lui ont été affectées par les utilisateurs qui lui sont similaires (i.e. $u_i \in S_{u_a}$).
- Moyenne pondérée : les notes sont pondérées proportionnellement au degré de similarité de leurs auteurs à l'utilisateur actif.

4.2.2 Filtrage collaboratif orienté item (IC-CF)

Pour prédire la note que donnerait l'utilisateur actif u_a à un produit x_c, l'approche procède par aggrégation des notes que u_a aurait affectée à des produits qui sont notés similairement à celui-ci. Pour celà, l'algorithme calcule $ISim(x_c, x_i) = ISim(\vec{r_c}, \vec{r_i})$, les similarités entre le produit candidat x_c et tout autre produit x_i en se basant sur les notes $\vec{r_c}$ et $\vec{r_i}$ que ces derniers ont eu de la

part des mêmes utilisateurs. Ensuite, la note estimée r_{ac} est calculée par l'agrégation des notes que u_a aurait données aux produits $x_i \in S_{x_c}$ les plus similaires à x_c (Sarwar *et al.*, 2001).

Les estimateurs de notes suivants ont été implémentés pour les différentes variantes du filtrage collaboratif orienté item :

- La moyenne : la note estimée est la moyenne des notes que u_a aurait affectées aux produits du voisinage de x_c.
- Moyenne pondérée : les notes affectées par u_a aux produits $x_i \in S_{x_c}$ sont pondérées proportionnellement à leurs similarité $ISIM(x_c, x_i)$ au produit candidat x_c.

4.2.3 Filtrage basé sur le contenu (CBF)

Cette approche part du calcul des similarité $ISim_{x_c,x_i}$ entre le produit candidat x_c et les autres x_i en se basant sur leurs descripteurs $\vec{f}_i = \{f_{i,1}, f_{i,2}, f_{i,3}, \dots\}$. Ensuite, la note prédite r_{ac} est estimée similairement au filtrage collaboratif orienté item en utilisant la moyenne et la moyenne pondérée comme estimateurs (Pazzani, 1999).

4.2.4 Filtrage démographique (DF)

Le filtrage démographique implémenté utilise à la fois les similarités entre les utilisateurs et les items afin de prédire les notes. L'approche commence par la sélection des voisinages d'utilisateurs S_{u_a} démographiquement similaires à u_i et des ressources S_{x_c} similaires à x_c sur la base de leurs attributs (Pazzani, 1999). Ensuite, la note est estimée par l'agrégation des notes attribuées par les membres de S_{u_a} aux éléments de S_{x_c} moyennant un des estimateurs suivants :

- Moyenne : si N est le nombre des notes $r_{i,j}$ tel que $u_i \in S_{u_a}$ et $x_j \in S_{x_c}$, la note estimée est calculée comme suit :

$$\hat{r}_{ac} = \frac{1}{N} \sum_{u_i \in S_{u_a}} \sum_{x_j \in S_{x_c}} r_{i,j} \tag{3}$$

- Moyenne pondérée : les notes sont pondérées par $USim(u_a, u_i)$, la similarité de l'utilisateur u_i à l'utilisateur actif u_a et/ou $ISim(x_c, x_j)$, la similarité d'un produit x_j par rapport au candidat x_c. l'estimateur est formalisé comme suit :

$$\hat{r}_{ac} = \frac{\sum_{u_i \in S_{u_a}} \sum_{x_j \in S_{x_c}} USim(u_a, u_i) \cdot ISim(x_c, x_j) \cdot r_{i,j}}{\sum_{u_i \in S_{u_a}} \sum_{x_j \in S_{x_c}} USim(u_a, u_i) \cdot ISim(x_c, x_j)} \tag{4}$$

4.2.5 Modèle de recommandation fréquentiste proposé (FM)

Afin d'adapter le modèle de recommandation proposé au jeu de données utilisé, les variables contexte et achat non fournies sont omises. Par conséquent, l'objectif du modèle adapté est de calculer la probabilité qu'une note de valeur e_k soit affectée par l'utilisateur u à un produit x afin de déterminer son intérêt à celui-ci. Cette probabilité s'écrit comme suit :

$$p(e_k|u,x,q) = p(e_k|g_u, c_x, q)p(g_u|u)p(c_x|x) \tag{5}$$

Afin de générer des recommandations, l'algorithme commence par la segmentation des utilisateurs et des produits. Dans ce travail, l'étape de segmentation a été effectuée par les algorithmes de Kmeans et de maximisation de l'espérance en utilisant les distances euclidienne et de

Manhattan. Pour cela, plusieurs variables ont été recodées en variables binaires tel que l'occupation des utilisateurs et les genres des films. Les premières expérimentations montrent que ces mesures génèrent des classifications similaires et donc des résultats de prédiction très proches. Cependant, les meilleurs résultats ont été obtenus en utilisant la distance euclidienne puisqu'elle prends en considération le caractère ordinal des variables ordinales (p.ex. l'âge de l'utilisateur et la date de sortie d'un film). Par ailleurs, plusieurs méthodologies ont été employées.

1. Segmentation par les caractéristiques intrinsèques : la segmentation se base sur les attributs démographiques pour les utilisateurs et sur les caractéristiques pour les produits.
2. Segmentation par les notes : la segmentation des utilisateurs est effectuée sur la base de la similarité de leurs notes. Cette approche est similaire à celle du filtrage collaboratif orienté utilisateur. Analogiquement, les produits sont segmentés sur la base de la similarité des notes qu'ils ont obtenues de la part des mêmes utilisateurs. Cette approche est analogique à celle utilisée dans le filtrage collaboratif orienté item.
3. Segmentation mixte : la segmentation se base à la fois sur les caractéristiques intrinsèques et les notes. Ceci permet de rassembler dans les mêmes groupes les utilisateurs appartenant aux mêmes classes démographiques et ayant les mêmes centres d'intérêt. De même, les catégories de produits rassembleraient des produits ayant des caractéristiques similaires et/ou appréciées par les mêmes utilisateurs.

Afin d'estimer la note $\hat{r}_{ux}$ affectée par un utilisateur u à un produit x, la probabilité $p(e_k|u,x,q)$ est utilisée pour calculer l'espérance de la variable note tel que :

$$\hat{r}_{ux} = E[p(e|u,x,q)] = \sum_k e_k \times p(e_k|u,x,q) \tag{6}$$

4.3 Principaux résultats expérimentaux

La figure 1 présente la distribution des erreurs de prédiction des notes des utilisateurs pour les meilleures variantes des approches étudiées. Les valeurs d'erreur obtenues ainsi sont résumées dans des boites à moustache afin d'étudier la variabilité de la qualité de chacune des approches. Les boites représentent ainsi les valeurs d'erreur de prédiction obtenues entre le premier et le troisième quartile, tandis que la ligne interne représente la valeur médiane de l'erreur. Les valeurs d'erreur extrêmes sont représentées par des points à l'extérieur des boites.

Le tableau 1 présente les résultats expérimentaux obtenus par validation croisée (2 échantillonnages), pour chacune des approches étudiées en termes de précision, de rappel, d'erreur absolue moyenne (MAE) et de racine de l'erreur quadratique moyenne (RMSE). Les mesures de précision et de rappel sont calculées à partir des notes estimées par chaque approche, en considérant chacune des valeurs possibles de cette variable (entre 1 et 5) comme étant une classe à prédire. Le tableau 1 montre que les meilleurs résultats sont obtenus par le modèle proposé ainsi que par les approches de filtrage collaboratif.

Les approches de filtrage par contenu (CBF) et collaboratif orienté item (IC-CF) présentent les résultats les plus proches des notres par rapport à l'erreur absolue moyenne. Cependant, leur pertinence en termes de rappel et de précision de ces approches varie largement en fonction de la taille du voisinage de similarité utilisée lors de la détermination des produits similaires au produit candidat à la recommandation.

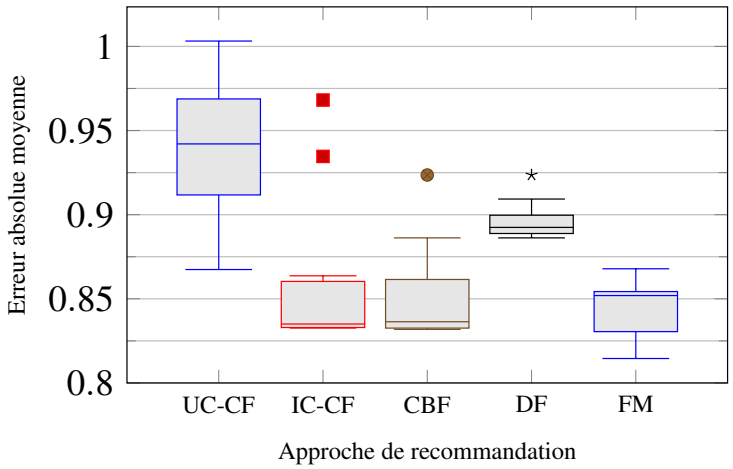


FIGURE 1 – Résultats comparatifs (erreurs de prédiction)

TABLE 1 – Résultats expérimentaux

	DF	CBF	IC-CF	UC-CF	FM
Précision	20,39%	23,85%	27,71%	28,47%	28,61%
Rappel	17,07%	29,11%	44,96%	43,21%	46,39%
MAE	0,8862	0,8318	0,8325	0,8674	0,8145
RMSE	1,1225	1,0446	1,0432	1,0981	1,0539

Le filtrage basé sur le contenu (CBF) effectue ses recommandations en tirant profit des similarités entre les produits. Cependant, la mesure de similarité ainsi que l'estimateur de note employés ont moins d'influence sur la qualité des recommandations que les données utilisées. En effet, lorsque les entités du domaine d'application sont homogènes et font partie du même concept sémantique, cette approche est capable de générer des recommandations pertinentes puisqu'elle favorise les contenus similaires à ceux déjà appréciés par l'utilisateur. Cependant, lorsque les ressources disponibles appartiennent à des catégories sémantiquement différentes, cette approche est incapable de recommander des produits appartenant à des catégories inconnues par l'utilisateur. En effet, les produits de différentes catégories ne sont pas comparables à cause de leurs différents descripteurs et limitent ainsi l'applicabilité des mesures de similarité.

Les expérimentations montrent que le filtrage collaboratif orienté utilisateur (UC-CF) est le plus sensible à la configuration. Sur le jeu de données de MovieLens, les meilleurs résultats ont été obtenus par l'estimateur de note basé sur la moyenne pondérée. La qualité de cet estimateur s'explique par le recours à la pondération des notes des utilisateurs par rapport à leurs similarités. Ces pondérations donnent ainsi plus d'importance aux avis des utilisateurs qui ont plus de similarité démographique et/ou comportementale avec l'utilisateur actif.

Le filtrage démographique (DF) donne des résultats homogènes indépendamment de la mesure de similarité et de l'estimateur de note employés. Cependant, la qualité de ses prédictions dépend des tailles des voisinages d'utilisateurs et de produits utilisés qui nécessitent une phase de configuration afin de maximiser la qualité et les performances de l'approche.

Les expérimentations présentées montrent que le modèle de recommandation proposé per-

met de décrire et d'anticiper les comportements des consommateurs. En effet, en unifiant et en généralisant les idées directrices des approches de recommandation existantes, notre modèle est capable de prédire et de quantifier les intérêts des consommateurs, qu'ils soient influencés par la démographie ou par les caractéristiques des produits. Dans ce modèle, le nombre de groupes d'utilisateurs N_G et de catégories de produits N_C influencent la qualité des recommandations en termes de rappel et de précision. En effet, sous-estimer N_g ou N_c peut conduire à une perte de précision, puisque les consommateurs (ou les produits) au sein du même groupe deviennent moins similaires. De même, lorsque le nombre des classes est surestimé, le rappel du modèle peut diminuer puisque les individus similaires (produits ou clients) peuvent être affectés à différents groupes et ne serait donc pas pris en compte lors de la recommandation. Dans les expérimentations présentées, N_g ou N_c ont été déterminés de manière empirique guidée par la visualisation des distributions des utilisateurs et des films. Par ailleurs, le seuil de probabilité au-dessus duquel un produit est recommandé pourrait être ajusté de manière à trouver un compromis entre l'exactitude des recommandations et leur diversité. En effet, l'augmentation de ce seuil est adaptée aux utilisateurs ayant des besoins spécifiques puisqu'elle conduit à moins de recommandations, mais avec une précision élevée. Par contre, le seuil peut être réduit pour promouvoir la diversité des recommandations pour les consommateurs impulsifs et les clients errants n'ayant pas de produits spécifiques à acheter.

5 Conclusion et perspectives

Le travail présenté dans cet article a pour objectif de proposer un modèle de recommandation unifié capable de prédire les intérêts et les achats des consommateurs. Le modèle proposé se base sur un ensemble de facteurs et d'hypothèses de dépendance afin de prédire de manière probabiliste les comportements de consommation. Les principales contributions de ce travail proviennent de (1) la modélisation des comportements de consommation, (2) de la généralisation des idées directrices des principales approches de recommandation existantes et enfin (3) de la formalisation statistique du modèle et de la méthodologie d'inférence des recommandations. L'expérimentation comparative réalisée sur des données réelles a permis de positionner notre proposition par rapport aux principales approches existantes et de valider son apport.

Comme perspective de ce travail, nous considérons l'intégration de la description visuelle des ressources dans le modèle de recommandation proposé afin de mieux couvrir les facteurs qui influenceraient les choix des consommateurs. Ceci nécessite auparavant l'étude de son influence et l'évaluation de son apport au modèle de recommandation proposé.

Références

AGRAWAL R. & SRIKANT R. (1994). Fast algorithms for mining association rules in large databases. In J. B. BOCCA, M. JARKE & C. ZANIOLO, Eds., *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, p. 487–499 : Morgan Kaufmann.

BAAZAOUI H., HADDAD M. R. & GHEZALA H. B. (2014). A personalised semantic and spatial information retrieval system based on user's modelling and accessibility measure. *International Journal of Multicriteria Decision Making*, **4**(2), 183–200.

BALABANOVIĆ M. & SHOHAM Y. (1997). Fab : content-based, collaborative recommendation. *Communications of the ACM*, **40**(3), 66–72.

BOUTEMEDJET S. & ZIOU D. (2008). A graphical model for context-aware visual content recommendation. *IEEE Transactions on Multimedia*, **10**(1), 52–62.

BURKE R. (2000). Knowledge-based recommender systems. In *ENCYCLOPEDIA OF LIBRARY AND INFORMATION SYSTEMS*, p. 2000 : Marcel Dekker.

BURKE R. D. (2007). Hybrid web recommender systems. In P. BRUSILOVSKY, A. KOBSA & W. NEJDL, Eds., *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, p. 377–408 : Springer.

DESHPANDE M. & KARYPIS G. (2004). Item-based top-*N* recommendation algorithms. *ACM Transactions on Information Systems*, **22**(1), 143–177.

GOLDBERG D., NICHOLS D., OKI B. M. & TERRY D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, **35**(12), 61–70.

HAWKINS D. I., BEST R. J. & CONEY K. A. (2003). *Consumer Behavior : Building Marketing Strategy*. McGraw-Hill/Irwin, 9 edition.

HSIEN LIAO S., HUI CHU P., JU CHEN Y. & CHANG C.-C. (2012). Mining customer knowledge for exploring online group buying behavior. *Expert Systems with Applications*, **39**(3), 3708 – 3716.

IYENGAR S. S. & LEPPER M. R. (2000). When choice is demotivating : can one desire too much of a good thing ? *Journal of Personality and Social Psychology*, **79**(6), 995–1006.

JACOBY J., SPELLER D. E. & BERNING C. A. K. (1974). Brand choice behavior as a function of information load : Replication and extension. *Journal of Consumer Research : An Interdisciplinary Quarterly*, **1**(1), 33–42.

JONES G. J. F. (2005). Challenges and opportunities of context-aware information access. In *UDM*, p. 53–62 : IEEE Computer Society.

KRULWICH B. (1997). Lifestyle finder : Intelligent user profiling using large-scale demographic data. *AI Magazine*, **18**(2), 37–45.

LIN W., ALVAREZ S. A. & RUIZ C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Min. Knowl. Discov*, **6**(1), 83–105.

LINDEN G., SMITH B. & YORK J. (2003). Amazon.com recommendations : Item-to-item collaborative filtering. *IEEE Internet Computing*, **7**(1), 76–80.

MOONEY R. J. & ROY L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, p. 195–204. New York : ACM Press.

PAZZANI M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev*, **13**(5-6), 393–408.

POUSSEVIN M., GUARDIA-SEBAOUN E., GUIGUE V. & GALLINARI P. (2014). Recommandation par combinaison de filtrage collaboratif et d'analyse de sentiments. In *CORIA-CIFED*, p. 27–42.

RESNICK P., IACOVOU N., SUCHAK M., BERGSTORM P. & RIEDL J. (1994). Grouplens : An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, p. 175–186 : ACM.

SARWAR B. M., KARYPIS G., KONSTAN J. A. & RIEDL J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW*, p. 285–295.

SCHWARTZ B. (2005). *The paradox of choice : Why more is less*. Harper Perennial.

WOERNDL W. & GROH G. (2007). Utilizing physical and social context to improve recommender systems. In *Web Intelligence/IAT Workshops*, p. 123–128 : IEEE.

ZHANG Y., QI J., SHU H. & CAO J. (2007). Personalized product recommendation based on customer value hierarchy. In *SMC*, p. 3250–3254 : IEEE.

Approche de découverte de nouvelles catégories dans un wiki sémantique basée sur les motifs fréquents

Yaya TRAORE[1,3], Sadouanouan MALO[2], Cheikh Talibouya DIOP[3], Moussa LO[3], OUARO Stanislas[1]

[1] Université de Ouagadougou ,
Ouagadougou – BP 7021, Burkina Faso
{yaytra,ouaro}@yahoo.fr
[2] Université Polytechnique de Bobo Dioulasso,
Bobo-Dioulasso – BP 1091, Burkina Faso
sadouanouan@yahoo.fr
[3] Université Gaston Berger de Saint -Louis,
Saint-Louis – BP 234, Sénégal
{cheikh-talibouya.diop,moussa.lo}@ugb.edu.sn

Résumé : Dans cet article, nous proposons une approche de découverte de nouvelles catégories potentiellement utiles dans un wiki sémantique. Les pages du wiki sont sémantiquement annotées et des tags (mots clés) peuvent être associés librement à celles-ci. Les pages sont créées par les utilisateurs autorisés à partager des informations sur le wiki. Les catégories permettent d'organiser les liens entre les pages dans le wiki. Elles sont créées par les experts. Notre contribution dans ce papier consiste à extraire parmi les tags qui sont associés librement aux pages, les motifs fréquents de tags qui sont identifiés comme de nouvelles catégories utiles qui guideront l'expert dans la création ou la modification de catégories dans le wiki. Nous utilisons l'ontologie associée au wiki pour bénéficier de plus d'informations structurées afin de sélectionner les tags de la fouille dans le prétraitement et d'éliminer certains motifs de tags de l'analyse dans la phase de fouille.

Mots-clés : Wiki sémantique, Ontologie, Motifs fréquents

1 Introduction

Nos travaux rentrent dans le cadre du projet[1] « Mise en place d'une plateforme web social et sémantique pour le partage de connaissances des communautés ouest-africaines» qui répond à un besoin de disposer d'un cadre de partage de connaissances sur les communautés ouest africaines en s'appuyant sur les technologies du web social et sémantique. Il s'agit de s'appuyer sur les méthodes de l'ingénierie de la connaissance et, en particulier sur les technologies du Web sémantique pour proposer des solutions de partage de connaissances à nos communautés. C'est ainsi qu'en vue de la mise en place d'une plateforme web social et sémantique, un wiki sémantique, est développé autour du moteur Semantic MediaWiki (Krötzsch et al., 2006). Un état de l'art sur les wikis sémantiques est disponible dans (Buffa et al., 2007) et (Meilender, 2013). La plateforme est organisée autour (i) des pages de catégories qui servent à l'organisation des informations dans le wiki ;(ii) des pages de propriétés qui servent à préciser les liens sémantiques qu'il y a entre les informations du wiki ; (iii) des pages (dites normales) qui sont les informations qu'on veut présenter sur le wiki. Les pages de catégories et de propriétés sont créés par l'expert du domaine. Les pages normales sont créées

1. http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/ED/pdf/SenegalGastonBergerFR.pdf

par les utilisateurs autorisés à partager des connaissances sur le wiki. La plateforme propose à tous ses utilisateurs un champ (Figure 1) qui permet d'associer des tags (mots clés) à chaque page normale à la création. La quantité immense de tags stockés sur les pages cache plusieurs connaissances qu'il faut extraire pour réorganiser les liens entre les pages. Cela suppose alors une maintenance de cette plateforme par les experts du domaine. Cette maintenance consiste en l'annotation de nouvelles ressources, l'organisation des liens entre les ressources existantes. Pour guider l'expert dans cette maintenance, nous proposons dans ce papier une approche de découverte de nouvelles catégories utiles à partir des tags stockés sur les pages normales du wiki.

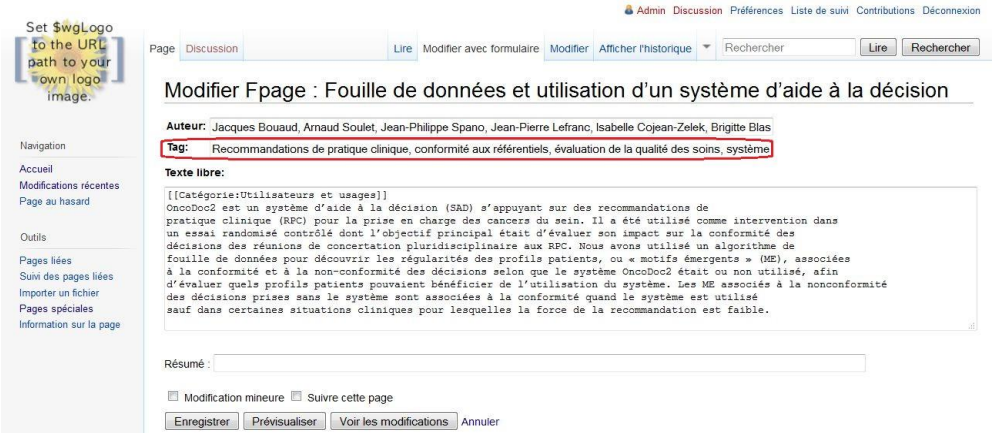


FIGURE 1 – Exemple de page tagguée.

Nous exportons l'ensemble des pages du wiki en RDF pour créer une base de connaissances du wiki. Cette base de connaissances est utilisée dans le processus de fouille pour bénéficier de plus d'informations structurées. La Figure 2 donne un extrait de la base de connaissances d'un wiki avec l'éditeur Protégé. Les concepts représentent les catégories du wiki, les relations représentent les propriétés du wiki et les instances représentent les pages du wiki. Les tags stockés sur les pages du wiki sont représentés par les valeurs des propriétés associées aux instances. SWIVT ontologie **(**Krözsch et al., 2012**)** fournit une base pour l'interprétation de données sémantiques exportées par Semantic MediaWiki.

Dans la suite de cet article, nous présentons à la section 2 les définitions et notations qui seront utiles dans l'article. La section 3 présente les travaux liés à notre approche. Nous développons notre approche dans la section 4. Nous terminons par une conclusion et des perspectives.

2 Définitions et notations

Dans cette section, nous définissons les différentes notions utilisées dans le reste de l'article.

Contexte d'extraction : Un contexte d'extraction est un triplet $CE= (P,T,R)$ où P représente l'ensemble fini des pages du wiki sémantique, T l'ensemble fini des tags, R une relation binaire entre T et P tel que R(p, t) = 1 si la page p ϵ P est tagguée par t ϵ T sinon 0. Nous définissons la fonction g qui permet d'avoir l'ensemble des pages associées à un tag comme suit : g : T → P tel que pour t ϵ T, g (t) = {p/ p ϵ P}.

Motif fréquent de tags : Un motif de tags est un sous ensemble de tags. Le support d'un motif de tags est la proportion de pages annotées par ce sous ensemble de motif. Un motif est fréquent si son support est supérieur à un seuil fixé σ. Soit $T_1 \subseteq T$ un motif de tags. Notons $Supp(T_1)$ (1) son support : T_1 est fréquent si $Supp(T_1) >= \sigma$.

$$Supp(T_1) = \frac{|g(T1)|}{|P|} \qquad (1)$$

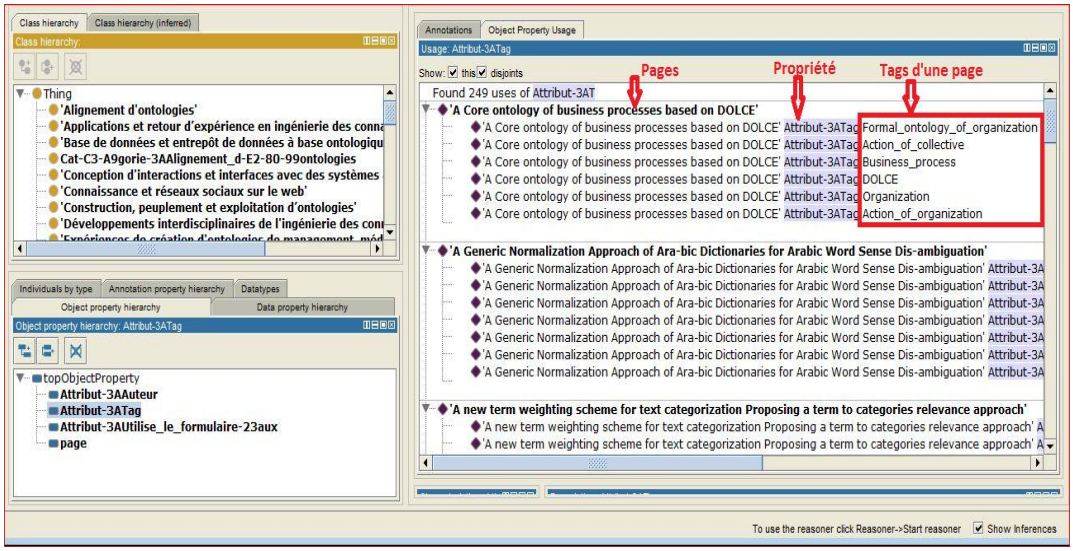


FIGURE 2 – Extrait de la base de connaissances du wiki.

Nouvelle catégorie : Dans notre contexte, une nouvelle catégorie est un motif fréquent de tags utilisés sur les pages du wiki et qui n'est pas dans la liste des catégories existantes de celui-ci. Soit C l'ensemble des catégories du wiki et f un motif fréquent de tag, f est une nouvelle catégorie si et seulement si elle vérifie la propriété suivante : $\forall t \in f \implies t \notin C$.

Distance sémantique de (Cilibrasi et al., 2006) : Pendant la fouille, certains tags peuvent ne pas être retenus à l'étape de prétraitement alors qu'ils peuvent avoir des corrélations avec les tags du domaine de la fouille. Nous utilisons la distance sémantique de (Cilibrasi et al., 2006) pour les sélectionner. Dans notre étude nous adaptons la distance sémantique proposée par (Cilibrasi et al., 2006) **(**notée DCV**)** entre deux termes (tags) t_1 et t_2 ainsi qu'il suit:

$$\boldsymbol{DCV}(t_1, t_2) = \frac{\max\{\log(fr(t_1)), \log(fr(t_2))\} - \log(fr(t_1, t_2))}{\log(M) - \min\{\log(fr(t_1)), \log(fr(t_2))\}} \qquad (2)$$

M désigne le nombre de page du wiki, $fr(t_1)$ la fréquence de t_1, $fr(t_2)$ la fréquence de t_2 et $fr(t_1, t_2)$ la fréquence de t_1 et t_2. Cette mesure désigne une mesure de la proximité sémantique entre t_1 et t_2 et varie entre 0 et 1. 0 indique que t_1 est « sémantiquement » proche de t_2 et 1 le contraire. Deux tags t_1 et t_2 cachent donc une relation si $\boldsymbol{DCV}$ (t_1, t_2)=0.

3 Travaux existants

Un certain nombre de travaux utilisant les ontologies dans le processus d'extraction de connaissances à partir des données (ECD) existent. (Euler et al., 2004) utilisent l'ontologie pendant la phase de prétraitement, (Brisson et al., 2006) utilisent l'ontologie dans le prétraitement et le post-traitement, (Marinica et al., 2010) l'utilisent dans le post-traitement pour réduire la quantité de règles extraites à partir des schémas de règles. Dans ces approches la disponibilité d'un expert du domaine est nécessaire pour valider les correspondances entre les concepts de l'ontologie et les sous-ensembles d'enregistrements de la base de données, ce qui n'est pas toujours possible. Dans notre étude, nous utilisons un wiki sémantique qui stocke une quantité immense de tags sur les pages. Dans ce contexte, les travaux de (Tobias et al., 2011) proposent une extension de Semantic Mediawiki pour extraire des motifs fréquents de nuages de tags à partir d'une propriété, mais cette extension ne permet pas de détecter de nouvelles catégories. Notre objectif est de fouiller dans les tags stockés sur les pages pour identifier de nouvelles catégories. Nous nous inspirons des travaux de (Yaya et al.,2014) sur

l'apport de l'ontologie dans le prétraitement de l'extraction des connaissances à partir d'un wiki sémantique, des travaux de (Christian et al., 2010) sur l'extraction des catégories, propriétés pour créer une ontologie et l'enrichir au fur et à mesure à partir d'un wiki, les travaux de (Jian et al., 2006), (Julien et al., 2010), (Fleischhacker et al., 2012) sur la fouille de données utilisant une base de données RDF. Sur la base de ces travaux, nous proposons d'utiliser la base de connaissances obtenue à partir du wiki sémantique pour bénéficier de plus d'informations structurées pour sélectionner automatiquement les tags de la fouille sur la base d'une proximité sémantique avec l'objectif de la fouille. Certains tags *a priori* rejetés peuvent avoir des corrélations avec les tags retenus que nous détectons grâce à la distance sémantique de (Cilibrasi et al., 2006). Pour construire le contexte d'extraction qui sera le point d'entrée de l'étape de la fouille, nous utilisons la relation entre les pages et les tags. Dans la phase de fouille en s'inspirant de (Antunes, 2007) nous utilisons la structure conceptuelle de l'ontologie comme une condition d'élagage pour enlever certains motifs de tags de l'analyse.

4 Description de l'approche

Notre approche (Figure 3) se situe dans le cadre global de l'extraction de connaissances à partir de données (ECD**)** (Fayyad et al., 1996) et se déroule en deux grandes étapes ci-dessous expliquées :

- Etape 1: Construction du contexte d'extraction (Algorithme 1) : (1) exportation de toutes les pages du wiki sémantique en RDF; (2) définition et expression de l'objectif de fouille par un mot clé ;(3) sélection des tags proches de l'objectif de la fouille en utilisant l'ontologie et ceux corrélés grâce à la distance DCV. (4) l'ensemble des tags du domaine et ceux corrélés forment les tags sélectionnés pour le CE; (5) construction du contexte d'extraction : sélection des pages de chaque tag et définition de la relation.
- Etape 2: Algorithme de découverte (Algorithme 2) : cette étape est basée sur l'algorithme Apriori (Agrawal ,1994**) :** (6) sélection de motifs de tags candidats et élagage en utilisant la structure conceptuelle de l'ontologie du wiki pour enlever des motifs de tags candidats qui sont sémantiquement proches des catégories existantes dans le wiki. (7) calcul des motifs fréquents de tags qui sont identifiés comme de nouvelles catégories utiles.

L'innovation de notre approche est la sélection non supervisée des tags du contexte d'extraction sans l'aide de l'expert et l'introduction d'une contrainte d'élagage basée sur la structure conceptuelle de l'ontologie associée au wiki dans algorithme Apriori (Agrawal ,1994**)**. A notre connaissance c'est la première fois que notre approche est utilisée dans la phase de prétraitement et de fouille de l'ECD à partir d'un wiki sémantique.

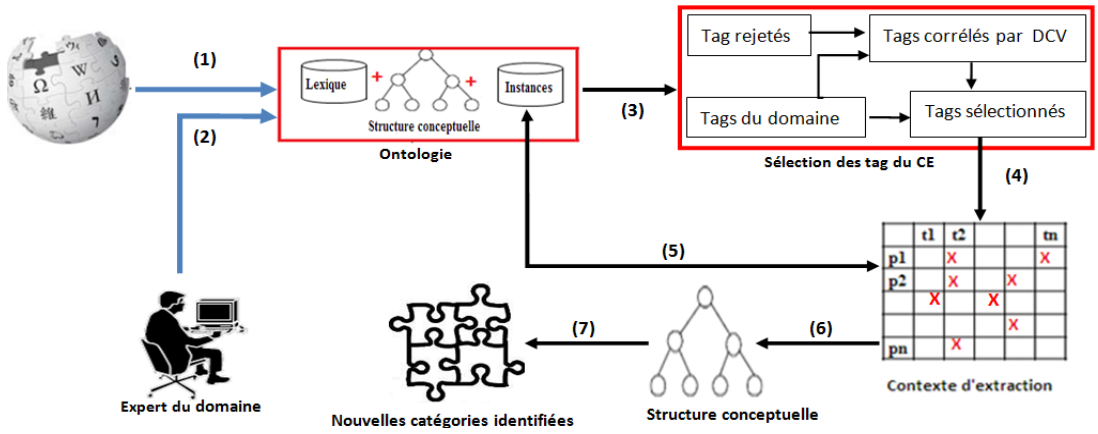


FIGURE 3 – Approche de découverte de nouvelles catégories dans un wiki sémantique.

Algorithme 1 : Algorithme de construction du CE

Entrée : M=mots clés de la fouille, BC=Base de connaissances
Sortie : CE: contexte d'extraction
Début

1. **/* Extraction de tags ,de pages par de requêtes sparql */**
2. T=ensemble des tags extraits de BC
3. P=ensemble des pages extraites de BC
4. T_O=ensemble des tags de BC similaire au mots clés M
5. T_R= T \ T_O **// T_R =Tags rejetés (non proches du domaine M)**
6. T_S = T_O **// T_s=Tags du contexte d'extraction CE**
7. **// application de la distance sémantique de Cilibrasi**
8. Pour chaque tag $t_r \epsilon$ T_R faire
9. Pour chaque tag $t_o \epsilon$ T_O faire
10. Si (DCV(t_r,,t_o) = 0) alors
11. T_S = $T_S \cup t_r$
12. Finsi
13. Finpour
14. Finpour
15. **/* Construction du contexte d'extraction CE */**
16. Pour chaque page p de P faire
17. Pour chaque tag t de Ts faire
18. Si (p ϵ g(t)) alors
19. CE(p, t)=1
20. Sinon
21. CE(p, t)=0
22. FinSi
23. Finpour
24. Finpour
25. Retourner CE

Fin

Algorithme 2 : Algorithme de découverte de nouvelles catégories

Entrée :CE:contexte d'extraction (Base de transaction),S:structure conceptuelle de l'ontologie,minsup:seuil minimum de support
Sortie : F:motifs fréquents de tags
Début

1. L_1=ensemble des 1-itemsets fréquents
2. K=2
3. Tant que($L_{K-1} \neq \emptyset$) faire
4. **// Phase de génération des candidats**
5. C_K = ensemble des K-itemsets C tels que : C = F1 $\cup$ F2 où F1 et F2 sont éléments de L_{k-1} et F1$\cap$F2 comporte (K-2) éléments
6. **//Phase d'élagage**
7. Supprimer de C_K tout candidat C tel qu'il existe un sous-ensemble de C de (K-1) éléments non présent dans L_{K-1}
8. **//Phase d'élagage sémantique**
9. Supprimer de C_K tout candidat C tel qu'il existe un élément de C qui est sémantiquement proche d'un concept de la structure conceptuelle S
10. **// Phase d'évaluation des candidats**
11. Calculer le support de chaque candidat C dans C_K
12. L_K ={C ϵ C_K / support(C) >= minsup}
13. K=K+1
14. Fintanque
15. Retourner F=$\cup L_K$

Fin

5 Conclusion

Cet article a présenté une approche de découverte de nouvelles catégories utiles dans un wiki sémantique en utilisant la base de connaissances à base ontologique du wiki dans le processus. De nombreuses perspectives s'offrent à la suite de nos travaux. La première d'entre elles est d'évaluer notre approche sur un wiki sémantique avec un volume important d'annotations afin d'analyser plus en détail l'impact de notre proposition. Les autres perspectives seront consacrées au développement des techniques exploitant les résultats de l'algorithme 2 pour réorganiser les pages du wiki.

Références

AGRAWAL R. AND SRIKANT. R.(1994) Fast algorithms for mining association rules in large databases, *Proc. VLDB conf., pp 478-499,* September 1994.

ANTUNES, C. (2007). ONTO4AR : A Framework for Mining Association Rules. In Proceedings of the International Workshop on Constraint-Based Mining and Learning (CMILE ″UPKDD), Warsaw, Poland, pp. 37–48.

BRISSON, L. ET M. COLLARD. (2008). An Ontology Driven Data Mining Process. In Proceedings of the 10th International Conference on Enterprise Information Systems, Barcelona, Spain,pp. 54–61.

BUFFA M., GANDON F., ERETEO G. (2007). Wiki et web sémantique. *In F. Trichet (Ed.),* IC'2007 : 18eJournées Francophones d'Ingénierie des connaissances.

CHRISTIAN SCHÖNBERG, HELMUTH PREE,BURKHARD FREITAG (2010). Rich ontology Extraction and Wikipedia Expansion Using Language Resources, Proc. of the 11th int.Conf. on Web-Age Information Management ,Jiuzhaigou,China, LNCS,volume 6184.

CILIBRASI R., VITANYI P. (2006). Similarity of Objects and the Meaning of Words. In Proceedings of the Third international conference on Theory and Applications of Models of Computatio TAMC'06, Beijing, China, pages 21-45.

DBPEDIA EN FRANÇAIS. (2014). Dernière consultation : Décembre 2014. http://fr.dbpedia.org/

EULER T. ET M. SCHOLZ (2004). Using Ontologies in a KDD Workbench. In In Workshop on Knowledge Discovery and Ontologies at ECML/PKDD, Pisa, Italy, pp. 103–108.

FAYYAD, U. M., PIATETSKY-SHAPIRO, G., & SMYTH, P. (1996a). The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM, 39(11), pp. 27-34.

FLEISCHHACKER D. VÖLKER J., AND STUCKENSCHMIDT H. (2012). Mining rdf data for property axioms. In OTM Conferences (2), pages 718-735.

JIANG T. ET TAN A. (2006). Mining rdf metadata for generalized association rules: knowledge discovery in the semantic web era. In WWW, pages 951-952.

JULIEN RABATEL, SANDRA BRINGAY AND PASCAL PONCELET. (2010). Contextual sequential pattern mining. In (ICDMW), ICDM, pages 981-988. IEEE.

KRÖZSCH M. ET VRANDECIC D. (2012). Swivt ontology specification. dernière consultation Janvier 2015, http://semantic-mediawiki.org/swivt/

KRÖTZSCH M., VRANDECIC D., VÖLKEL M. (2006).Semantic Mediawiki. ISWC 2006:5th International Semantic Web Conference,Athens,Ga, USA, November 5-9.

MARINICA, C. ET F. GUILLET. (2010). Knowledge-Based Interactive Postmining of Association Rules Using Ontologies. IEEE Transactions on Knowledge and Data Engineering 22,784–797.

MEILENDER T.,(2013). Un wiki sémantique pour la gestion des connaissances décisionnelles – Application à la cancérologie, Thèse de Doctorat , Université de Lorraine, 2013.

TOBIAS BECK, ANDREAS FAY. (2011). FrequentPattern TagCloud, Semantic MediaWiki Extension, Documentation, University of Heidelberg.

YAYA TRAORE, SADOUANOUAN MALO, CHEIKH TALIBOUYA DIOP, MOUSSA LO, STANISLAS OUARO.(2014). Extraction des connaissances dans un wiki sémantique : apport des ontologies dans le prétraitement,5th Journées Francophones sur les Ontologies (JFO), pp.127-138,14-16 Nov. 2014, Hammamet, Tunisie.

Amélioration continue d'une chaîne de traitement de documents avec l'apprentissage par renforcement

Esther Nicart[1,2], Bruno Zanuttini[2], Bruno Grilhères[1], Patrick Giroux[1]

[1] IPCC, Airbus Defence and Space, France, prenom.nom@airbus.com

[2] MAD, GREYC, Université de Caen, Basse-Normandie, France, prenom.nom@unicaen.fr

Résumé : Nous nous intéressons au problème de l'amélioration continue d'une chaîne de traitement de documents, visant à extraire des événements dans des documents provenant de sources ouvertes. Il s'agit de tirer parti des corrections effectuées par les opérateurs humains pour que la chaîne de traitement apprenne de ses erreurs, et s'améliore de façon générale.

Nous appliquons l'apprentissage par renforcement (en l'occurrence, le *Q-Learning*) à ce problème, où les actions sont les services d'une chaîne de traitement d'extraction de l'information. L'objectif est de profiter du *feedback* utilisateur pour permettre au système d'apprendre la configuration idéale des services (ordonnancement, *gazetteers* et règles d'extraction) en fonction des caractéristiques des documents à traiter (langue, type, etc.). Nous menons de premières expériences avec des données de *feedback* générées automatiquement à partir d'un oracle, et les résultats sont encourageants.

Mots-clés : Intelligence artificielle, Apprentissage par renforcement, Extraction et gestion des connaissances, Interaction homme-machine, Renseignement en sources ouvertes (ROSO)

1 Introduction

Nous nous intéressons au problème générique de l'amélioration continue d'une chaîne de traitement de documents, plus précisément d'extraction d'événements d'intérêt. L'application qui nous intéresse particulièrement est le renseignement à partir de sources ouvertes. Dans cette application, des documents provenant essentiellement du *web* sont fournis en continu à une chaîne de traitement, qui vise à extraire des événements (par exemple, une attaque terroriste) et leurs caractéristiques (date, lieu, acteurs, etc.) et à les intégrer à une base de données.

Dans de telles applications, il est clair que l'extraction ne peut pas être parfaite. On peut ainsi imaginer qu'une dépêche relatant « le *bombardement*, par des ions, d'une *cible* d'or par la physique *atomique* lors d'une *manifestation* pour la fête de la science », puisse induire une chaîne de traitement en erreur et lui fasse insérer dans la base un attentat à l'arme atomique. Par ailleurs, la volonté de traiter des documents provenant du monde entier entraîne le besoin de traiter des documents dans des langues très diverses, pour lesquelles des dictionnaires peuvent être de qualité très variable. Pour toutes ces raisons, le cadre typique implique des opérateurs humains, qui peuvent corriger *a posteriori* les événements placés automatiquement en base de données.

L'application qui nous intéresse utilise la plateforme WebLab (2015) pour le renseignement en sources ouvertes. La chaîne de traitement est définie par des experts, et consiste en un enchaînement figé (mais potentiellement conditionnel) de traitements atomiques, tels que l'extraction de la langue ou du format, la traduction, la détection d'événements en utilisant des mots ou verbes déclencheurs, etc. Aucun mécanisme ne permet d'améliorer la chaîne au fil du temps.

Notre objectif est de combler ce manque, en fournissant un mécanisme d'amélioration continue de la chaîne de traitement, tirant parti des retours (du *feedback*) exprimés implicitement par les opérateurs lorsqu'ils corrigent des événements dans la base de données. Il s'agit de faire en sorte que la chaîne « apprenne de ses erreurs ». Ainsi, le système pourrait apprendre des règles telles que « si le document est en breton, il est préférable de le traduire d'abord en français puis d'extraire des événements, plutôt que d'appliquer directement une phase d'extraction sur le breton », en constatant que des corrections sont souvent apportées sur les événements extraits par la deuxième option.

À moyen terme, nous visons la prise en compte directe du *feedback* de l'utilisateur. Son expertise (le *feedback*) est manifestée par les traces de ses actions (Bratko & Suc, 2003). Ces traces seront captées à travers l'interface graphique qui donne les détails des événements en synthèse. La chaîne de traitement restera alors une « boîte noire » pour l'utilisateur. Il s'agira de retours qualitatifs, en particulier sur les événements extraits (« corrigé », « consulté et non corrigé », « non consulté », etc.).

Nous restreignons ici le cadre, en supposant que le système reçoit un *feedback* quantitatif sur la qualité des extractions. Nous proposons une formalisation du problème en apprentissage par renforcement, et rendons compte de premières expériences très encourageantes. Pour celles-ci, le *feedback* est basé sur une distance entre les événements désiré et extrait et donc, implicitement, sur le temps qui serait nécessaire à la correction des erreurs.

Culotta *et al.* (2006) ont montré l'intérêt de solliciter des corrections à l'utilisateur, et de guider la mise à jour du modèle. Pourtant le modèle pourrait être bon, mais mal appliqué, et les utilisateurs n'ont pas l'expertise pour modifier le système. Chai *et al.* (2009) ont essayé de combler ce manque d'expertise en proposant un langage permettant aux utilisateurs de corriger directement l'application d'extraction de l'information sans l'intervention d'un expert technique, mais cela suppose que l'amélioration s'applique à tous les documents à traiter, c'est-à-dire, dans une chaîne de traitement figée. Or, les documents de renseignement sont hétérogènes, et sont traités dans des grands volumes.

Nous proposons de permettre à l'utilisateur de se distancer complètement du modèle, en offrant un système modulaire [*cf.* Fromherz *et al.* (2003) qui construisent de chaînes de façon adaptative et modulaire pour des photocopieurs Xerox], qui apprend à modifier son propre comportement en temps réel [*cf.* Doucy *et al.* (2008) qui modifient également une chaîne de traitement à la volée], à partir du *feedback* offert par l'utilisateur [*cf.* Dupont *et al.* (2011) qui montrent l'utilisation de RL pour la sélection d'outils de recherche basée sur l'analyse des actions de l'utilisateur], et qui plus est, s'adapte à chaque document source unique.

2 La plateforme WebLab

La chaîne de traitement qui motive notre travail s'appuie sur la plateforme *open-source* WebLab (2015). WebLab intègre des services *web* qui peuvent être interchangés ou permutés afin de créer une chaîne de traitement. Cette chaîne peut ensuite être utilisée pour analyser des documents multimédia *open-source*, et en extraire l'information.

Une chaîne de traitement typique de WebLab (Figure 1) commence par convertir le document source en une ressource XML. Cette ressource est ensuite transmise de service en service. Chaque service analyse le contenu de la ressource telle qu'il la reçoit, et l'enrichit avec des annotations. Enfin, les résultats sont stockés pour consultation par l'utilisateur.

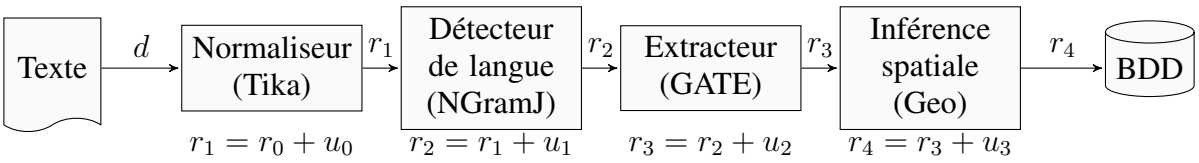


FIGURE 1 – Une chaîne WebLab typique : un document, d, est converti en une ressource XML, r_0, et des annotations, u_j, sont ajoutées par chaque service. Enfin, les résultats sont stockés.

```
1  <resource type="Document" uri="weblab:aaa">
2    <annotation uri="weblab:aaa#a0">
3      <Description about="weblab:aaa">
4        <wp:hasNativeContent resource="file:weblab.content"/>
5        <wp:hasOriginalFileSize>255</wp:hasOriginalFileSize>
6        <dc:source>documents/event.txt</dc:source>
7        <wp:hasOriginalFileName>event.txt</wp:hasOriginalFileName>
8        <dcterms:modified>2015-02-14T19:52:21+0100</dcterms:modified>
9        <wp:hasGatheringDate>2015-02-10T00:11:00+0200</wp:hasGatheringDate>
10     </Description>
11   </annotation>
12   <annotation uri="weblab:aaa#a1">
13     <Description about="weblab:aaa">
14       <wp:isProducedBy resource="weblab:tika"/>
15       <dc:format>text/plain</dc:format>
16     </Description>
17   </annotation>
18   <annotation uri="weblab:aaa#a2">
19     <Description about="weblab:aaa">
20       <wp:isProducedBy resource="weblab:ngramj"/>
21       <dc:language>en</dc:language>
22     </Description>
23   </annotation>
24   <mediaUnit type="wl:Text" uri="weblab:aaa#0">
25     <content>4/29/1971: In a series of two incidents that might have been part of a multiple attack, suspected members of the
       Chicano Liberation Front bombed a Bank of America branch in Los Angeles, California, US. There were no casualties but
       the building sustained $1,600 in damages.</content>
26   </mediaUnit>
27 </resource>
```

FIGURE 2 – Flux XML simplifié d'une ressource WebLab, qui montre les informations extraites telles que le format (ligne 15), la langue (ligne 21) et le contenu original (ligne 25).

Exemple 1
Considérons le document textuel suivant :

> `4/29/1971: In a series of two incidents that might have been part of a multiple attack, suspected members of the Chicano Liberation Front bombed a Bank of America branch in Los Angeles, California, US. There were no casualties but the building sustained $1,600 in damages.`

La ressource XML (simplifiée) de la Figure 2 est produite après le passage du document par le normaliser, *Tika (2015), qui a ajouté l'annotation « text/plain » (ligne 15) et le contenu original, et le détecteur de la langue, NGramJ (2015), qui a ajouté la langue « en » (ligne 21).*

Nous considérons l'utilisation d'une telle chaîne pour l'extraction d'événements d'intérêt pour la veille économique, stratégique, ou militaire. En travaillant avec WebLab, nous nous situons dans la continuité du travail de Serrano (2014), qui propose la définition suivante d'un événement, formalisé dans l'ontologie *WOOKIE*. Notre travail est indépendant de WebLab et *WOOKIE*, et nous aurions pu utiliser une autre définition, par exemple, celle de van Hage *et al.* (2011).

Définition 1 (Un événement)
Un événement E *est un quadruplet* $E =< C, T, S, A >$*, où :*

- $C \subseteq \mathbb{C}$ *est la dimension conceptuelle de* E*, donnée par un ensemble d'atomes pris dans un domaine* $\mathbb{C}$ *commun à tous les événements ;*
- T *est la dimension temporelle de* E*, c'est-à-dire la date à laquelle* E *est survenu (potentiellement ambiguë, telle que « mardi dernier ») ; pour modéliser l'ambiguïté, on prend* $T \subseteq \mathbb{T}$*, où* $\mathbb{T}$ *est l'ensemble des dates ;*
- S *est la dimension spatiale de* E*, potentiellement ambiguë également, avec* $S \subseteq \mathbb{S}$ *;*
- A *est la dimension agentive de* E*, càd l'ensemble des participants impliqués (*$A \subseteq \mathbb{A}$*).*

Si la définition est générale, on s'intéresse dans le cadre de cet article à des domaines précis : $\mathbb{C}$ est un ensemble fixé et fini d'atomes dans *WOOKIE* ; $\mathbb{T}$ est l'ensemble de toutes les « dates » qui peuvent être extraites, par exemple « mardi dernier », « 2001 », « 2001/9/11, 8:46 » ; $\mathbb{S}$ est l'ensemble des entités utilisées par GeoNames (2015) ; enfin, $\mathbb{A}$ est l'ensemble (infini) de tous les participants pouvant être extraits, vus comme des chaînes de caractères.

Exemple 2 (suite de l'exemple 1)
Le document au dessus donnera lieu à l'extraction d'un événement $E =< C, T, S, A >$ *avec* $C = \{$*AttackEvent*, *BombingEvent*$\}$, $T = \{$*4/29/1971*$\}$, $S = \{$*Los Angeles*, *California*, *United States*, *America*$\}$, $A = \{$*Chicano Liberation Front*, *Bank of America*$\}$.

3 Apprentissage par renforcement

Pour atteindre notre objectif, nous appliquons les techniques de l'*apprentissage par renforcement* (*Reinforcement Learning*, RL). Pour une introduction détaillée au sujet, nous renvoyons le lecteur à Sutton & Barto (1998), mais nous en rappelons les grands principes dans cette section.

En RL, l'apprenant reçoit une récompense, basée sur les résultats des actions qu'il a choisies. Plus les résultats sont proches des objectifs, plus la récompense est élevée. Le système essaie de maximiser ces récompenses, typiquement en *exploitant* ce qu'il connaît déjà pour continuer à recevoir de bonnes récompenses, et en *explorant* de nouvelles actions avec l'espoir d'obtenir des récompenses encore plus importantes. Prenons l'exemple d'un robot qui doit naviguer sur une grille. Son objectif est d'atteindre une case spécifique, qui est la seule à donner une récompense. Typiquement, au fil des épisodes, il renforcera la valeur des cases depuis lesquelles il sera arrivé rapidement au but, et apprendra ainsi le chemin idéal.

Le RL est généralement formalisé comme un processus de décision markovien (*Markov Decision Process*, MDP). Un tel processus modélise l'environnement en termes d'*états*, dans lesquels des *actions* sont possibles, qui mènent à d'autres états de manière stochastique. Le fait que l'environnement soit dans un état donné à un certain instant apporte une récompense immédiate à l'agent. L'objectif d'un apprenant est de choisir ses actions de façon à maximiser son espérance de récompenses cumulées, sans connaître, initialement, ni les distributions sur les états résultant d'une action, ni les récompenses associées aux états. Bien entendu, ce cadre générique admet de nombreuses variantes (pour un aperçu récent, voir Szepesvári (2010)).

Définition 2 (MDP)
Un processus de décision markovien *(Puterman, 1994) est un 5-uplet* (S, A, P, R, γ)*, avec*

- S *un ensemble (fini, discret) d'états possibles de l'environnement,*

— *A un ensemble (fini, discret) d'actions (que l'agent peut effectuer),*
— *P un ensemble de distributions $\{P_a(s,\cdot) \mid s \in S, a \in A\}$; $P_a(s,s')$ est la probabilité que l'environnement soit dans l'état s' après que l'agent a effectué l'action a en s,*
— *R une fonction de récompense, que nous supposons définie sur les états ; $R(s)$ est la récompense obtenue par l'agent pour se trouver dans l'état s,*
— *$\gamma \in [0,1]$ un facteur d'atténuation, qui contrôle l'importance des récompenses espérées dans le futur, relativement aux récompenses espérées dans l'immédiat.*

Dans le cadre du RL, l'agent (apprenant) connaît initialement seulement les espaces d'états et d'actions S, A, ainsi que le facteur γ. À tout instant t, il connaît l'état courant s_t de l'environnement, et choisit une action a_t. L'environnement passe dans un état s_{t+1} tiré selon la distribution $P_{a_t}(s_t,\cdot)$, et l'agent est informé de l'état s_{t+1} et de la récompense $r_{t+1} = R(s_{t+1})$. Le processus continue en s_{t+1}. L'agent doit, au fil de ces interactions avec l'environnement, apprendre une série de *politiques* $\pi_0, \pi_1, \ldots, \pi_t, \ldots$, une *politique* $\pi_t : S \to A$ donnant, pour l'instant t, l'action $\pi_t(s)$ à effectuer si l'état courant est $s_t = s$, pour tout $s \in S$. Son objectif est à tout instant de maximiser l'espérance de la récompense cumulée, c'est-à-dire l'espérance de la quantité $\sum_{t'=t}^{\infty} \gamma^t R(s_t)$.

Dans l'exemple du robot, à l'instant t, l'état courant s_t est la case sur laquelle il se trouve, et son action a_t est prise parmi *nord*, *ouest*, etc. La probabilité $P_{a_t}(s_t, s')$ qu'il arrive sur la case s' à l'instant suivant dépend de sa case de départ s_t et de l'action a_t.

Il n'est pas si trivial de définir des états et actions dans une chaîne de traitement, pourtant, il semble naturel d'utiliser le RL sur notre problématique. Les seules informations connues avant de commencer une chaîne de traitement sont les services disponibles, leurs paramètres, et les états potentiels des documents et du système (*cf.* section 2). L'apprenant ne connaît ni la forme, ni le contenu des documents à l'avance, ni si une extraction sera possible, donc ses décisions sont prises dans l'incertain. Nous voulons que le système apprenne une série de *politiques* adaptées aux besoins de l'utilisateur, pour améliorer en continu l'extraction des événements.

De nombreux algorithmes ont été proposés dans la littérature pour les problèmes de RL. Dans cet article, nous utilisons une des approches les plus standards, le *Q-learning* (Watkins, 1989), avec une exploration ϵ-gloutonne. Cette approche consiste à maintenir, pour chaque couple état/action (s,a), une valeur notée $\hat{Q}(s,a)$ qui représente intuitivement l'estimation courante, par l'agent, de l'espérance de récompense s'il se trouve dans s, exécute a, puis suit une *politique* optimale. En résumé, lorsque l'agent est dans l'état s_t, choisit a_t, se retrouve en s_{t+1} et reçoit une récompense r_{t+1}, il met à jour son estimation de la valeur $\hat{Q}(s_t,a_t)$ de la manière suivante : $\hat{Q}(s_t,a_t) \leftarrow (1-\alpha)\hat{Q}(s_t,a_t) + \alpha\gamma(r_{t+1} + \gamma\, max_{a_{t+1}} Q(s_{t+1},a_{t+1}))$ où α, le *taux d'apprentissage*, est un coefficient dans $[0,1]$ qui fixe l'importance de la dernière expérience $(r_{t+1} + \gamma\, max(\ldots))$ par rapport à l'expérience déjà accumulée (l'ancienne valeur de $\hat{Q}(s_t,a_t)$).

Enfin, dans le Q-learning avec exploration ϵ-gloutonne, le *taux d'exploration* $\epsilon \in [0,1]$ règle le dilemme exploitation/exploration de la manière suivante. À chaque pas de temps t, l'agent tire un nombre aléatoire dans $[0,1]$; s'il fait entre 0 et ϵ, alors il choisit une action aléatoirement (il *explore*) ; sinon, l'agent *exploite* et choisit simplement l'action a qui maximise $\hat{Q}(s_t,a)$.

Le *Q-learning* est un algorithme simple, dont les paramètres α et ϵ peuvent être réglés de façon intuitive, et c'est pourquoi nous l'utilisons dans la suite. Toutefois, notre contribution consiste à modéliser le problème de l'amélioration continue d'une chaîne de traitement comme un problème de RL, et tout algorithme pour ce problème pourrait également être utilisé.

4 Amélioration continue via l'apprentissage par renforcement

Dans la pratique, la chaîne de traitement utilisée est complexe. Elle est écrite et calibrée par des experts qui choisissent les services constituant la chaîne, leur ordonnancement et leurs paramètres (par exemple, les *gazetteers* de mots déclencheurs pour les services de détection d'événements). L'ordonnancement peut être conditionnel (par exemple, si le document est en format pdf, passer au service 1 pour le convertir en XML, et au service 2 sinon), mais il est figé.

Malgré l'expertise, il est très difficile d'obtenir une chaîne parfaite, parce que l'utilisation des documents *open source* provenant du *web* apporte des difficultés : leurs format et contenu ne sont pas standards, les pages sources elles-mêmes ne sont pas contrôlables, les urls changent, ou sont piratés, et il y a du « bruit » (publicité par exemple). On observe des erreurs d'extraction pouvant être des événements d'intérêt manqués, des événements mal extraits (les informations non connexes dans la même phrase associées faussement, par exemple).

Par exemple, l'utilisateur voudrait de l'information sur les accords entre pays. Il est impossible de dire avec certitude que le mot « alliance » dans un document y fait référence. Ce n'est qu'après l'extraction de l'événement déclenchée par le mot « alliance » que l'utilisateur se rend compte que la page parle de mariages, par exemple. Même si le document provient d'un journal politique, il se peut que l'on parle d'une coalition entre partis politiques, ou que les filtres de publicité n'aient pas réussi à attraper une vente de bagues. Le mot « union » a pu être utilisé au lieu d'« alliance », et l'événement n'a pas été reconnu. Avec ces incertitudes, les experts qui paramètrent la chaîne essaient d'envisager les situations les plus communes. Il est inconcevable qu'ils puissent construire des chaînes à la main en examinant chaque document source.

Notre objectif est donc l'amélioration continue de la chaîne de traitement, de sorte que le système apprenne de ses erreurs. Pour cela, on peut tirer parti du fait que des opérateurs humains consultent les fiches synthétiques produites par le système, qui contiennent les événements extraits et les pointeurs vers les documents sources. Ces utilisateurs ont la possibilité de corriger le contenu extrait, fournissant ainsi, indirectement, un *feedback* sur le traitement effectué.

À moyen terme, nous cherchons à développer un système qui réponde aux besoins réels de la communauté *Open-source intelligence* (OSINT). L'utilisateur corrigera les erreurs, et le système pourra prendre en compte ce *feedback* implicite sur les traitements qu'il a effectués, afin d'améliorer ces derniers pour les documents suivants.

Toutefois, dans cet article, nous nous intéressons à un objectif simplifié, dans lequel le *feedback* est supposé donné de façon explicite (simulé dans nos expériences), et sur une échelle numérique. Ce cadre simplifié est une première étape vers la résolution du problème, et nos expériences fournissent ainsi une preuve de concept pour notre objectif à moyen terme.

4.1 Formalisation comme un problème de décision

Puisqu'une chaîne unique pour traiter parfaitement chaque type de document est impossible à construire, l'idéal serait une chaîne faite sur mesure pour chaque document. Nous choisissons de modéliser le problème du traitement d'un document comme un processus de décision markovien (MDP). La stochasticité nous permet de prendre en compte, en particulier, le fait que des actions menées dans un contexte apparemment similaire, peuvent ne pas produire le même résultat. À titre d'exemple, choisir d'extraire la langue peut résulter en l'extraction de langues différentes, ce qui est pris en compte directement par les actions stochastiques des MDP.

Le système a une perception de la tâche sous la forme d'*états* du processus : document courant, informations déjà extraites, temps déjà passé sur ce document, etc. Chaque passage par un service modifie l'état courant. Par ailleurs, le système dispose d'un certain nombre d'*actions* qu'il peut appliquer dans l'état courant : ces actions correspondent au service suivant à lancer (ou à l'arrêt du traitement et l'enregistrement des événements extraits en base). La répétition d'une même action sur un même document est techniquement autorisée, mais cela sera pénalisé par le système de récompense (car induisant un temps de traitement plus long). La chaîne de traitement n'est plus figée, mais contrôlée par un algorithme de RL.

Les actions font transiter le système d'un état à l'autre : par exemple, l'action consistant à extraire la langue, appliquée dans un état donné s_t correspondant à un document en cours de traitement, fera transiter le système vers un état s_{t+1} égal à s_t à ceci près qu'il contiendra l'information « langue extraite » et l'annotation « fr ». Enfin, les récompenses $r(s)$ sont données au système en fonction du *feedback* sur les événements extraits (simulé dans notre cas), et donc seulement pour les états terminaux d'un traitement. Les volumes de documents traités en production (potentiellement tous les documents possibles du *web*) étant très importants, le temps passé à traiter le document influe également sur la récompense (*cf.* section 5).

Plus précisément, les états que perçoit la chaîne sont des états combinatoires, formés par les valeurs d'un certain nombre de *descripteurs* des documents. Ces états permettent une généralisation en apprentissage, par exemple, nous avons déjà vu que le *type* de document (politique vs. mariage) influence en grande mesure l'utilité du mot « alliance » pour l'extraction de l'information. On peut espérer que la chaîne apprenne au fil des interactions que

— si l'état courant a la valeur « true » pour le descripteur « typeExtrait » et la valeur « politique » pour le descripteur « type », alors la meilleure action à effectuer consiste à lancer un service d'extraction qui utilise « alliance » parmi les mots déclencheurs,
— dans les autres cas où le type est extrait, la meilleure action consiste à arrêter le traitement (inutile d'essayer d'extraire des accords entre pays dans des documents non politiques),
— sinon, la meilleure action consiste à lancer un service de reconnaissance de type.

5 Cadre expérimental

L'objectif de cet article est de donner une preuve de concept de notre approche. Pour cela, nous avons considéré un corpus de textes, dont les événements d'intérêt sont déjà connus. En appliquant notre formalisme, nous avons utilisé un algorithme de *Q-learning* pour contrôler le traitement, en simulant un *feedback* en utilisant la vérité terrain.

Nous nous basons sur une chaîne simple mais typique (*cf.* section 2). La chaîne est écrite comme une route Camel (2015) en XML. Chaque service est défini comme un *endpoint*, et nous utilisons le *Dynamic Router* pour donner à l'IA le contrôle sur les services appelées, leur ordonnancement, et leurs paramètres, spécifiquement, le choix des *gazetteers* (les mots déclencheurs) de GATE (Cunningham *et al.*, 2011) pour la détection des événements dans un texte.

5.1 Corpus

Nous nous intéressons à l'extraction d'événements correspondant à des attentats à la bombe (*bombings*) dans le monde entier. Le *Global Terrorism Database* (GTD (2014)) est une base

de données *open source* composée des détails de plus de 125 000 événements terroristes mondiaux de 1970 à 2013. Le corpus est formé d'un ensemble des synthèses de ces événements $\{d_1 \dots d_N\}$ d'où une chaîne d'extraction parfaite extrairait les événements $E_1, \dots, E_N \in GTD$, respectivement (*cf.* exemple 1, exemple 2). Nous attendons de notre système qu'il apprenne une chaîne qui s'approche de ce but, en apprenant non seulement le bon ordonnancement des services, mais aussi le fait que certains services (dans notre cas, le service d'inférence de l'information géographique) et certains *gazetteers* de GATE ne sont pas utiles.

5.2 États et actions

Un état est représenté par une affectation des caractéristiques : *language* $\in$ {"en"," "}, *format* $\in$ {"*text/plain*"," "}, *durée*, *nbServices* $\in \{0-5, 5-20, 20+\}$, *bombing* $\in$ {*true, false*}, *any* $\in$ {*true, false*} où *durée* est le nombre de secondes écoulées depuis le début du traitement du document courant (arrondie à la dizaine de secondes), *nbServices* est le nombre de services déjà utilisés sur ce document, *bombing* est *true* si et seulement si un événement de ce type a déjà été extrait, et de même, *any* est *true* si et seulement si un événement quelconque a déjà été extrait. Ces caractéristiques sont choisies à titre illustratif pour la preuve de concept, en lien avec un ensemble restreint d'actions. Un système opérationnel prendrait évidemment en compte de nombreuses autres caractéristiques (type de document, liste complète de langues, etc.).

Les actions disponibles consistent à choisir le prochain service parmi {Tika, NGramJ, GATE, Geo}. Quand le système choisit GATE, il a le choix parmi six *gazetteers* : *bombing* (verbes et noms), *injure* (verbes et noms), *HarryPotter* (verbes et noms). Les mots contenus dans les listes de *bombing* et *injure* peuvent déclencher l'extraction des événements. Les listes *HarryPotter* contiennent exclusivement des mots qui ne sont pas présents dans les documents du GTD, tels que *dragon*, et l'IA devrait donc apprendre que ce paramètre du service est inutile. Nous cherchons ainsi à vérifier que, pour la tâche d'extraction de *bombings* qui lui est confiée, le système réussit bien à apprendre que le *gazetteer* le plus pertinent est *bombing*, et que les deux autres *gazetteers* l'induisent en erreur (si les mots d'*injure* sont utilisés pour détecter des *bombings*) ou le ralentissent inutilement (*gazetteer HarryPotter*). Une dernière action disponible permet au système d'arrêter le traitement du document, et de retourner les éventuels événements extraits.

5.3 Protocole

Nous avons pris deux jeux de documents du GTD qui contiennent de l'information sur des *bombings*. Le premier consiste en 100 documents d'entraînement que nous avons traités 30 fois par lots, utilisant un taux d'exploration ϵ de 0.4 qui est divisé par 2 jusqu'à 0.1 tous les 5 lots pour réduire de plus en plus l'exploration et augmenter l'exploitation. Les documents étant hétérogènes, nous avons utilisé un taux d'apprentissage α de 0.2. Cela ne garantit pas la convergence de l'apprentissage, mais permet de réduire l'effet négatif des grandes variations, par exemple, si plusieurs documents à la suite ne contiennent pas d'information extractible.

En contrôle, nous comparons la performance de deux chaînes « expertes » (*Mixte* qui accède à toutes les *gazetteers* de GATE, et *Bombe* qui n'accède qu'au *gazetteer* de *bombing*) à celle de l'IA sur le même jeu de 100 documents. Le deuxième jeu consiste en 1000 documents aléatoirement choisis que l'IA « voit » pour la première fois. Nous les traitons avec les deux chaînes « expertes » sans IA ; avec une IA paramétrée avec $\epsilon = \alpha = 0$ et les Q-valeurs apprises à l'issue

du lot 30 ; et avec une IA « vierge » avec un $\epsilon = 0.4$ initialement, divisé par 2 tous les 100 documents jusqu'à $\epsilon = 0.05$, et $\alpha = 0.2$. Pour éviter que l'IA tourne à l'infini, si le temps de traitement d'un document dépasse 30 secondes [1], la chaîne est arrêtée dès que le service courant a fini, induisant une récompense réduite ou négative, et les éventuels événements extraits sont retournés. Les documents sont toujours traités dans le même ordre.

5.4 Objectifs et récompenses

Le *feedback* donné au système prend en compte la similarité entre les événements potentiellement extraits par la chaîne sur le document et l'événement du GTD qui correspond à ce document (voir ci-dessous), et le temps passé à traiter le document. Précisément, en notant $Sim(E_e, E_b)$ la similarité moyenne entre les événements extraits (quand ils existent) et un *bombing* réel, et t le temps passé par la chaîne sur le document, nous donnons le *feedback* F suivant : Si $Sim(E_e, E_b) = 0$ (càd si $E_e = \emptyset$ ou $E_e \neq E_b$) alors F est $-t$, sinon, F est $\frac{Sim(E_e,E_b)}{\max(t,25)}$.

Nous formalisons ainsi le fait que l'extraction d'événements corrects est primordiale, et doit se faire dans un temps raisonnable. D'autre part, si aucun événement n'a été extrait, la chaîne est pénalisée, et ce d'autant plus que le temps passé est long. Nous encourageons ainsi la chaîne à détecter des événements, ou à détecter rapidement qu'il n'y a aucun événement d'intérêt dans le document.

Nous définissons la similarité entre deux événements $E_1 =< C_1, T_1, S_1, A_1 >$ et $E_2 =< C_2, T_2, S_2, A_2 >$ par $Sim(E_1, E_2) = (\alpha \cdot Sim(C_1, C_2) + \beta \cdot Sim(T_1, T_2) + \gamma \cdot Sim(S_1, S_2) + \delta \cdot Sim(A_1, A_2))/(\alpha + \beta + \gamma + \delta)$

Puisque nous simulons un utilisateur intéressé principalement par les événements de type *bombing*, nous donnons à α une valeur plus élevée, spécifiquement $\alpha = 20; \beta = \gamma = \delta = 1$.

La similarité conceptuelle $Sim(C_1, C_2)$ est de 1 s'il y a un atome commun entre les dimensions conceptuelles de E_1 et E_2, et de 0 sinon. Par exemple, pour $C_1 = \{BombingEvent, AttackEvent\}$ et $C_2 = \{BombingEvent\}$, on obtient $C_1 \cap C_2 = \{BombingEvent\}$, et donc $Sim(C_1, C_2) = 1$. Nous procédons de même pour la similarité géographique $Sim(S_1, S_2)$.

Pour la similarité temporelle $Sim(T_1, T_2)$, s'il y a au moins un élément en commun nous donnons une similarité de 1, mais si une comparaison directe ne donne pas un résultat, nous utilisons l'information dérivée (jour, mois, année, jour de la semaine). Par exemple, pour $T_1 = \{7\ October\ 1969\}$ et $T_2 = \{Tuesday\}$, l'intersection est vide, mais en notant que le 7 octobre 1969 a été un mardi, nous obtenons $Sim(T_1, T_2) = \frac{1}{7}$.

Enfin, pour la similarité entre les dimensions agentives, nous utilisons la distance de Levenshtein sur chaque paire d'agents a_1, a_2 pris dans A_1, A_2 (nombre minimal de caractères à supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre), rendue « floue » ($FSLS(a_i, a_j)$) en considérant les sous-séquences de la chaîne principale (Ginstrom, 2015). Nous définissons $Sim(A_1, A_2)$ comme $\max\{FSLS(a_1, a_2) \mid a_1 \in A_1, a_2 \in A_2\}$, si elle est au-dessus d'un certain seuil θ, et 0 sinon (en pratique, $\theta = 0.45$ donne de bons résultats). Par exemple, pour $A_1 = \{Le\ Professeur\ Tournesol\ PhD\}$ et $A_2 = \{Tournessol\}$, on obtient

$Sim(A_1, A_2) = FSLS(\cancel{Le\ Professeur}\ Tournesol\ \cancel{PhD}, Tournessol) = 1 - (\frac{1}{9}) = 0.89$ (1 suppression)

1. Ce seuil est mal calibré : il est trop haut pour que l'IA apprenne à s'arrêter aussi rapidement que la chaîne experte. En observant qu'une chaîne experte prend en moyenne 25 secondes pour traiter de documents similaires, nous avons ajouté une marge de 5 secondes pour permettre à l'IA d'apprendre qui s'est finalement avérée inutile.

Nous ne nous intéressons pas ici à l'association d'entités telles que *François Hollande / le président*, mais la modularité du système permettrait de prendre en compte cette similarité facilement en s'appuyant sur des ressources adéquates.

6 Résultats expérimentaux

Les diagrammes $1a$ et $1b$ dans Figure 3 montrent les récompenses reçues (par document et par ordre décroissant des récompenses respectivement) par une IA, et par les deux chaînes « expertes » sur les 100 documents d'entraînement. Dans le diagramme $1a$, les documents où l'IA a fait au moins une exploration sont marqués sur les axes en haut et en bas, et seulement les courbes des lots $1, 10, 20, 30$ sont montrées (les autres étaient similaires).

La courbe *IA Lot30* et la courbe *IA Lot1* montrent comme attendu que plus l'IA traite de documents, plus elle s'améliore. La comparaison avec la courbe *Bombe* montre que l'IA est capable d'une performance proche de celle de la chaîne experte. Notons que les documents où l'IA reçoit de moins bonnes récompenses que la chaîne experte sont dues aux explorations (p.ex. document 13), ou à la mauvaise calibration du seuil de temps (document 5). Parfois l'IA surpasse la chaîne experte (document 87) qui se traduit par une récompense (réduite) pour l'extraction d'un événement autre que *bombing*.

Pour nous approcher du cas d'utilisation réel, nous avons ensuite traité 1000 documents inconnus, choisis aléatoirement. Les diagrammes $2a$ et $2b$ montrent les récompenses reçues (par document, et par ordre décroissant des récompenses respectivement) par l'*IA formée* sur les 100 documents mentionnés ci-dessus, par les deux chaînes « expertes » (*Bombe* et *Mixte*), et par l'*IA non formée* » (c'est-à-dire sans *a priori* sur les besoins de l'utilisateur ni sur les documents). Nous voyons que, sauf mauvaise calibration du seuil, la courbe *IA formée* suit exactement celle de la chaîne experte *Bombe*, montrant qu'elle a bien appris à n'extraire que des événements *bombing*. $2c$ montre la moyenne cumulative (lissée pour les valeurs négatives) des différences entre les récompenses reçues par les IAs (*formée* et *non formée*), et la chaîne experte *Bombe*. Nous voyons que l'IA *formée* est capable d'une performance constante, et que l'IA *non formée* se stabilise au bout d'environ 600 documents, et donc généralise bien en apprentissage. Il s'avère également que l'IA a bien appris l'ordonnancement de la chaîne. Par exemple, dans l'état correspondant à un document sans type ni langue extraits, 0 secondes et 2 services déjà écoulés, et aucun événement extrait, la meilleure action apprise est de passer le document au service Tika, et dans l'état correspondant à un document où la langue, le type, et au moins un événement sont extraits, quelles que soient les valeurs des autres attributs, l'IA arrête le traitement de ce document. L'IA a aussi appris à optimiser la chaîne pour ce type de document. Elle n'appelle pas le service *Geo* et n'utilise que la liste de verbes de *bombing*, comme espéré. Cependant, elle a trouvé une solution inattendue avec le choix des noms *injure*, ce qui reflète peut être l'importance relative donnée par GATE lui-même aux noms et verbes pour les extractions.

7 Conclusion et perspectives

Nous avons proposé une formalisation du problème de l'amélioration continue d'une chaîne de traitement de documents, visant à extraire des événements dans des documents provenant de sources ouvertes, ainsi qu'une solution basée sur l'apprentissage par renforcement.

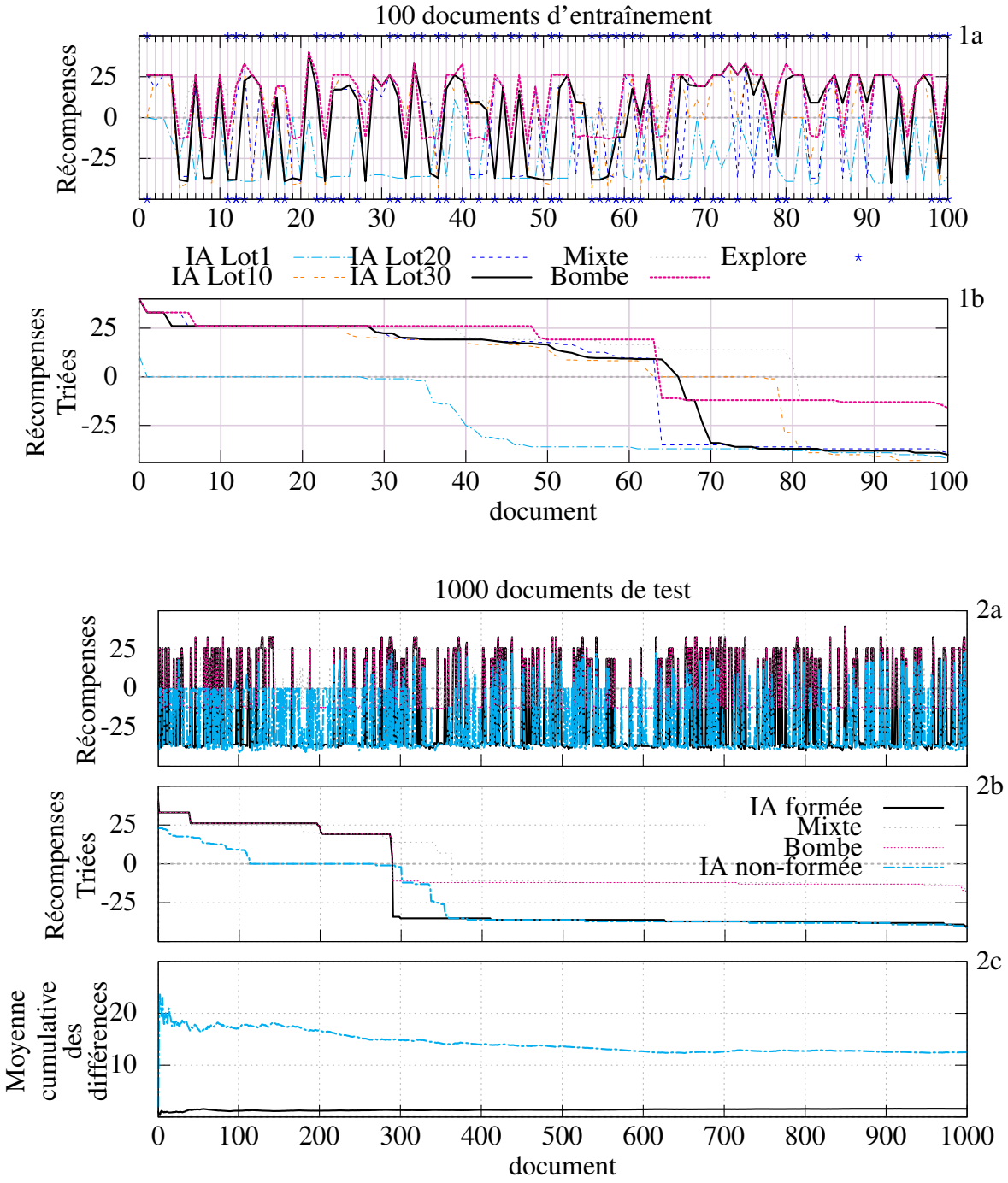


FIGURE 3 – Résultats expérimentaux : $1a, 1b$ montrent les résultats pour les documents d'entraînement, et $2a, 2b, 2c$ pour les documents de test ; $1a, 2a$ montrent les récompenses reçues par document (toujours rencontrés dans le même ordre), $1b, 2b$ montrent les récompenses triées, et $2c$ la moyenne cumulative lissée des différences entre le système experte et les IA.

Des premières expériences nous permettent d'apporter une preuve de concept à notre approche, et à court terme nous voulons nous approcher d'une situation de production en augmentant la taille des expériences, avec plusieurs types d'événements, une affectation plus complète de caractéristiques aux états, etc. Bien que nous ayons présenté ici un système simple, le temps de calcul ne sera pas un frein avec un algorithme de type *Q-learning*, dont les calculs sont instantanés à chaque étape.

Notre objectif final est d'intégrer le *feedback* implicite. Pour cela nous nous appuierons notamment sur Weng & Zanuttini (2013) et Weng *et al.* (2013), qui traitent de la prise en compte de récompenses qualitatives, de façon interactive, dans la prise de décision séquentielle.

Références

BRATKO I. & SUC D. (2003). Learning qualitative models. *AI magazine*, **24**(4), 107.

CAMEL (2015). Apache camel. `http://camel.apache.org/`.

CHAI X., VUONG B.-Q., DOAN A. & NAUGHTON J. F. (2009). Efficiently incorporating user feedback into information extraction and integration programs. In *Proc. SIGMOD 2009*, p. 87–100 : ACM.

CULOTTA A., KRISTJANSSON T., MCCALLUM A. & VIOLA P. (2006). Corrective feedback and persistent learning for information extraction. *Artif. Intell.*, **170**(14-15), 1101–1122.

CUNNINGHAM H., MAYNARD D., BONTCHEVA K., TABLAN V., ASWANI N., ROBERTS I., GORRELL G., FUNK A., ROBERTS A., DAMLJANOVIC D., HEITZ T., GREENWOOD M. A., SAGGION H., PETRAK J., LI Y. & PETERS W. (2011). *Text Processing with GATE (Version 6)*.

DOUCY J., ABDULRAB H., GIROUX P. & KOTOWICZ J.-P. (2008). Méthodologie pour l'orchestration sémantique de services dans le domaine de la fouille de documents multimédia.

DUPONT G., ADAM S. & LECOURTIER Y. (2011). Apprentissage par renforcement pour la recherche d'information interactive. In *Actes des JFPDA 2011*, p. Actes–électroniques.

FROMHERZ M. P., BOBROW D. G. & DE KLEER J. (2003). Model-based computing for design and control of reconfigurable systems. *AI magazine*, **24**(4), 120.

GEONAMES (2015). Geonames. `http://www.geonames.org/`.

GINSTROM R. (2015). The GITS Blog. `http://ginstrom.com/`.

GTD (2014). Global Terrorism Database. `http://www.start.umd.edu/gtd/`.

NGRAMJ (2015). Ngramj. `http://ngramj.sourceforge.net/`.

PUTERMAN M. L. (1994). *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. New York : Wiley, first edition.

SERRANO L. (2014). *Vers une capitalisation des connaissances orientée utilisateur. Extraction et structuration automatiques de l'information issue de sources ouvertes*. Thèses, UNICAEN, France.

SUTTON R. S. & BARTO A. G. (1998). *Reinforcement Learning — An Introduction*. MIT Press.

SZEPESVÁRI C. (2010). *Algorithms for Reinforcement Learning*. Morgan and Claypool.

TIKA (2015). Apache tika. `http://tika.apache.org/`.

VAN HAGE W. R., MALAISÉ V., SEGERS R., HOLLINK L. & SCHREIBER G. (2011). Design and use of the Simple Event Model (SEM). *Web Semantics : Science, Services and Agents on the World Wide Web*, **9**(2), 128–136.

WATKINS C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, University of Cambridge.

WEBLAB (2015). Weblab. `http://weblab-project.org/`.

WENG P., BUSA-FEKETE R. & HÜLLERMEIER E. (2013). Interactive Q-Learning with Ordinal Rewards and Unreliable Tutor. In *ECML/PKDD Workshop on RL with Generalized Feedback*.

WENG P. & ZANUTTINI B. (2013). Interactive Value Iteration for Markov Decision Processes with Unknown Rewards. In *International Joint Conference on Artificial Intelligence*.

Détection d'informations vitales pour la mise à jour de bases de connaissances

Rafik Abbes, Nathalie Hernandez, Karen Pinel-Sauvagnat, Mohand Boughanem

INSTITUT DE RECHERCHE EN INFORMATIQUE DE TOULOUSE
118, ROUTE DE NARBONNE, F-31062 TOULOUSE CEDEX 9, FRANCE
RAFIK.ABBES, NATHALIE.HERNANDEZ, KAREN.SAUVAGNAT, MOHAND.BOUGHANEM @IRIT.FR

Résumé : Mettre à jour une base de connaissances est une problématique actuelle qui suit l'évolution permanente du web de données liées. De nombreuses approches ont été proposées afin d'extraire dans des documents textuels la connaissance à mettre à jour. Ces approches arrivent à maturité mais reposent sur l'hypothèse selon laquelle le corpus adéquat a déjà été constitué. Dans la majorité des cas, les documents à prendre en compte sont sélectionnés manuellement ce qui rend difficile une mise à jour exhaustive de la base. Dans cet article nous proposons une approche originale visant à identifier automatiquement dans un flux de documents du web les éléments pouvant apporter de la connaissance nouvelle sur des instances déjà représentées dans une base.
Mots-clés : Informations vitales, Mise à jour de bases de connaissances, TREC Temporal Summarization

1 Introduction

Les bases de connaissances telles que *DBpedia* sont devenues des sources indispensables pour rendre accessibles à tout un chacun la connaissance relative aux instances du monde réel comme les personnes, les organisations, les événements, etc. Au cours du temps, la connaissance relative à ces instances peut évoluer lorsque, par exemple, dans le cas de personnes, ces instances réalisent de nouvelles actions, ou se trouvent dans de nouvelles situations. Ceci implique un travail permanent de suivi pour maintenir les bases de connaissances à jour.

L'extraction de connaissances à partir de documents textuels est une approche couramment utilisée pour la constitution de base de connaissances [Petasis *et al.* (2011)]. Ces approches reposent souvent sur l'hypothèse selon laquelle le corpus à partir duquel la connaissance est extraite est identifié, que ce soit à partir des pages Wikipedia dans le cas de DBpedia, ou constitué manuellement [Augenstein *et al.* (2012); Exner & Nugues (2012)]. Dans le contexte de la mise à jour de bases de connaissances, la tâche d'identification des textes d'où extraire la connaissance n'est pas triviale. D'une part, certaines approches d'extraction de connaissance à partir du texte analysent les documents dans leur intégralité, or la connaissance sur une instance donnée est souvent décrite uniquement dans quelques phrases du document. D'autre part, lorsqu'on considère en particulier certaines instances comme des instances de type événement (catastrophe naturelle, ...) dont la connaissance établie peut évoluer fréquemment au cours de la période englobant la date de l'événement, les textes sur lesquels ces approches sont appliquées doivent reporter des informations nouvelles et à jour.

Dans cet article nous proposons une approche visant à identifier dans un flux de documents du Web les phrases qui reportent des informations opportunes et pertinentes sur les instances représentées d'une base de connaissances. Nous appelons ces phrases *phrases vitales*.

Détecter en temps réel les phrases vitales est une tâche complexe qui soulève plusieurs problématiques pouvant être vues comme des étapes du processus :

– comment détecter si un document reporte une information vitale sur une instance ?
– étant donné un document vital, comment extraire les phrases vitales reportant les informations vitales ?
– comment détecter si deux phrases vitales reportent la même information ?

La *première étape* est importante et nécessite la mise en place d'un processus riche que nous avons détaillé dans des travaux précédents [Abbes *et al.* (2013, 2015)]. Dans cet article, nous simplifions cette étape en nous focalisant sur des instances de type événement et en analysant le flux de documents uniquement dans les périodes de déroulement de ces événements. Par conséquent, les documents mentionnant le ou les labels associés à l'instance dans la base de connaissances ont tendance à reporter des informations vitales. La *deuxième étape* est primordiale puisqu'elle permet de choisir les phrases vitales candidates. Les travaux de l'état de l'art se fondent généralement sur la présence de mots spécifiques pour calculer un score de pertinence. Ces mots sont choisis soit manuellement, soit sélectionnés automatiquement. Dans ce travail, nous proposons d'exploiter la connaissance déjà représentée dans la base de connaissances. Nous cherchons à identifier le vocabulaire propre à chaque type d'instances. Concernant la *dernière étape*, pour détecter la nouveauté d'une phrase par rapport à une autre, nous cherchons là encore à évaluer l'apport de la connaissance déjà représentée dans la base à enrichir.

En résumé, nous souhaitons répondre à la question suivante : à quel point l'exploitation de connaissances déjà représentées peut servir à détecter les informations vitales et non redondantes relatives aux instances dans un flux de documents Web ? Nous voyons ces travaux comme une première étape qui devra ensuite être complétée par une phase d'extraction de connaissances des phrases vitales dont nous montrons le processus d'identification dans ce papier.

Cet article est organisé comme suit. Nous présentons dans la section 2 un état de l'art des travaux liés à l'identification d'informations vitales sur le Web. La section 3 présente notre approche reposant sur la prise en compte de connaissances connues sur l'instance. Dans la section 4, nous présentons et discutons l'intérêt de notre approche par rapport aux méthodes de l'état de l'art. Nous concluons et énonçons quelques perspectives en section 5.

2 État de l'art

Accélérer la mise à jour des bases de connaissances est une problématique actuelle dont le premier enjeu est d'identifier un besoin d'évolution. L'analyse de documents d'où extraire la connaissance à mettre à jour est une solution pour identifier ce besoin [Zablith *et al.* (2015)]. La phase d'identification de ces documents est souvent laissée aux concepteurs de la base dans les travaux d'ingénierie de connaissances. Cependant lorsque la base de connaissances comporte des instances largement mentionnées sur le Web, il est dommage de ne pas tirer profit de ces informations. Le module *DBpedia Live*, par exemple, vise à mettre à jour en temps quasi réel la base de connaissances lorsque les *infobox* des pages Wikipedia sont modifiées [Lehmann *et al.* (2014)]. Cependant, comme souligné dans [Frank *et al.* (2012)], un certain temps de latence est constaté pour la mise à jour alors que l'information est publiée en temps réel sur le Web.

Pour faire face à ce problème et aider à l'identification d'informations vitales lorsqu'un document de référence régulièrement mis à jour n'est pas disponible, nous nous sommes tournés vers les travaux de recherche d'information qui s'intéressent à ces aspects. La campagne

d'évaluation TREC a notamment lancé la tâche *Knowledge Base Acceleration (KBA)* [Frank *et al.* (2013, 2012)]. Plusieurs méthodes ont été proposées [Wang *et al.* (2013)] [Abbes *et al.* (2013)] [Abbes *et al.* (2015)] afin d'identifier, en temps réel, les documents vitaux reportant des informations nouvelles sur des instances de type personnes, organisations et établissements. Cependant, malgré leur utilité, ces méthodes renvoient des documents entiers ce qui oblige les éditeurs de la base de connaissances ou les outils d'extraction à parcourir tous leurs contenus pour chercher les nouvelles informations vitales. En outre, elles ne traitent pas le problème de redondance entre les documents, c'est à dire qu'elles renvoient tous les documents apportant des informations vitales même s'ils contiennent des informations vitales redondantes.

D'autres approches se sont intéressées à identifier à partir d'un flux de documents Web, les phrases vitales relatives à des événements largement connus comme les catastrophes naturelles [Aslam *et al.* (2013)] [McCreadie *et al.* (2014)]. Liu *et al.* (2013) s'appuient sur des données d'apprentissage pour apprendre les mots importants permettant d'identifier les phrases vitales. Xu *et al.* (2013) utilisent un classifieur afin de détecter les phrases contenant de nouvelles informations. Zhang *et al.* (2013) sélectionnent les phrases contenant les mots les plus représentatifs selon leur fréquence d'occurrences. Dans ce travail, nous nous intéressons aussi aux instances d'événements. Nous détectons les phrases vitales d'un nouvel événement émergent en exploitant des mots importants récupérés à partir des connaissances déjà représentées dans la base.

3 Détection en temps réel des informations vitales relatives à une instance

Notre approche a pour but de détecter en temps réel les phrases reportant de nouvelles informations vitales (pertinentes et opportunes) relatives à une instance donnée d'une base de connaissances à partir d'un flux de documents issus du Web. Ces phrases vitales peuvent servir à mettre à jour la connaissance sur cette instance. Par conséquent, elles doivent être pertinentes (concerner l'instance), exhaustives (couvrir les différentes informations publiées sur l'instance), non redondantes (reportées une seule fois) et émises sans trop de latence.

Formellement, considérons un flux continu F composé de documents d ayant chacun une date de publication $t(d)$ et une séquence de phrases s_j tels que $0 \leq j < l(d)$ où $l(d)$ désigne la longueur du document d en nombre de phrases. Soient $h_0, h_1, \ldots, h_n$ des instants séparés par un intervalle de temps constant (par exemple une heure). Nous désignons par F_{h_i} l'ensemble de documents du flux tel que $\forall d \in F_{h_i}, h_{i-1} \leq t(d) < h_i$.

L'algorithme 1 décrit le fonctionnement général de notre approche de détection des phrases vitales relatives à une instance donnée I. A chaque instant h_i, nous distinguons 3 étapes principales que nous détaillons dans les sous-sections suivantes :

1. sélection des documents vitaux D_{h_i} par rapport à I en utilisant comme requête le ou les labels associés à l'instance dans la base de connaissances,
2. sélection des phrases vitales candidates (contenant une information vitale),
3. vérification de la nouveauté des phrases candidates par rapport aux phrases déjà sélectionnées ($\in V(I)$).

Algorithm 1 Détection des phrases vitales relatives à une instance

ENTRÉES: F : Flux de documents
ENTRÉES: I : Instance à mettre à jour, ayant une étiquette *I.label*
ENTRÉES: h_0 (h_n) : Début (Fin) de la période d'analyse du flux
SORTIE: $V(I) \leftarrow \{\}$: Historique des phrases vitales relatives à I

pour chaque $i \in [1, n]$ **faire**
  $D_{h_i} \leftarrow sélection\_des\_documents(F_{h_i}, I.label)$
  pour chaque $d \in D_{h_i}$ **faire**
    pour chaque $s_j \in d$ **faire**
      si $est\_vitale(s_j, I)$ ***ET*** $est\_nouvelle(s_j, V(I))$ **alors**
        $enrichir(V(I), s_j)$
      fin si
    fin pour
  fin pour
fin pour

3.1 Sélection des documents vitaux

Nous n'analysons que la période "chaude" durant laquelle les informations vitales sur une instance donnée sont publiées dans le flux de documents du Web. Dans ce travail, nous supposons que cette période est connue. A chaque instant h_i, nous analysons les nouveaux documents apparus dans le flux entre h_{i-1} et h_i et nous attribuons à chacun d'eux un score de vitalité par rapport à l'instance I considérée. Ce score est calculé par la probabilité que le ou les termes composant le label de l'instance I soient générés par un modèle de langue probabiliste estimé à partir du document analysé [Zhai & Lafferty (2001)]. Le ***top-h*** des documents est sélectionné afin d'être analysé dans l'étape suivante.

3.2 Sélection des phrases vitales

Dans cette étape, nous analysons les phrases contenues dans les documents sélectionnés. Pour chaque phrase, nous devons décider si elle est vitale (reportant une information pertinente et opportune) par rapport à l'instance à surveiller I. Notre intuition est de considérer une phrase comme vitale si :

– elle est à proximité de l'instance I (des termes de l'étiquette de I),
– elle contient des mots "importants" relativement à l'instance I.

La proximité d'une phrase par rapport à l'instance I peut refléter sa pertinence. Une phrase mentionnant l'instance a plus de chance de parler de celle ci. Nous traduisons ainsi la proximité entre une phrase s_j et l'instance I en un score calculé selon l'équation suivante :

$$score\_proximité(s_j, I) = \frac{1}{|I.label|} \sum_{t \in I.label} \sum_{dist=0}^{dmax} e^{-dist * match\_s(t, s_{j+dist}, s_{j-dist})} \quad (1)$$

$I.label$ est l'étiquette décrivant l'instance I. $|I.label|$ est le nombre de mots qu'elle contient.
$match\_s(t, s_x, s_y)$ est égal à 1 si t est contenu dans l'une des phrases s_x et s_y, 0 sinon.
$dmax$ est la distance maximale à considérer (calculée en nombre de phrases).

Nous considérons uniquement les phrases à proximité de l'instance I en favorisant celles qui sont proches de l'ensemble des termes composant l'étiquette de l'instance I, c.à.d, ayant un *score proximité* supérieur à un seuil τ_p (la valeur de τ_p peut être déterminée expérimentalement).

En plus de la proximité, nous supposons que pour une instance I, il existe un ensemble de mots "importants" qui peuvent refléter la vitalité d'une phrase. Nous appelons ces mots des *mots déclencheurs*. Nous posons l'hypothèse selon laquelle les instances de même type (représenté dans la base de connaissances) partagent les mêmes mots déclencheurs. Afin d'identifier ces mots déclencheurs, nous proposons d'exploiter toutes les annotations (description en langage naturel) qui ont pu être renseignées sur des instances du type considérées. Nous considérons comme étant une annotation le texte associé à une instance par les propriétés d'annotation de OWL, ou les propriétés du Dublin Core, ou encore le résumé associé dans DBpedia par la propriété dbpedia-owl :abstract. Par exemple, les mots tels que *effets*, *force*, *tempête*, *blessés*, *dommages* pourront être très utiles pour décrire les instances de type *ouragan* comme ils sont présents dans les annotations associées aux instances *ouragan Sandy* et *ouragan Isaac*.

Formellement, soient $X(I) = \{A(I_1), A(I_2), \ldots, A(I_m)\}$ l'ensemble des m extraits des valeurs des annotations associées aux instances de même type que I. Nous pondérons les mots t par l'équation suivante :

$$\omega(t) = \frac{\sum_{i=1}^{m} TF(t, A(I_i))}{IIF(t)} \tag{2}$$

$TF(t, A(I_i))$ est le nombre d'occurrences du terme t dans l'annotation $A(I_i)$

$IIF(t) = log(\frac{m+1}{IF(t)})$ est un facteur utilisé pour donner la priorité aux termes se trouvant dans la plupart des annotations des instances de même type que l'instance I

$IF(t)$ est le nombre d'instances du type dont l'annotation contient le terme t

Les ***top-k*** premiers mots seront considérés comme des mots déclencheurs pour l'instance I. Pour qu'une phrase soit considérée comme une phrase vitale candidate, il faut :

- que le score de proximité de la phrase soit $> \tau_p$,
- qu'elle contienne un mot déclencheur.

3.3 Détection de la nouveauté

Les phrases sélectionnées à l'étape précédente pourraient contenir des informations vitales redondantes déjà émises. Afin d'éliminer la redondance, nous comparons chaque phrase vitale candidate à toutes les phrases vitales déjà ajoutées à l'ensemble incrémental $V(I)$. Détecter la nouveauté n'est pas une tâche facile. Comme le montre le tableau 1, les deux phrases s1 et s2 contiennent un grand nombre de chaînes de caractères en commun, mais reportent deux informations différentes. Inversement, les phrases s2 et s3 sont divergentes textuellement mais portent la même information.

Dans notre approche, nous considérons qu'une phrase vitale candidate s_j est nouvelle par rapport aux phrases déjà émises $V(I)$ si son texte est divergent et/ou présente une instance liée nouvelle (non détectée dans les phrases précédentes $V(I)$). Formellement, s_j est nouvelle si elle respecte la fonction de nouveauté suivante :

n°	Date	Texte
s1	26 Oct. 2012 - 07 :27	Hurricane Sandy leaves 21 people dead in Caribbean
s2	26 Oct. 2012 - 20 :50	Hurricane Sandy leaves 41 people dead in Caribbean
s3	30 Oct. 2012 - 06 :17	Hurricane Sandy is continuing to head north from the Caribbean where it has killed a total of 41 people in the Caribbean

TABLE 1 – Exemple de phrases vitales

$$est\_nouvelle(s_j, V(I)) = \textit{texte\_divergent}(s_j, V(I)) \circ \textit{instance\_liée\_nouvelle}(s_j, V(I)) \quad (3)$$

$$texte\_divergent(s_j, V(I)) = \begin{cases} faux & si\ \exists s_k \in V(I),\ cos(s_j, s_k) > \tau_n(V(I)) \\ vrai & sinon \end{cases} \quad (4)$$

$$instance\_liée\_nouvelle(s_j, V(I)) = \begin{cases} vrai & si\ \exists x \in IL(s_j, I),\ \forall s_k \in V(I)\ x \notin IL(s_k, I) \\ faux & sinon \end{cases} \quad (5)$$

$IL(s_i, I)$ est l'ensemble des instances liées reconnues dans la phrase s_i. Dans notre méthode, nous proposons de prendre en compte les propriétés définies dans l'ontologie pour le concept type de l'instance I que nous considérons. Nous cherchons à identifier dans la phrase, des instances ou des valeurs potentiellement liées sémantiquement car leur type correspond au domaine ou co-domaine des propriétés définies dans l'ontologie pour le concept.

$\tau_n(V)$ est un seuil de nouveauté textuelle. Au fur et à mesure que l'ensemble de phrases vitales $V(I)$ s'enrichit, le risque de redondance augmente, d'où l'idée de faire décroître le seuil τ_n selon une fonction gaussienne :

$$\tau_n(V(I)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{|V(I)|^2}{\delta^2}} \quad (6)$$

Les paramètre σ a un impact sur la tolérance de la similarité, et le paramètre δ contrôle le taux de décroissance du seuil. $|V(I)|$ est le nombre de phrases de l'ensemble $V(I)$.

Le symbole $\circ$ de l'équation 3 peut être un opérateur ***ET*** pour rendre le système orienté Précision en limitant la redondance (dans ce cas, la phrase s3, ne présentant aucune instance liée nouvelle par rapport aux phrases s1 et s2, sera considérée comme redondante malgré que le fait qu son texte diverge), ou bien un opérateur ***OU*** pour privilégier le Rappel (dans ce cas, la phrase s2, malgré le fait qu'elle diverge peu par rapport à s1, elle sera considérée comme nouvelle car elle présente une nouvelle valeur).

4 Expérimentations

Trouver un cadre expérimental pour évaluer notre approche n'est pas trivial car il n'existe pas à notre connaissance de jeux de données constitués d'une base de connaissances dont plusieurs versions sont disponibles ainsi que les corpus de documents desquels la connaissance a été extraite pour mettre en place les différentes versions. Bien qu'étant une tâche de Recherche d'Information, la tâche *Temporal Summarization* (TS) de la campagne d'évaluation TREC est à notre sens la plus adaptée.

ID	Ressource dans dbpedia	Requête (label)	Type	Début	Fin
1	2012_Buenos_Aires_rail_disaster	buenos aires train crash	accident	2012-02-22-12	2012-03-03-11
9	2012_Guatemala_earthquake	guatemala earthquake	earthquake	2012-11-07-16	2012-11-17-16
19	2012_Romanian_protests	romanian protests	protest	2012-01-12-00	2012-01-26-00

TABLE 2 – Exemples d'instances proposées dans la tâche TS en 2013 et 2014

4.1 Cadre expérimental

Le but de cette tâche est de concevoir des systèmes capables de surveiller les événements en détectant à la volée toutes les nouvelles informations publiées dans un flux qui comporte 500 millions de documents en anglais, issus de différentes sources du Web (Presse, Blog, etc.) et associés à des dates de publications (*timestamp*) allant du mois d'octobre 2011 au mois d'avril 2013. Les systèmes doivent extraire les phrases contenant des informations vitales tout en évitant la redondance.

4.1.1 Instances

Les topics à considérer dans le cadre de cette tâche correspondent à des événements d'actualité tels que des manifestations, des accidents ou des catastrophes naturelles. Le tableau 2 illustre quelques exemples parmi les 24 événements [1] proposés par les organisateurs de la tâche en 2013 et 2014. Les colonnes *Début* et *Fin* définissent la période à surveiller pour chaque événement. En analysant ces topics, nous avons remarqué qu'ils correspondaient à des instances de DBpedia pour lesquelles un label est défini. Nous considérons que les instances pour lesquelles nous souhaitons identifier des phrases vitales dans notre approche peuvent être apparentées aux topics. Le label défini dans DBpedia est utilisé pour constituer la requête dans notre approche. Il est présenté dans la troisième colonne du tableau.

4.1.2 Informations vitales à retrouver et jugements de pertinence

La figure 1 illustre le nombre d'informations vitales à retrouver pour les instances proposées. Ces informations sont extraites à partir des différentes mises à jour des pages Wikipedia de ces événements. Ces informations ayant été rajoutées manuellement au cours d'une mise à jour de la page de Wikipedia, nous considérons qu'elles auraient également mené à une mise à jour manuelle de la connaissance représentée dans DBpedia sur l'instance. Cette référence nous paraît donc pertinente pour évaluer notre approche.

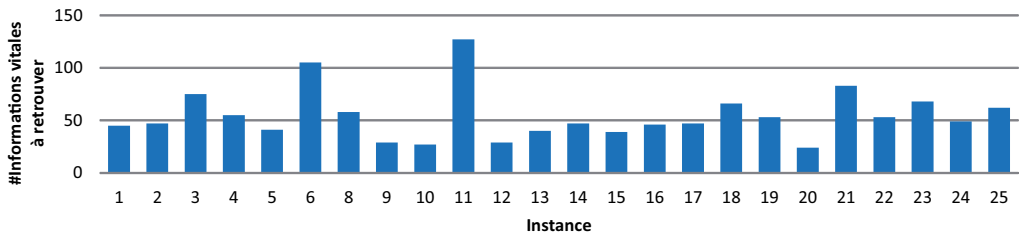


FIGURE 1 – Nombre d'informations vitales à retrouver pour les instances proposées dans la tâche TS en 2013 et 2014

1. Les *topics* sont disponibles dans ce lien *www.trec-ts.org/documents*

Une phrase est jugée pertinente si elle peut être associée à au moins une information vitale. Cette association est faite par les juges de la tâche. Dans l'exemple de l'instance *2012_Buenos_Aires_rail_disaster*, la phrase *49 dead, over 500 wounded in Buenos Aires !*, émise le *23-02-2012 ; 03 :21*, est associée à trois informations vitales : *"train accident in Buenos Aires, Argentina"*, *"550 injured"* et *"49 confirmed deaths"*.

4.1.3 Mesures d'évaluation

Pour analyser les résultats, nous utilisons les mesures suivantes classiques de Rappel et Précision, ainsi que :

$$Précision_N = \frac{Nombre\ d'informations\ vitales\ détectées}{Nombre\ total\ de\ phrases\ émises} \tag{7}$$

$$H = 2 * \frac{Précision * Rappel}{Précision + Rappel} \tag{8}$$

$$H_N = 2 * \frac{Précision_N * Rappel}{Précision_N + Rappel} \tag{9}$$

Nous utilisons le rappel, la precision et (8) pour mesurer la capacité d'un système à renvoyer les phrases vitales, sans pénaliser la redondance. Pour considérer la nouveauté (pénaliser la redondance), nous utilisons le rappel, (7) et (9).

4.2 Configuration de notre système

Pour l'étape de ***sélection des phrases vitales candidates***, nous appliquons la méthode expliquée dans la section 3.2 qui repose sur la détection du ***topK*** des mots déclencheurs relatif à l'instance traitée (Eq. 2). Nous désignons cette stratégie par ***Gen-Auto***.
Nous évaluons aussi deux autres stratégies :

- La stratégie ***Gen-Man*** : sélection manuelle de 15 mots génériques qui peuvent caractériser tous ou la plupart des événements évalués. Parmi ces mots-clés, nous listons : *died*, *dead*, *death*, *kill*, *injuries*, *damage*, *victims*, *survivor* etc.
- La stratégie ***QW*** : considération des termes du label de l'instance etudiée comme mots déclencheurs.

Pour la ***détection de la nouveauté***, nous évaluons les méthodes suivantes :

- ***Texte*** : utilisant uniquement la nouveauté textuelle (Eq. 4)
- ***NER*** : utilisant uniquement la reconnaissance d'instances liées (Eq. 5). Les instances considérées dans la tâche sont du type Event. Puisque les propriétés [2] définies pour ce concept dans l'ontologie DBpedia sont très discutables, nous avons choisi d'exploiter la définition de l'ontologie Event [3] en recherchant dans les phrases des instances de lieux, de personnes et des valeurs numériques [4].
- ***NER*Texte*** : utilisant la fonction de nouveauté combinée avec un opérateur ET (Eq. 3)
- ***NER+Texte*** : utilisant la fonction de nouveauté combinée avec un opérateur OU (Eq. 3)
- ***Sans*** : Sans appliquer la nouveauté (émettre toutes les phrases sélectionnées à l'étape 2)

2. http ://mappings.dbpedia.org/server/ontology/classes/Event
3. http ://motools.sourceforge.net/event/event.html
4. nous utilisons l'outil réalisé par le groupe *NER Stanford* (http ://nlp.stanford.edu/ner/)

Paramètres de notre système

Nous avons appliqué la validation croisée afin de fixer les paramètres de notre système, en faisant varier le nombre de documents sélectionnés par heure entre 1 et 20 avec un pas de 1, *top-k* entre 4 et 40 avec un pas de 2, τ_p entre 0.4 et 1 avec un pas de 0.1, δ entre 10 et 300 avec un pas de 10, σ entre 0.5 et 1 avec un pas de 0.1. Les valeurs optimales obtenues sont : *top-h=10* et *top-k=15*, $\tau_p = 0.8$, $\delta = 200$ et $\sigma = 0.5$.

4.3 Analyses des résultats

A l'issue de la première étape, notre système renvoie $20\,800$ documents pour les 24 instances (soit 866 documents par instance) permettant d'atteindre un rappel moyen de 0.65.

4.3.1 Stratégies de sélection des mots déclencheurs

La figure 2 compare les différentes stratégies de sélection des mots déclencheurs pour la détection des phrases vitales sans tenir compte de la redondance (***sans***). Considérer comme mots déclencheurs uniquement les termes du label associé à l'instance (***QW***) permet de capturer environ **63%** (0.407/0.650) des informations vitales contenues dans les documents sélectionnés avec une précision ne dépassant pas ***0.161***. La condition de proximité (Eq. 1) avec un seuil $\tau_p = 0.8$ semble être stricte car elle exige la présence de la plupart des termes du label dans les phrases vitales ce qui peut expliquer la perte de 37% d'informations vitales. L'utilisation des mots-clés **Gen-Auto** revient à vérifier la présence simultanée des termes du label et d'un mot générique. Comme résultat, on constate une amélioration légère de la précision par rapport à QW "pratiquement" sans baisse du rappel. Cette stabilité du rappel prouve que les mots saillants récupérés automatiquement des annotations associées aux instances du même type permettent de couvrir les différents aspects de la nouvelle instance traitée. L'amélioration de la précision montre l'importance de ces mots. La sélection manuelle de mots génériques (**Gen-Man**) améliore la précision (surtout pour les instances de 2014) mais le rappel est relativement faible par rapport à la méthode automatique *Gen-Auto*.

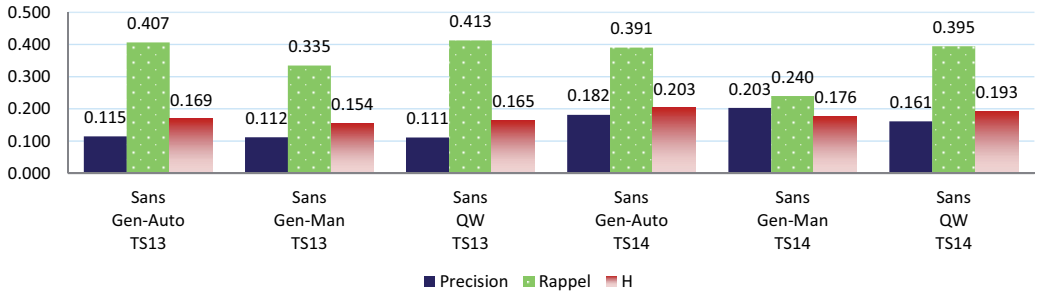


FIGURE 2 – Comparaison des différentes stratégies de sélection des mots déclencheurs (Gen_Auto, Gen_Man, QW) pour la détection des phrases vitales

4.3.2 Comparaison des différentes configurations de détection de la nouveauté

La figure 3 compare les différentes configurations de détection de la nouveauté. L'application du module de nouveauté permet d'améliorer la $precision_N$ en pénalisant le rappel. Combiner

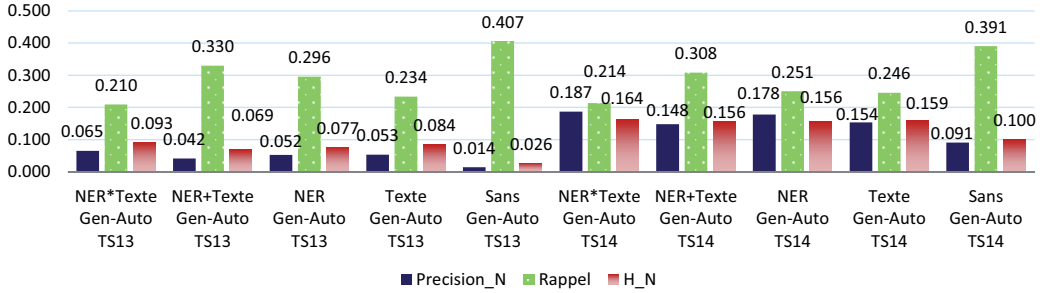


FIGURE 3 – Comparaison des différentes méthodes de détection de la nouveauté

la similarité textuelle avec la reconnaissance d'instances liées ***NER*Texte*** donne une meilleure moyenne harmonique (H_N) entre le rappel et la précision pour les instances de 2013 et 2014. Utiliser la stratégie *NER+Texte* reste utile si nous privilégions l'exhaustivité de la connaissance à extraire pour l'instance.

4.3.3 Comparaison de notre système par rapport aux systèmes participant à la tâche

Dans cette section, nous comparons notre système par rapport aux meilleurs systèmes ayant participé à la tâche en utilisant l'outil d'évaluation officiel [5] avec les jugements de pertinences officiels. Les mesures *ELG* et *LC* sont similaires aux mesures de précision et rappel respectivement mais en pénalisant la redondance et la latence lors de la détection d'informations [Aslam *et al.* (2013)]. Notre système aurait pu être classé premier (/7 participants) dans la tâche de TS 2013, et troisième (/6 participants) pour l'année 2014. Notre système apparaît donc comme efficace pour la détection de phrases vitales pour la mise à jour de bases de connaissances.

TS 2013				**TS 2014**			
Système	**ELG**	**LC**	**H-ts**	**Système**	**ELG**	**LC**	**H-ts**
*Gen-Auto ; Text*NER*	0.1102	0.1986	**0.1355**	cunlp	0.0631	0.322	**0.1162**
Gen-Auto ; Text+NER	0.0768	0.2619	0.1188	BJUT	0.0657	0.4088	0.1110
ICTNET	0.0794	0.3636	0.1078	*Gen-Auto ; Text*NER*	0.0881	0.1646	0.1047
PRIS	0.136	0.195	0.1029	uogTr	0.0467	0.4453	0.0986
HLTCOE	0.0522	0.2834	0.0827	*Gen-Auto ; Text+NER*	0.0712	0.2181	0.0963

TABLE 3 – Comparaison de notre système par rapport aux systèmes participant à la tâche TS 2013 et 2014. H-ts est la moyenne harmonique entre ELG et LC.

4.3.4 Rapidité de notre approche par rapport aux mises à jour de Wikipedia

La figure 4 compare la rapidité de notre système (Gen-Auto ; NER*Texte) à détecter les informations vitales pour les 24 événements par rapport aux mises à jour Wikipedia. Notre système permet de détecter 67% d'informations vitales avant que celles-ci soient mises à jour dans Wikipedia. La moitié des informations sont détectées par notre système 7 heures (au moins) avant qu'elles ne soient mises à jour dans Wikipedia. En moyenne, notre système permet de gagner 18 heures.

Dans le tableau 4, nous illustrons quelques exemples d'informations vitales détectées par notre système avant qu'elles soient ajoutées dans Wikipedia.

5. http ://www.trec-ts.org/documents

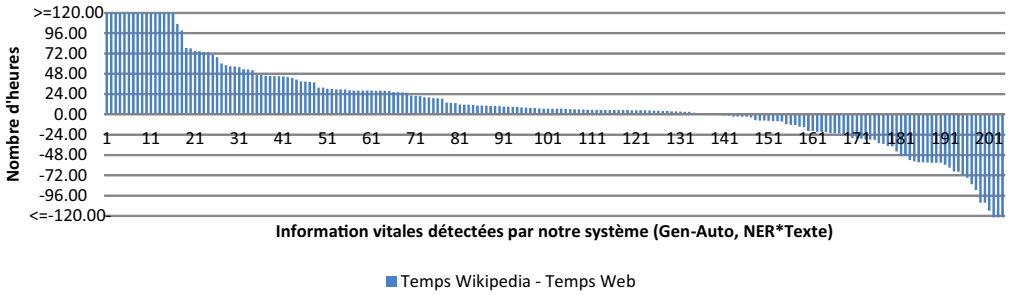


FIGURE 4 – Évaluation de la rapidité de notre système (Gen-Auto ; NER*Texte) par rapport aux mises à jours Wikipedia

Id de l'instance	*Information vitale détectée*	t_{web}	t_{wp}	t_{IB}	*Gain*
1	*550 injured*	22-02-12 16 :05	22-02-12 22 :49	22-02-12 22 :49	6.7h
1	*crashed at speed of 26 kilometers per hour*	22-02-12 22 :21	22-02-12 23 :01	-	0.67h
9	*39 casualties reported in Guatamala*	08-11-12 00 :33	08-11-12 04 :33	08-11-12 04 :33	1h
9	*48 casualties reported*	08-11-12 07 :42	08-11-12 07 :55	08-11-12 07 :55	0.22h
19	*Early modest estimates put over 5000 people in the streets of Romanian cities*	16-01-12 03 :58	18-01-12 02 :28	-	46.5h
19	*Queensland floods*	27-01-13 11 :35	24-01-13 22 :42	-	60.8h

TABLE 4 – Exemple d'informations vitales détectées par notre approche (Gen-Auto, NER*Texte). t_{web}, t_{wp}, t_{IB} représentent les temps de la disposition de l'information par notre système, dans Wikipedia et dans les infoboxes de Wikipedia respectivement.

La figure 4 et les exemples du tableau 4 montrent que les informations sont généralement publiées dans le documents Web (presse, blogs, etc.) avant qu'elles soient éditées dans les encyclopédies collaboratives comme Wikipedia. Notons que la mise à jour n'est pas forcément reportée dans les InfoBoxes principalement exploités pour enrichir DBpedia. Bien que les instances analysées représentent des événements largement connus, qui intéressent plusieurs contributeurs, nous remarquons toujours un temps de latence. Ce temps de latence devrait être plus grand pour des instances "moins connues".

5 Conclusion

Dans ce travail, nous avons proposé une méthode qui permet d'extraire les informations vitales au fur et à mesure qu'elles apparaissent dans le Web. L'importance de ce type d'approches nous paraît cruciale afin d'accélérer la mise à jour des bases de connaissances. Ces approches sont utiles non seulement pour aider à la mise à jour de documents collaboratifs décrivant des instances comme les pages wikipedia, mais aussi pour la mise à jour des bases de connaissances elles-mêmes car elles permettent d'identifier les phrases spécifiques pouvant ensuite être analysées par des extracteurs (tels que ceux décrit dans Zablith *et al.* (2015)) pour enrichir la base. L'expérimentation que nous avons menée montre que des mises à jour plus fines de DBpedia pourraient notamment être mises en oeuvre par l'identification en temps réel de phrases vitales issues du web dont l'information ne se trouve pas toujours dans l'infoBox. Nous souhaitons à très court terme poursuivre les évaluations de notre système en utilisant des outils d'extractions de connaissances sur les phrases vitales retrouvées. Nous souhaitons également trouver un autre cadre d'évaluation pour lequel des documents et une base de connaissances dans laquelle des connaissances plus formellement représentées sont disponibles.

Références

ABBES R., PINEL-SAUVAGNAT K., HERNANDEZ N. & BOUGHANEM M. (2013). IRIT at trec knowledge base acceleration 2013 : Cumulative citation recommendation task. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA.

ABBES R., PINEL-SAUVAGNAT K., HERNANDEZ N. & BOUGHANEM M. (2015). Leveraging temporal expressions to filter vital documents related to an entity. In *ACM Symposium on Applied Computing (SAC) (to appear)*.

ASLAM J., DIAZ F., EKSTRAND-ABUEG M., PAVLU V. & SAKAI T. (2013). Trec 2013 temporal summarization. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, USA.

AUGENSTEIN I., PADÓ S. & RUDOLPH S. (2012). Lodifier : Generating linked data from unstructured text. In *Proceedings of the 9th International Conference on The Semantic Web : Research and Applications*, ESWC'12, p. 210–224, Berlin, Heidelberg.

EXNER P. & NUGUES P. (2012). Entity extraction : From unstructured text to dbpedia rdf triples. WoLE'12.

FRANK J. R., BAUER S. J., KLEIMAN-WEINER M., ROBERTS D. A., TRIPURANENI N., ZHANG C. & RE C. (2013). Evaluating stream filtering for entity profile updates for trec 2013. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA.

FRANK J. R., KLEIMAN-WEINER M., ROBERTS D. A., NIU F., ZHANG C., RE C. & SOBOROFF I. (2012). Building an Entity-Centric stream filtering test collection for TREC 2012. In *Proceedings of the Text REtrieval Conference (TREC)*.

LEHMANN J., ISELE R., JAKOB M., JENTZSCH A., KONTOKOSTAS D., MENDES P., HELLMANN S., MORSEY M., VAN KLEEF P., AUER S. & BIZER C. (2014). DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

LIU Q., LIU Y., WU D. & XUEQI C. (2013). Ictnet at temporal summarization track trec 2013. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA.

MCCREADIE R., MACDONALD C. & OUNIS I. (2014). Incremental update summarization : Adaptive sentence selection based on prevalence and novelty. In *Proceedings of the 23rd International Conference on Conference on Information and Knowledge Management*, p. 301–310, New York, USA.

PETASIS G., KARKALETSIS V., PALIOURAS G., KRITHARA A. & ZAVITSANOS E. (2011). Ontology population and enrichment : State of the art. In *Knowledge-driven multimedia information extraction and ontology evolution*, p. 134–166 : Springer-Verlag.

WANG J., SONG D., LIN C.-Y. & LIAO L. (2013). BIT and MSRA at trec kba ccr track 2013. In *Notebook of the TExt Retrieval Conference 2013 (TREC 2013)*, Gaithersburgh, MD, USA.

XU T., MCNAMEE P. & W.QARD D. (2013). HLTCOE at TREC 2013 : Temporal summarization. In *Proceedings of the Text REtrieval Conference*, Gaithersburgh.

ZABLITH F., ANTONIOU G., D'AQUIN M., FLOURIS G., KONDYLAKIS H., MOTTA E., PLEXOUSAKIS D. & SABOU M. (2015). Ontology evolution : a process-centric survey. *The Knowledge Engineering Review*, **30**(01), 45–75.

ZHAI C. & LAFFERTY J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, p. 334–342, New York, NY, USA.

ZHANG C., XU W., MENG F., LI H., WU T. & XU L. (2013). The information extracion systems of pris at temporal summarization track. In *Proceedings of the Text REtrieval Conference*, Gaithersburgh.

SPARE-LNC : un langage naturel contrôlé pour l'interrogation de traces d'interactions stockées dans une base RDF

Bryan Kong Win Chang, Marie Lefevre, Nathalie Guin, Pierre-Antoine Champin

UNIVERSITÉ DE LYON, CNRS
Université Lyon 1, LIRIS, UMR 5205, F-69622, France

Résumé : Les traces issues de l'interaction d'un utilisateur avec un environnement informatique sont une source d'informations non négligeable concernant l'utilisation de celui-ci, si l'on est en mesure de les étudier correctement. Dans cet article, nous proposons un langage naturel contrôlé permettant à des analystes non-informaticiens d'interroger des traces stockées dans une base RDF. Nous expliquons les apports de ce langage, sa mise en œuvre ainsi que les évolutions envisagées suite aux premières mises à l'essai.

Mots-clés : Langage Naturel Contrôlé, Génération de Requêtes SPARQL, RDF, Interrogation de Traces

1 Introduction

Les traces sont communément définies comme le résultat d'une activité. Si l'on considère par exemple un logiciel informatique, une première définition des traces serait tout simplement ce que le logiciel produit, ce que le logiciel permet de réaliser comme action, ce qui résulte de cette action, comme par exemple un affichage visuel, mais aussi d'autres informations qui ne sont pas directement transmises à l'utilisateur. On retrouve ainsi dans ces traces au sens large les *logs*, ou encore les paramétrages d'un utilisateur concernant le logiciel.

Ces traces qui sont des restes, enregistrements ou témoins de l'activité qui a eu lieu, sont une source d'information primordiale pour différents utilisateurs de ces logiciels. Elles peuvent être utilisées en tant que source brute d'information, ou au contraire être traitées par un système tiers (ou même le système les générant) afin de réaliser d'autres actions.

Un exemple simple d'utilisation concerne par exemple les *logs*, qui peuvent contenir des éléments divers sur la manière dont s'est déroulé le programme et ainsi aider à le corriger lorsqu'il ne s'est pas exécuté correctement. Ici, l'expert accède directement au contenu des traces et agit en conséquence. D'autres systèmes utilisent également des mécanismes automatisés afin de fournir des données personnalisées aux différents acteurs du logiciel. Par exemple dans le domaine du jeu vidéo, les traces du joueur peuvent être utilisées pour adapter le comportement des personnages non-joueurs (Rubin & Ram, 2012). Sur le web, les traces peuvent être analysées pour fournir diverses recommandations (Zarka, 2013). Dans le domaine des *Environnements Informatiques pour l'Apprentissage Humain* (*EIAH*), l'analyse des traces représente un enjeu majeur pour la compréhension du comportement de l'apprenant et l'évaluation de ses connaissances afin de pouvoir fournir à ce dernier un accompagnement personnalisé lors de son apprentissage.

Cependant, malgré l'omniprésence des traces dans les logiciels, il est rare que celles-ci soient récoltées et stockées selon le même modèle et dans le même format. Or pour certains éléments de traces, et pour les traces en général, étant donné un domaine d'utilisation, on retrouve des mêmes besoins. Pour les logiciels éducatifs, ces besoins ont d'ailleurs donné lieu à des études afin de formaliser ce que sont ces données (nommées *indicateurs* (Dimitrakopoulou *et al.*, 2006)) que l'on calcule ou récupère afin d'instancier une représentation de ce que l'on sait sur un apprenant donné, le *profil d'apprenant*. Cependant, même si ces éléments ont pu faire l'objet d'une formalisation et d'une classification, la récupération des informations contenues dans les traces reste à la discrétion de ceux qui conçoivent le système.

Un premier pas vers l'homogénéisation des traces et l'explicitation de leur modèle a été proposé avec la notion de Système à Base de Traces (*SBT*). Le nom de Système à Base de Traces provient d'un parallèle avec les Systèmes à Base de Connaissances (Settouti *et al.*, 2009), où les traces viennent remplacer les connaissances. Une première proposition de ce formalisme est présenté par (Settouti *et al.*, 2009). Le méta-modèle proposé accorde une place prépondérante à la notion du temps, les traces étant représentées comme un ensemble d'observables situés dans le temps. Ce formalisme s'accompagne d'une étude des technologies potentielles pour l'implémentation de ces traces, qui a abouti à la création du kTBS.

KTBS, *kernel for Trace-Based System*, est ainsi une implémentation d'un SBT permettant de stocker les traces dans un ensemble de triplets RDF, l'ensemble agissant comme un RDF-store pour l'accès aux données (Champin *et al.*, 2011). Les requêtes de modification, ajout et récupération des données se font alors *via* le langage SPARQL. Toutefois, bien que le RDF et son langage associé SPARQL (version 1.1) satisfaisaient la majeure partie des besoins liée à l'exploitation des traces (Settouti, 2011), l'utilisation de ces technologies pose un nouveau problème à la démocratisation de l'usage du kTBS : comment permettre à des utilisateurs non-informaticiens d'accéder et d'utiliser le contenu des traces pour récupérer les informations qui les intéressent ?

Afin de répondre à cette problématique, nous présentons dans cet article un langage naturel contrôlé permettant l'interrogation des traces respectant le modèle SBT dans des triplestores [1]. La deuxième section de cet article est ainsi consacrée à un état de l'art concernant les outils existants pour l'interrogation des traces et des triplestores. Ensuite, nous présentons le langage en lui-même et son implémentation, avant de conclure par une discussion sur les utilisations actuelles puis par des perspectives de recherche.

2 État de l'art

Dans le but d'interroger une architecture basée sur RDF, il est pertinent de se pencher sur les différentes techniques utilisées pour faciliter l'accès de ces données à ce qui est appelé classiquement "l'utilisateur lambda" du web. Abstraire des requêtes SPARQL est en effet une problématique qui a déjà vu de nombreuses propositions plus ou moins réussies, et fait l'objet de plusieurs concours récompensant les outils de recommandation et de recherche permettant à l'utilisateur d'obtenir le plus facilement l'information qu'il cherche (Lopez *et al.*, 2013).

1. Un triplestore est une base de données spécialement conçue pour le stockage et la récupération de données RDF.

Les approches proposées afin de résoudre ce problème sont diverses. Une d'entre elle est l'approche des moteurs de recherche, dont l'utilisation intuitive fonctionne par la saisie d'un certain nombre de mots liés à la recherche, que chaque moteur interprétera à sa manière pour en sortir une liste de suggestions que l'utilisateur final est libre de choisir.

Une autre approche consiste à considérer un ensemble de requêtes que l'utilisateur final sera potentiellement amené à vouloir réaliser sur une base de données (Pradel *et al.*, 2012). Les principes de l'approche sont indépendants du domaine d'application, mais l'ensemble des patterns à définir est un travail d'ingénierie supplémentaire lourd pour chaque nouveau domaine.

Une dernière approche consiste à proposer des Langages Naturels Contrôlés (*LNC*). Il s'agit ici de restreindre le langage naturel pour en faire un langage utilisable sans ambiguïté. Les approches de ce genre sont beaucoup étudiées en anglais, où sont développés depuis longtemps des analyseurs sémantiques du langage (Montague, 1973). Par exemple, SQUALL (Ferré, 2012) est un langage, qui s'inspire de la théorie de Montague "Universal Grammar", théorie appuyant la possibilité de considérer les langages naturels et machines de la même manière (Montague, 1973). Sa théorie explique cette possibilité *via* la mise en application de procédés algorithmiques simples pour passer d'un langage naturel à des langages formels simples, ayant mené aux parsers de Montague. SQUALL utilise ainsi cette base pour traduire des questions posées en Anglais sur des données de DBpedia.

L'article présentant SQUALL est suivi en 2013 d'un article (Ferré, 2013) où les difficultés rencontrées lors de la traduction du langage au SPARQL sont explicités et où il est montré que l'expressivité de son langage est similaire à celui du SPARQL .

Dans le cadre d'une utilisation associée au kTBS, le langage SQUALL propose un langage indépendant de tout domaine, en se calquant sur le côté triplet "sujet prédication objet" du SPARQL lors de la traduction en langage formel. Cependant, le temps ou la date des informations ne sont pas considérées de manière particulière. Par exemple, le "Quand" ("When") est ainsi absent du langage. Cependant il reste un des langages les plus proches de la formulation utilisateur.

En parallèle des approches du web sémantique, de nombreux travaux ont été proposés dans le domaine des *EIAH* concernant la définition des indicateurs à partir des traces. On peut citer la capitalisation des requêtes, comme dans les travaux concernant la génération des indicateurs (Choquet & Iksal, 2007; Diagne, 2009; Gendron, 2010; Djouad, 2011). Il y a également les modèles qui servent à guider l'utilisateur dans son raisonnement pour la construction d'indicateurs (Dimitrakopoulou *et al.*, 2006) utilisés dans ces mêmes outils. Toutes ces approches présentent le défaut d'être liées à leur application ou à des contraintes supplémentaires concernant le contenu des traces, là où le kTBS recherche une indépendance au domaine.

Dans la section suivante, nous présentons notre Langage Naturel Contrôlé, indépendant du domaine et permettant de répondre à la limite de la prise en compte du temps induite par l'utilisation de SPARQL.

3 SPARE-LNC

Afin de présenter le langage, nous commençons par aborder le modèle des données à interroger puis le type du langage et les choix effectués afin de représenter les règles contraignant le LNC. Une fois les règles introduites, nous présenterons par des exemples les bases du langage,

avant d'introduire progressivement les différentes spécificités de ce dernier, pour terminer par l'implémentation du langage que nous avons effectuée.

3.1 Les données interrogées

Les traces du kTBS présentent un certain nombre de spécificités qui sont héritées du méta-modèle (Settouti *et al.*, 2009) dont il est issu. Une trace dans le kTBS est ainsi principalement composée d'un ensemble d'éléments indépendants, que l'on nomme des *obsels* (par contraction de *"observable elements"*). Chacun de ces obsels dispose d'attributs. Certains attributs sont présents dans tous les obsels, comme par exemple une date de début et une date de fin, ou encore le type de l'obsel. Chaque obsel peut ensuite posséder un nombre non défini d'attributs constituant les éléments considérés comme dépendants du domaine. Dans une logique de représentation RDF, ces obsels forment alors un ensemble de triplets liés par un sujet RDF unique pour un obsel donné. Les traces ayant pour but de récolter l'activité d'un individu, et selon ce que l'expert cherche à analyser, le détail et la durée d'une expérimentation, les traces peuvent rapidement atteindre une masse importante en terme de nombre de triplets RDF.

Afin de définir les bases de notre langage SPARE, il a fallu principalement s'intéresser à deux choses. La première concerne les fonctionnalités que le langage doit proposer afin de satisfaire les besoin exprimés concernant les interrogations de traces. La seconde est la forme que doit revêtir le langage, afin d'assurer autant que possible une utilisabilité convenable pour des non-informaticiens.

Dans le kTBS, bien qu'un modèle de traces décrivant les obsels pouvant être rencontrés dans la trace puisse être fourni, il n'est pas obligatoire. Il est possible de générer ce dernier à partir des traces, mais la génération *a posteriori* implique le risque de voir apparaitre des obsels dont le contenu n'est pas décrit dans le modèle généré car absent lors de sa génération. Par contre, le modèle de ces traces étant contraint par le méta-modèle du kTBS, nous avons choisi de nous intéresser plus particulièrement à ce dernier afin d'exprimer les besoins du langage.

Pour la forme, l'approche retenue est de se baser sur un langage d'interrogation textuel, et plus précisément un *LNC*. Si des interfaces graphiques permettant l'interrogation de traces existent (Settouti *et al.*, 2009), ces dernières sont généralement d'une manière ou d'une autre liées à leur domaine d'utilisation voire à l'application elle-même. Dans la mesure où aucune interface générique n'est associée au kTBS pour l'interrogation des traces, utiliser un langage textuel parsé tel qu'un *LNC* présente un certain nombre d'avantages, le premier étant la liberté de l'interface. À partir d'un langage n'ayant besoin que de texte, il est possible de définir des interfaces allant d'un simple champ textuel à compléter, à une interface graphique plus complète où le langage remplace simplement le SPARQL dans son utilisation. De plus, pour un domaine particulier, il est possible de reprendre le langage et de bâtir une interface adaptée. Le texte offre également une facilité supplémentaire pour l'échange de requêtes, en fournissant un format directement capitalisable, et dont le contenu, se basant sur le langage de l'utilisateur, est facilement compréhensible par l'utilisateur.

En résumé, dans le cadre d'une utilisation typique de notre approche (*cf. Figure 1*), afin d'utiliser le langage, l'utilisateur, à travers une interface personnalisée répondant aux besoins de l'application, choisit de manière directe (en écrivant les requêtes en LNC) ou indirecte (en utilisant des requêtes générées ou pré-écrites) une requête ou un ensemble de requêtes. Ces requêtes sont ensuite analysées par l'interpréteur du langage (ou parser), qui traduit la requête

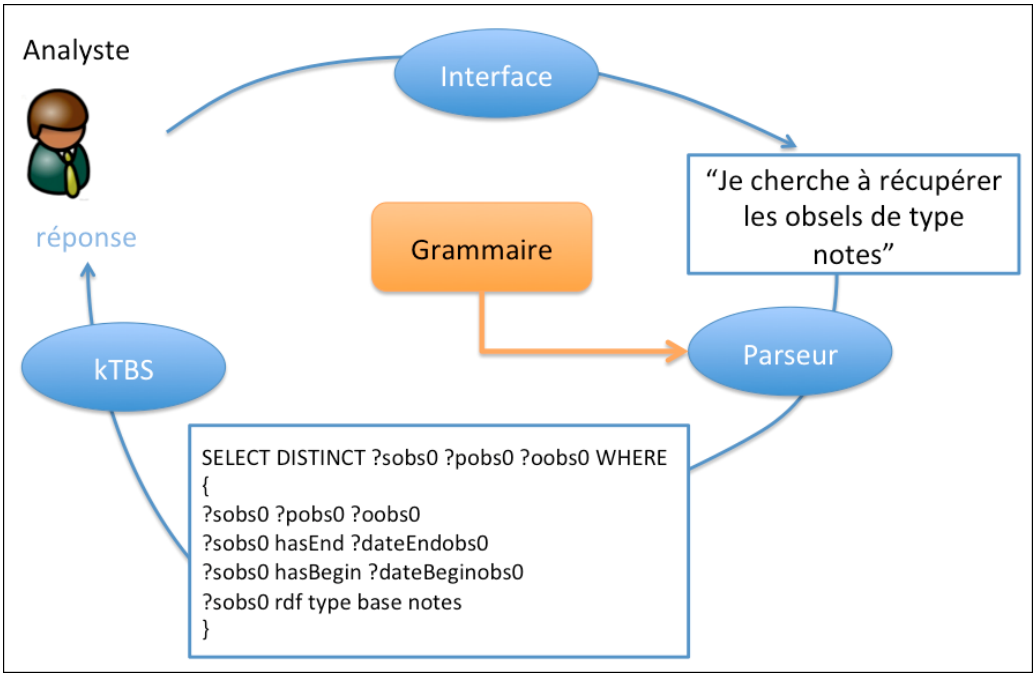


FIGURE 1 – Interrogation d'une trace RDF par l'utilisateur *via* SPARE-LNC

de l'utilisateur dans le langage compréhensible par le kTBS, le SPARQL. En réponse, le kTBS interprète la requête et retransmet la réponse à l'utilisateur.

3.2 Une grammaire pour SPARE-LNC

SPARE-LNC est l'abréviation pour SPArql REquest en Langage Naturel Contrôlé. Le principe du langage est ainsi de proposer une alternative au SPARQL pour interroger les traces stockées dans le système à base de traces kTBS. Pour formaliser le *LNC*, nous avons choisi d'opter pour un langage guidé par une grammaire. La formalisation du langage se base ainsi sur une grammaire algébrique. Le langage utilisé comme base est le français, mais l'adaptation de la grammaire pour de l'anglais, après première étude, ne semble pas poser de problèmes majeurs. L'entrée de l'utilisateur est analysée dans son intégralité *via* l'ensemble de règles composant la grammaire. L'ordre des règles n'est pas absolu, permettant certaines libertés. La *Figure 2* montre une version simplifiée de la grammaire de SPARE-LNC. La version complète de la grammaire est disponible en ligne (Kong Win Chang *et al.*, 2014). On peut différencier dans la définition de la grammaire deux groupes de règles. Le premier groupe (partie haute de la *Figure 2*) permet la création de phrases exprimant des conditions sur des éléments à récupérer dans les traces. Le deuxième permettant la définition de phrases gérant les éléments récupérés, par exemple en opérant des calculs ou des sélections sur ce qui est récupéré.

Le langage utilisé pour requêter la base est ainsi composé d'un ensemble de ces phrases qui satisfont la grammaire proposée. Chacune de ces phrases correspond ainsi à une sous-requête du langage, formant un texte décrivant les données à récupérer en énonçant un ensemble de contraintes et un ensemble d'actions à réaliser.

3.3 Contraintes sur les données

Les sous-requêtes exprimant les contraintes sur les données sont à la base du système d'interrogation. La phrase doit être simple et compréhensible par tout un chacun. La sous-requête la plus simple est ainsi composée de l'unique texte suivant : "Je cherche à récupérer les obsels.". Afin de faciliter l'utilisation *via* le texte, un certain nombre de phrases équivalentes existe dans le langage. Ainsi "Je cherche à récupérer les obsels" dispose d'un autre équivalent "Je veux récupérer les obsels".

Bien sûr, cet exemple de sous-requête ne permet pas vraiment de contraindre le contenu récupéré depuis la base. Toutefois il peut être intéressant de s'intéresser à une variante utilisable, "Je cherche à compter les obsels.", qui permet elle plus d'utilisations, comme par exemple, calculer une moyenne de notes ou compter le nombre d'évènements de la base.

```
liste_requête  = début* select?
début          = début_select1
               / début_nommé1
               / début_soit
               / début_parmi

début_select1  = "Je cherche à " action nommage point
début_nommé1   = "Je nomme" NOM_RES "les obsels" conditions_obsels* point
début_parmi    = "Parmi " NOM_RES début_select1

action         = " récupérer les obsels " conditions_obsels*
               / "récupérer les attributs" NOM_ATT? conditions_attributs*

conditions_obsels = " de type " NOM_TYPE
                  / " ayant un attribut " NOM_ATT?
                  / "suivi" within? "par un obsel " condobs_temp*

conditions_attributs = VALL_ATT
                     / "de valeur" VALL_ATT
                     / VALL_ATT "de " NOM

within           = " dans l'heure "
                 / " dans les " VALEUR " secondes qui suivent"
                 / " dans les " VALEUR " milisecondes qui suivent"

point            = " . "
point_virgule    = " ; "
```

```
select         = "Je sélectionne les requêtes : " (NUM, " : ")
                 + ( NOM, " : " ) "."

début_soit     = " Soit " NOM_RES action point
               / " Soit " NOM_RES " = " equation point_virgule
```

FIGURE 2 – Extrait court de la grammaire de SPARE-LNC

3.3.1 Les attributs

À ces sous-requêtes de base s'ajoutent les conditions les plus simples : les conditions sur les attributs, par exemple, ou sur le type de l'obsel. Pour ajouter une condition sur un attribut, l'utilisateur doit globalement pouvoir, d'une part, rechercher les obsels possédant un attribut particulier, et d'autre part, pouvoir poser une condition sur sa valeur. Le premier cas s'exprime par "Je veux récupérer un obsel ayant un attribut Notes", à qui on peut rajouter "de valeur

supérieure à 10." par exemple. Il est à noter que le nom de l'attribut est optionnel, et l'utilisateur pourra écrire : "Je veux récupérer les obsels ayant un attribut de valeur supérieure à 10.". Bien sur, cela implique que l'on récupérera potentiellement, si on ne précise rien d'autre, d'autres événements qui disposent d'un attribut numérique répondant à la condition.

Certains attributs sont communs à toutes les traces. Ces derniers étant des éléments importants pour l'interrogation des traces, le langage propose des formulations propres afin d'interroger ces derniers. Par exemple pour récupérer un ensemble d'obsel d'un type donné, il faut interroger l'attribut commun à toutes les traces nommé *"hasType"*. L'utilisateur pourra écrire "Je veux récupérer les obsels de type contrôle." au lieu de "Je veux récupérer les obsels ayant un attribut hasType de valeur contrôle.". Un autre ensemble d'attributs communs a eu droit à une considération toute particulière de par leur place centrale pour les traces du kTBS : l'ensemble des attributs temporels.

3.3.2 Les opérateurs temporels

Dans le but de pouvoir extraire des données des traces, les notions de position temporelle relative ou absolue d'un événement par rapport à un autre sont cruciales pour l'expression des conditions. L'expression des opérateurs permettant la vérification temporelle n'est pas impossible en SPARQL. Cependant, si l'on souhaite exprimer "un obsel A suivant un obsel B", l'utilisateur se retrouvera à écrire un ensemble de triplets conséquent, qui de plus ne sont pas forcément intuitifs.

Pour palier ce problème, nous avons décidé d'implémenter ces opérateurs temporels dans notre langage, en prenant comme base les opérateurs de Allen (Allen, 1983) qui expriment un ensemble de 13 relations possibles entre deux événements exprimés sous la forme d'intervalles. Ces treize opérateurs sont en fait sept opérateurs et leurs opposés, l'un d'entre eux étant son propre opposé (l'égalité). Ces opérateurs sont : *before, contains, overlaps, meets, starts, finishes, equals, after, during, overlapped by, met by, started by, finished by*. Chacun des opérateurs a son équivalent dans le langage SPARE-LNC. Un obsel de type A suivant un obsel de type B s'écrira tout simplement : "Je cherche à récupérer les obsels de type A suivi par un obsel de type B."

3.3.3 Liens et références

Si l'on utilise l'ensemble des contraintes exprimées ci-dessus dans une seule et même sous-requête, on risque de se retrouver avec des sous-requêtes extrêmement longues. C'est pourquoi nous proposons des possibilités de définir des sous-résultats durant la définition de la requête *via* un système de nommage. La sous-requête "Je veux récupérer les obsels de type contrôle ayant un attribut note de valeur supérieure à 10." peut se diviser et devenir : "Je nomme Controle les obsels de type contrôle. Parmi Controle, je cherche à récupérer les obsels ayant un attribut note de valeur supérieure à 10.". *Controle* devient l'identifiant de la sous-requête, et peut ainsi être repris comme entrée pour d'autres sous-requêtes.

Il existe ainsi dans le langage trois moyens de nommer un résultat. Le premier est de commencer la phrase par *"Je nomme"* avant de donner l'identifiant. La deuxième manière est de finir par *"que je nomme"*. Bien que la formulation ne soit pas forcément la plus lisible, cela permet de rajouter rapidement un identifiant à une phrase n'en disposant pas sans toutefois trop altérer la lisibilité. Enfin, la troisième façon est similaire à *"Je nomme"*, et utilise le mot clef *"Soit"*.

"Soit" a une autre utilité. Une phrase commençant par *"Soit"* a les mêmes possibilités de base qu'un *"Je nomme"*, mais permet également d'introduire des calculs mathématiques.

Par exemple, soit la requête suivante : "Je nomme Notes les obsels de type note. Parmi Notes, je cherche à récupérer les attributs 'resultat' que je nomme A. Soit Moyenne = AVG(A) ;" On récupère les obsels nous intéressant dans les deux premières phrases avant de calculer la moyenne dans la troisième. Les expressions mathématiques acceptées par *"Soit"* sont les mêmes que celles du SPARQL, et ont les mêmes contraintes de variables que le langage originel. Du fait que le point est utilisé en RDF pour l'expression de certaines valeurs numériques, une expression mathématique utilisant le mot clef *"Soit"* se termine par un point virgule au lieu d'un point.

Les contraintes sur les opérations possibles entre deux types de variables SPARQL se retrouvent aussi lors de la spécification de sous-requêtes dépendantes d'une précédente. Si on reprend le résultat d'une sous-requête à l'aide du mot clef *"Parmi"*, comme dans l'exemple précédent, "Notes" fait référence à l'ensemble des observables de type "note" et ne peut pas être l'objet d'une opération arithmétique. De même, "A" est un ensemble de variables numériques, et ne peut pas faire l'objet de filtres supplémentaires.

Définir un ensemble de sous-requêtes n'est cependant pas suffisant pour renvoyer un résultat en particulier. Le résultat voulu par l'utilisateur n'est pas forcément le résultat d'une des sous-requêtes, mais potentiellement plusieurs. Par défaut, le langage considère tous les résultats des sous-requêtes comme résultat potentiel pour l'utilisateur. Le résultat par défaut est ainsi l'union des résultats des sous-requêtes. Pour définir ce qu'il faut récupérer, il faut utiliser une requête spécifique indiquant quelles variables récupérer, qui s'écrit tout simplement *"Je garde uniquement ..."* suivi de la liste des éléments à récupérer. Ces éléments de la liste peuvent être exprimés de deux manières, soit par l'utilisation des identificateurs proposés par l'utilisateur lors de la définition de la sous-requête (*via* le mécanisme de nommage), soit par des numéros de phrase dans l'ordre de la déclaration.

3.4 Implémentation

Pour une première implémentation du langage, et voulant appliquer les technologies du web aux traces stockées sous format RDF sur un kTBS en ligne dans le cadre d'un MOOC (Massive Open Online Course), le choix s'est dirigé pour une implémentation javascript du parser de grammaire avec du html pour les interfaces. Afin d'utiliser notre langage, l'utilisateur manipule ainsi une interface html qui lui permet de définir une requête du langage, qui est ensuite traduite en requête compréhensible par le kTBS, qui interprète et renvoie à l'utilisateur sa réponse (*cf. Figure 1*) .

Pour implémenter le parseur, nous avons utilisé un outil de génération de parser pour javascript [2] utilisant en entrée un pseudo-code javascript contenant les règles du langage. Le parseur résultant est structurellement une fonction javascript facilement utilisable prenant en paramètre une chaine de caractères contenant la requête en langage naturel contrôlé et ayant pour sortie une requête SPARQL directement utilisable sur le kTBS.

Le parser procède par sous-requêtes, avant de construire les liens entre les différentes sous-requêtes grâce au système d'identifiants. Ainsi, le parser extrait l'ensemble des conditions expri-

2. http://pegjs.org/

mées par chaque phrase et construit une première table de liens qui contient les identifiants référençant des sous-requêtes trouvés lors de l'analyse et une deuxième table contenant l'expression SPARQL des conditions. Après une première analyse du texte pour récupérer l'ensemble de ces tables, une deuxième lecture est réalisée pour résoudre les liens entre les sous-requêtes. Cette deuxième passe opère par un parcours simple dans les tables de liens en ajoutant les tables référées à la table en cours de traitement. La requête finale SPARQL est ensuite extraite soit par la fusion des tables dans le cas où la sous-requête de sélection est absente, soit par la sélection exprimée par la sous-requête de sélection.

Nous avons également codé deux interfaces pour le langage [3]. Une première est une simple interface textuelle, qui se caractérise par l'usage de plusieurs champs textes, dont un pour l'écriture de la requête et l'autre pour l'affichage de la requête SPARQL correspondante. Cette première interface textuelle est pratique, mais l'utilisation nécessite de connaître le langage. La deuxième interface est une interface graphique pour construire des sous-requêtes du langage. Les différents mots du langages sont accessible *via* des boutons qui les ajoutent à une liste pour construire la requête. Chacune des sous-requêtes peut être sélectionnée par l'utilisateur et ce dernier peut rajouter autant de sous-requêtes qu'il veut par l'utilisation des boutons ajoutant les débuts de sous-requête. Lorsque une sous-requête est sélectionnée, l'interface désactive les boutons qui ne permettent pas de construire une sous-requête grammaticalement correcte. Une fois la requête terminée, l'utilisateur peut interroger un kTBS. Il est également possible pour l'utilisateur de consulter la requête en langage naturel contrôlé et la requête SPARQL. L'affichage de la requête à l'utilisateur lui permet d'avoir accès au contenu de la requête et de la copier ou de la modifier directement s'il le souhaite.

4 Utilisation du langage et limitations

Le développement du parser a suivi la logique du développement progressif des fonctionnalités du langage. Les premières fonctionnalités ont été la récupération des obsels sans conditions et la possibilité de compter le nombre d'obsels d'une trace. Chaque étape d'ajout d'une nouvelle fonctionnalité se suivait d'une étape de validation de la requête SPARQL générée. La validation consistant à vérifier que la requête générée est valide et génère le résultat souhaité exprimé par SPARE-LNC. La vérification des fonctionnalités a eu lieu en premier lieu sur des bases de traces jouets contenant l'ensemble des cas recherchés pour tester la fonctionnalité nouvellement implémentée, puis sur des bases de traces d'un MOOC [4] utilisant le kTBS. La mise à disposition de ces bases de traces a notamment permis de débuter les tests du langage dans le cadre de la création d'indicateurs d'EIAH (trois pour le moment), retranscrits *via* les deux interfaces.

Cependant l'utilisation du langage par des informaticiens ou des non-informaticiens a posé un certain nombre de problèmes et de questions concernant l'expressivité du langage. Un des premiers points concerne l'équivalence entre SPARE-LNC et le SPARQL. Si le langage naturel contrôlé SPARE permet de s'affranchir du SPARQL et de n'écrire qu'un ensemble de phrases en français, il existe des opérations que SPARE-LNC ne couvre pas. L'une de ces opérations

3. Les interfaces de manipulation du langage SPARE-LNC ainsi que des exemples de requêtes SPARE-LNC associées à leur équivalent SPARQL sont disponibles sur http://liris.cnrs.fr/~bkongwin/SPARE/

4. MOOC FOVEA, Session primtemps 2014, environ 4000 inscrits : http://anatomie3d.univ-lyon1.fr/webapp/website/website.html?id=3346735&pageId=223206

manquantes est par exemple la possibilité d'ordonnancer la réponse de la requête par rapport à certains critères. Trier un ensemble de notes directement *via* une seule sous-requête n'est pas possible, bien qu'il soit possible d'utiliser un moyen dévié, comme récupérer une par une les notes de l'ensemble en séparant l'ensemble entre la meilleure note et le reste. Ces opérations sont réalisables de façon simple sur les données finales, et doivent être considérées comme un prochain ajout au langage, bien qu'il faille réaliser un certain travail en amont. Car s'il est facile d'ajouter une règle au langage par le simple ajout de règles supplémentaires à la grammaire, il convient de s'assurer qu'une phrase grammaticalement correcte au sens de SPARE n'ait pas plusieurs interprétations.

Ce point tout particulier est un point d'importance majeure dans SPARE, où l'on cherche à permettre plusieurs formulations similaires. Par exemple, si l'on veut exprimer qu'un élément est précédé par un autre et se déroule en même temps qu'un troisième, la phrase SPARE-LNC correcte serait : "Je cherche à récupérer les obsels de type A suivi par un obsel de type B, pendant un obsel de type C.". Si maintenant on veut rajouter que l'événement de type B de notre exemple se déroule en même temps qu'un événement de type D et est suivi par un événement de type E, on devrait écrire : "Je cherche à récupérer les obsels de type A suivi par un obsel de type B pendant un obsel de type D et suivi par un obsel de type E, pendant un obsel de type C.". Nous voyons ici qu'il est possible d'avoir plusieurs interprétations d'une même phrase. Afin d'éviter des confusions, le changement d'un référant dans la phrase se fait *via* l'utilisation du mot clef *"lui-même"* que l'on ajoute après la déclaration du nouvel obsel. Bien sûr, cette solution ne fait que déplacer le problème à une ou deux conditions temporelles plus tard, c'est pourquoi nous conseillons de séparer la grande sous-requête en plusieurs requêtes plus petites pour éviter toute ambiguïté.

Une autre contrainte forte a aussi dû être posée lorsque l'on doit introduire des conditions sur des éléments RDF. Si le parser reconnaît les chaînes de caractères sans espace et les nombres sans problèmes, tout élément inclus dans les conditions contenant un caractère spécial ou un espace doit être mis entre guillemets simples. Un bon exemple serait si l'on veut récupérer tous les obsels du 21 janvier d'une personne du nom de Francis Dac. La requête s'exprimerait "Je veux récupérer les obsels ayant un attribut nom de valeur 'Francis Dac' et ayant un attribut date de valeur '21-01-2015'". Sans la contrainte concernant les guillemets afin de délimiter les valeurs des attributs, la phrase pourrait alors avoir plusieurs interprétations différentes, l'une d'elle étant de rechercher tous les obsels ayant un attribut appelé "nom" et ayant une valeur "Francis Dac et ayant un attribut date de valeur 21-01-2015".

Or cette utilisation d'informations provenant directement de la base RDF dans le langage SPARE-LNC a été entre autres l'un des principaux points de discussion lors de rencontres avec des utilisateurs potentiels. En effet, les éléments de la base du MOOC contenaient un très grand nombre de liens http qui étaient difficiles à manipuler pour un utilisateur non-informaticien. Ceci a mené à l'éventualité d'affecter à un ingénieur expert de la trace la tâche de mettre en place un équivalent entre certains éléments contenus dans la trace et une formulation adaptée pour l'utilisateur. Cependant, les formulations de ces équivalents étant dépendants à la fois du vocabulaire de l'utilisateur et de la formulation choisie dans les traces, ces adaptations seraient à effectuer du côté des interfaces de l'application ou du groupe d'applications utilisant ces adaptations.

Concernant la compréhension des sous-requêtes, la présence obligatoire des termes du méta-modèle ('obsels', 'attributs'...) dans les requêtes a également soulevé des inquiétudes. Ces no-

tions nouvelles d'obsels et d'attributs d'un obsel peuvent dans l'absolu être supprimées sans changer le sens de la requête. Cependant, si la phrase en SPARE-LNC "Je veux les obsels de type note." n'est pas différente de la phrase en français "Je veux les notes", il existe l'ambiguïté de savoir si on parle des notes comme d'un type d'obsel ou comme d'un attribut ou encore comme la valeur d'un des attributs. Cette simplification n'est pas sans solution, la plus simple étant de récupérer dans ce cas les éléments de type notes, les éléments ayant un attribut notes et les éléments ayant un attribut de valeur notes, mais nous considérons que familiariser l'utilisateur avec les éléments de la logique des traces lui permettra de mieux comprendre à l'usage le contenu de ses traces et ainsi mieux les appréhender.

5 Conclusion et perspectives

Le langage SPARE-LNC et son implémentation en tant que parser permet la génération de requêtes SPARQL fonctionnelles. Les premières utilisations du langage directement par des utilisateurs non impliqués dans la genèse de ce dernier ont donné lieu à des discussions dont il faut tenir compte pour espérer une utilisation plus large de SPARE-LNC dans le cadre du kTBS. Ces discussions mettent en avant l'importance de proposer un certain nombre d'évolutions pour SPARE-LNC. L'inquiétude soulevée par l'utilisation du vocabulaire spécifique aux traces est par exemple un point qui peut être effacé par la réalisation d'une interface spécifique à l'application ciblée. La question se pose alors d'évaluer le coût de la réalisation d'une telle interface par rapport à une utilisation directe du langage d'origine que SPARE-LNC est censé remplacer. Pour un programme n'ayant besoin que d'une ou deux requêtes SPARQL, il sera probablement plus rapide de définir directement ces requêtes dans le programme plutôt que d'utiliser SPARE-LNC. SPARE-LNC est par contre plus utile lorsqu'écrire des requêtes pour un kTBS occupe une place importante dans les fonctionnalités du logiciel. C'est le cas par exemple de l'outil Samotrace-me (Cordier *et al.*, 2015) qui intègre SPARE-LNC pour calculer des valeurs à partir d'éléments récupérés dans les traces d'un kTBS.

Dans les deux cas, il reste des avantages à l'utilisation de SPARE-LNC. Du fait de son rapprochement avec le langage naturel, SPARE-LNC propose une description intelligible de ce que fait la requête, pour peu que la personne lisant la requête soit familière avec les notions de trace et d'obsel. Cette description propose aussi la particularité interessante d'être le format de stockage de base de la requête, et peut ainsi être échangé directement en format textuel, facilitant la réutilisation des requêtes déjà définies.

La réutilisation de requêtes amène également la question de la pertinence des requêtes d'une trace à une autre. En se basant sur le méta-modèle du kTBS, SPARE-LNC décrit des requêtes toujours valides pour un kTBS, mais il n'assure pas que le résultat soit cohérent d'une trace à l'autre. En effet, les modèles des observables contenus dans deux traces différentes ne sont pas forcément les mêmes. Il serait du coup intéressant de se pencher sur la possibilité de créer des patrons de requêtes pour SPARE-LNC, décrivant la requête en langage naturel contrôlé et les conditions que doivent satisfaire les différents obsels faisant l'objet de conditions dans le patron.

Le fait que l'utilisateur puisse potentiellement adapter les requêtes à ses propres besoins présuppose cependant une certaine compréhension de ce que contient la trace et de ce qu'il peut y récupérer. Dans les interfaces actuelles de SPARE, l'interface textuelle ne propose pas de représentation de la trace, et l'interface graphique propose des récupérations systématiques

de toutes les données, ce qui est peu envisageable sur des bases de traces très peuplées. La recherche et la proposition de nouvelles interfaces viables intégrant le contenu des traces pour le langage SPARE-LNC apparait alors comme une priorité pour son utilisation.

Références

ALLEN J. F. (1983). Maintaining knowledge about temporal intervals. *Commun. ACM*, **26**(11), 832–843.

CHAMPIN P.-A., PRIÉ Y., AUBERT O., CONIL F. & CRAM D. (2011). *kTBS : Kernel for Trace-Based Systems*. Software, LIRIS.

CHOQUET C. & IKSAL S. (2007). Modélisation et contruction de traces d'utilisation d'une activité d'apprentissage : une approche langage pour la réingénierie d'un eiah. *Revue STICEF*.

CORDIER A., DERBEL F. & MILLE A. (2015). *Observing a web based learning activity : a knowledge oriented approach*. Research report, hal-01128536, LIRIS.

DIAGNE F. (2009). *Instrumentation de la supervision de l'apprentissage par la réutilisation d'indicateurs : Modèles et Architecture*. Thèse de doctorat en informatique, Université Joseph Fourier - Grenoble.

DIMITRAKOPOULOU A., PETROU A., MARTINEZ A., MARCOS J. A., KOLLIAS V., JERMANN P., HARRER A., DIMITRIADIS Y. & BOLLEN L. (2006). *State of the art of interaction analysis for Metacognitive Support and Diagnosis*. Research report, Kaleidoscope.

DJOUAD T. (2011). *Ingénierie des indicateurs d'activités à partir de traces modélisées pour un Environnement Informatique d'Apprentissage Humain*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1.

FERRÉ S. (2012). Squall : A controlled natural language for querying and updating rdf graphs. In S. B. HEIDELBERG, Ed., *Controlled Natural Language*, p. 11–25.

FERRÉ S. (2013). squall2sparql : A translator from controlled english to full sparql 1.1. In *Proceeding of the Question Answering over Linked Data lab QALD-3 at CLEF lab*.

GENDRON E. (2010). *Cadre conceptuel pour l'élaboration d'indicateurs de collaboration à partir des traces d'activité*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1.

KONG WIN CHANG B., GUIN N., LEFEVRE M. & CHAMPIN P.-A. (2014). *Conception d'un langage d'interrogation des traces accessible à des non-informaticiens*. Research Report RR-LIRIS-2014-015, LIRIS.

LOPEZ V., UNGER C., CIMIANO P. & MOTTA E. (2013). Evaluating question answering over linked data. *Web Semantics Science Services And Agents On The World Wide Web*, **21**, 3–13.

MONTAGUE R. (1973). The proper treatment of quantification in ordinary english. In H. ET AL. (EDS.), Ed., *Approaches to Natural Language*, p. 221–242.

PRADEL C., HAEMMERLÉ O. & HERNADEZ N. (2012). A semantic web interface using patterns : The swip system. In M. CROITORU, S. RUDOLPH, N. WILSON, J. HOWSE & O. CORBY, Eds., *Graph Structures for Knowledge Representation and Reasoning*, volume 7205 of *Lecture Notes in Computer Science*, p. 172–187.

RUBIN J. & RAM A. (2012). Capturing and adapting traces for character control in computer role playing games. In *Proceedings of the ICCBR 2012 Workshop TRUE*, p. 193–201.

SETTOUTI L. S. (2011). *Systèmes à Base de traces modélisées - Modèles et langages pour l'exploitation des traces d'Interactions*. Thèse de doctorat en informatique, Université Claude Bernard Lyon 1.

SETTOUTI L. S., PRIÉ Y., CHAMPIN P.-A., MARTY J.-C. & MILLE A. (2009). *A Trace-Based Systems Framework : Models, Languages Semantics*. Research report, inria-00363260-v2, LIRIS.

ZARKA R. (2013). *Trace-Based Reasoning for User Assistance and Recommendations*. Thèse de doctorat en informatique, INSA de Lyon.

Diffusion de systèmes de préférences par confrontation de points de vue, vers une simulation de la Sérendipité

Guillaume Surroca[1], Philippe Lemoisson[1,2],
Clément Jonquet[1], Stefano A. Cerri[1]

[1] Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM),
Université de Montpellier & CNRS
nom@lirmm.fr

[2] UMR Territoires, Environnement, Télédétection et Information Spatiale (TETIS),
CIRAD, Montpellier, France
philippe.lemoisson@cirad.fr

Résumé : Le Web d'aujourd'hui est formé, entre autres, de deux types de contenus que sont les données structurées et liées du Web sémantique et les contributions d'utilisateurs du Web social. Notre ambition est d'offrir un modèle pour représenter ces contenus et en tirer communément avantage pour l'apprentissage collectif et la découverte de connaissances. En particulier, nous souhaitons capturer le phénomène de Sérendipité (i.e., de l'apprentissage fortuit) à l'aide d'un formalisme de représentation des connaissances subjectives où un ensemble de points de vue forment un graphe de connaissances interprétable de façon personnalisée. Nous établissons une preuve de concept sur la capacité d'apprentissage collectif que permet ce formalisme appelé Viewpoints en construisant une simulation de la diffusion de connaissances telle qu'elle peut exister sur le Web grâce à la coexistence des données liées et des contributions des utilisateurs. A l'aide d'un modèle comportemental paramétré pour représenter diverses stratégies de navigation Web, nous cherchons à optimiser la diffusion de systèmes de préférences. Nos résultats nous permettent d'identifier les stratégies les plus adéquates pour l'apprentissage fortuit et d'approcher la notion de Sérendipité. Une implémentation du noyau du formalisme Viewpoints est disponible ; le modèle sous-jacent permet l'indexation de tous types de jeux de données.

Mots-clés : représentation des connaissances, découverte et diffusion de connaissances, Sérendipité, ingénierie des connaissances centrée utilisateurs, apprentissage, intelligence collective, Web 2.0, agents.

1 Introduction

Depuis que le Web 2.0 a démocratisé la création, le partage et la recommandation de contenus notamment grâce aux réseaux sociaux, aux blogs et aux forums et depuis que les technologies du Web sémantique ont commencé à structurer la connaissance du Web, deux formes de contenus s'y sont dégagées. Celles-ci sont différentes dans leurs façons de s'établir et dans leurs niveaux de structuration. D'une part les plateformes contributives du Web social permettent la production d'une profusion de données peu ou pas structurées mais au cycle d'évolution et d'entretien très rapide (e.g., les folksonomies [15]). D'autre part des connaissances très structurées sont constituées par consensus par des cercles d'experts, acteurs de la construction du Web sémantique (e.g., les ontologies [8] ou les données liées [3]). Avec l'approche Viewpoints, notre ambition est de créer un formalisme de représentation des connaissance qui intègre aussi bien les jeux de données structurées et liées du Web sémantique que l'abondance d'interactions du Web social afin de tirer le double avantage de la structuration qui caractérise les jeux de données du Web sémantique et de la vitesse d'évolution et d'entretien des connaissances partagées sur le Web social comme cela a été envisagé dans [2, 7,

23]. Notre objectif est de remettre au centre du modèle de représentation des connaissances la contribution des agents du Web (humains ou artificiels) sous forme de points de vue reliant des ressources (identifiées par une URI). Nous nous posons les questions suivantes :

1. Quelles sont les stratégies de navigation sur le Web qui permettent la diffusion optimale des systèmes de préférences des utilisateurs ?
2. Comment positionner les conditions propres à l'apprentissage fortuit, c'est-à-dire à la Sérendipité, dans l'étude des systèmes de préférences ?

Nous parlerons de système de préférences d'un agent[1] pour identifier l'ensemble des goûts ou attirances qu'il exprime sous forme de relations de proximité ou de distance entre ressources du Web. Dans une précédente contribution [10] nous avons démontré la capacité d'apprentissage d'une base de connaissances construite à partir d'une première ébauche de notre formalisme. Toutefois, cette preuve de concept ne se basait que sur une modélisation très pauvre du comportement des agents qui naviguaient au hasard au sein de la base de connaissances pour y contribuer ; nous nous intéressions alors seulement à la satisfaction des utilisateurs sans prendre en compte leurs systèmes de préférences. Dans une autre contribution, nous avons montré comment Viewpoints permet la recherche et la découverte de connaissances grâce à un prototype de recherche de publications scientifique [18]. Dans la modélisation du comportement des agents que nous proposons aujourd'hui, nous incluons un paramètre d'« ouverture à la Sérendipité » qui est la propension d'un agent à s'orienter vers des ressources hors de son système de préférences pour guider sa recherche ; cela nous permet d'évaluer la diffusion des systèmes de préférences selon si agent est plutôt ouvert d'esprit ou plutôt focalisé sur ce qu'il connait et préfère. A partir de cette modélisation, nous construisons une simulation dans laquelle nous créons des règles de comportement individuel (niveau microscopique) et observons l'effet sur l'apprentissage collectif et la diffusion des systèmes de préférences (niveau macroscopique). Cette simulation donne les grandes lignes de l'effet de l'utilisation de Viewpoints pour encapsuler des données du Web sémantique et social.

L'article est structuré de la façon suivante : la section 2 pose le contexte et les inspirations de notre approche en présentant la notion de Sérendipité dans un cadre informatique. L'état de l'art présente aussi un positionnement de Viewpoints par rapport aux autres approches de représentation des connaissances. Ensuite, nous rappelons brièvement le formalisme Viewpoints dans la Section 3. La section 4 explicite notre modèle du comportement des utilisateurs du Web et notre représentation des systèmes de préférences, montre comment nous simulons l'évolution du Web en tant que graphe de connaissances et expose nos hypothèses concernant l'impact des stratégies individuelles de navigation. La section 5 présente une simulation mettant en œuvre trois agents (les princes de Serendip) contribuant à tour de rôle à un graphe de connaissances 'jouet' construit avec des ressources de différentes formes, tailles et couleurs, puis ouvre sur une discussion des résultats au regard de nos hypothèses et de notre problématique. La section 6 conclue et présente des perspectives possibles à ce travail.

2 État de l'art et inspirations

2.1 Représentation des connaissances

Plusieurs travaux se sont positionnés sur le rapprochement du Web sémantique et du Web social [23]. Nous positionnons notre approche par rapport à cet héritage de la manière suivante : il s'agit d'une représentation des connaissances qui en plus d'intégrer l'agent comme présenté dans [13, 15] le considère de manière centrale. Qui plus est, il s'agit d'une représentation des connaissances qui considère le point de vue comme micro-expression des sémantiques individuelles tel que [11]. Cependant, le mécanisme d'arbitrage et de confrontation des points de vue ne fait appel à aucune contribution supplémentaire. Ensuite, l'accent est mis sur l'émergence au sein du graphe biparti, de même que [1, 16] qui étudiaient la possibilité de

[1] Dans cet article, nous utilisons le mot agent au sens d'entité autonome et intelligente tel que proposée dans le domaine des systèmes multi-agents. Cependant, nous ne nous intéressons pas aux représentations internes des agents, e.g. BDI, mais bien à la mise en commun de leurs préférences (« point de vues ». Nous ne considérons pas d'interaction directe entre agent, mais seulement via l'environnement.

l'émergence d'une représentation des connaissances collective dans une vision « bottom-up », à partir des interactions d'un système. Pour finir, nous définissons une distance métrique sur l'ensemble des ressources formé par les fournisseurs, descripteurs et supports de connaissance alors que les distances sémantiques que l'on trouve dans la littérature s'appliquent à des sous-classes homogènes telle que des distances entre tags ou concepts [9].

2.2 La Sérendipité ou l'apprentissage fortuit

Ce mot est dérivé d'un ancien comte persan *Les trois princes de Serendip* [14] Merton écrit à propos du phénomène de Sérendipité qu'il « concerne l'expérience assez générale de l'observation d'une donnée non-anticipée, anormale et stratégique qui devient l'occasion du développement d'une nouvelle théorie, ou l'extension d'une théorie existante. » Plus récemment, Perriault disait « L'effet Serendip (…) consiste à trouver par hasard et avec agilité une chose que l'on ne cherche pas. On est alors conduit à pratiquer l'inférence abductive, à construire un cadre théorique qui englobe grâce à un bricolage approprié les informations jusqu'alors disparates » [17]. Nous notons que la notion de hasard (termes 'hasard' ou 'accident' dans les définitions) est importante dans le phénomène de Sérendipité. Toutefois, elle ne dépend pas uniquement du 'divin jet de dés' dans [6] et n'a lieu qu'à la frontière de ce que l'on sait déjà[2]. Ainsi, les apprentissages fortuits sont grandement facilités lorsque les nouvelles connaissances se situent au voisinage de connaissances existantes et qu'elles peuvent être interprétées par quelqu'un qui connaît ce voisinage. Nous partageons cette vision selon laquelle la préparation, l'entraînement et la connaissance ne garantissent pas la découverte par Sérendipité mais elles la rendent plus probable. Nous pouvons ainsi parler de *zone proximale de Sérendipité* de façon similaire à la notion de zone proximale de développement [21]. Nous montrerons par la suite comment nous avons traduit *l'ouverture à la Sérendipité* dans notre modèle.

La Sérendipité existe d'autant plus sur le Web au vu de l'immense quantité de données qu'il contient et des chances que l'on a de s'y perdre. Nous pouvons donc parler d'apprentissage sérendipiteux sur le Web tel qu'expliqué ci-après. La recherche de connaissances par l'apprentissage sérendipiteux peut aboutir par chance ou comme sous-produit d'une tâche principale [4]. Par exemple, un utilisateur fait une recherche initiale qui le mène, au fur et à mesure de l'exploration des résultats, sur une trajectoire tangente non-prévue initialement qui in fine s'avère plus productive que sa première requête. Dans de tels cas Bowles écrit que l'apprentissage sérendipiteux a lieu [4]. C'est exactement le phénomène que nous modélisons et observons dans notre simulation section 4 à l'aide de stratégie de navigation. En addition, selon Allen Tough, presque 80% de l'apprentissage est informel et non planifié [20]; la navigation sérendipiteuse est une « loterie intellectuelle (…) peu de probabilité mais gros gain potentiel » [12]. Dans ce dernier travail, le parallèle avec notre approche Viewpoints est évident : « nous gagnons aussi de nouveaux points de vue ou associations pour notre problème en parcourant des sources alternatives utilisant des outils, des techniques et des structures de données différentes ».

De ces réflexions sur la Sérendipité nous retiendrons les principes suivants : (i) Les esprits préparés sont mieux disposés pour reconnaître la découverte fortuite (principe de Pasteur) c'est-à-dire ce qui est dans la *zone proximale de Sérendipité*. (ii) Le hasard joue un rôle fondateur car il permet de générer assez de chaos pour permettre l'innovation et la découverte. Comme écrit Toubia, «… l'anarchie de la pensée augmente la probabilité d'avoir des idées créatives …» [19].

La Sérendipité intéresse de plus en plus les acteurs des systèmes de recommandation car même si la précision des recommandations est importante leur variété l'est tout autant. La Sérendipité permet d'aller au-delà de ce que faisaient les systèmes de recommandation en

[2] Comme le disait Pasteur « la chance favorise les esprits préparés » et elle a en l'occurrence favorisé celui d'Alexandre Fleming qui s'il n'avait pas été expert n'aurait pas reconnu la pénicilline comme résultat accidentel de son travail qui consistait à l'origine à faire des cultures de staphylocoques dans le but d'étudier l'effet antibactérien du lysozyme. Ses boîtes de Pétri furent contaminées accidentellement et il se rendit compte qu'autour des champignons qui avait contaminé ses boîtes les staphylocoques ne poussaient plus.

créant la surprise, la variété et la nouveauté dans les résultats proposés. D'ailleurs plusieurs systèmes de recommandation ont commencé à mettre en œuvre ces principes. Par exemple, StumbleUpon.com permet par exemple à ses utilisateurs de « trébucher » sur une ressource du Web au hasard tout en appliquant le principe de Pasteur car la recommandation se fait en fonction de ses activités récentes ou des goûts qu'il a exprimé ce qui nécessite de modéliser ses préférences. L'utilisateur est donc préparé à « trébucher ». La recommandation basée sur une folksonomie présentée dans [22] permet par exemple aux utilisateurs en associant des livres à des tags en plus de dépasser la classification traditionnelle de rajouter de nouveaux livres dans la *zone proximale de Sérendipité* d'autres utilisateurs. Cependant, à notre connaissance, en dehors de [5] proposant un cadre théorique du phénomène de Sérendipité, la littérature sur la formalisation et la mesure de ce phénomène est pratiquement inexistante. Il n'existe pas aujourd'hui de modèle de la Sérendipité.

3 Le formalisme Viewpoints

Viewpoints est un formalisme de représentation des connaissances subjectives, c'est à dire que toute relation de proximité ou distance entre deux *ressources* est exprimée par un *agent* sous la forme d'une connexion sémantique (un *viewpoint*) typée reliant ces deux *ressources*. L'exploitation des *viewpoints* est assujettie à une évaluation a posteriori, selon une *perspective* choisie par l'utilisateur/contributeur, en fonction de qui a émis les *viewpoints*, de quand ils ont été émis, de leurs types ou d'autres critères plus complexes. Cela fait de Viewpoints un formalisme de représentation des connaissances centré sur les agents (qui sont eux même des ressources) au sens large : humain ou artificiel (e.g., automate de fouille de données, extracteur de connaissances, ontologie) qui sont tous deux considérés de la même manière. L'ensemble des ressources (fournisseurs, descripteurs et supports de connaissances) liées par les points de vue forment le graphe de connaissances. Par exemple, dans [19] nous avions illustré le formalisme dans un prototype de moteur de recherche sur données de publications scientifiques indexées à l'aide des métadonnées bibliographiques (auteurs, articles, mots-clés). Le graphe de connaissances (KG) est un graphe biparti constitué d'une part d'un ensemble de ressources R et d'un ensemble de viewpoints V reliant ces ressources entre elles. Les ressources de R sont soit des agents (fournisseurs de connaissances, c'est à dire créateurs de viewpoints), soit des descripteurs de connaissances (des tags de folksonomies ou bien des concepts d'ontologies) ou bien des supports de connaissances (vidéos, pages Web, message, post, etc.). Un viewpoint est un quadruplet ($a \rightarrow \{r_1, r_2\}, \theta, t$) contenant les informations suivantes :

- a, l'*agent* qui a exprimé ce viewpoint
- $\{r_1, r_2\}$, le couple de *ressources* sémantiquement connectées par a
- θ, le type du *viewpoint*, qui va permettre d'interpréter la relation créée
- t, la date de création du *viewpoint*.

Par exemples : `(Guillaume→{Diffusion de systèmes […] points de vue, acm : Knowledge representation and reasoning}, dc:subject, 27/02/2015)` signifie que l'agent Guillaume associe par la relation DublinCore subject cet article au concept Knowledge representation and reasoning d'ACM. `(Mario→{Mario, Luigi}, foaf:knows, 1985)` signifie que Mario exprime le fait qu'il connaisse Luigi, il émet donc un viewpoint qui le rapproche de Luigi. Pour identifier le sens des données représentées sous formes de viewpoints, nous réutilisons tant que possible les types existants du Web sémantique. L'utilisation de ces URIs rend possible la mise en relation du graphe de connaissance avec d'autres jeux de données RDF. A moyen terme un export direct au format RDF du graphe de connaissance permettra d'interconnecter Viewpoints au Web de données.

4 Exploitation des points de vue

L'ensemble des connexions entre deux ressources dues aux différents agents constitue un lien de proximité nommé synapse. La force de cette synapse est fonction de l'agrégation des poids résultant des évaluations de chaque viewpoint. Les deux fonctions d'*évaluation* (map) et

d'*agrégation* (reduce) des *viewpoints* sont au cœur de la notion de *perspective* qui permet d'exploiter le graphe de connaissances. C'est-à-dire, que d'un même KG, plusieurs interprétations – des Knowledge Maps (KM) – peuvent être faites qui dépendent de la façon dont un agent évalue et/ou agrège les viewpoints de chacun. Le KG interprété est un graphe $G_{R,S}$ composé de ressources (R) et de synapses (S). Ainsi, les algorithmes s'exécutant sur KG peuvent être directement adaptés sans effort à des algorithmes de graphe classiques s'exécutant sur KM. La perspective est propre à chaque utilisateur de KG qui décide d'interpréter KG de la manière qu'il souhaite. Par exemple, une perspective simple (telle que présentée dans [19]) donnerait un poids de 1 à tous les viewpoints (map) et calculerait la valeur d'une synapse en faisant une simple somme (reduce). Les deux fonctions d'évaluation et d'agrégation des viewpoints peuvent être étendues à volonté pour correspondre mieux aux usages de notre formalisme.[3] Parmi les algorithmes de graphes nous pouvons mentionner l'algorithme de Dijkstra, du plus court chemin, que nous utilisons dans le calcul de la distance sémantique entre deux ressources quelconques ou l'algorithme de détection de communauté de Louvain que nous utilisons quand nous cherchons une partition de KM. La Figure 1 illustre le processus d'interprétation de KG. Dans la simulation ci-après, nous utilisons : (i) une fonction de voisinage direct qui renvoie pour une ressource r_i toutes les ressources r_j directement reliées par des viewpoints à r_i, ainsi que les poids des synapses liant r_i et r_j ; (ii) une fonction de voisinage indirect qui se base sur l'algorithme de Dijkstra et renvoie pour une ressource r_i toutes les ressources r_j sur tous les chemins partant de r_i et de longueur inférieure ou égale à m (paramètre fixé pour la simulation). Le noyau du formalisme Viewpoints est implémenté en Java et nous utilisons Neo4j pour le stockage de KG. Une interface programmatique (API) existe pour l'indexation de n'importe quel jeu de données[4].

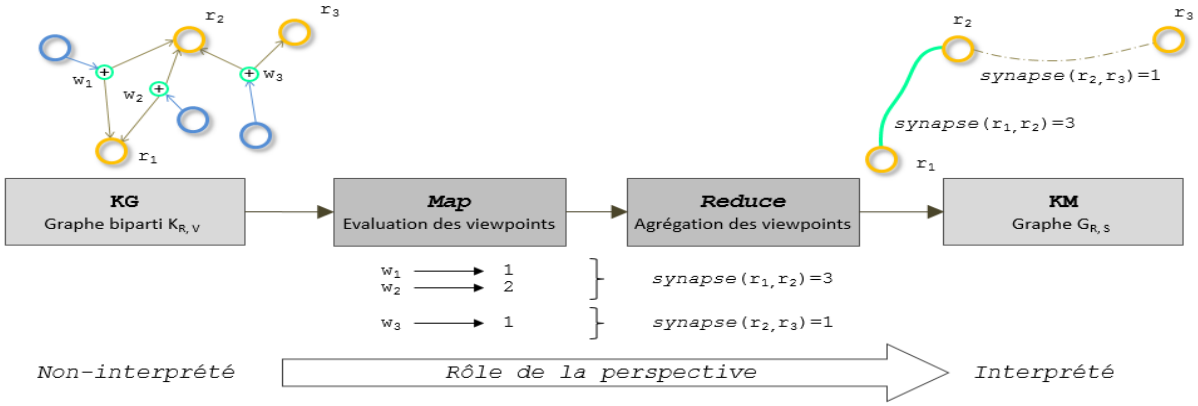


FIGURE 1 – Illustration du processus d'interprétation de KG

5 Simulation des princes de Serendip

Nous souhaitons simuler l'évolution d'une base de connaissances telle que le Web à partir de règles de comportement individuelles qui décrivent les navigations d'agents sur le Web et la diffusion de leurs systèmes de préférences respectifs. Nous commençons par expliquer comment nous représentons les systèmes de préférences dans un graphe de connaissances Viewpoints, puis nous proposons un modèle du comportement simulant différentes stratégies de navigation paramétrables. Ce modèle est fondé sur les calculs de voisinage direct et indirect. Finalement, nous observerons l'effet de cet ensemble de règles individuelles sur le niveau macroscopique de la connaissance représentée dans KG.

[3] Le fait de concevoir l'exploitation de notre graphe à l'aide d'une approche map/reduce nous ouvre un grand potentiel en termes de traitement de gros volume de données et de passage à l'échelle. Cela n'a pas était formellement mesuré à ce jour, mais nous obtenons d'ores et déjà des performances intéressantes sur les jeux de données que nous avons testés.

[4] https://github.com/sifrproject/viewpoints_kernel

5.1 Représentation des systèmes de préférences

Chaque ressource de KG est caractérisée par une forme, une taille et une couleur qui serviront à les rapprocher. Les informations de forme et de taille seront déjà présentes dans le graphe de départ de la simulation ; ces informations simulent les données du Web sémantique. Les informations de couleur des ressources seront ajoutées au fur et à mesure de la simulation par 3 agents (rouge, vert, bleu), les princes de Serendip, qui connaissent et aiment respectivement une couleur distincte ; ces informations simulent les contributions du Web social. Le système de préférences d'un prince est traduit par les viewpoints qu'il émet pour se rapprocher des ressources de sa couleur ou rapprocher entre elles des ressources de même couleur (la sienne). La diffusion d'un système de préférences est donc équivalente à la diffusion de l'information de couleur dans le graphe. Ainsi, l'apprentissage de la couleur par le graphe représente l'émergence d'une intelligence collective de la communauté. Nous considérons ici deux types de viewpoints : (i) rapprochant deux ressources de même couleur (`vps:knows`) (ii) rapprochant un prince d'une couleur à une ressource de la même couleur (`vps:likes`). Par exemple, si le prince rouge fait une recherche sur une ressource r qui est rouge et obtient parmi les résultats une ressource r' qui est aussi rouge alors il créera les deux types de viewpoints : `(prince rouge→{prince rouge, r}, vps:likes, τ)` et `(prince rouge→{r, r'}, vps:knows, τ)`. Dans la section suivante nous présenterons les stratégies de navigation dans KG qui permettent à un prince de diffuser la connaissance de sa couleur.

5.2 Modèle comportemental des princes

L'automate à état (Figure 2) décrit le comportement des princes quand ils naviguent dans KG, et diffusent au fur et à mesure de leurs feedbacks (émissions de viewpoints) leurs systèmes de préférences. Plus généralement, cet automate nous permet de décrire le comportement d'un utilisateur explorant le contenu d'une base de connaissances telle que le Web. Nous capturons ainsi des comportements tels que : la requête sur moteur de recherche, l'exploration des résultats, l'exploration des liens inclus dans ces résultats et le retour éventuel au moteur de recherche avec une autre requête, etc. Les probabilités qui conditionnent les transitions dans cet automate dépendent de trois paramètres :

- β, qui est la probabilité de revenir en arrière pendant la navigation, c.-à-d. soit de revenir à la recherche d'origine (état de départ) ou à la dernière recherche effectuée (état précèdent).
- μ, qui est le choix parmi les outils de navigation disponibles : soit l'utilisation du moteur de recherche opérant globalement sur le graphe soit l'exploration locale des résultats de proche en proche en suivant les liens qui les connectent.
- σ, qui est la capacité à diriger sa navigation vers des ressources qui n'appartiennent pas forcément à son propre système de préférences : l'ouverture à la Sérendipité.

Dans notre simulation le comportement d'un prince de Serendip correspond à un paramétrage spécifique de β, μ et σ ; nous parlerons de *stratégie de navigation*. Ces stratégies simulent des stratégies de navigation sur le Web (ou autre base de connaissances). La simulation se divise en cycles qui correspondent à des explorations successives de KG. Au début d'un cycle, un prince commence par une interaction avec KG qui simule l'utilisation d'un moteur de recherche : une ressource de KG est sélectionnée aléatoirement et nous utilisons la fonction de voisinage indirect pour obtenir une liste de résultats (autres ressources) triés. A partir des résultats proposés, le prince poursuit (β faible) ou abandonne cette recherche et en fait une nouvelle (β fort). S'il poursuit, il va évaluer (relativement à la couleur correspondant à son système de préférences) ces résultats un par un et opter pour le premier non-visité en fonction du paramètre σ. Si le prince est ouvert à la Sérendipité (σ fort), alors il ne se dirigera pas systématiquement vers une ressource de même couleur que lui, sinon (σ faible[5]) il

[5] S'il avait choisi le moteur de recherche Qwant.com il aurait donc commencé par le premier résultat qui lui semble correspondre le plus à ses goûts (σ faible).

privilégiera sa couleur. Ayant choisi une ressource, le prince passera à la prochaine étape de son cheminement, en fonction de µ, soit en faisant une recherche sur cette ressource (µ fort) soit en explorant localement autour de cette ressource (µ faible). La première interaction simule le fait d'ouvrir une page Web après avoir cliqué sur une des URL proposées par le moteur de recherche ; l'interaction suivante simule soit une nouvelle recherche sur par exemple le titre de la page, soit le clic sur un lien inclus dans celle-ci. Dans la simulation, un prince dispose d'un budget d'interactions qui diminue à chaque interaction (recherche ou exploration). Ce budget représente la quantité d'effort qu'il est prêt à faire dans sa navigation. Si au moment du retour en arrière il n'y a plus d'étapes précédentes, s'il n'y a plus de ressources non-visitées ou si son budget d'interaction a été dépensé alors le cycle s'achève.

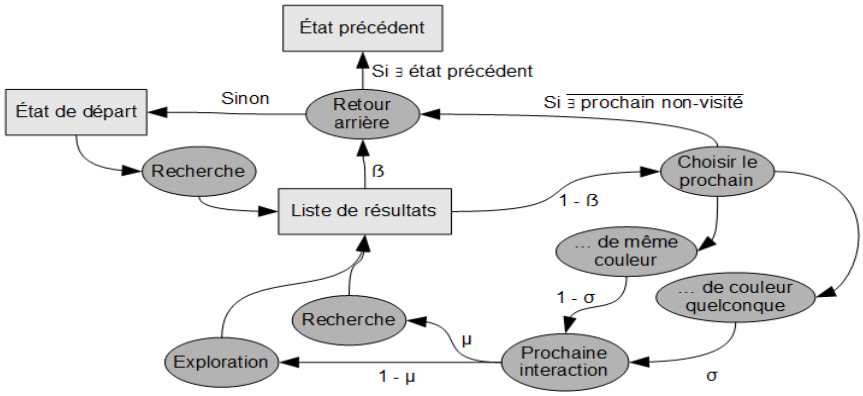


FIGURE 2 – Automate de comportement des princes : stratégies de navigation.

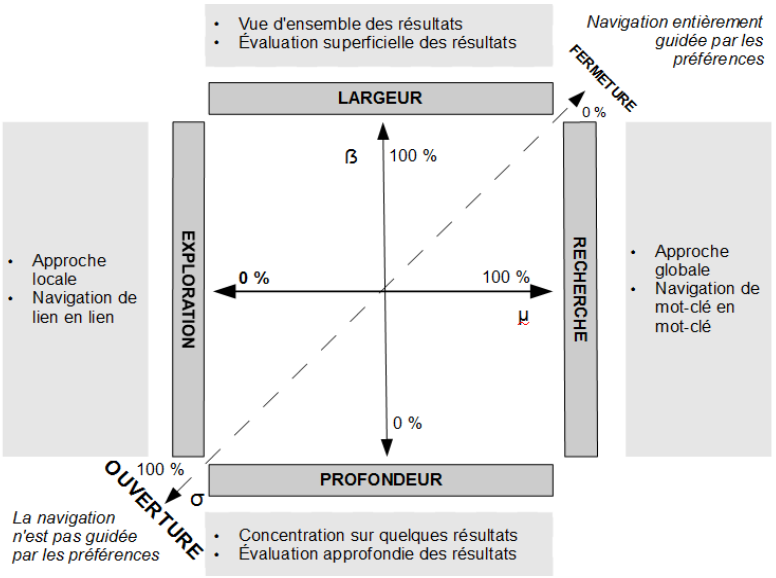


Figure 3 – Différentes stratégies de navigation en fonction des paramètres β, µ et σ.

Nous représentons dans la Figure 3 les trois paramètres relatifs aux stratégies de navigation dans un espace à trois dimensions. Ces stratégies mises en place dans la simulation des princes de Serendip simulent des stratégies de navigation sur le Web. En terme de parcours de graphe plus β est élevé plus on se rapproche d'un parcours en LARGEUR et plus β est faible plus il s'agit d'un parcours en PROFONDEUR. Dans une démarche de recherche d'information le parcours en LARGEUR reviendrait à évaluer de manière superficielle l'ensemble des résultats pour avoir une idée d'ensemble de tous les résultats et l'approche par PROFONDEUR reviendrait plutôt à se concentrer sur ce qui paraîtrait être la meilleure réponse et la creuser plus en profondeur. µ conditionne le style de navigation. Quand µ est élevé on utilise

majoritairement des moteurs de RECHERCHE renvoyant des résultats triés et indirectement liés tandis que quand μ est faible on explore de proche en proche en récupérant des résultats non-triés et directement liés (EXPLORATION). Par exemple, la navigation entre vidéos suggérées sur Youtube est un bon cas de figure d'une exploration de proche en proche tandis que l'utilisation répétitive de Google dans une recherche est plutôt un exemple de parcours en LARGEUR. Nous représentons l'ouverture à la Sérendipité (σ) comme une troisième dimension. Quand σ est grand c'est que l'utilisateur est dans une démarche d'OUVERTURE et qu'il est disposé à cheminer aussi bien parmi des ressources qui correspondent à ses préférences que d'autres ressources qui n'y correspondent pas mais qui pourrait l'amener à la découverte fortuite. Dans le cas opposé (FERMETURE), l'utilisateur parcourt le Web entièrement guidé par ses préférences.

5.3 Déroulement de simulation

5.3.1 Conditions initiales

Un KG de taille déterminée est généré. Les ressources de ce KG sont des ressources caractérisées par une taille (petit, moyen, grand), une forme (carré, cercle, triangle) en plus de posséder une couleur (rouge, vert, bleu). Pour chaque combinaison possible de taille, forme et couleur N ressources sont créées. Il y a donc initialement 27N ressources dans KG. Deux agents artificiels que nous appellerons péons sont ajoutés à KG. L'un d'entre eux partage son appréciation de la notion de forme au graphe de connaissances en reliant tous les couples de ressources de même forme par des viewpoints de type `svp:initial`. L'autre péon fera de même pour la caractéristique de taille. Ainsi, après le passage des péons, KG ne connaît pas la couleur car les ressources ne sont liées que par les deux caractéristiques de taille et de forme. Pour finir la phase d'initialisation trois autres agents sont ajoutés à KG : les princes. Chacun est caractérisé par une couleur unique et a la capacité d'apprécier cette couleur et de partager cette appréciation en émettant de nouveaux viewpoints de type `svp:like` et `svp:knows` dans le graphe de connaissances. Il y a donc une connaissance implicite que les princes sont seuls aptes à partager, par émission de viewpoints de feedback.

5.3.2 Diffusion des systèmes de préférences par confrontation de points de vue

Les paramètres de la simulation sont résumés dans le Tableau 1. Le prince suit le modèle comportemental que nous avons précédemment défini et diffuse ses préférences (la connaissance de sa couleur) en émettant des viewpoints de type svp:like et svp:knows. Le poids associé à chaque type de viewpoint est indiqué dans le Tableau 1. La fonction d'agrégation des viewpoints pour le calcul de la valeur des synapses est la somme. A la fin de chaque cycle les mesures suivantes sont effectuées. Elles nous permettent d'évaluer la diffusion de la connaissance des couleurs dans KG :

- `M1 Couleur X` : Il s'agit du ratio : distance[6] moyenne entre ressources quelconques / distance moyenne entre ressources de couleur X.
- `M2 Couleur X` : Il s'agit de la probabilité d'obtenir au voisinage d'une ressource de couleur spécifique des ressources de la même couleur.

Étant donné le nombre important de paramètres (Tableau 1), nous ne présenterons des résultats obtenus (courbes) que pour certaines simulations, que nous avons jugées les plus significatives pour l'étude des stratégies de navigation. Cependant, nous expliquerons les effets de tel ou tel paramètres dans la section discussion. Dans la suite, les valeurs fixes des paramètres sont précisées dans le Tableau 1.

[6] La mesure de distance employée est une distance aux propriétés métriques (symétrie, séparation et inégalité triangulaire) basée sur le calcul du plus court chemin de Dijsktra (cf. [19]).

TABLE 1 – *Résumé des paramètres de la simulation et de leurs valeurs fixes.*

Catégorie	Paramètre	Valeur (si fixée)	
Paramètres d'échelle	Facteur d'échelle (N)	3	
	Nombre de cycles	100	
	Nombre d'interactions par cycle	50	
Paramètres de perspective	Poids associé aux viewpoints de type `vps:initial`	1	
	… de type `vps:knows`	2	
	… de type `vps:like`	1	
Paramètres de stratégie de navigation	ß		
	µ		
	σ		
Répartition de l'activité	Prince rouge	33%	80%
	Prince vert	33%	10%
	Prince bleu	33%	10%
Paramètres d'algorithme	Borne de distance pour le calcul de voisinage sémantique (m)	2	

5.4 Hypothèses

Au fur et à mesure que les princes contribuent à KG ils partagent leurs appréciations des couleurs avec les autres utilisateurs grâce au mécanisme de feedback. Nous souhaitons observer comment, après leurs contributions, KG aura « appris » au niveau global la notion de couleur qui n'était pas dans les connaissances originalement représentées par les viewpoints. Chaque système de préférences individuel d'un prince devient ainsi, grâce aux viewpoints, une part de la connaissance collective représentée dans KG où il cohabite avec les systèmes de préférences des autres princes. Nous souhaitons expérimenter différentes stratégies de navigation et démontrer que les systèmes de préférences diffusés de façon concurrente ne se neutralisent pas. Nous souhaitons également mesurer l'effet de la Serendipité. Ainsi, nous nous attendons à ce que la mesure M1 augmente, c'est-à-dire à ce que la distance moyenne entre ressources de même couleur décroisse plus vite que la distance moyenne entre ressource quelconques. En effet, les princes rapprochent les ressources de même couleur d'eux-mêmes et les unes des autres, sans jamais rapprocher deux ressources de couleurs différentes. Pour les mêmes raisons, la mesure M2 devrait augmenter aussi car elle reflète la probabilité de trouver une ressource de même couleur dans le m-voisinage d'une ressource.

6 Résultats et discussions

Dans un premier temps, nous faisons varier les stratégies de navigation en conservant la symétrie dans le comportement des trois princes et dans leur répartition de l'activité. Nous observons comment KG « apprend » la couleur rouge en mettant l'accent sur le paramètre σ (ouverture à la Sérendipité). Dans un second temps, nous nous restreignons à deux stratégies de navigation contrastées et jouons sur des répartitions d'activité différentes pour les trois princes ; nous comparons alors les apprentissages respectifs des trois couleurs par KG.

6.1 Impact de l'ouverture à la Sérendipité

Nous commençons par évaluer l'impact de σ sur la diffusion de la couleur rouge grâce aux mesures `M1 Rouge` et `M2 Rouge`. Nous remarquons (Figure 4) que dans le cas d'une utilisation majoritaire du moteur de recherche M1 et M2 croissent plus vite quand l'ouverture est faible mais qu'inversement quand l'ouverture est élevée elles atteignent des valeurs finales

plus élevées. La recherche renvoie des résultats indirectement liés et permet de créer des viewpoints originaux. L'ouverture à la Sérendipité permet au final une diffusion plus grande de la connaissance des couleurs grâce à l'exploration de plus de ressources qui n'auraient pas été rencontrées avec des stratégies fermées. Par exemple, le prince rouge peut rencontrer une autre ressource rouge cachée derrière une ressource verte s'il ose explorer la ressource verte. Par contraste, nous observons (Figure 5) que dans une approche d'exploration locale où seuls sont renvoyés des résultats directement liés l'ouverture à la Sérendipité n'apporte rien ni en terme de croissance des valeurs M1 et M2, ni en terme de valeur finale obtenue. L'idée, avec une telle stratégie, est d'explorer localement et en profondeur les résultats, ainsi le fait de passer par des résultats moins intéressants en chemin a plutôt tendance à freiner la diffusion des systèmes de préférences. L'effet de μ (outil de navigation) est donc très important sur la Serendipité. Nous nous rendons toutefois compte de la relative homogénéité de notre graphe par rapport à la structure du Web. Nous pensons que la Sérendipité peut apporter en condition réelle un saut qualitatif plus substantiel que celui que nous mesurons sur ce graphe 'jouet'. Dans cette simulation les trois princes sont également actifs (33%) et ß=10%[7]

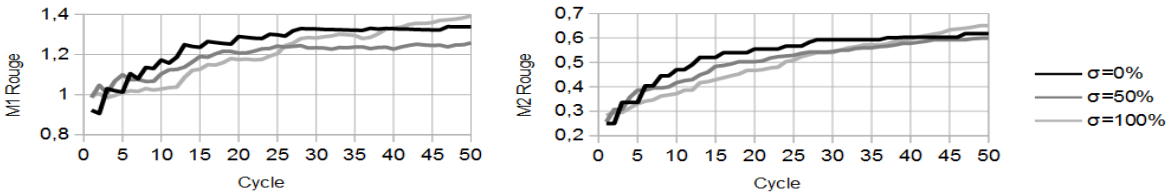


FIGURE 4 – Pour μ=70% et ß=10% (Recherche Profondeur plus ou moins Ouverte).

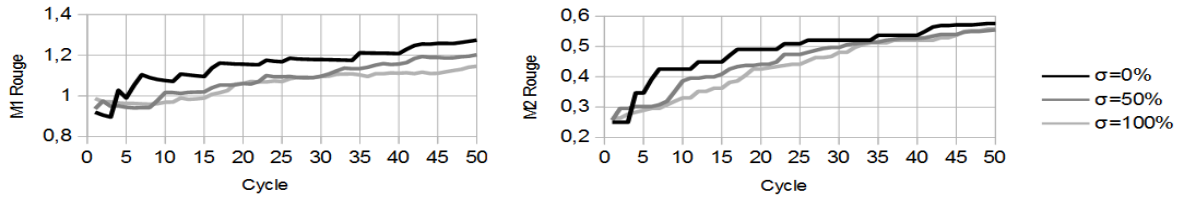


FIGURE 5 – Pour μ=30% et ß=10% (Exploration Profondeur plus ou moins Ouverte).

6.2 Impact de la répartition de l'activité parmi les princes

Dans cette partie nous analysons l'impact de la répartition de l'activité des princes sur la diffusion des couleurs. Pour cela nous observons comparativement `M1 Rouge`, `M1 Vert` et `M1 Bleu` qui évaluent chacune la diffusion d'une couleur dans le graphe. Dans cette simulation, nous faisons varier les probabilités associées aux degrés d'activité respectifs des princes et considérons successivement deux configurations contrastées pour les stratégies de navigation : Recherche Largeur Fermée (μ=80%, ß=40%, σ =10%) et Exploration Profondeur Ouverte (μ=20%, ß=10%, σ =70%). Nous comparons les résultats obtenus pour ces stratégies avec une réparation homogène de l'activité des princes (Figure 6) et avec une répartition non-homogène (Figure 7). Dans les deux cas, nous remarquons que la diffusion de chaque couleur se fait même si la concurrence ralentit cette diffusion. Lorsque les princes sont en concurrence, l'apprentissage d'une couleur se fait bien au détriment d'une autre (lorsque M1 augmente pour un cycle donnée, les autres diminuent) et le prince le plus actif diffuse plus efficacement sa couleur. Cependant, la somme des `M1 Rouge`, `M1 Vert` et `M1 Bleu` finales a une valeur plus élevée quand toutes les connaissances sur les couleurs peuvent être diffusées (Figure 7, la somme des valeurs finales vaut respectivement 3.9 et 3.6) que quand une couleur domine dans la diffusion des couleurs (Figure 6, la somme des valeurs finales vaut respectivement 3.6 et 3.5). D'après ces résultats, on peut déduire que les contributions des utilisateurs du Web social

[7] Nous avons pu étudier au fur et à mesure de nos simulations que la variation du paramètre ß ne change pas les résultats que nous présentons ci-après. Ainsi, nous le fixons dans toutes les simulations présentées à 10% donnant ainsi priorité aux stratégies en profondeur.

ne neutralisent pas celles des autres mais peut les occulter en passant au premier plan.

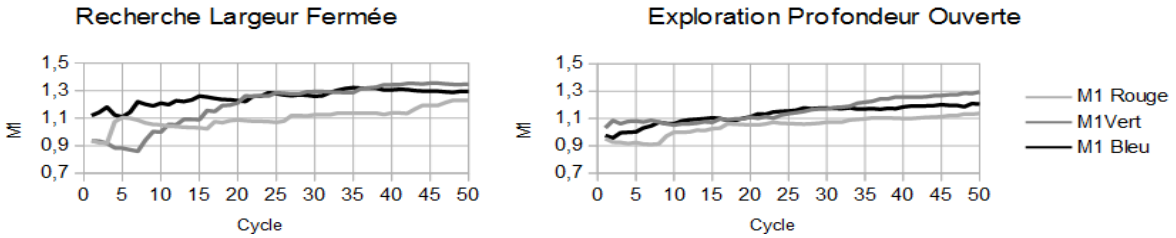


FIGURE 6 – Tous les princes contribuent autant (33%)

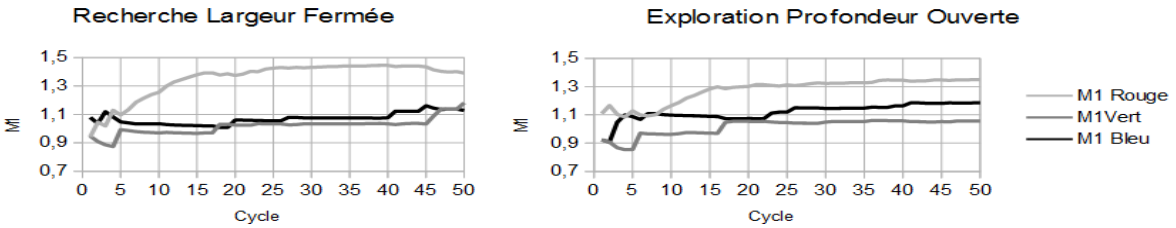


FIGURE 7 – Le prince rouge est plus actif (80%) que les autres (10%).

7 Conclusion et perspectives

Après avoir présenté et positionné notre approche de représentation des connaissances subjectives nous avons étudié le phénomène de Sérendipité en son influence pour le Web d'aujourd'hui. A travers la simulation des princes de Serendip nous présentons un essai de modélisation de la Sérendipité sur le Web. Nous sommes toutefois conscients que ce modèle du comportement des utilisateurs du Web ne rend pas complètement compte de la réalité et de la diversité des méthodes d'explorations du Web. Malgré cela, nous espérons avoir démontré la capacité d'apprentissage du graphe de connaissances de Viewpoints. Les résultats de la simulation nous permettent d'évaluer l'apport de l'ouverture à la Sérendipité dans diverses stratégies de navigation ainsi que son impact sur la diffusion des systèmes de préférences ; nous avons donc consolidé la preuve de concept de Viewpoints en le confrontant à une modélisation d'usage plus réaliste que lors de nos dernières simulations.

Toutefois la preuve de concept ultime reste de nous confronter aux vrais usages et données du Web. Nous avons d'ores et déjà commencé la transition vers les cas d'études sur des données réelles du Web social et sémantique en indexant des données cinématographiques contenant 1M de notations de films d'utilisateurs de MovieLens[8]. Nous souhaitons surtout mettre à l'épreuve l'approche face à des usages et utilisateurs réels. C'est pourquoi nous prévoyons deux cas d'utilisation orientés vers la facilitation de la découverte scientifique transversale dans des contextes de représentation des connaissances agronomiques (Cirad) et biomédicales dans le cadre du projet SIFR (http://www.lirmm.fr/sifr). En outre, en plus d'évaluer l'approche par les usages, nous comptons nous comparer aux algorithmes de recherche d'information en utilisant des benchmarks spécialisés comme on peut en trouver sur LETOR[9] et les mesures de rappel et de précision. Ces cas d'utilisation permettront également de confronter notre approche à de grandes quantités de données et aux problèmes de passage à l'échelle. En outre, ils nous permettront de se confronter à des approches d'ingénierie des connaissances qui considèrent plus l'expertise comme fournie a priori. Alors que Viewpoints laisse émerger les expertises par la plasticité du graphe au fur et à mesure des interactions.

Remerciements

Ce travail a bénéficié des soutiens du Cirad et du projet SIFR financé en partie par le

[8] http://datahub.io/dataset/movielens

[9] http://research.microsoft.com/en-us/um/beijing/projects/letor//

programme JCJC de l'Agence nationale de la Recherche (ANR-12-JS02-01001), l'Université de Montpellier, le CNRS et l'Institut de Biologie Computationnelle de Montpellier.

Références

[1] Aberer, K., Cudr, P., Catarci, T., Hacid, M., Illarramendi, A., Mecella, M., Mena, E., Neuhold, E.J., De, O., Risse, T. and Scannapieco, M. 2004. Emergent Semantics Principles and Issues. *Database Systems for Advanced Applications*. D. Lee, YoonJoon and Li, Jianzhong and Whang, Kyu-Young and Lee, ed. Springer Berlin Heidelber. 25–38.

[2] Ankolekar, A. and Krötzsch, M. 2007. The two cultures: Mashing up Web 2.0 and the Semantic Web. *16th international conference on World Wide Web* (2007), 825–834.

[3] Bizer, C., Health, T. and Berners-Lee, T. 2009. Linked Data - The Story So Far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*. 1–22.

[4] Bowles, M. 2004. Relearning to E-learn: Strategies for Electronic Learning and Knowledge. *Educational Technology & Society*. 7, 4 (2004), 212–220.

[5] Corneli, J., Pease, A. and Colton, S. 2014. Modelling serendipity in a computational context. *arXiv preprint arXiv:1411.0440*. (2014).

[6] Fine, G.A. and Deegan, J.G. 1996. Three principles of Serendip: insight, chance, and discovery in qualitative research. *International Journal of Qualitative Studies in Education*. 9, 4, 434–447.

[7] Gruber, T. 2008. Collective knowledge systems: Where the Social Web meets the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*. 6, 1, 4–13.

[8] Karapiperis, S. and Apostolou, D. 2006. Consensus building in collaborative ontology engineering processes. *Journal of Universal Knowledge Management*. (2006), 199–216.

[9] Lee, W.-N., Shah, N., Sundlass, K. and Musen, M. 2008. Comparison of ontology-based semantic-similarity measures. *AMIA, Annual Symposium 2008*. (Jan. 2008), 384–8.

[10] Lemoisson, P., Surroca, G. and Cerri, S.A. 2013. Viewpoints: An Alternative Approach toward Business Intelligence. *eChallenges e-2013 Conference* (Dublin, Irland, 2013), 8.

[11] Limpens, F. and Gandon, F. 2011. Un cycle de vie complet pour l ' enrichissement sémantique des folksonomies. *Extraction Gestion de Connaissance EGC 2011* (2011), 389–400.

[12] Marchionini, G. 1997. *Information Seeking in Electronic Environments*. Cambridge university press.

[13] Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A. and Stumme, G. 2009. Evaluating similarity measures for emergent semantics of social tagging. *18th international conference on World wide web - WWW '09* (Madrid, Espagne, Apr. 2009), 641–650.

[14] Merton, R.K. and Barber, E. 2006. *The Travels and Adventures of Serendipity: A Study in Sociological Semantics and the Sociology of Science*. Princeton University Press.

[15] Mika, P. 2005. Ontologies Are Us: A Unified Model of Social Networks and Semantics. *4th International Semantic Web Conference, ISWC'05* (Galway, Ireland, 2005), 522–536.

[16] Noh, T., Park, S., Park, S. and Lee, S. 2010. Learning the emergent knowledge from annotated blog postings. *Web Semantics: Science, Services and Agents on the World Wide Web*. 8, 4, 329–339.

[17] Perriault, J. 2000. Effet diligence, effet serendip et autres défis pour les sciences de l'information. *Pratiques collectives distribuées sur Internet* (2000).

[18] Surroca, G., Lemoisson, P., Jonquet, C. and Cerri, S.A. 2014. Construction et évolution de connaissances par confrontation de points de vue : prototype pour la recherche d'information scientifique. *IC - 25èmes Journées francophones d'Ingénierie des Connaissances*.

[19] Toubia, O. 2006. Idea Generation, Creativity, and Incentives. *Marketing Science*. 25, 5, 411–425.

[20] Tough, A. 1999. Reflections on the Study of Adult Learning. *WALL Working Paper*. (1999).

[21] Vygotski, L. 1933. Apprentissage et développement: tensions dans la zone proximale. *Paris: La dispute (2ème éd. Augmentée)*. 233.

[22] Yamaba, H., Tanoue, M. and Takatsuka, K. 2013. On a serendipity-oriented recommender system based on folksonomy. *Procedia Computer Science*. 22, (2013), 276–284.

[23] Zacklad, M., Cahier, J. and Pétard, X. 2003. Du Web Cognitivement Sémantique au Web Socio-Sémantique. *Journée «Web Sémantique et SHS»*. (2003).

Intégration de données cliniques et omiques pour la recherche d'information dans le Dossier Patient Informatisé

Chloé Cabot[1], Lina F. Soualmia[1,2], Stéfan J. Darmoni[1,2]

[1] CISMeF & TIBS, LITIS EA4108 NormaSTiC CNRS FR3638, Rouen University Hospital, France
chloe.cabot@chu-rouen.fr

[2] INSERM, LIMICS UMR 1142, Paris, France

Résumé : Nous décrivons dans cet article le modèle de données générique Information Retrieval for Omic and Clinical Sciences (IROmiCS) que nous proposons pour gérer les principaux types de données omiques (données d'expression, de méthylation de l'ADN et variants génomiques). Nous décrivons également le langage de requêtes que nous avons développé qui repose sur le modèle IROmiCS et qui est dédié à l'interrogation des données cliniques et omiques. Pour valider le modèle de données proposé, ainsi que le langage de requêtes associé, des données omiques expérimentales ont été intégrées dans ce modèle ainsi que des données de référence telle que les bases Gene du NCBI, Uniprot/Swissprot et la Gene Ontology. Plusieurs types de requêtes ciblant des données cliniques et des données omiques ont été réalisées sur les données intégrées. Une interface graphique facilite la visualisation des données intégrées par les cliniciens et les chercheurs. L'outil de recherche a permis de traiter des données symboliques, textuelles, numériques et chronologiques.

Mots-clés : Dossier Patient Informatisé, Stockage de données et Recherche d'Information, Intégration de données, Médecine personnalisée.

1 Introduction

Un Dossier Patient Informatisé (DPI) est défini comme "une version électronique du dossier papier traditionnel utilisé par les professionnels de santé" (Sewell & Thede, 2012). Hebda et Czar Hebda *et al.* (2005) décrivent le DPI comme une ressource d'information électronique utilisée en santé pour stocker les données patient. L'International Organization for Standardization (ISO) définit un DPI comme "*un entrepôt d'information sur la santé d'un individu dans une forme traitable par informatique, stockée et transmise avec une sécurité adéquate, et accessible par de multiples utilisateurs autorisés. Il respecte un modèle d'information logique communément admis, dépendant des systèmes de DPI. Son but premier est le support de la continuité, de l'efficacité et de la qualité des soins et il contient une information rétrospective, concurrente et prospective*". Selon cette dernière définition, le DPI joue un rôle central puisqu'il comporte les informations à long terme relatives aux soins et évènements de soin de tous types, mais aussi des instructions, informations prospectives comme des plans, ordres et évaluations (Garde *et al.*, 2007).

Ainsi, la communauté médicale fait face à un nouveau paradigme dans sa manière d'interagir avec les données cliniques. Les DPI permettant de gérer et partager tous les différents types de données cliniques (texte, numérique, synthétique, chronologique). Quelques entrepôts de données cliniques proposent des architectures, outils et services permettant l'utilisation des données du DPI, en particulier en recherche translationnelle. En effet, durant la dernière décennie, le séquençage de nouvelle génération (NGS) a été considérablement amélioré et ces techniques haut débit sont aujourd'hui communément utilisées pour répondre à de nombreuses questions

biologiques à l'échelle du génome : identification de variations, analyse d'expression ou encore modification de la chromatine. Alors que le génome humain avait demandé dix ans à être complété et coûté des milliards de dollars, aujourd'hui les scientifiques peuvent réaliser un séquençage de génome ou d'exome en moins d'une semaine pour moins de mille dollars (Fernald *et al.*, 2011). Les données omiques générées par l'usage croissant de ces techniques ouvrent de nouvelles perspectives dans la recherche d'applications biomédicales. Aujourd'hui, elles sont déjà utilisées pour identifier de nouveaux biomarqueurs, des mutations génétiques permettant de prédire la susceptibilité ou la prédisposition génétique à certaines pathologies ou à évaluer une réponse personnalisée à un médicament. Prochainement, la réunion de données cliniques et omiques pourrait mener à des applications innovantes comme de nouveaux tests diagnostiques ou encore des thérapies ciblées ainsi que des avancées significatives dans la compréhension de certaines maladies génétiques complexes et des mécanismes impliqués.

Plusieurs outils et frameworks dédiés à la recherche d'information dans les DPI ont été proposés. Ces outils ont été adaptés selon chaque format de données : structuré, non structuré ou mixte. Principalement deux différents types d'outils existent : (a) des Systèmes de Recherche d'Information (SRI) orientés population basés sur des entrepôts de données cliniques pour n patients et (b) des SRI au sein des DPI dédiés à un seul patient. Pour cette dernière catégorie, plusieurs outils ont été développés. CISearch (Natarajan *et al.*, 2010) a été développé et implémenté à l'hôpital universitaire de Columbia. L'utilisateur peut interroger tous les rapports textuels (imagerie, pathologie, décharge) en utilisant des outils Apache Lucène. Medical Information Retrieval System (MIRS) (Spat *et al.*, 2008) est également basé sur les outils Apache Lucène. Dans le système OpenEHR (Kalra *et al.*, 2005), un langage de requête dédié est utilisé, lié à une structure orientée archétype. Le language Archetype Query Langage (AQL) est construit pour interroger ce type de modèle de données dans les DPI. La plateforme Stanford Translational Research Integrated Database Environment (STRIDE) fournit également un système d'interrogation nommé Anonymous Cohort Tool, dédié à la création de cohortes (Lowe *et al.*, 2009). Le moteur de recherche EMERSE (Hanauer, 2006) permet la recherche plein texte avec des options avancées adaptées aux DPI comme la recherche par troncatures ou par synonymie. Enfin, XOntoRank Farfan *et al.* (2009) est un moteur de recherche sémantique permettant de réaliser des requêtes sémantiques dans des documents médicaux structurés, suivant la norme Health Level Seven International (HL7) CDA [1]. Ces documents peuvent contenir à la fois des données codées, structurées ou libres.

Integrating Biology and the Bedside (I2B2) (Murphy *et al.*, 2010) est un framework libre permettant de réutiliser les données cliniques existantes dans les DPI à des fins de recherche, et si combinées à des données génomiques, à faciliter la conception de thérapies ciblées. Cette plateforme profite actuellement d'une large adoption par la communauté scientifique académique et industrielle. L'un des composants les plus visibles d'I2B2 est le I2B2 workbench, un outil dédié à la sélection de patients permettant l'interrogation et la visualisation des données cliniques (Deshmukh *et al.*, 2009). Cependant, le modèle de données d'I2B2 n'inclut pas un point de vue centré sur le patient. Transmart (Sarkar *et al.*, 2011) est une plateforme de recherche translationnelle supportée par une communauté croissante de développeurs. Ce logiciel est directement basé sur le modèle de données d'I2B2. Il permet d'explorer des données phénotypiques, de réaliser des méta-analyses et de tester et valider de nouvelles hypothèses.

1. `www.hl7.org/`

Depuis 2011, un projet en cours appelé RAVEL (Recherche d'Information et Visualisation dans le Dossier Patient Informatisé) est dédié au développement d'outils efficaces et productifs permettant aux utilisateurs de situer, en temps réel, les éléments pertinents des DPI et de les visualiser grâce à des modèles de présentation synthétiques et intuitifs (Thiessard *et al.*, 2012).

Nous décrivons dans cet article le modèle de données omiques Information Retrieval for Omic and Clinical Sciences (IROmiCS) que nous avons développé. Il repose en partie sur le modèle de données cliniques RAVEL de façon à accomplir plusieurs tâches : (i) la représentation des données omiques, (ii) l'intégration, la gestion et le stockage du plus grand nombre de types de données omiques, (iii) la recherche d'information à deux échelles : la première à l'échelle d'un patient unique, qui est axée soin, et la deuxième à l'échelle de plusieurs patients, axée épidémiologie (comme par exemple la création de cohortes) et recherche clinique (comme par exemple la sélection automatique de patients pour des essais cliniques basés sur des critères d'inclusion et d'exclusion).

Cet article est organisé comme suit. La section 2 est dédiée à la description du modèle de données IROmiCS ainsi que les données qui y ont été intégrées. La section 3 présente l'outil de recherche d'information dans les DPIs via le langage de requêtes que nous avons développé, ainsi que la visualisation des données. Nous comparons nos approches avec les solutions existantes dans la section 4. Enfin, nous concluons et donnons quelques perspectives de travail dans la section 5.

2 Modèle de données IROmiCS et sources de données

2.1 Modélisation des données omiques

La conception d'un modèle de données générique gérant à la fois des données omiques et des données cliniques nécessite d'établir une revue complète et cohérente des différents types de données omiques aujourd'hui utilisés. Les données pertinentes pour l'intégration avec des données cliniques doivent être sélectionnées en fonction de leur utilité et de leur intérêt dans le cadre du DPI. Plusieurs problèmes doivent être résolus, incluant le volume de données à considérer et le manque de consensus sur les informations pertinentes à retenir.

Quatre niveaux de données ont été posés afin de décrire les types de données, en accord avec les conventions adoptées par les bases de données internationales comme ArrayExpress EBI (2013), GEO NCBI (2013) ou TCGA NIH (2013) (TABLE 1). Les données *brutes* (niveau 1) correspondent aux données non normalisées. Pour un séquençage, ce niveau correspond aux données brutes sorties du séquenceur. Elles peuvent être accessibles par des fichiers textes ou binaires, dont le format dépendra fréquemment du matériel utilisé. Le plus souvent, le volume de données est très important (jusqu'à plusieurs gigaoctets pour une seule analyse) et ces données ne peuvent pas être interprétées manuellement.

Les données *traitées* (niveau 2) correspondent aux données normalisées par une méthode statistique de régression non paramétrique comme la LOWESS par exemple. Il s'agit du signal d'une sonde ou d'un groupe de sondes pour une analyse d'expression, ou encore d'un variant supposé pour un échantillon. Ces données sont accessibles par des fichiers textes sur des banques de dépôt comme GEO ou ArrayExpress. Le volume de données est réduit, mais reste important. Pour une analyse d'expression de gènes, le fichier de résultats concernant un seul échantillon peut aller jusqu'à une centaine de mégaoctets. L'interprétation manuelle reste

Niveau	Type	Description	Exemple
1	Données brutes	Données de bas niveau par échantillon, non normalisées	Fichiers BAM ou CEL Signal brut par sonde
2	Données traitées	Données normalisées par échantillon	Signal normalisé par sonde ou set de sondes
3	Données interprétées	Données traitées agrégées par échantillon	Signal d'expression d'un gène, par échantillon
4	Régions d' intérêt	Associations quantifiées entre classes d'échantillons	Un gène X est impliqué dans 10% des lymphomes

TABLE 1 – Les quatre niveaux de données omiques permettant leur classement en fontion de leur traitement

délicate, l'information concernant des sondes ou des variants non validés.

Les données *interprétées* (niveau 3) regroupent des données qui ont été agrégées pour un échantillon. Par exemple, pour l'analyse d'expression de gènes, il s'agira du signal d'expression d'un gène, les signaux des sondes correspondant à ce gène ayant été agrégés ou encore d'un variant validé. Ce type de données est disponible en fichier texte, le plus souvent tabulé. Cependant, il n'existe pas de standard établi. Le volume de données est dans ce cas réduit, un fichier de résultats pour une analyse d'expression peut représenter de quelques kilooctets jusqu'à 1 Mo, selon le nombre de gènes analysés. Ce niveau de données n'est pas accessible dans les banques de dépôt ArrayExpress et GEO, qui ne proposent que des données de niveau 1 et 2. Peu de banques de données proposent ces données interprétées. Le portail TGCA offre les données recueillies dans une vingtaine d'études impliquant jusqu'à plusieurs centaines de patients. Les techniques utilisées sont variées et couvrent tous les types de données vus précédemment. Dans ce cas, on dispose d'informations validées et exploitables. Ce niveau de données paraît donc pertinent à intégrer dans un dossier médical. La TABLE 2 présente pour chaque type de données l'information de niveau 3 correspondante.

Enfin, les données *interprétées et agrégées* correspondent au niveau d'interprétation le plus élevé (niveau 4). Il s'agit de réaliser des associations quantifiées et croisées entre différents types d'échantillons afin d'isoler une région d'intérêt. Cette interprétation approfondie des données omiques nécessite une expertise biostatistique et biologique humaine pointue. L'aboutissement à ce niveau d'interprétation est notamment l'un des buts de la plateforme Transmart (Sarkar *et al.*, 2011). Très peu de ressources sont disponibles de façon standardisée et formalisée. De plus, de telles données ne s'appliquent plus avec l'échelle du patient, mais à celle du phénotype, il n'est donc pas adapté à l'intégration avec des données cliniques. Cependant, les régions d'intérêt isolées représentent une information pertinente, notamment à des fins de diagnostic ou de recherche.

Nous avons évalué les quatre niveaux de données afin de sélectionner les données pertinentes à modéliser. La comparaison directe de ces données et leur intégration impose de considérer certains points : (i) la normalisation des données brutes, pour exclure des biais liés à l'étude, la plateforme ou la préparation des échantillons, (ii) l'interprétation des données brutes pour améliorer la lisibilité des résultats par les cliniciens et les chercheurs et (iii) le volume de données.

Les deux premiers niveaux regroupent données brutes et traitées, qui sont de trop bas niveau et volumineuses pour être considérées. Ces données ne correspondent pas à un point de vue

centré sur un patient puisqu'elles ne sont ni agrégées ni interprétées. Cependant, le troisième niveau de données désigne des données agrégées et interprétées comme des signaux d'expression par gène par échantillon ou des variants validés. De plus, le volume de données est limité. Enfin, le quatrième niveau de données ne correspond pas avec l'échelle d'un patient puisqu'il désigne les observations faites sur une population de patients et échantillons.

Le modèle de données omique a été conçu selon le niveau 3 de données décrit. Ce modèle se compose de trois parties gérant (i) les données liées aux laboratoires et études, (ii) aux données de variants et (iii) aux données d'expression. Le détail des types de données gérées est donné dans la TABLE 2. Le modèle complet est disponible à `http://www.chu-rouen.fr/cismef/papers/omic_mld.pdf`

2.1.1 Données des études et laboratoires

La première partie du modèle vise à gérer les données des laboratoires, responsables et études. Les informations liées aux laboratoires sont leur `nom`, `code`, `adresse`, `email` et `numéro de téléphone`. Les informations stockées relatives aux études ont pour but la conservation et la traçabilité des `protocoles`, `équipements`, `échantillons` ou encore `version d'assemblage du génome` utilisé dans l'expérience ainsi que la `source des données`. Cette partie gère également les `données administratives` des responsables d'une étude.

2.1.2 Données de variants

Pour appréhender les données communément collectées liées aux variants génomiques, des collections de métadonnées de variants génomiques comme le National Center for Biomedical Ontology (NCBO) SNP Ontology [2] et la dbSNP ont été étudiées. La NCBO SNP Ontology liste 23 classes pour décrire un variant génomique, allant de la classification des acides aminés, aux données de séquençage jusqu'au type du variant. Un sous-ensemble de ces métadonnées qui peuvent être extraites des systèmes de reporting des laboratoires génomiques a été retenu. Cette partie du modèle gère ainsi des données liées aux Single Nucleotide Variants (SNV) et insertions/délétions (indels). Pour chaque variant, les `noms systématiques` (nucléiques et protéiques), les `codons` et `bases de référence et mutés`, la `catégorie` de la variation, sa `localisation` et la `région impliquée` sont retenus. Pour chaque patient, les `variations détectées` et son `génotype` pour la variation correspondante sont stockés.

2.1.3 Données d'expression, variants structuraux, méthylation de l'ADN

La base de données contient des données concernant les `gènes et protéines` extraites de la base NCBI Gene et Uniprot KB utilisées comme référence. Les analyses de méthylation de l'ADN, perte d'hétérozygotie (LOH) ou les variants du nombre de copies sont gérés grâce à une entité générique unique. Cette entité possède plusieurs attributs comme le `type de segment génomique analysé`, sa `localisation` et ses `données de référence`. Chaque gène, protéine ou segment est lié au patient concerné et au `résultat de l'analyse`.

2. `bioportal.bioontology.org/ontologies/SNPO`

Type de données	Niveau 3 : Description
Variants structuraux	Altération d'une région segmentée par échantillon
Analyse du nombre de copies	Altération du nombre de copies pour une région segmentée par échantillon
Méthylation de l'ADN	Valeurs bêta calculées pour une région génomique par échantillon
Expression : exon	Signal d'expression normalisé par exon par échantillon
Expression : gène	Signal d'expression normalisé par gène par échantillon
Expression : miRNA	Signal d'expression normalisé par miRNA par échantillon
Expression : jonction	Signal d'expression normalisé par jonction par échantillon
Expression : transcrit	Signal d'expression normalisé par transcrit par échantillon
Expression : protéine	Signal d'expression normalisé par protéine par échantillon
Variants (SNP, indels)	Variants validés par échantillon

TABLE 2 – Description du niveau d'interprétation 3 pour les principaux types de données omiques sélectionnés pour concevoir le modèle de données IROmiCS

2.2 Modèle de données clinique

Le modèle de données cliniques est basé sur un modèle conceptuel intégré à un modèle physique générique (Cabot *et al.*, 2014). Ce modèle conceptuel compact contient seulement une dizaine d'entités (patients, séjours, analyses et actes médicaux), alors qu'un modèle de données cliniques en comporte habituellement plus d'une centaine. Il repose sur un modèle physique générique Entité-Attribut-Valeur (EAV) composé de deux parties : le modèle définissant le modèle de données conceptuel et l'instance du modèle stockant les données. Ce modèle compact est dédié à la recherche d'information. Il permet de gérer des types de données hétérogènes. Ce "méta-modèle" intègre l'ensemble des ressources terminologiques et documents indexés. Les sources de données cliniques et omiques ont été intégrées dans le modèle de données, créant ainsi un entrepôt de données clinomiques (voir FIGURE 1).

Les données du modèle de données cliniques sont réparties en onze tables dédiées aux informations administratives du patient (table `DM_PAT`), aux analyses biologiques (table `DM_ANA`), aux prescriptions (table `DM_PRESCR`), aux actes médicaux (table `DM_ACT`), séjours (table `DM_STAY`) et comptes-rendus (table `DM_REC`). Le modèle complet est disponible à `http://www.chu-rouen.fr/cismef/papers/model_ravel.png`.

Ainsi, pour un patient donné, la base de données contiendra ses informations administratives (nom, âge, genre) et les différents séjours passés dans le centre hospitalier (avec les dates d'en-

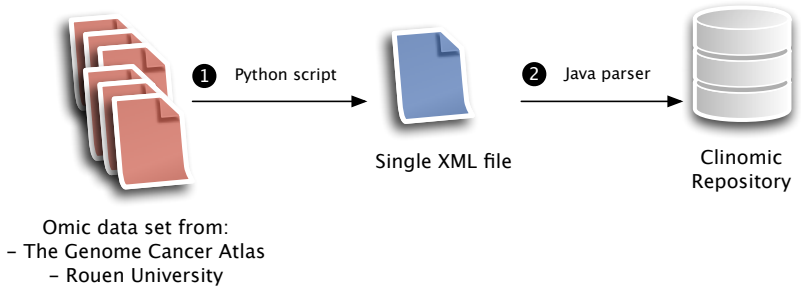


FIGURE 1 – Processus d'intégration des données omiques

Chaque jeu de données expérimentales a été traité afin de regrouper tous les fichiers de données au sein d'un fichier XML unique. Ensuite, le fichier XML a été traité et intégré dans l'entrepôt de données.

trée et de sortie et l'unité médicale d'accueil). À chaque séjour correspondra un ou plusieurs compte-rendus d'hospitalisation, ainsi que les actes médicaux et analyses biologiques réalisés ainsi que des prescriptions. Les séjours, actes médicaux, analyses biologiques, prescriptions et compte-rendus sont indexés automatiquement (Pereira *et al.*, 2009; Chebil *et al.*, 2012) par diverses terminologies (Classification Commune des Actes médicaux (CCAM), Classification Internationale des Maladies - 10e édition (CIM-10), SNOMED CT, Terminologie Unifiée du Vidal).

2.3 Sources de données

Les données d'un corpus composé de 2 000 patients et 200 000 comptes-rendus ont été utilisées dans cette étude, approuvée par la Commission Nationale de l'Informatique et des Libertés (CNIL). Toutes les informations cliniques disponibles dans les DPI ont été intégrées dans le modèle RAVEL, comme les codes de la CIM10 qui permettent les codages de données comme "Cancer du colon", les données des patients (âge, genre), les résultats de tests et les comptes-rendus médicaux.

Les données omiques ont été obtenues à partir de plusieurs sources comme des bases de données internationales (Gene Expression Omnibus (GEO) (Edgar *et al.*, 2002), ArrayExpress (Rustici *et al.*, 2013), The Cancer Genome Atlas (TCGA) (NIH, 2013)) et grâce à des collaborationss avec les laboratoires du Centre Hospitalier de Rouen (INSERM U1079 et INSERM U918) principalement spécialisés en oncologie. Les données omiques sont couplées avec des données de référence concernant les gènes, protéines et phénotypes. Pour cela, des données de référence du NCBI et Uniprot/Swissprot ont été utilisées. ces deux bases de données internationales sont supervisées et reconnues. Leurs données ont été filtrées pour ne retenir que les gènes et protéines humains dans l'entrepôt clinomique. La description des phénotypes repose sur le catalogue Online Mendelian Inheritance in Man (OMIM), et les bases Human Phenotype Ontology (HPO) (Grosjean *et al.*, 2013) et Human Rare Diseases Ontology (HRDO) (Aimé *et al.*, 2012). OMIM fournit les informations concernant les phénotypes liés aux maladies génétiques. La HRDO comporte des données sur les maladies orphelines. Enfin, la Gene Ontology complète la description des gènes et protéines. Environ (i) 9 Go de données extraites de NCBI Gene, (ii)

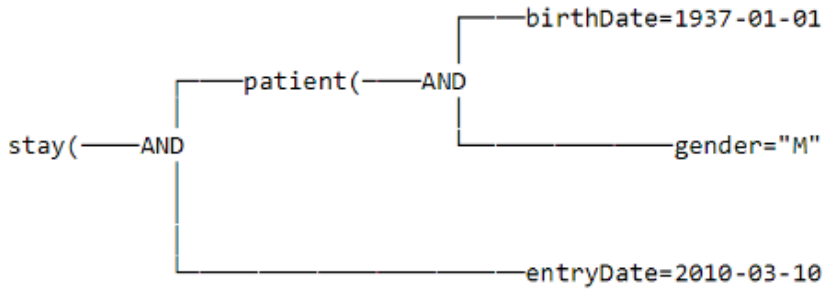


FIGURE 2 – Représentation en arbre de la requête `stay(patient(birthDate=1937-01-01 AND gender="M") AND entryDate=2010-03-10)`

530 Mo d'Uniprot/Swissprot/KB et (iii) 165 Mo d'OMIM ont été initialement intégrées dans l'entrepôt de données clinomiques. La mise à jour automatique des données est assurée quotidiennement. Pour cette étude, 88 % des maladies OMIM et 90 % des termes HPO ont été manuellement et automatiquement traduits en français et inclus dans le portail HeTOP (Grosjean *et al.*, 2013).

3 Recherche d'information

En se basant sur le modèle de données IROmiCS décrit, nous avons conçu un langage d'interrogation spécifique, dédié à la recherche d'information dans les données omiques et cliniques. Le moteur de recherche a été conçu pour être générique, rapide, multilingue et aligné avec de multiples terminologies. Il utilise un langage d'interrogation spécifique visant à faciliter la recherche d'information. Il présente trois caractéristiques principales : (i) c'est un langage orienté objet, (ii) il est flexible et (iii) il a des capacités d'interrogation complète (toutes les données contenues dans la base de données peuvent être interrogées). Le langage est composé d'unités syntaxiques, respectant la syntaxe suivante :

`ENTITÉ(CLAUSE_CONTRAINTES)`

où :

- `ENTITÉ` correspond à une entité du modèle conceptuel IROmiCS
- `CLAUSE_CONTRAINTES` correspond aux contraintes appliquées à cette entité, construite en utilisant des attributs ou des objets liés (voir TABLE 3).

Plus de détails sont disponibles sur le langage d'interrogation dans (Lelong *et al.*, 2014). Dans le modèle actuel, trois `ENTITÉS` principales sont modélisées à trois niveaux : le niveau du patient, le niveau du séjour, et le niveau le plus bas (comme l'analyse biologique ou l'analyse omique). Par exemple, les requêtes `patient()` et `medicalUnit()` retournent respectivement tous les patients et toutes les unités médicales contenues dans la base données. La clause de contrainte `CLAUSE_CONTRAINTES` permet d'appliquer des contraintes à l'entité spécifiée. C'est une expression booléenne, ainsi les opérateurs booléens `AND, OR, NOT` et les parenthèses sont utilisées pour construire des liens logiques entre contraintes uniques.

Requête en langage naturel	**Traduction dans le langage d'interrogation**

Les patients de l'étude 12 ayant une expression de HRNR supérieure à 3	`patient(`
	`study(id="OMI_STUDY_12")`
	`AND quantification(gene(geneSymbol="HRNR")`
	`AND numericValue > 3)`

Patients ayant des variations faux-sens sur HOMER1 et un taux de glucose sanguin supérieur à 1,1g/L	`patient(`
	`study(id="OMI_STUDY_1")`
	`AND variant(gene(geneSymbol="HOMER1")`
	`AND variantCategory="Missense" )`
	`AND bioTest(`
	`bioResultEXECode(label="Glucose")`
	`AND numericValue > 1.1))`

Tous les segments génomiques délétés dans l'étude 10	`quantification(interpretation="deletion"` `AND study(id="OMI_STUDY_10"))`

Tous les variants faux sens sur le gène HRNR sans l'étude 1	`variant(study(id=”OMI_STUDY_1”)`
	`AND gene(geneSymbol=”HRNR”)`
	`AND variantCategory=”Missense”)`

TABLE 3 – Exemples de requêtes omiques

Cette clause de contrainte peut être construite en utilisant les attributs de l'entité spécifiée. Par exemple, la requête suivante `patient(birthDate=1937-01-01 AND gender='M')` utilise deux attributs `birthDate` et `gender` de l'entité `patient` et retournera tous les patients masculins, nés le 01/01/1937. Les opérateurs booléens, parenthèses et comparateurs sont définis explicitement dans la grammaire du langage alors que les entités sont déduites automatiquement par auto-complétion à partir du modèle de données IROmiCS. Le moteur de recherche permet d'interpréter les requêtes pour extraire les données correspondantes de la base de données. Le processus d'interprétation contient trois étapes : (i) le parsing de la requête, (ii) sa représentation sous la forme d'un arbre (voir FIGURE 2) et (iii) la construction de la requête SQL correspondant à l'arbre généré, le modèle de données étant intégré dans une base de données relationnelle. Différentes données peuvent être extraites : (i) données symboliques (absence, présence), (ii) données numériques (avec les opérateurs >, < et =) et (iii) données chronologiques. Le moteur de recherche a été adapté pour permettre aux utilisateurs d'interroger à la fois données cliniques et données omiques (variant, gène, protéine ou segment génomique). Des mots-clés ont été définis pour interroger chaque entité du modèle omique conceptuel et élaborer des contraintes. Le temps de réponse moyen pour un patient est inférieur à deux secondes, ce qui est considéré comme satisfaisant pour un clinicien ou un chercheur. Pour n patients, le temps de réponse moyen est inférieur à dix secondes. Il est possible de réaliser la RI simultanément sur les données cliniques et omiques dans la même requête. Par exemple, les patients qui ont des variations faux-sens sur le gène HOMER1 et un taux de glucose sanguin supérieur à 1,1g/L peuvent être extraits. Chaque entité du modèle de données IROmiCS conceptuel est interrogeable. Les requêtes sont réalisables à plusieurs échelles : au niveau du patient, du séjour

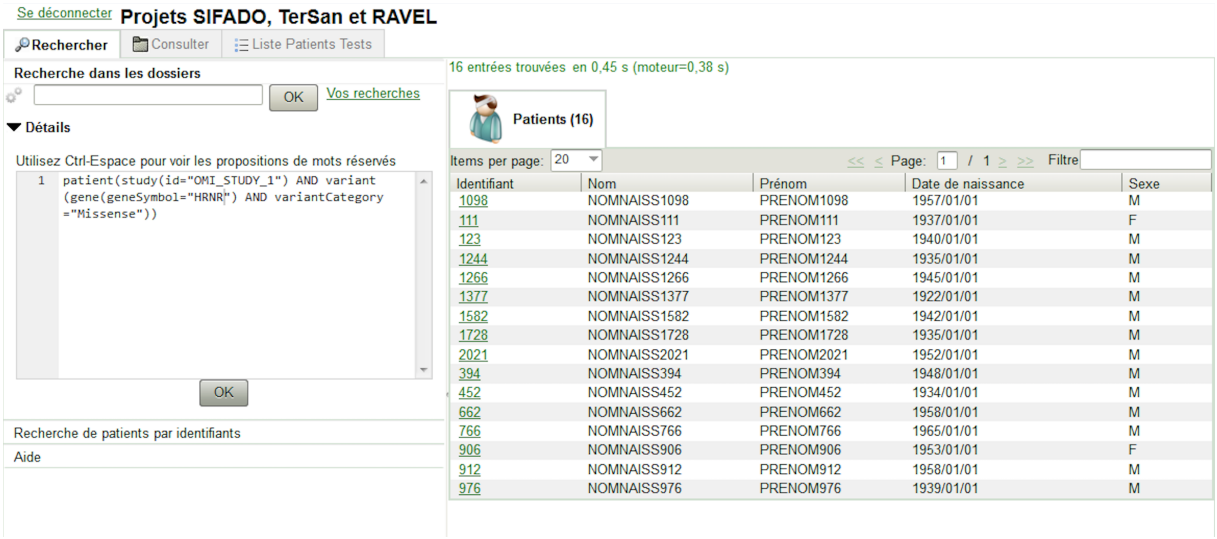


FIGURE 3 – Interface d'interrrogation

ou de l'étude. Les variants et régions génomiques peuvent également être extraits.

Pour faciliter la visualisation des données et l'utilisation de l'outil de recherche, une interface web a été conçue. Elle permet de visualiser et d'interroger toutes les données décrites dans le modèle au sein d'une interface conviviale. Les données peuvent être consultées via l'outil de recherche décrit ici et propose également une vue mono-patient où toutes les données d'un patient unique sont regroupées et visualisables (voir FIGURE 3).

4 Discussion et Conclusion

Alors que certaines solutions logicielles existent déjà en sciences translationnelles pour intégrer des données biologiques et cliniques, aucune ne gère tous les types de données omiques, données de séquences, données d'expression et variants. Bien que certaines problématiques se posent pour intégrer différents types de données omiques à partir de différentes études de sources diverses dans un même modèle de données, ce type de données peut intéresser à la fois recherche clinique et pratique clinique. Le modèle de données omique IROmiCS proposé dans cet article gère les types de données les plus communs. Il a été testé avec plusieurs jeux de données de neuf études omiques différentes. Des données d'expression (gènes, protéines, microARN), CGH-array, méthylation de l'ADN ont été intégrées avec succès. De plus, environ 25 000 variants, incluant des SNV et indels ont également été insérés avec succès dans la base de données implémentant le modèle décrit. Cependant, les variants insérés ont été extraits d'une seule étude, du fait du manque de données publiques accessibles.

Alors que la solution de référence i2b2 est largement adoptée à la fois par la communauté académique et l'industrie, notre modèle apporte certains avantages clés. En effet, IROmiCS, étendant le modèle de données clinique RAVEL peut gérer un grand nombre de types de données (numériques, dates) et est extensible et adaptable à de futurs nouveaux types de données omiques. Une interface graphique utilisateur est dédiée à la visualisation et la recherche d'in-

formation dans ces données et se base sur IROmiCS. Cette interface permet l'interrogation des données cliniques et des données omiques. De plus, le moteur de recherche développé dans le cadre du projet RAVEL peut gérer les opérateurs logiques permettant d'interroger des données numériques, et des mots clés permettant des requêtes chronologiques comme décrit précédemment. Le moteur de recherche peut gérer à la fois des requêtes multi et mono patients, alors que i2b2 ne gère que les requêtes multi-patients.

Nous envisageons d'évaluer l'ergonomie et l'utilisabilité de l'outil de recherche par un ensemble de médecins et cliniciens (avec et sans formation au langage d'interrogation). Enfin, dans le cadre de la création de cohortes, la réponse de l'outil à des critères d'inclusion et d'exclusion de patients de diverses études cliniques sera prochainement évaluée. En perspective, il pourrait être intéressant de déterminer un standard pour les données de niveaux 3, basé sur la norme HL7 RIM V3. Un tel standard serait essentiel à l'industrialisation de notre solution.

5 Remerciements

Cette étude a été financée par l'Agence Nationale de la Recherche (ANR-11-TECS-012) et la région Haute-Normandie dans le cadre du projet PLaIR2.

Références

AIMÉ X., CHARLET J., FURST F., KUNTZ P., TRICHET F. & DHOMBRES F. (2012). Rare diseases knowledge management : the contribution of proximity measurements in ontoorpha and omim. *Stud Health Technol Inform*, **180**, 88–92.

CABOT C., GROSJEAN J., LELONG R., LEFEBVRE A., LECROQ T., SOUALMIA L. & DARMONI S. (2014). Omic data modelling for information retrieval. In *Proceedings of the 2nd International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO*, p. 415–424.

CHEBIL W., SOUALMIA L., DAHAMNA B. & DARMONI S. (2012). Indexation automatique de documents en santé : évaluation et analyse de sources d'erreurs. *IRBM*, **33**(5), 316–329.

DESHMUKH V. G., MEYSTRE S. M. & MITCHELL J. A. (2009). Evaluating the informatics for integrating biology and the bedside system for clinical research. *BMC Med Res Methodol*, **9**, 70.

EBI (2013). Array express.

EDGAR R., DOMRACHEV M. & LASH A. E. (2002). Gene expression omnibus : Ncbi gene expression and hybridization array data repository. *Nucleic Acids Res*, **30**(1), 207–10.

FARFAN F., HRISTIDIS V., RANGANATHAN A. & WEINER M. (2009). Xontorank : Ontology-aware search of electronic medical records. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, p. 820–831 : IEEE.

FERNALD G. H., CAPRIOTTI E., DANESHJOU R., KARCZEWSKI K. J. & ALTMAN R. B. (2011). Bioinformatics challenges for personalized medicine. *Bioinformatics*, **27**(13), 1741–8.

GARDE S., KNAUP P., HOVENGA E. & HEARD S. (2007). Towards semantic interoperability for electronic health records. *Methods Inf Med*, **46**(3), 332–43.

GROSJEAN J., MERABTI T., SOUALMIA L. F., LETORD C., CHARLET J., ROBINSON P. N. & DARMONI S. J. (2013). Integrating the human phenotype ontology into hetop terminology-ontology server. *Stud Health Technol Inform*, **192**, 961.

HANAUER D. A. (2006). Emerse : The electronic medical record search engine. *AMIA Annu Symp Proc*, p. 941.

HEBDA T., CZAR P. & MASCARA C. (2005). *Handbook of informatics for nurses and health care professionals*. Pearson Prentice Hall.

KALRA D., BEALE T. & HEARD S. (2005). The openehr foundation. *Stud Health Technol Inform*, **115**, 153–73.

LELONG R., MERABTI T., GROSJEAN J., JOULAKIAN M., GRIFFON N., DAHAMNA B., CUGGIA M., PEREIRA S., GRABAR N., THIESSARD F., MASSARI P. & DARMONI S. (2014). Moteur de recherche sémantique au sein du dossier du patient informatisé : langage de requêtes spécifique. In *15es Journées francophones d'informatique médicale*.

LOWE H. J., FERRIS T. A., HERNANDEZ P. M. & WEBER S. C. (2009). Stride–an integrated standards-based translational research informatics platform. *AMIA Annu Symp Proc*, **2009**, 391–5.

MURPHY S. N., WEBER G., MENDIS M., GAINER V., CHUEH H. C., CHURCHILL S. & KOHANE I. (2010). Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *J Am Med Inform Assoc*, **17**(2), 124–30.

NATARAJAN K., STEIN D., JAIN S. & ELHADAD N. (2010). An analysis of clinical queries in an electronic health record search utility. *International Journal of Medical Informatics*, **79**(7), 515–522.

NCBI (2013). Gene expression omnibus.

NIH (2013). The genome cancer atlas.

PEREIRA S., SAKJI S., NÉVÉOL A., KERGOULAY I., KERDELHUÉ G., SERROT E., JOUBERT M. & SJ D. (2009). Multi-terminology indexing for the assignment of mesh descriptors to medical abstracts in french. In *AMIA symp.*, p. 521–525 : IOS Press. PSIP.

RUSTICI G., KOLESNIKOV N., BRANDIZI M., BURDETT T., DYLAG M., EMAM I., FARNE A., HASTINGS E., ISON J., KEAYS M., KURBATOVA N., MALONE J., MANI R., MUPO A., PEDRO PEREIRA R., PILICHEVA E., RUNG J., SHARMA A., TANG Y. A., TERNENT T., TIKHONOV A., WELTER D., WILLIAMS E., BRAZMA A., PARKINSON H. & SARKANS U. (2013). Arrayexpress update–trends in database growth and links to data analysis tools. *Nucleic Acids Res*, **41**(Database issue), D987–90.

SARKAR I. N., BUTTE A. J., LUSSIER Y. A., TARCZY-HORNOCH P. & OHNO-MACHADO L. (2011). Translational bioinformatics : linking knowledge across biological and clinical realms. *J Am Med Inform Assoc*, **18**(4), 354–7.

SEWELL J. & THEDE L. (2012). Informatics and nursing : Opportunities and challenges. online glossary of terms.

SPAT S., CADONNA B., RAKOVAC I., GÜTL C., LEITNER H., STARK G. & BECK P. (2008). Enhanced information retrieval from narrative german-language clinical text documents using automated document classification. *Stud Health Technol Inform*, **136**, 473–8.

THIESSARD F., MOUGIN F., DIALLO G., JOUHET V., COSSIN S., GARCELON N., CAMPILLO B., JOUINI W., GROSJEAN J., MASSARI P., GRIFFON N., DUPUCH M., TAYALATI F., DUGAS E., BALVET A., GRABAR N., PEREIRA S., FRANDJI B., DARMONI S. & CUGGIA M. (2012). Ravel : retrieval and visualization in electronic health records. *Stud Health Technol Inform*, **180**, 194–8.

Ontopy : programmation orientée ontologie en Python*

Jean-Baptiste Lamy[1], Hélène Berthelot[1]

LIMICS, Université Paris 13, Sorbonne Paris Cité, 93017 Bobigny, France, INSERM UMRS 1142, UPMC Université Paris 6, Sorbonne Universités, Paris
jean-baptiste.lamy@univ-paris13.fr, helene.berthelot@orange.fr

Résumé : Les ontologies et les modèles objets partagent un vocabulaire commun mais diffèrent dans leurs utilisations : l'ontologie permet d'effectuer des inférences et les modèles objets sont utilisés pour la programmation. Il est souvent nécessaire d'interfacer ontologie et programme objet. Plusieurs approches ont été proposées, de OWL API à la programmation orientée ontologie. Dans cet article, nous présentons Ontopy, un module de programmation orientée ontologie dynamique en Python, et nous prendrons pour exemple la comparaison des contre-indications des médicaments.
Mots-clés : Ontologies, Programmation orientée ontologie, Programmation dynamique

1 Introduction

Les ontologies formelles, par exemple au format OWL (*Ontology Web Language*), structurent un domaine de connaissance pour réaliser des inférences logiques et relier les connaissances entre elles. Des éditeurs comme Protégé rendent facile la construction d'ontologies, mais leur intégration à des logiciels existants est plus compliquée (Goldman NM, 2003). Il existe des similitudes entre ontologie et modèle objet (Koide *et al.*, 2005) : les classes, propriétés et individus des ontologies correspondent aux classes, attributs et instances des modèles objets (Knublauch *et al.*, 2006). Cependant, les principaux outils comme OWL API (Horridge & Bechhofer, 2011) n'en tirent pas parti : avec ces outils une classe de l'ontologie *ne correspond pas* à une classe du langage de programmation. Ces outils sont par conséquents complexes à mettre en oeuvre et difficilement compatibles avec les méthodes de développement agile. Une approche différente consisterait à aller vers le rapprochement, voire l'unification, des ontologies et des modèles objets : c'est la *programmation orientée ontologie* (Goldman NM, 2003). Sur un exemple du W3C, cette approche a permis de réduire de moitié le volume de code source (Knublauch *et al.*, 2006).

Cet article présente Ontopy, un module Python pour la programmation orientée ontologie dynamique. Ontopy permet de créer et manipuler les classes et les instances OWL comme des objets Python, et de classifier automatiquement des classes et des instances *via* un raisonneur externe. Nous présentons ensuite le problème de la comparaison des contre-indications des médicaments, que nous réalisons avec une ontologie et un programme objet. Nous montrerons un exemple d'utilisation d'Ontopy dans ce contexte. OWL API n'a pas été utilisé car peu adapté à nos méthodes de développement agile, de plus nous souhaitions réutiliser des outils terminologiques mis au point précédemment en Python (Lamy *et al.*, 2015). Nous terminerons en comparant notre approche à la littérature.

⋆. Ce travail a été financé par l'ANSM au travers du projet de recherche VIIIP (AAP-2012-013).

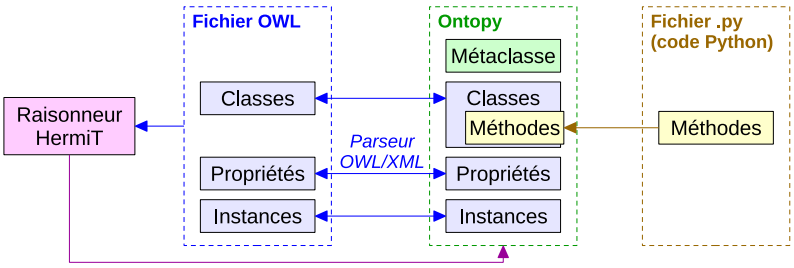


FIGURE 1 – Architecture générale d'Ontopy.

2 Ontopy : un module Python pour la programmation orientée ontologie

Ontopy est un module Python en logiciel libre (licence GNU LGPL v3, `https://bitbucket.org/jibalamy/ontopy`) pour la programmation orientée ontologie et le développement agile d'application à base d'ontologie. Le langage Python 3.4 a été choisi car il s'agit d'un langage objet dynamique avec héritage multiple. En particulier, il permet de changer la classe d'un objet ou les superclasses d'une classe en cours d'exécution, par exemple suite à la classification, ce que ne permet pas un langage statique comme Java. Ontopy permet (a) de charger des ontologies au format OWL 2 XML, (b) d'accéder au contenu de l'ontologie comme s'il s'agissait d'objets Python, (c) de créer des classes OWL en Python, (d) d'ajouter des méthodes Python aux classes OWL, et (e) d'effectuer la classification automatique des instances, classes et propriétés. Les types de données suivants sont gérés : booléen, entier, flottant, date, chaîne de caractères.

Ontopy (Figure 1) n'effectue aucune inférence (hors mise à jour des propriétés inverses) tant que le raisonneur n'est pas appelé explicitement. Ce comportement est similaire à celui de Protégé. Nous avons utilisé le raisonneur HermiT 1.3.8 (Motik *et al.*, 2009) auquel nous avons ajouté une option en ligne de commande pour obtenir en sortie la classification des instances. La classification se fait en 3 étapes : (1) exporter l'ontologie dans un fichier OWL temporaire, (2) exécuter HermiT sur ce fichier, (3) récupérer la sortie d'HermiT et appliquer les résultats en changeant les classes des instances et les superclasses des classes.

Python permet de modifier son modèle objet via un système de *métaclasses* (classe de classe). La Table 1 montre les méthodes spéciales que nous avons redéfinies pour adapter le modèle objet Python à OWL. Deux autres différences ont demandé un traitement particulier : (1) dans une ontologie, une instance peut appartenir à plusieurs classes, ce que ne permettent pas les langages objets ; dans ce cas, une classe intersection héritant des différentes classes est créée automatiquement et associée à l'objet, (2) les annotations ne sont pas héritées dans les ontologies, alors que tous les attributs le sont dans les langages objets ; c'est pourquoi nous avons placé les annotations dans un dictionnaire à part qui fait correspondre une entité (ou un triplet) à un second dictionnaire, lequel fait correspondre les propriétés d'annotation à leurs valeurs.

3 Le problème de la comparaison des contre-indications des médicaments

Le processus complexe de rédaction, structuration et codage des propriétés des médicaments conduit à une grande hétérogénéité dans les bases de données, qui com-

Méthode	Effet	Raison de la redéfinition
C.__new__	Crée un nouvel objet	Combiner la nouvelle classe à la classe OWL de même nom, si elle existe
C.__instancecheck__	Teste si un objet est une instance de la classe	Prendre en compte les classes équivalentes OWL
C.__subclasscheck__	Teste si une classe est une sous-classe de la classe	Prendre en compte les classes équivalentes OWL
C.mro	Calcule l'ordre de résolution des méthodes (*method resolution order*, MRO) notamment en cas d'héritage multiple	Ne pas déclencher d'erreur en cas de MRO temporairement incorrect lors du chargement de l'ontologie (les classes parentes étant ajoutées une à une)
i.__setattr__	Modifie un attribut de l'objet	Mettre à jour les propriétés inverses
i.__getattr__	Obtient un attribut de l'objet (appelé uniquement pour les attributs inexistants)	Retourner une liste vide si la propriété n'a pas été renseignée, ou None pour une propriété fonctionnelle

TABLE 1 – Méthodes spéciales du modèle objet de Python qui ont été redéfinies pour le rendre compatible avec OWL. Pour chaque méthode est indiqué si elle s'applique aux classes (C.) ou aux instances (i.), son effet et la raison de sa redéfinition.

Condition clinique	ticagrélor	aspirine	héparine
maladie hémorragique	CI		
maladie hémorragique acquise		CI	
maladie hémorragique constitutionnelle		CI	CI

Condition clinique	ticagrélor	aspirine	héparine
maladie hémorragique	CI	CI	CI/ok
maladie hémorragique acquise	CI	CI	ok
maladie hémorragique constitutionnelle	CI	CI	CI

TABLE 2 – Trois contre-indications pour trois médicaments, issues de la base médicament Thériaque en haut, et telles qu'interprétées par un expert en bas (CI : contre-indiqué, ok : absence de contre-indication, CI/ok : contre-indiqué dans certaines situations seulement).

plique la comparaison entre médicaments. La Table 2 (haut) montre trois exemples de situations de contre-indication pour trois médicaments, extraits de la base Thériaque (http://theriaque.org). Cependant, bien que cela n'apparaisse pas dans ce tableau, le ticagrélor est contre-indiqué avec les maladies hémorragiques acquises et constitutionnelles, car contre-indiqué dans l'ensemble des maladies hémorragiques (héritage). Et l'aspirine est contre-indiquée dans les maladies hémorragiques car contre-indiquée à la fois dans celles acquises et constitutionnelles (partition). Enfin, il est possible de déduire les situations dans lesquelles un médicament *n'est pas* contre-indiqué, par exemple les maladies hémorragiques acquises pour l'héparine (à ne pas confondre avec l'absence de mention de contre-indication dans la base). La Table 2 (bas) montre l'interprétation que ferait un expert ; nous souhaitons automatiser ce raisonnement.

Nous avons structuré les contre-indications à l'aide d'une ontologie formelle, dans laquelle les conditions cliniques associées aux contre-indications sont décrites par un code dans une terminologie et un ou plusieurs qualifieurs tels que "acquise", "constitutionnelle", "antécédent",... Ces conditions cliniques sont représentées par des classes et non des instances, afin de pouvoir prendre en compte les relations *est-un* existant entre conditions cliniques (par exemple maladie hémorragique acquise *est une* maladie hémorragique).

4 Exemple d'utilisation d'Ontopy

Nous donnons ici un exemple d'application d'Ontopy au problème de la comparaison des contre-indications. Ontopy charge les ontologies à partir des répertoires locaux définis dans la variable globale `onto_path`, ou à défaut à partir de leur URL. `onto_path` se comporte comme le `classpath` de Java ou le `pythonpath` de Python, mais pour les fichiers OWL.

```
from ontopy import *
onto_path.append("/chemin/local/des/ontos")
onto_ci = get_ontology("http://test.org/onto_ci.owl").load()
#charge /chemin/local/des/ontos/onto_ci.owl ou http://test.org/onto_ci.owl
```

L'ontologie peut ensuite être utilisée comme un module Python, et la notation pointée usuelle permet d'accéder aux éléments de l'ontologie. Des attributs (`imported_ontologies`, `classes`, `properties`, etc) permettent de récupérer la liste des éléments d'un type donné.

```
onto_ci.Médicament # La classe http://test.org/onto_ci.owl#Médicament
```

Les classes de l'ontologie peuvent être instanciées en Python. La notation pointée permet d'accéder aux relations des instances. Les relations fonctionnelles ont une valeur unique, les autres sont des listes.

```
aspirine = onto_ci.Médicament("aspirine") # onto_ci.owl#aspirine
aspirine.noms_de_marque = ["Aspirine du Rhône", "Aspirine UPSA"]
```

Il est possible de créer des classes OWL en Python, en héritant de `Thing` ou d'une classe fille. Les attributs `is_a` et `equivalent_to` sont des listes correspondant aux superclasses et aux classes équivalentes OWL. Ces listes peuvent contenir des classes, mais aussi des restrictions portant sur une propriété (définies de manière similaire à Protégé), des énumérations d'instances (*one of*), ou plusieurs de ces éléments reliés par des opérateurs logiques ET (&), OU (|) ou NON (NOT). Les classes présentes dans `is_a` sont ajoutées aux superclasses Python, en revanche les autres éléments ne sont pas traités comme des classes par Ontopy. L'exemple ci-dessous crée la classe des maladies hémorragiques acquises, fille de `Condition_clinique`, et définie comme équivalente à une condition clinique associée au terme "maladie hémorragique" et ayant pour qualifieur Acquis.

```
class Maladie_hémorragique_acquise(onto_ci.Condition_clinique):
    equivalent_to = [ onto_ci.Condition_clinique
    & onto_ci.a_pour_terme     (SOME, onto_ci.Terme_maladie_hémorragique)
    & onto_ci.a_pour_qualifieur(SOME, onto_ci.Acquis) ]
```

Nous pouvons ensuite créer la première contre-indication et la relier à l'aspirine.

```
ci1 = onto_ci.Contre_indication()
aspirine.a_pour_contre_indication.append(ci1)
```

Relier cette contre-indication aux maladies hémorragiques acquises est un peu plus compliqué, car il s'agit d'une classe et non d'une instance. Pour cela nous modifions les attributs `is_a` de la classe `Maladie_hémorragique_acquise` et de l'instance `ci1`. L'attribut `is_a` d'une instance fonctionne de manière similaire à celui d'une classe, mais contient les classes auxquels appartient l'instance. Ci-dessous, nous spécifions que la contre-indication est reliée seulement à des maladies hémorragiques acquises, et que la classe des maladies hémorragiques acquises est reliée à notre contre-indication.

```
ci1.is_a.append(
    onto_ci.a_pour_condition_clinique(ONLY, Maladie_hémorragique_acquise) )
Maladie_hémorragique_acquise.is_a.append(
    onto_ci.est_condition_clinique_de(VALUE, ci1) )
```

Créons ensuite la classe définie des conditions cliniques contre-indiquées avec l'aspirine.

```
class Condition_CI_avec_aspirine(onto_ci.Condition_clinique):
    equivalent_to = [ onto_ci.Condition_clinique
    & onto_ci.est_condition_clinique_de(SOME, onto_ci.Contre_indication
        & onto_ci.est_contre_indication_de(VALUE, aspirine) ) ]
```

Ontopy permet aussi l'ajout de méthodes Python aux classes OWL, en redéfinissant les classes dans un module Python. Ce module peut être lié à l'ontologie via une annotation, de sorte à être chargé automatiquement avec l'ontologie. L'exemple suivant ajoute une méthode `teste_ci` à la classe `Médicament`. Elle prend en paramètre une classe de condition clinique et retourne une chaîne de caractères. La méthode récupère la classe des conditions cliniques contre-indiquées avec le médicament, en se basant sur son nom, et teste si la condition clinique est une classe fille avec l'opérateur `issubclass` de Python. Puis nous lançons le raisonneur et nous affichons les résultats.

```
class Médicament(Thing):
    def teste_ci(self, Condition):
        Condition_CI = onto_ci["Condition_CI_avec_" + self.name]
        if issubclass(Condition, Condition_CI): return "CI"
        [...] # XXX tester si le médicament est OK

onto_ci.sync_reasoner() # Lance HermiT et effectue la classification
print(aspirine.teste_ci(Maladie_hémorragique)) # => "CI"
```

5 Discussion et conclusion

La programmation orientée ontologie n'est pas une idée nouvelle et le W3C a déjà suggéré l'intégration de méthodes dans des classes OWL (Knublauch *et al.*, 2006). Des approches statiques ont été proposées (Kalyanpur *et al.*, 2004; Goldman NM, 2003), qui génèrent le code source de classes Java ou C# correspondant à une ontologie en OWL. Ces approches permettent d'accéder à l'ontologie et de vérifier le typage à la compilation, mais leur nature statique n'est pas adaptée à la classification automatique. Plus récemment, une approche semi-dynamique en Java (Stevenson & Dobson, 2011) a permis la classification des instances mais pas celle des classes. Une approche dynamique a été proposée en Common Lisp (Koide *et al.*, 2005), en utilisant un algorithme de subsomption spécifique pour l'inférence et non un raisonneur externe. Un prototype en Python a aussi été réalisé (Babik & Hluchy, 2006), mais ne va pas jusqu'à une syntaxe "entièrement Python" pour

définir les restrictions ou les relations. Une troisième approche consiste à concevoir de nouveaux langages, tel que Go ! (Clark & McCabe, 2006).

Au final, peu d'approches sont allées aussi loin dans l'unification entre modèle objet et ontologie que la nôtre. Ontopy n'a pas été optimisé en terme de performance car nous n'en avons pas ressenti le besoin : le temps consommé par la manipulation de l'ontologie en Python reste négligeable comparé au temps de raisonnement. La totalité de l'ontologie est chargée en mémoire, ce qui peut poser problème sur des ontologies volumineuses. Nous avons cependant réussi à charger IDOSCHISTO, une ontologie complexe sur la schistosomiase (Camara *et al.*, 2014). Une autre limite d'Ontopy est la prise en compte d'espaces de nom multiples et d'assertions présentes dans une ontologie mais portant sur des éléments d'une autre ontologie, qui enfreignent le principe d'*encapsulation* des langages objets (l'ensemble des informations d'un objet sont placées dans une seule "capsule").

Les perspectives de développement d'Ontopy incluent (a) la liaison à un *triple store*, afin de ne pas charger la totalité des ontologies en mémoire, (b) la traçabilité de l'ontologie d'origine de chaque assertion, afin de faciliter l'emploi d'ontologies modulaires, ainsi que (c) la génération automatique de boîtes de dialogue pour éditer les instances.

Références

Babik M. & Hluchy L. (2006). Deep Integration of Python with Web Ontology Language. In *Proceedings of the 2nd workshop on scripting for the semantic web*, Budva, Montenegro.

Camara G., Despres S. & Lo M. (2014). IDOSCHISTO : une extension de l'ontologie noyau des maladies infectieuses (IDO-Core) pour la schistosomiase. In *Actes du congrès d'Ingénierie des Connaissances (IC2014)*, p. 39–50, Clermont-Ferrand, France.

Clark K. L. & McCabe F. G. (2006). Ontology oriented programming in Go. *Applied Intelligence*, **24**, 3–37.

Goldman NM (2003). Ontology-oriented programming : static typing for the inconsistent programmer. In *Lecture notes in computer science : the SemanticWeb, ISWC*, volume 2870, p. 850–865.

Horridge M. & Bechhofer S. (2011). The OWL API : A Java API for OWL ontologies. *Semantic Web 2*, p. 11–21.

Kalyanpur A., Pastor D., Battle S. & Padget J. (2004). Automatic mapping of OWL ontologies into Java. In *Proceedings of the Sixteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2004)*, p. 98–103.

Knublauch H., Oberle D., Tetlow P. & Wallace E. (2006). A Semantic Web Primer for Object-Oriented Software Developers. *W3C Working Group Note*.

Koide S., Aasman J. & Haflich S. (2005). OWL vs. Object Oriented Programming. In *the 4th International Semantic Web Conference (ISWC 2005), Workshop on Semantic Web Enabled Software Engineering (SWESE)*.

Lamy J. B., Venot A. & Duclos C. (2015). PyMedTermino : an open-source generic API for advanced terminology services. *Stud Health Technol Inform*.

Motik B., Shearer R. & Horrocks I. (2009). Hypertableau reasoning for description logics. *Journal of Artificial Intelligence Research*, **36**, 165–228.

Stevenson G. & Dobson S. (2011). Sapphire : Generating Java Runtime Artefacts from OWL Ontologies. In *Lecture Notes in Business Information Processing, Advanced Information Systems Engineering Workshops* , volume 83, p. 425–436.

Traitement des incompatibilités de candidats issus d'alignements entre plusieurs bases de connaissances

Fabien Amarger[1,2], Jean-Pierre Chanet[1], Ollivier Haemmerlé[2], Nathalie Hernandez[2], Catherine Roussey[1]

[1] UR TSCF, Irstea, 9 av. Blaise Pascal CS 20085, 63172 Aubière, France
prénom.nom@irstea.fr

[2] IRIT, UMR 5505, Université de Toulouse, UT2J 5 allées Antonio Machado, F-31058 Toulouse Cedex, France
prénom.nom@univ-tlse2.fr

Résumé : De nombreux travaux ont été proposés dans la littérature dans le but de construire des ontologies à partir de sources telles que les thesaurus ou les classifications. Certaines de ces sources sont disponibles sur le Web de données, au format SKOS. Dans nos travaux, nous proposons de construire une base de connaissances destinée à un besoin applicatif particulier, en exploitant un ensemble de sources disponibles sur le domaine considéré. L'originalité de notre approche réside dans le fait d'exploiter la redondance entre les sources afin d'en extraire des candidats (classes, individus, propriétés...). Nous présentons dans cet article la notion d'incompatibilité entre candidats, qui résulte de l'hypothèse de travail selon laquelle nous ne considérons que des relations d'équivalence simple entre les sources. Nous présentons également la génération de sous-ensembles de candidats compatibles afin d'obtenir un consensus cohérent entre les sources. Cette approche a été évaluée sur un cas d'étude réel concernant le domaine de la taxonomie du blé, réalisée en collaboration avec un expert.

Mots-clés : Acquisition de connaissances, candidat d'élément ontologique, traitement d'incompatibilités, taxonomie du blé

1 Introduction

Les données relatives à des domaines particuliers sont, le plus souvent, disponibles sur le Web dans des formes structurées (comme les bases de données ou les thésaurus) et sont consacrées à un usage donné. Les utilisateurs finals peuvent être déroutés face à l'abondance de données disponibles, de qualité différente, provenant de sources différentes, exprimées dans des formalismes différents. L'intérêt du Web de données liées ou Linked Open Data (LOD) est de faciliter l'interrogation de ces données ouvertes en permettant d'établir des liens entre elles. Des approches (Soergel *et al.* (2004); Villazón-Terrazas *et al.* (2010)) ont été proposées afin de formaliser la transformation de données structurées dans le but de les publier sur le LOD. Néanmoins, ces approches nécessitent beaucoup de temps et d'interactions avec l'utilisateur pour être efficaces. De plus, très peu d'approches exploitent l'intérêt de la transformation multi-sources, Amarger *et al.* (2013). Nous avons donc proposé une méthode permettant la construction d'une base de connaissances (BC) pour un domaine spécifique, réutilisant plusieurs sources non-ontologiques, telles que des thésaurus ou des classifications. L'idée principale est de modéliser un domaine d'étude en construisant un module ontologique. Chaque source est ensuite analysée pour enrichir le module et ainsi le peupler avec de nouveaux éléments extraits de la source. Nous obtenons donc plusieurs versions du module, chaque version du module est une BC issue de la source. Les BC sont ensuite alignées par des outils d'alignement. Un regroupement d'éléments alignés provenant de différentes BC est appelé un *candidat*. L'objectif de nos

travaux est d'extraire les connaissances communes à plusieurs BC, Amarger *et al.* (2014).

Nous proposons ici une évolution de cette approche en partant d'une hypothèse liée aux alignements entre BC. Un alignement entre deux BC est un ensemble de correspondances entre éléments ontologiques des BC considérées. Nous ne considérons comme correspondances entre éléments ontologiques que des relations d'équivalence de cardinalité 1 :1, dites correspondances simples. Rappelons qu'un regroupement d'éléments ontologiques alignés constitue un candidat. Les outils d'alignements sont capables d'obtenir des correspondances d'équivalence ayant une cardinalité 1 :n ce qui nous amène une ambiguïté. Les correspondances 1 :n peuvent générer au moins n candidats. Nous nommons ces ensembles de candidats des candidats incompatibles. L'objectif du travail présenté est de faciliter la validation des candidats. Nous allons donc chercher à découvrir des sous-ensembles de candidats compatibles entre eux. Cet ensemble est nommé une extension. Une extension ne doit pas être incluse dans une autre. L'objectif est de trouver l'extension la plus grande validée par un expert.

Cet article est organisé de la façon suivante : (1) présentation du processus général, (2) génération d'extension de candidats compatibles, (3) travaux connexes, (4) évaluation sur la taxonomie des blés.

2 Processus général

Notre méthode se compose de trois étapes : (1) "analyse de sources" qui permet de déterminer quelles sources vont être utilisées dans le processus, (2) "transformation des sources" qui permet d'obtenir une base de connaissances source pour chaque source par transformation automatique, en se fondant sur un module ontologique donné et (3) "la fusion des bases de connaissances sources". C'est sur cette dernière étape que se fonde en grande partie l'originalité de notre approche, puisqu'elle permet la fusion de ces différentes bases de connaissances sources en se basant sur l'idée que plus une connaissance apparaît dans plusieurs sources et plus sa confiance augmente. Nous avons détaillé ces processus dans des travaux précédents, Amarger *et al.* (2014).

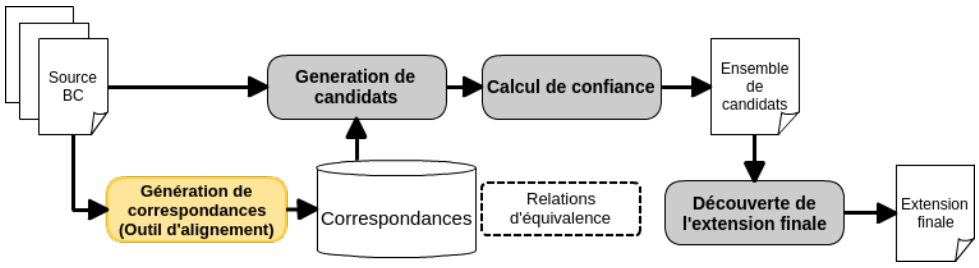


FIGURE 1 – Processus de fusion de bases de connaissances

La figure 1 présente le processus de fusion des bases de connaissances sources en 4 étapes. Les trois premières ayant été décrites dans nos travaux précédents (Amarger *et al.* (2014) et Roussey *et al.* (2013)), nous nous focalisons dans cet article sur la dernière activité (Découverte de l'extension finale).

3 Génération d'extension de candidats compatibles

Partant du constat que les outils d'alignements proposent des correspondances (Euzenat & Shvaiko (2007)) d'équivalence de cardinalité 1 :n, nous présentons dans un premier temps notre méthode afin de générer des candidats. Nous étudierons ensuite comment gérer les incompatibilités entre candidats par rapport au constat précédent et nous verrons enfin comment générer une extension.

Candidats : Un candidat $cand$ représente un élément susceptible d'appartenir à la base de connaissances finale que nous cherchons à construire. Cet élément est le résultat de la fusion d'éléments issus de plusieurs BC, jugés équivalents par des outils d'alignement. Un candidat $cand = (V_{cand}, E_{cand})$ est un graphe non-orienté connexe dont les sommets V_{cand} sont des éléments ontologiques provenant de BC différentes et les arêtes E_{cand} sont les correspondances d'équivalence. Nous considérons ici qu'un candidat n'existe que s'il contient au moins deux éléments ontologiques provenant de deux BC différentes. Les candidats avec un seul élément ontologique ne sont pas considérés car nous cherchons les connaissances communes. De plus nous cherchons à générer les candidats maximaux, ce qui signifie que nous ne considérons pas les candidats inclus dans un autre.

Incompatibilités : Nous posons une incompatibilité entre les candidats partageant un élément ontologique, ce qui implique qu'un seul de ces candidats peut faire partie de la base de connaissances finale. Nous obtenons ici un graphe d'incompatibilité dont les sommets sont des candidats et les arêtes des incompatibilités.

Extensions : Une extension est le sous-ensemble de candidats n'ayant pas de lien d'incompatibilités entre eux. Trouver toutes les extensions revient à résoudre un problème connu en théorie des graphes : MCE (Maximum Clique Enumeration). Il faut pour cela considérer le graphe complémentaire au graphe d'incompatibilité et donc rechercher toutes les cliques maximales. Pour résoudre ce problème de MCE, plusieurs algorithmes existent. Le plus courant est le Bron Kerbosch (Bron & Kerbosch (1973)). Il existe néanmoins un certain nombre d'améliorations de cet algorithme, tel que l'algorithme de Tomita (Tomita *et al.* (2006)) ou, plus récemment, l'algorithme de Eppstein et Strash (Eppstein & Strash (2011)). Ce problème étant NP-difficile, il est inenvisageable de découvrir toutes les cliques maximales possibles. Notre problème est donc de trouver un moyen d'obtenir une extension et de la faire valider par un utilisateur dans ces conditions de complexité. Pour résoudre ce problème, nous utilisons un solver CSP (Constraint Satisfaction Problem), GLPK (GNU Linear Programming Kit – `https://www.gnu.org/software/glpk/`), pour utiliser la technique de Branch and Bound qui nous permettra de trouver une solution au problème en maximisant une fonction objective. Nous cherchons ici à maximiser le nombre de candidats dans une extension. De cette manière, la première extension rencontrée qui maximise le nombre de candidat sera retournée en un temps fini.

Validation : Une phase de validation des candidats est utile pour ajouter des contraintes au problème afin de converger vers une solution optimale, autrement dit, pour ce qui nous concerne, vers une extension dont tous les candidats sont validés. L'idée étant de présenter à un expert les candidats d'une extension, un à un, pour validation. Si l'expert valide le candidat présenté, alors on peut ajouter la contrainte selon laquelle l'extension doit contenir ce candidat. Inversement, si l'expert ne valide pas le candidat, alors on ajoute la contrainte selon laquelle l'extension ne doit pas contenir ce candidat. À chaque candidat non validé, l'algorithme est relancé afin de trouver

une nouvelle extension avec les nouvelles contraintes. L'algorithme s'arrête quand il n'y a plus de candidats à valider dans l'extension. Grâce à cette méthode, nous pouvons être assurés de ne présenter à l'expert qu'un minimum de candidats à valider en déduisant automatiquement que tous les candidats incompatibles avec un candidat validé ne peuvent pas apparaître dans l'extension finale. Ceci permet de réduire le nombre d'interactions pour faciliter le travail d'un expert. Le temps nécessaire pour la validation par l'expert des candidats est, dans le pire des cas, égal, en nombre d'interactions, au nombre de candidats générés. En d'autres termes, dans le pire des cas, l'expert aura à valider tous les candidats s'ils sont tous compatibles les uns avec les autres. Dès qu'une incompatibilité apparaît, ce nombre d'interactions est forcément diminué.

4 Travaux connexes

Les travaux connexes concernant la fusion de bases de connaissances et plus particulièrement la gestion des incompatibilités (ou incohérences) sont assez récents. La plupart des travaux (Trojahn *et al.* (2011); Abbas & Berio (2013); Raunich & Rahm (2014)) cherchent à détecter des incompatibilités entre les différentes correspondances établies entre deux sources par une ou plusieurs approches d'alignement. L'idée sous-jacente est de déterminer quelles correspondances peuvent être ignorées dans le but de lever l'incompatibilité. Dans ces travaux, une incompatibilité est détectée lorsque les correspondances établies rendent la base de connaissances inconsistantes d'une point de vue logique. Certains travaux (Abbas & Berio (2013),Trojahn *et al.* (2011)) considèrent également les préférences des deux agents utilisant chacun des ontologies pour établir ces incompatibilités. Le traitement des incompatibilités se fait soit par l'utilisation de règles (Raunich & Rahm (2014)), soit en cherchant des sous-ensembles compatibles, notamment en utilisant la théorie de l'argumentation (Trojahn *et al.* (2011)). Notre approche se place très clairement dans cette deuxième catégorie : nous pouvons assimiler nos candidats à des arguments et nos incompatibilités à des attaques entre arguments en suivant la théorie de Dung (1995). La différence est ici que nous manipulons des incompatibilités entre candidats d'éléments ontologiques composés de correspondances entre plusieurs sources. Nous ne remettons en question que les correspondances $1 : n$ ayant mené à la génération de plusieurs candidats incompatibles. Nous identifions le candidat valide en exploitant les correspondances.

5 Évaluation sur la taxonomie des blés

Les jeux de données que nous avons utilisés proviennent d'un projet de construction d'une base de connaissances sur les céréales Roussey *et al.* (2013) et Amarger *et al.* (2014). Nous avons utilisé uniquement les données concernant la taxonomie des blés. Pour ce faire, nos experts ont sélectionné les sources suivantes :
Agrovoc (http ://aims.fao.org/standards/agrovoc/about),
TaxRef (http ://inpn.mnhn.fr/programme/referentiel-taxonomique-taxref),
NCBI Taxonomy (http ://www.ncbi.nlm.nih.gov/).

En utilisant le module ontologique AgronomicTaxon [1](Roussey *et al.* (2013)) et des patrons de transformations adaptés à chacune des trois sources, nous avons utilisé notre approche pour générer 3 BC. Ces BC ont été alignées avec LogMap, Jiménez-Ruiz & Grau (2011). Nous avons

1. https://sites.google.com/site/agriontology/home/irstea/agronomictaxon

généré un ensemble de candidats. Comme présenté précédemment, nous générons le graphe d'incompatibilités des candidats. Le tableau 1 présente quelques données sur ce graphe.

Sources	nb_{eo}	nb_{cand}	nb_{incomp}
Agrovoc	11	150	1555
TaxRef	19		
NCBI	130		

TABLE 1 – Graphe d'incompatibilités

$nb_{ext_{max}}$	$nb_{ext_{finale}}$	$nb_{interaction}$	$Ratio$
25	23	62	0.41

TABLE 2 – Résultats

eo : élément ontologique, ext_{max} *: extension maximum possible,* ext_{finale} *: extension finale,* $Ratio = \frac{nb_{interaction}}{nb_{cand}}$

En utilisant notre méthode, nous avons demandé à un expert de valider ces candidats en comptant le nombre d'interactions qui ont été nécessaires pour obtenir l'extension finale. Nous obtenons les résultats présentés dans le tableau 2.

Nous pouvons observer plusieurs faits notables dans ce tableau. Tout d'abord, la taille de l'extension maximale possible au début de l'exercice est de 25 candidats, alors que la taille de l'extension validée est de 23 candidats. Ceci vient du fait que 2 candidats n'ont pas été validés par l'expert ainsi que tous les candidats incompatibles avec ceux-ci. De plus, il a fallu 62 interactions de l'expert. Il a donc validé ou invalidé 62 candidats sur les 150 présents initialement. On observe donc un ratio d'interactions de 0.41, ce qui signifie que 41% des candidats ont dû être observés par l'expert afin d'obtenir l'extension finale. Cette technique est donc avantageuse puisque l'on gagne plus de 50% des interactions nécessaires à la validation de tous les candidats générés.

6 Conclusion

Nous avons présenté dans cet article un moyen de générer des incompatibilités entre des candidats provenant de l'extraction multi-source d'éléments ontologiques. Ces incompatibilités peuvent être utilisées pour générer des extensions, des sous-ensembles cohérents de candidats. Nous avons aussi présenté un moyen de valider les candidats en exploitant ces incompatibilités pour limiter les interactions de l'expert afin d'obtenir l'extension optimale. Cette méthode a été validée sur un jeu de données réel provenant de plusieurs sources pour la création d'une base de connaissances sur la taxonomie des plantes.

Il serait intéressant de faire évoluer ces travaux en utilisant des scores associés aux candidats Amarger *et al.* (2014) et des pondérations des correspondances proposés par les outils d'alignement, pour améliorer la fonction objective à optimiser. L'utilisation des scores des candidats dans la fonction objective permettrait de présenter en premier les candidats les plus pertinents.

Durant les évaluations, un phénomène est apparu qui pourrait permettre de réduire considérablement le nombre d'interactions de l'expert pour la validation des candidats. Parmi les candidats incompatibles issus d'une même correspondance $1 : n$, les candidats non validés par l'expert comportent moins de labels communs que le candidat validé par l'expert. Il serait donc intéressant de pouvoir présenter en premier les candidats partageant le plus de label. Cette idée est généralisable en considérant tout le voisinage des candidats (et pas seulement les labels) dans la fonction objective à maximiser, ce qui reviendrait à privilégier les candidats partageant le plus de voisins communs (sommets ou arcs) : par exemple des sommets labels alignés ou des candidats sommets liés par la même relation.

Références

ABBAS M. & BERIO G. (2013). Creating ontologies using ontology mappings : Compatible and incompatible ontology mappings. *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, p. 143–146.

AMARGER F., ROUSSEY C., CHANET J., HAEMMERLÉ O. & HERNANDEZ N. (2013). Etat de l´art : Extraction d´information à partir de thésaurus pour générer une ontologie. *INFORSID*, p. 29–44.

AMARGER F., ROUSSEY C., CHANET J., HAEMMERLÉ O. & HERNANDEZ N. (2014). Skos sources transformations for ontology engineering : Agronomical taxonomy use case. *MTSR*.

BRON C. & KERBOSCH J. (1973). Algorithm 457 : finding all cliques of an undirected graph. *Communications of the ACM*, p. 575–577.

DUNG P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, p. 321–357.

EPPSTEIN D. & STRASH D. (2011). Listing all maximal cliques in large sparse real-world graphs. *Experimental Algorithms*, p. 364–375.

EUZENAT J. & SHVAIKO P. (2007). *Ontology matching*.

JIMÉNEZ-RUIZ E. & GRAU B. C. (2011). Logmap : Logic-based and scalable ontology matching. *The Semantic Web–ISWC 2011*, p. 273–288.

RAUNICH S. & RAHM E. (2014). Target-driven merging of taxonomies with atom. *Information Systems*, p. 1–14.

ROUSSEY C., CHANET J., CELLIER V. & AMARGER F. (2013). Agronomic taxon. In *WOD*, p.5̃.

SOERGEL D., LAUSER B., LIANG A., FISSEHA F., KEIZER J. & KATZ S. (2004). Reengineering thesauri for new applications : The AGROVOC example. *Journal of Digital Information*, p. 1–23.

TOMITA E., TANAKA A. & TAKAHASHI H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, p. 28–42.

TROJAHN C., EUZENAT J., TAMMA V. & PAYNE T. R. (2011). Argumentation for reconciling agent ontologies. p. 89–111.

VILLAZÓN-TERRAZAS B., SUÁREZ-FIGUEROA M. C. & GÓMEZ-PÉREZ A. (2010). A pattern-based method for re-engineering non-ontological resources into ontologies. *Int. J. Semantic Web Inf. Syst.*, p. 27–63.

LOVMI : vers une méthode interactive pour la validation d'ontologies

Marion Richard[1,4], Xavier Aimé[1,4], Marie-Odile Krebs[2,4], Jean Charlet[1,3,4]

[1] INSERM UMRS 1142, LIMICS, F-75006, Paris, France
Sorbonne Universités, UPMC Univ. Paris 06, UMR_S 1142, LIMICS, F-75006, Paris, France
Université Paris 13, Sorbonne Paris Cité, LIMICS, (UMR_S 1142), F-93430, Villetaneuse, France
marion.richard1@etu.upmc.fr, xavier.aime@inserm.fr

[2] Laboratoire de Pathophysiologie des Troubles Psychiatriques, Centre Hospitalier Sainte-Anne
mo.krebds@ch-sainte-anne.fr

[3] Assistance publique - Hôpitaux de Paris, France
jean.charlet@inserm.fr

[4] GDR Psychiatrie CNRS 3557, Paris, France.

Résumé :
Les méthodes de construction d'ontologies se sont fortement développées au travers du traitement automatique du langage et de l'intérêt croissant aux corpus de données volumineux, engendrant un effacement progressif des acteurs du domaine au profit du traitement des données du domaine. Cependant, quelle que soit la ressource utilisée, la validation des ontologies demeure une question centrale de l'ingénierie des connaissances. Elle s'articule autour de deux problématiques complémentaires : (1) la validation structurelle et (2) la validation sémantique (de l'adéquation au domaine modélisé). Dans le premier cas, de nombreuses méthodes ont vu le jour offrant des supports réalisant automatiquement les tâches de validation. A contrario, les méthodes pour la recherche du second cas sont encore peu nombreuses. Nous proposons dans cet article la méthode LOVMI, mise en œuvre pour la validation structurelle et sémantique du module « facteurs sociaux et environnementaux des maladies psychiatriques » de notre ontologie ONTOPSYCHIA.

Mots-clés : Ontologies, validation d'ontologies, évaluation d'ontologies, psychiatrie, facteurs sociaux et environnementaux.

1 Introduction

Une ontologie, suivant la définition consensuelle de Gruber (1995), est une formalisation d'une conceptualisation partagée. C'est un artefact qui permet la recherche sémantique, les raisonnements formels ou encore l'intégration de données. Les méthodes de construction d'ontologies se sont fortement développées au travers du traitement automatique du langage et de l'intérêt croissant pour les corpus de données volumineux. De nombreuses méthodes de construction semi-automatiques ont vu le jour telles que, ONTOLEARN (Velardi *et al.*, 2013), ARCHONTE (Bachimont *et al.*, 2002; Charlet *et al.*, 2006) ou TERMINAE (Aussenac-Gilles *et al.*, 2008). On peut ainsi constater que pour créer une ontologie de domaine, la place des acteurs – jusque-là centrale – se trouve petit à petit effacée au profit du traitement des données. Outre les capacités de l'ontologue à capter la modélisation adéquate au domaine, ces méthodes soulèvent donc le problème de la validité des modélisations, sans l'intervention systématique de spécialistes qui utilisent quotidiennement les connaissances ainsi modélisées.

Le domaine médical est de plus en plus demandeur de systèmes fondés sur des ontologies, afin de permettre notamment le codage d'actes médicaux ou l'aide au diagnostic. De nombreuses ontologies ont été réalisées à ce jour comme, par exemple GENE ONTOLOGY en génétique (Ashburner *et al.*, 2000) ou encore FOUNDATIONAL MODEL OF ANATOMY en anatomie

(Rosse *et al.*, 2003). Dans le domaine de la psychiatrie, on peut citer les travaux de Kola *et al.* (2010) qui se sont intéressés aux problèmes d'hétérogénéité des données et au besoin d'interopérabilité. Hastings *et al.* (2012) ont développé une « ontologie réaliste » modélisant les maladies mentales. Plus récemment, Silva *et al.* (2014) ont relancé les discussions sur la modélisation des diagnostics en s'intéressant au développement d'un *ontology-based system* pour l'aide au diagnostic des maladies mentales, et pour permettre une description plus rationnelle des diagnostics.

Dans nos travaux, nous nous intéressons à la fois aux problématiques d'hétérogénéité, d'interopérabilité et de modélisation des troubles mentaux. Nous développons une ontologie de la psychiatrie, OntoPsychia, fondée sur les comptes rendus d'hospitalisation (CRH). Cette ontologie, associée à des outils de fouilles sémantiques, permettra à terme (1) une meilleure représentation de la comorbidité, (2) la recherche sémantique dans les CRH, (3) une indexation des CRH pour constituer des cohortes et (4) l'identification de profils de patients résistants. Nous considérons, en outre, que ce genre d'approche peut contribuer au développement d'un consensus autour des catégories descriptives des troubles psychiatriques.

La validation des ontologies est devenue une problématique centrale de l'Ingénierie des Connaissances (IC). Malgré un manque de consensus, l'utilisation et la volonté de réutilisation des ontologies au sein d'applications dédiées – tels que les systèmes de questions/réponses ou d'extraction de relations/concepts – poussent au développement de méthodes de validation. La validation s'articule aujourd'hui autour de deux axes : (1) la validation de la structure de l'ontologie et (2) la validation vue sous l'angle du contenu sémantique en adéquation avec la réalité modélisée.

Dans notre article, nous proposons la méthode LOVMI (Les Ontologies Validées par Méthode Interactive) pour la validation d'ontologies, en nous fondant sur le module « facteurs sociaux et environnementaux » de notre ontologie OntoPsychia. En section 2, nous présentons un état de l'art sur les méthodes et outils pour la validation d'ontologies. La section 3 décrit le matériel sur lequel notre expérimentation se base. Enfin, la section 4 est consacrée à la présentation de notre méthode de validation structurelle et sémantique d'ontologies. Nous discutons cette méthode en 5 avant de conclure et d'introduire les perspectives de ce travail.

2 L'art de valider une ontologie

Le développement des méthodes de construction d'ontologies fondées sur l'extraction de termes spécialisés au sein de corpus, a engendré un effacement progressif des acteurs du domaine et placé l'ontologue au centre du processus. Cependant, ces acteurs du domaine demeurent les détenteurs de la connaissance encyclopédique et pratique qui peut faire défaut à l'ontologue. On observe également que ces méthodes automatiques ont permis de développer des ontologies plus volumineuses, entraînant du même coup une plus grande difficulté à assurer une modélisation adéquate et correcte. La validation d'ontologies est par conséquent devenue une problématique à part entière de l'IC. Dans un ouvrage dédié à cette problématique, Vrandečić (2009) a relevé trois scénarios qui justifient la validation et que nous résumons comme tels : (1) une ontologie adéquate permettra une meilleure réutilisation des données ; (2) les ontologues ont besoin de méthodes pour évaluer et valider leurs modèles afin de les encourager à partager leurs résultats avec la communauté et également leur permettre de réutiliser avec confiance le travail des autres à leurs propres fins ; (3) les méthodes d'évaluation d'ontologies

permettent de vérifier automatiquement si les contraintes et les exigences sont remplies et de révéler ainsi les problèmes de plausibilité. Cela diminue les coûts d'entretien des ontologies.

On note également dans la littérature, que la validation se définit sous deux aspects complémentaires : (1) la *validation structurelle* qui peut être réalisée automatiquement grâce au développement d'outils dédiés et (2) la *validation sémantique* qui peine encore à trouver des méthodes consensuelles.

2.1 La définition des critères de validation

Avant même que les méthodes de validation d'ontologies ne commencent à se développer, les chercheurs se sont intéressés aux critères permettant d'affirmer qu'une ontologie est valide.

Guarino & Welty (2000) énoncent des critères qui visent à définir un cadre permettant d'affirmer qu'une ontologie est valide, et qui sont basés sur des principes philosophiques existants : l'*essence* et la *rigidité* ; l'*identité* et l'*unité* ; la *dépendance*.

Plus récemment, Poveda-Villalón *et al.* (2012) ont réalisé une analyse des outils de validation disponibles. Ils ont ainsi défini six dimensions qui permettent de conclure à une ontologie de qualité : (1) la *compréhension humaine* qui définit si l'ontologie fournit suffisamment d'informations pour être comprise par un humain ; (2) la *consistance logique* qui se réfère au fait qu'il y ait (i) des inconsistances logiques ou (ii) des bouts de l'ontologie qui puissent potentiellement mener à une inconsistance, sans pour autant être détectables par un raisonneur ; (3) les *problèmes de modélisation* qui se posent si l'ontologie n'est pas définie en utilisant correctement les primitives données par les langages d'implémentation d'ontologies, ou si des choix de modélisation pourraient être améliorés ; (4) la *spécification du langage ontologique* qui indique si l'ontologie est conforme aux spécifications du langage ontologique utilisé pour implémenter l'ontologie ; (5) la *représentation du monde réel* qui renvoie à la précision de la modélisation ontologique du domaine – cette dimension doit être vérifiée par des humains ; et (6) l'*intégration à des applications sémantiques* qui indique si l'ontologie est adaptée pour les applications qui lui sont destinées.

2.2 Validation de la structure

La validation de la structure a mené à de nombreuses études et applications. Le raisonneur HERMIT [1] (Shearer *et al.*, 2008) permet de vérifier la consistance et la cohérence d'un modèle et donc de répondre aux questions : « existe-t-il un monde qui soit représenté par l'ontologie ? », ou « toutes les classes sont-elles satisfaites ? ». On peut également citer les raisonneurs PELLET [2] et FACTPLUS [3] parmi les plus utilisés en raison de leur présence en tant que module d'extension dans PROTÉGÉ [4].

ONTOCHECK [5] (Schober *et al.*, 2012) se présente lui aussi comme un module d'extension à l'éditeur d'ontologie PROTÉGÉ et vise à contrôler le respect des conventions de nommage, ainsi

1. http://hermit-reasoner.com/
2. https://github.com/complexible/pellet
3. http://owl.man.ac.uk/factplusplus/
4. http://protege.stanford.edu/products.php
5. http://protegewiki.stanford.edu/wiki/OntoCheck

que l'exhaustivité des méta-données. Sous l'éditeur NEON [6], le module d'extension XD ANALYZER [7] a été développé, pour permettre de faire un retour qualitatif à l'utilisateur en suivant la méthodologie XD. Cette dernière fournit une liste de bonnes pratiques (concernant les labels, les commentaires, les concepts non utilisés, etc.) à respecter pour la construction d'ontologies.

ONTOCLEAN [8] (Guarino & Welty, 2000) est une méthodologie permettant la validation de l'adéquation des relations taxonomiques d'une ontologie. Elle consiste à annoter les concepts selon les méta-propriétés de rigidité, d'unité et de dépendance. Ensuite une analyse fondée sur des règles de contraintes prédéfinies est réalisée sur les annotations, afin de mettre en avant les erreurs taxonomiques. ONTOCLEAN a été implémentée dans deux principales applications : ODECLEAN et AEON [9]. ODECLEAN (Fernández-López & Gómez-Pérez, 2002) est un module d'extension sous l'éditeur d'ontologies WebODE [10]. Pour son développement, les auteurs ont créé une top ontologie des universaux en suivant la méthode ONTOCLEAN. Ils y ont donc inclus les méta-propriétés. Ensuite, ils ont ajouté à la top ontologie des universaux, les règles de contraintes associées aux méta-propriétés d'ONTOCLEAN, via leur éditeur d'axiomes et de règles WAB. L'utilisateur de WEBODE peut ainsi choisir d'utiliser les principes d'ONTOCLEAN et assigner à chaque concept sa valeur en tant que méta-propriété. L'ontologie peut ensuite être validée grâce au module ODECLEAN qui implémente les contraintes associées aux méta-propriétés. AEON (Völker *et al.*, 2008) met en avant les contraintes de temps liées à la méthode ONTOCLEAN qui oblige à l'annotation manuelle des concepts et à une intervention d'ontologues particulièrement expérimentés. Le but d'AEON est donc d'annoter automatiquement les concepts de l'ontologie en suivant la méthode ONTOCLEAN, puis de réaliser la vérification des contraintes. Pour annoter automatiquement les concepts, ils réalisent une concordance lexico-syntaxique sur le Web.

OntOlogy Pitfall Scanner ! (OOPS !) [11] (Poveda-Villalón *et al.*, 2012) est un outil indépendant de tout éditeur d'ontologies. Le but de OOPS ! est l'identification des anomalies ou mauvaises pratiques dans une ontologie. Pour cela les auteurs ont défini un certain nombre de « pitfalls » (embûches, pièges) répertoriés en langage naturel dans un catalogue. On en compte actuellement 40, dont 32 sont implémentés en tant que classes java et ajoutés au module d'analyse des pitfalls. En entrée, l'application prend l'URI d'une ontologie ou bien le code source en RDF [12]. L'ontologie est chargée via l'API Jena avant d'être analysée pour en extraire les erreurs potentielles. Le résultat est une page Web sur laquelle sont répertoriées les pitfalls (les erreurs identifiées) accompagnées d'une proposition de résolution. Les pitfalls peuvent concerner des éléments individuels, plusieurs éléments ou toute l'ontologie. Une méthodologie du même type avait déjà été utilisée avec succès pour valider une ontologie développée au sein de notre équipe (Charlet *et al.*, 2012).

6. http://neon-toolkit.org/wiki/Main_Page.html
7. http://neon-toolkit.org/wiki/XDTools.html
8. http://c2.com/cgi/wiki?OntoClean
9. https://code.google.com/p/aeon-project/downloads/list
10. http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/old-technologies/60-webode
11. http://oops.linkeddata.es/
12. Pour respect de la confidentialité, il est possible de déclarer qu'on ne souhaite pas que le code chargé sur la page soit conservé.

2.3 Validation de la sémantique

La validation de la sémantique est un sujet nettement moins riche dans la littérature. Cette étape met en jeu des aspects communicationnels entre acteurs de domaines d'expertise différents : les ontologues et les spécialistes du domaine modélisé dans l'ontologie. Ghidini *et al.* (2009) précisent que dans le cas de la description de modèles d'entreprises, une seule personne ne peut posséder toutes les connaissances et compétences lui permettant de modéliser l'entièreté du domaine. Les auteurs se sont alors fondés sur un système de wiki pour développer un outil de construction collaborative d'ontologies. Ainsi, à chaque élément du modèle est associé une page MOKI contenant des informations structurées sur l'élément et qui peut être comprise par n'importe quel acteur ayant des connaissances techniques ou non. En effet, chaque page contient une description informelle de l'élément sous forme de texte libre, d'image ou de dessin et une partie structurée (par le biais d'un formulaire) dans laquelle les éléments sont décrit sous forme de triplets (sujet, relation, objet).

Le développement collaboratif a également été adopté par Ressad-Bouidghaghen *et al.* (2013), qui expliquent qu'il est plus facile de travailler de façon collaborative lors de grands projets de construction de ressources sémantiques. En effet, cela permet d'établir dès la construction une « modélisation consensuelle » acceptée et validée par les différents acteurs. Ils mettent également en avant la construction modulaire afin que chaque acteur puisse intervenir dans son domaine ou sous-domaine de compétence. Pour chaque module, un ontologue est désigné comme acteur responsable. Lors de l'intégration des modules, les choix qui font débat ou qui se heurtent à la modélisation d'autres modules sont discutés et votés, pour établir un consensus autour du point de vue adopté pour la modélisation. Cette méthode permet de garder une trace des décisions, de développer plusieurs modules en parallèle, et d'établir rapidement un consensus autour des questions de lexique, de sémantique, ou de modélisation.

Concernant les outils pour le développement collaboratif d'ontologies, on peut citer : le serveur ONTOLINGUA [13] qui propose un environnement collaboratif pour parcourir, créer, éditer, modifier ou utiliser les ontologies (Farquhar *et al.*, 1997) ; WEBPROTÉGÉ [14] qui a été développé en reprenant l'architecture de PROTÉGÉ et qui est accessible via n'importe quel navigateur web (Tudorache *et al.*, 2013) ; et Ressad-Bouidghaghen *et al.* (2013) qui indique qu'un module est en cours de construction pour permettre le développement collaboratif avec TERMINAE [15].

Cependant, le développement collaboratif a ses limites. Dans le cas des méthodes de construction entièrement collaboratives, il demande une grande disponibilité des acteurs. Et dans le cas de la construction d'ontologies, il implique de posséder un certain nombre de compétences techniques non négligeable, notamment en formalisme logique. C'est le point de vue adopté par Abacha *et al.* (2013) qui constatent que la collaboration avec des médecins, oblige à trouver des stratagèmes leur permettant de s'impliquer dans la validation de l'ontologie, sans avoir à toucher au modèle. Ils proposent donc une approche par un système de validation par questions/réponses et invalidation par texte libre. Le médecin se trouve face à une liste de questions booléennes. Lorsqu'il invalide une question, il peut fournir une justification sous forme de texte libre afin de permettre à l'ontologue de corriger le formalisme incriminé. À noter que les auteurs partent du postulat qu'une réduction de la communication entre experts et ontologues

13. http://www.ksl.stanford.edu/software/ontolingua/
14. http://webprotege.stanford.edu/
15. http://lipn.fr/terminae/index.php/Download

réduit les erreurs. Cependant, en état actuel leur méthode se trouve limitée par le grand nombre de questions générées pour des ontologies de taille de plusieurs milliers de concepts. Pour pallier ce problème, les auteurs proposent de mettre en place une validation au fur et à mesure de la construction de l'ontologie.

3 Matériel : le module « facteurs sociaux et environnementaux » d'OntoPsychia

Le développement de notre ontologie se fonde sur une méthodologie combinant approche modulaire et modélisation des aspects sociaux de la vie d'un patient. Nous avons suivi la méthodologie illustrée dans Charlet *et al.* (2006), qui combine une approche top-down et une approche botom-up. Cette méthode permet d'avoir accès aux termes qui représentent le concept en usage. Nous avons extrait un ensemble de termes spécialisés (via le logiciel Yatea (Aubin & Hamon, 2006)) dans un corpus composé de 8 000 CRH anonymisés avec le logiciel Medina (Grouin & Névéol, 2014) et issu de l'hôpital psychiatrique parisien Sainte-Anne. Nous avons fait le choix de ne pas tenir compte des classifications existantes en psychiatrie (telles que la CIM-10 [16] ou le DSM-IV [17]) durant la première étape de construction d'OntoPsychia, pour nous concentrer uniquement sur les informations contenues dans les CRH (approche bottom-up). La seconde étape de construction permettra d'enrichir notre modèle avec les classifications (approche top-down). Pour faire face à l'abondance de connaissances disponibles en psychiatrie, nous avons fait le choix d'une approche modulaire et ainsi décidé de construire trois modules : (1) facteurs sociaux et environnementaux, (2) troubles/maladies mentales et (3) traitements. Chaque concept est dénoté par un label préférentiel en anglais et en français, et un ou des label(s) alternatif(s) (synonyme, acronyme). Les labels sont issus du corpus et des nomenclatures (DSM, CIM et SNOMED).

Nous avons fait le choix de ce module sur les facteurs sociaux et environnementaux afin d'établir une corrélation entre contexte/événement et santé mentale. OntoPsychia pour les facteurs sociaux et environnementaux contient actuellement 1450 classes, une profondeur maximale de 6, un nombre maximale de frères de 14, un nombre moyen de frères de 5. Plus d'informations sont données dans Richard *et al.* (2015).

4 Validation d'OntoPsychia

4.1 Validation structurelle

4.1.1 Validation de la consistance à l'aide d'HermiT

Le choix du raisonneur HermiT pour valider la consistance d'OntoPsyhia s'est imposé naturellement, étant donné que nous utilisons l'éditeur d'ontologies Protégé 4.2 auquel il est intégré. HermiT nous a ainsi permis de vérifier que notre ontologie ne contenait pas de classes contradictoires et ce au fur et à mesure du développement d'OntoPsychia.

16. http://apps.who.int/classifications/icd10/browse/2015/en

17. Le Manuel diagnostique et statistique des troubles mentaux de la Société américaine de psychiatrie (APA) : http://www.psychiatry.org/practice/dsm

4.1.2 Validation de la structure avec OOPS !

Suite à notre étude de l'état de l'art sur la validation de la structure d'une ontologie, nous avons donc opté pour l'outil OOPS ! pour plusieurs raisons : disponibilité de l'outil et mise à jour régulière [18], critères utilisés qui ont été définis suite à une étude du domaine (voir 2.1), possibilité de conserver son code source privé, gratuité, indépendance du module, utilisation sous n'importe quel navigateur Web. L'analyse de notre ontologie (composée de 1 450 concepts et 218 relations) a permis d'identifier 10 pitfalls, répertoriés dans la Table 1. Nous avons pu en déduire les erreurs suivantes : (1) des problèmes d'import/export sous PROTÉGÉ qui ne rattache pas toujours les concepts à la racine de l'ontologie mère ; (2) OOPS ! n'évalue que le modèle RDF et ne prend donc pas en charge le vocabulaire Simple Knowledge Organization System (SKOS), nous ne pouvons pas évaluer les prefLabel et altLabel – ce point a donc été réalisé à l'aide de requêtes en SPARQL Protocol and RDF Query Language ; (3) presque la moitié de nos propriétés n'étaient pas complètement définies en terme de domaine ou/et co-domaine ; (4), (8) et (9) indiquent des erreurs dans les relations inverses et symétriques, OOPS ! a renvoyé une liste des cas problématiques qui concernaient la gémellité et a également proposé des résolutions ; (5) certains co-domaines et domaines ont été définis par l'intersection de classes au lieu d'être définis par l'union de ces classes ; (6) plusieurs conventions de nommage (typographies) ont été utilisées au sein de l'ontologie (ex : « Enseignement_Secondaire » versus « PostCure ») ; (7) un élément de l'ontologie est utilisé dans sa propre définition. (exemple « tutrice légale » définie par l'axiome « Tutrice_legale and est_tutrice_legale_de some individu », dont le domaine est « Tutrice_Legale »).

4.1.3 Validation des labels à l'aide de requêtes SPARQL et validation du choix du label préférentiel

Ces deux étapes sont essentielles à la validation de notre ontologie, car notre but est d'utiliser ONTOPSYCHIA pour l'annotation de texte libre. Nous devons donc être certain que la terminologie associée à notre ontologie est complète et en adéquation avec le domaine modélisé.

4.1.3.1 Vérification des labels à l'aide de requêtes SPARQL

PROTEGE ne propose pas de module pour la vérification des annotations, en particulier celles concernant les labels préférentiels et alternatifs. Comme nous venons de le voir, OOPS ! propose une vérification des labels, mais uniquement pour le modèle RDF. Afin de vérifier nos labels, nous avons donc utilisé le langage SPARQL qui permet de faire des requêtes sur le modèle SKOS. Nous avons ainsi pu vérifier que pour chaque classe et chaque relation, était associé un label préférentiel unique pour l'anglais et pour le français.

4.1.3.2 Vérification du choix des labels

Le choix des labels préférentiels et alternatifs a été réalisé à l'aide de la méthode développée par Aimé & Charlet (2012). Cette méthode s'appuie sur la biomimétique cognitive. Les auteurs précisent que chaque individu ou groupe d'individus se représente différemment les termes associés à un concept. Leur méthode vise donc à l'évaluation du gradient de prototypicalité

18. `http://oops-ws.oeg-upm.net/`

lexical pour chaque terme de l'ontologie. Le but est de pouvoir déterminer le meilleur label préférentiel associé à un concept, en adéquation avec l'usage en contexte et dans un temps donné.

L'évaluation est réalisée avec une ontologie de domaine et un corpus de textes représentant le domaine de l'ontologie. Les auteurs se basent ensuite sur le calcul de la saillance des termes, dépendant de deux mesures : (1) le calcul du poids selon la position du terme dans la structure d'un document ; (2) le calcul du poids selon la nature du document dans lequel apparaît le terme. Plus un terme est saillant, plus il est considéré proche du concept.

Numéro et nom du Pitfall	Catégorie(s) du pitfall	Nombre de cas	Importance :
(1) P04 : Création d'éléments non-connectés à l'ontologie	Problèmes de modélisation	2	Mineure
(2) P08 : Annotation manquante	Compréhension humaine	1667	Mineure
(3) P11 : Domaine et co-domaine des propriétés non-défini	Compréhension humaine Problèmes de modélisation	110	Importante
(4) P13 : Relation inverse non-définie	Compréhension humaine Problèmes de modélisation	209	Mineure
(5) P19 : Permutation d'intersection et d'union	Compréhension humaine Consistance logique Problèmes de modélisation	3	Critique
(6) P22 : Utilisation de différentes conventions de nommage	Compréhension humaine	ontologie*	Mineure
(7) P24 : Utilisation de définitions récursives	Problèmes de modélisation	12	Importante
(8) P25 : Définition d'une relation inverse à elle-même	Problèmes de modélisation	4	Importante
(9) P26 : Définition d'une relation inverse au lieu d'une relation symétrique	Problèmes de modélisation	4	Importante

TABLE 1 – Résultats de l'analyse de OOPS ! réalisée sur le module « facteurs sociaux et environnementaux des maladies psychiatriques » d'ONTOPSYCHIA.

4.2 Validation de l'adéquation au domaine : l'intervention humaine

4.2.1 Préparation de la rencontre avec les acteurs du domaine et déroulement d'une rencontre

La méthode interactive de validation que nous avons mise en place avec les acteurs du domaine, a pour but d'établir un consensus sur la modélisation. Elle repose sur une communication entre les acteurs et les ontologues ayant participé au développement d'ONTOPSYCHIA. Pour impliquer les acteurs du domaine, nous avons communiqué sur notre projet par le biais de réunions, présentations des outils, présentations des retombés, etc. Suite à cela, nous avons mis notre ontologie à disposition des acteurs du domaine via WEBPROTÉGÉ. Chaque acteur pouvait visualiser l'arborescence conceptuelle de l'ontologie (incluant axiomes et relations). Nous

leur avons ensuite proposé de se retrouver par petits groupes pour leur permettre de discuter ensemble, de manière interactive, de la modélisation et d'apporter leurs critiques et résolutions en cas de désaccord.

Les séances de validation se sont déroulées par groupes de deux (un psychologue clinicien et un psychiatre) et ont duré environ deux heures. Chaque acteur disposait de son propre ordinateur et donc d'un accès à l'ontologie mise en ligne sur WEBPROTÉGÉ. Chaque groupe devait travailler en interaction sur les mêmes concepts, afin de lancer des discussions et débats. Les conversations étaient enregistrées pour permettre à l'ontologue de conserver une trace de la totalité de l'entretien. Les acteurs étaient invités à laisser un commentaire en texte libre sur WEBPROTÉGÉ, sous la forme d'un résumé des points abordés au cours des discussions sur un concept ou une branche de concepts. Cela contribue à l'interaction entre les acteurs. Ceux non présents durant la séance de validation, peuvent avoir accès aux discussions et y répondre ou participer. Une fois ces recommandations posées, l'ontologue n'a pas donné plus d'indications aux acteurs. Il interagissait avec eux uniquement pour expliciter des choix de modélisation jugés ambiguës par les acteurs.

4.2.2 Résultats

Actuellement, deux séances de validation avec chacune un psychologue clinicien et un psychiatre ont été réalisées. Elles ont permis de valider environ 500 concepts, avec une moyenne d'environ 125 concepts validés par heure. Nous avons pu tirer plusieurs constats de ces séances de validation. Dans ces groupes de deux, aucun dominant n'est apparu, chacun intervenait selon son expérience et ses compétences professionnelles. La visualisation totale de la hiérarchie conceptuelle les a aidés à comprendre le sens des concepts et donc leur importance ou non dans l'ontologie. Ce point leur a permis de comprendre les stratégies de modélisation et dans certains cas, de valider des branches entières de concepts, comme ce fut pour les concepts liés à l'éducation scolaire. Cela permet un gain de temps non négligeable. Enfin, la visualisation des axiomes qui définissent les classes leur a permis de constater des manques (par exemple, pour l'ontologue, les concepts « maison » et « foyer » sont des lieux d'habitation, pour les acteurs il était essentiel d'ajouter la définition du type de logement en tant que « collectif » ou « individuel »). Enfin, chaque concept ou branche de concept ne demande pas le même temps de validation. D'après notre modélisation, nous avons pu constater que les concepts se répartissent en deux catégories, selon le degré d'interprétation auquel ils sont soumis. Plus il est important, plus la validation prend du temps. Les temps retranscrit ici, sont issus des enregistrements.

4.2.2.1 Les concepts peu soumis à interprétation

La première catégorie concerne les branches qui répertorient des concepts peu soumis à interprétation. Nous pouvons citer par exemple ceux qui modélisent « l'éducation scolaire », tel « établissement scolaire » ou « formation scolaire ». Nous comptons 124 de ces concepts. Ils ont été validés très rapidement, en survolant la hiérarchie. Nous avons enregistré 43 secondes de conversation et cinq commentaires écrit pour l'ensemble de ces concepts.

4.2.2.2 Les concepts soumis à une interprétation définitoire, contextuelle ou personnelle

La deuxième catégorie concerne les concepts qui selon notre modélisation, sont soumis à une interprétation définitoire, contextuelle ou personnelle. Nous devons par exemple définir avec les acteurs du domaine si un « compagnon » est perçu différemment d'un « mari », et donc utilisé différemment dans leur langage du domaine (interprétation définitoire). Ou encore si le terme « relation intime » indique dans leur contexte, leur référentiel, une « relation affective très proche entre deux personnes » ou une « relation sexuelle » (interprétation contextuelle). Dans le cas d'interprétation définitoire et contextuelle nous avons estimé un temps de validation dans la moyenne. Enfin, le sens de certains concepts peut être perçu très différemment d'un individu à un autre (interprétation personnelle). Par exemple le concept « licenciement » est une « rupture du contrat de travail », mais peut être ressenti de façon négative ou au contraire de façon positive – soulagement – dans le cas d'une personne traversant un burn-out. Ces doubles interprétations peuvent entraîner des modélisations incorrectes si elles ne sont pas discutées avec les acteurs. Ces concepts ont amené des conversations plus denses pour qu'un consensus autour de leur modélisation s'établisse au sein du groupe. Nous avons compté 13 de ses concepts, qui ont entraîné environ 15 minutes de discussions et généré sept commentaires (soit plus d'une minute de validation pour chaque concept).

5 Discussions

5.1 Une structure correcte et adéquate pour point de départ

L'étude de la littérature a mis en avant l'importance de la validation de la structure. Sans structure valide, il ne serait pas possible d'utiliser correctement le modèle. La validation de la structure garantit l'utilisation de l'ontologie au sein d'applications dédiées, ainsi que la réutilisation à d'autres fins. Elle permet de s'assurer que les inférences sont correctes, que le contenu informatif en terme de méta-propriétés peut être compris par n'importe quel ontologue, que le langage ontologique est correctement utilisé, que les conventions sont respectées, etc. Cependant une structure correcte ne garantie pas une sémantique valide et adéquate au domaine. Le critère indiqué par Poveda-Villalón *et al.* (2012) concernant la précision de la modélisation ontologique du domaine, doit donc être vérifié par des humains.

5.2 Proposition de la méthode LOVMI pour la validation d'ontologies

En se fondant sur l'état de l'art, les six dimensions définies par Poveda-Villalón *et al.* (2012) et notre expérience pour la validation d'OntoPsychia, nous proposons la méthode LOVMI pour la validation structurelle et sémantique d'ontologies, en cinq étapes.

1. *Validation de la consistance* à l'aide d'un raisonneur, au fur et à mesure du développement de l'ontologie. Les plus populaires d'entre eux : HermiT, Pellet, FactPlus.
2. *Validation de la structure* avec OOPS ! Comme développé dans ce papier, l'outil OOPS ! permet de réaliser une relecture complète du modèle et d'en extraire les erreurs avec les indications de correction associées.
3. *Validation des labels* à l'aide de requêtes SPARQL. Pour combler les manques des outils de validation existants, les requêtes SPARQL peuvent s'avérer satisfaisantes. Dans notre

cas, elles nous ont permis de vérifier que chaque concept possédait un label unique en SKOS pour chaque langue de l'ontologie.

4. *Validation du choix du label préférentiel*. Cette étape permet de vérifier la validité du choix du label préférentiel, suivant la méthode utilisant les prototypicalités lexicales.
5. *Validation de la sémantique* en collaboration avec les acteurs du domaine modélisé. Cette dernière étape permet de s'assurer que les choix sémantiques de l'ontologue, sont en adéquation avec la sémantique du domaine modélisé.

6 Conclusion

Nous avons présenté dans ce papier les problématiques liées à la validation d'ontologies et les solutions actuellement proposées pour les résoudre. En se fondant sur notre étude du domaine et notre expérience dans la validation du module « facteurs sociaux et environnementaux des maladies psychiatriques » d'ONTOPSYCHIA, nous travaillons à l'élaboration de la méthode LOVMI pour la validation structurelle et sémantique d'ontologies. Cette méthode s'appuie sur des outils déjà existants, et une collaboration entre les ontologues et les acteurs du domaine modélisé.

Références

ABACHA A. B., DA SILVEIRA M. & PRUSKI C. (2013). Une approche pour la validation du contenu d'une ontologie par un système à base de questions/réponses. In *IC-24èmes Journées francophones d'Ingénierie des Connaissances*.

AIMÉ X. & CHARLET J. (2012). Preferred label validation by lexical prototypicality gradient : a use case on a rare diseases ontology. *on Capturing and Refining Knowledge in the Medical Domain (K-MED 2012)*, p. 36.

ASHBURNER M., BALL C., BLAKE J., BOTSTEIN D., BUTLER H., CHERRY J., DAVIS A., DOLINSKI K., DWIGHT S., EPPIG J. *et al.* (2000). Gene ontology : tool for the unification of biology. *Nature genetics*, **25**(1), 25–29.

AUBIN S. & HAMON T. (2006). Improving term extraction with terminological resources. In *Advances in Natural Language Processing*, p. 380–387. Springer.

AUSSENAC-GILLES N., DESPRES S. & SZULMAN S. (2008). The terminae method and platform for ontology engineering from texts. *Bridging the Gap between Text and Knowledge-Selected Contributions to Ontology Learning and Population from Text*, p. 199–223.

BACHIMONT B., ISAAC A. & TRONCY R. (2002). Semantic commitment for designing ontologies : A proposal. In *Knowledge Engineering and Knowledge Management : Ontologies and the Semantic Web*, volume 2473 of *Lecture Notes in Computer Science*, p. 114–121. Springer Berlin Heidelberg.

CHARLET J., BACHIMONT B. & JAULENT M.-C. (2006). Building medical ontologies by terminology extraction from texts : an experiment for the intensive care units. *Computers in biology and medicine*, **36**(7), 857–870.

CHARLET J., DECLERCK G., DHOMBRES F., GAYET P., MIROUX P. & VANDENBUSSCHE P.-Y. (2012). Construire une ontologie médicale pour la recherche d'information : problématiques terminologiques et de modélisation. In *IC-23èmes Journées francophones d'Ingénierie des Connaissances*, p. 33–48.

FARQUHAR A., FIKES R. & RICE J. (1997). The ontolingua server : A tool for collaborative ontology construction. *International journal of human-computer studies*, **46**(6), 707–727.

FERNÁNDEZ-LÓPEZ M. & GÓMEZ-PÉREZ A. (2002). The integration of ontoclean in webode. In *CEUR Workshop Proceedings*.

GHIDINI C., KUMP B., LINDSTAEDT S., MAHBUB N., PAMMER V., ROSPOCHER M. & SERAFINI L. (2009). Moki : The enterprise modelling wiki. In *The Semantic Web : Research and Applications*, p. 831–835. Springer.

GROUIN C. & NÉVÉOL A. (2014). De-identification of clinical notes in french : towards a protocol for reference corpus development. *Journal of biomedical informatics*, **50**, 151–161.

GRUBER T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, **43**(5), 907–928.

GUARINO N. & WELTY C. (2000). A formal ontology of properties. In *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, p. 97–112. Springer.

HASTINGS J., SMITH B., CEUSTERS W., JENSEN M. & MULLIGAN K. (2012). Representing mental functioning : Ontologies for mental health and disease. In *ICBO 2012 : 3rd International Conference on Biomedical Ontology*.

KOLA J. S., HARRIS J., LAWRIE S., RECTOR A., GOBLE C. & MARTONE M. (2010). Towards an ontology for psychosis. *Cognitive Systems Research*, **11**(1), 42–52.

POVEDA-VILLALÓN M., SUÁREZ-FIGUEROA M. C. & GÓMEZ-PÉREZ A. (2012). Validating ontologies with OOPS ! In *Knowledge Engineering and Knowledge Management*, p. 267–281. Springer.

RESSAD-BOUIDGHAGHEN O., SZULMAN S., ZARGAYOUNA H. & PAUL E. (2013). Construction collaborative d'une ressource termino-ontologique (rto) pour le droit des collectivités territoriales. In *IC-24èmes Journées francophones d'Ingénierie des Connaissances*, IC 2013, Lille, France.

RICHARD M., AIMÉ X., KREBS M.-O. & CHARLET I. (2015). Enrich classifications in psychiatry with textual data : an ontology for psychiatry including social concepts. In *Studies in health technology and informatics (**in press**)*.

ROSSE C., JR J. L. M. *et al.* (2003). A reference ontology for biomedical informatics : the foundational model of anatomy. *Journal of biomedical informatics*, **36**(6), 478–500.

SCHOBER D., TUDOSE I., SVATEK V. & BOEKER M. (2012). Ontocheck : verifying ontology naming conventions and metadata completeness in protege 4. *Journal of Biomedical Semantics*, **3**(Suppl 2), S4.

SHEARER R., MOTIK B. & HORROCKS I. (2008). Hermit : A highly-efficient owl reasoner. In *OWLED*, volume 432.

SILVA C., MARREIROS G. & SILVA N. (2014). Development of an ontology for supporting diagnosis in psychiatry. In *Distributed Computing and Artificial Intelligence, 11th International Conference*, p. 343–350 : Springer.

TUDORACHE T., NYULAS C., NOY N. F. & MUSEN M. A. (2013). Webprotégé : A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic web*, **4**(1), 89–99.

VELARDI P., FARALLI S. & NAVIGLI R. (2013). Ontolearn reloaded : A graph-based algorithm for taxonomy induction. *Computational Linguistics*, **39**(3), 665–707.

VÖLKER J., VRANDEČIĆ D., SURE Y. & HOTHO A. (2008). Aeon–an approach to the automatic evaluation of ontologies. *Applied Ontology*, **3**(1), 41–62.

VRANDEČIĆ D. (2009). *Ontology evaluation*. Springer.

Décomposition sous-directe d'un treillis en facteurs irréductibles

Jean-François Viaud[1], Karell Bertet[1], Christophe Demko[1], Rokia Missaoui[2]

[1] Laboratoire L3i, Universite de La Rochelle, France
{jviaud, kbertet, cdemko}@univ-lr.fr

[2] Universite du Québec en Outaouais, Canada
rokia.missaoui@uqo.ca

Abstract : La taille d'un treillis de concepts peut augmenter de façon exponentielle avec la taille du contexte. Lorsque le nombre de noeuds devient important, l'étude et la génération d'un tel treillis devient impossible. Décomposer le treillis en petit sous-treillis est un moyen de contourner ce problème. Dans la décomposition sous-directe, les petits sous-treillis générés sont des quotients qui ont une interprétation intéressante dans le cadre de l'Analyse de Concepts Formels (FCA).

Dans ce papier, nous donnons les étapes pour obtenir une décomposition sous-directes en treillis irréductibles, en partant d'un contexte fini et réduit. Cette décomposition est obtenue en utilisant trois points de vue : les treillis quotients, les relations flèches et les sous-contextes compatibles. Cette approche est essentiellement algébrique car elle repose sur la théorie des treillis, sauf pour le dernier point. Nous donnons un algorithme polynomial permettant de générer cette décomposition à partir d'un contexte initial. Cette méthode peut être étendue pour permettre l'exploration interactive ou la fouille dans de grands contextes.

Mots-clés : treillis de concepts, relation de congruence, treillis quotient, relation flèche, sous-contexte flèche fermé, sous-contexte compatible

1 Introduction

Au cours de la dernière décennie, l'augmentation de capacités de calcul ont permis le développement de l'Analyse des Concepts Formels (FCA) à travers de nouvelles méthodes basées sur les treillis de concepts. Bien qu'ils soient de taille exponentielle en espace et temps dans le pire des cas, dans la pratique, les treillis de concepts sont d'une taille polynomiale et permettent une représentation intuitive des données stockées dnas un contexte qui relie les objets aux attributs à travers une relation binaire. Les méthodes basées sur les treillis de concepts ont été développées dans différents domaines tels que la découverte et la représentation des connaissances, les bases de données ou la recherche d'information où certains concepts pertinents, par exemple les correspondances possibles entre les objets et les attributs sont considérés soit comme des classifieurs, soit comme des hiérarchies.

Avec l'accroissement de la taille des données, un ensemble de méthodes ont été développées pour permettre de générer seulement quelques concepts (plutôt que tous) et leurs voisinages de manière interactive (Ferré, 2014; Visani *et al.*, 2011) et en ligne, ou pour permettre de meilleures visualisations à l'aide des "nested line diagrams" (Ganter & Wille, 1999). Ces approches deviennent inefficaces lorsque les contextes sont de grande taille. Cependant, l'idée principale d'une décompostion du treillis ou du contexte en plus petits éléments reste valable à condition de les propriétés de classifications du treillis initial soient conservées. Beaucoup de décompostions de treillis ont été défines et étudiées, à la fois du point de vue algébrique (Demel, 1982; Mihók & Semanišin, 2008) et du point de vue de la FCA (Ganter & Wille, 1999; Funk *et al.*, 1995).

Parmi ces décompositions, nous avons en particulier : le théorème de factorisation (Mihók & Semanišin, 2008), la décomposition atlas (Ganter & Wille, 1999), la décomposition subtensorielle (Ganter & Wille, 1999), les méthodes de doublement de convexe (Day, 1994; Nation, 1995; Geyer, 1994; Bertet & Caspard, 2002) ou bien encore la décomposition sous-directe. Cette dernière a été largement étudiée, il y a quelques années, dans le cadre de l'algèbre universelle (Demel, 1982; Freese, 2008), mais aussi dans le cadre de la FCA (Wille, 1983, 1987) et (Funk *et al.*, 1995). A notre connaissance, il n'y a pas de nouveaux développements ou de nouveaux algorithmes pour la décomposition sous-directe de contextes.

Dans ce papier, nous étudions la décomposition sous-directe d'un treillis de concepts, considérée comme une première étape à l'exploration interactive et à la fouille dans de grands contextes. La décomposition sous-directe d'un treillis L en treillis quotients $(L_i)_{i\in\{1,\ldots,n\}}$, peut être notée $L \hookrightarrow L_1 \times \cdots \times L_n$, et est définie par deux propriétés (résultats importants dans (Ganter & Wille, 1999)) : (i) L est un sous-treillis du treillis produit-direct $L_1 \times \cdots \times L_n$ et (ii) chaque projection de L sur un facteur est surjective (Ganter & Wille, 1999). Dans un premier temps, il est établi que chaque facteur est le treillis de concepts d'un sous-contexte flèche-fermé, c'est à dire fermé en regard de relations flèches entre objets et attributs. Cela signifie que cette décomposition peut être obtenue en calculant certains sous-contextes particuliers. Dans un second temps, l'équivalence entre les sous-contextes flèche-fermés et les relations de congruence est établie; une relation de congruence étant une relation d'équivalence compatible avec les loi sup et inf du treillis. Cela signifie que les concepts de L peuvent être retrouvés à partir des treillis quotients et que la propriété de classification du treillis initial est maintenue puisque les relations d'équivalence forment une partition de l'ensemble des concepts. Enfin, est formulée l'équivalence entre les sous-contextes flèche-fermés et les sous-contextes compatibles, c'est à dire les sous-contextes dont les concepts correspondent aux concepts du treillis initial. Ce resultat nous permet de calculer le morphisme de L dans le produit direct et ainsi de retrouver les concepts de L dans les treillis quotients. Dans ce papier, nous déduisons de ces résultats un lien très fort, entre les notions suivantes, qui, à notre connaissance, n'a pas été utilisé :

- Les treillis intervenant dans la décomposition sous-directe
- Les relations de congruences
- Les sous-contextes flèche-fermés et
- Les sous-contextes compatibles.

Comme suggéré dans (Ganter & Wille, 1999), les contextes définissant les treillis d'une décomposition sous-directe particulière, i.e. les contextes irréductibles, peuvent être obtenus à l'aide d'un traitement polynomial sur les lignes/objets (ou colonnes/attributs) du contexte initial. Ainsi, la décomposition sous-directe d'un treillis peut être étendue à une décomposition sous-directe de son contexte réduit en sous-contextes irreductibles.

Dans ce papier, nous proposons une décomposition sous-directe polynomiale d'un contexte en sous-contextes en étendant la décomposition sous-directe d'un treillis. Cette décomposition conduit à une écomonie de stockage des données pour les grands contextes. En effet, la génération de l'ensemble complet des treillis quotients peut être évitée en fournissant de manière interactive quelques (mais pas tous) concepts et leurs voisinages dans un grand contexte. De plus, il est possible de proposer à l'utilisateur de se concentrer sur un treillis quotient particulier et de générer ce treillis intégralement ou partiellement, ainsi que ses bases d'implications.

Il y a au moins deux raisons d'étudier ce cas de gestion des connaissances. La première tient au fait que l'utilisateur peut être submergé par la connaissance extraite des données, même dans le cas où la taille des données en entrée est faible. La seconde raison vient des progrès de la communauté FCA dans le construction et l'exploration des treillis de concepts. Les solutions existantes peuvent être désormais adaptées et enrichies pour cibler uniquement la connaissance utile.

Ce papier est organisé de la manière suivante. La section 2 introduit la décomposition sous-directe et les trois différents points de vue : les treillis quotients, les relations flèches et les sous-contextes compatibles. La section 3 présente la construction complète de la décomposition sous-directe et les algorithmes. La section 4 donne les conclusions et perspectives.

2 Cadre structurel

Tout au long de ce papier, tous les ensembles (et en particulier les treillis) seront supposés finis.

2.1 Treillis et Analyse des Concepts Formels

2.1.1 Treillis algébriques

Commençons par rappeler qu'un *treillis* $(L, \leq)$ est un ensemble ordonné dans lequel toute paire (x, y) d'éléments possède une borne supérieure, notée $x \vee y$, et une borne inférieure, notée $x \wedge y$. Comme nous ne considérons que des structures finies, toute partie $A \subset L$ possède une borne supérieure et une borne inférieure (i.e. les treillis finis sont complets).

Un élément $j \in L$ est dit *sup-irréductible* s'il n'est pas borne supérieure d'un ensemble qui ne le contient pas. L'ensemble des sup-irréductibles est noté J_L. Les *inf-irreductibles* sont définis duallement et leur ensemble est M_L. En conséquence directe de la définition, un élément $j \in L$ est sup-irréductible si et seulement s'il possède un unique prédécesseur; ce dernier est alors noté j^-. Duallement, un élément $m \in L$ est inf-irréductible si et seulement s'il admet un unique successeur qui est alors noté m^+.

Sur la figure 1, les sup-irréducible sont notés avec des nombres et les inf-irréductibles avec des lettres.

2.1.2 Treillis de Concepts ou de Galois

Un *contexte* (formel) (O, A, R) est défini par la donnée d'un ensemble O d'objets, d'un ensemble A d'attributs, et d'une relation binaire $R \subset O \times A$, entre O et A. On déduit deux opérateurs de la donnée d'un tel contexte :

- pour toute partie $J \subset O$, on pose $J' = \{a \in A, j \; R \; a \; \forall j \in J\}$ et duallement,
- pour toute partie $M \subset A$, on pose $M' = \{o \in O, o \; R \; m \; \forall m \in M\}$.

Un *concept* (formal) correspond à un rectangle maximal de la relation R et est défini par un paire (X, Y) telle que $X' = Y$ et $Y' = X$. Les ensembles X et Y sont respectivement appelés *extension* et *intention* du concept. L'ensemble des concepts issus d'un context est ordonné par la relation : $(J_1, M_1) \leq (J_2, M_2) \iff J_1 \subset J_2 \iff M_2 \subset M_1$ Cet ensemble de concepts

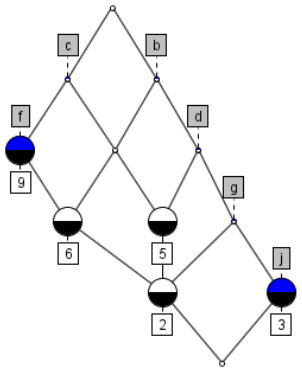


Figure 1: Un treillis avec ses noeuds irréductibles. L'élément le plus petit est en bas, le plus grand est en haut.

formels muni de cette relation d'ordre constitue un treillis complet, appelé *treillis de concepts* ou *treillis de Galois* du context (O, A, R).

Des contextes formels différents peuvent engendrer des treillis isomorphes. Parmi tous les contextes ayant même treillis de concepts, il en existe un unique et minimal (en nombre d'observations et attributs), appelé *contexte réduit*.

2.1.3 Bijection fondamentale

Un résultat fondamental (Barbut & Monjardet, 1970) affirme que tout treillis $(L, \leq)$ est isomorphe au treillis de concepts du contexte $(J_L, M_L, \leq)$, où J_L est l'ensemble des sup-irréductibles de L et M_L celui de ses inf-irréductibles. De plus, ce contexte est le contexte réduit. On en déduit immédiatement une bijection entre l'ensemble des treillis et l'ensemble des contextes réduits où les objets correspondent aux concepts sup-irréductibles et les attributs aux concepts inf-irréductibles. La figure 2 montre le contexte reduit du treillis de la figure 1.

2.2 Sous-contextes flèche-fermés et compatibles

Dans cette section est présentée l'équivalence entre les sous-contextes compatibles et les sous-contextes flèche-fermés.

2.2.1 Sous-contextes Compatibles

Un *sous-contexte* d'un contexte formel (O, A, R) est un triplet $(J, M, R \cap J \times M)$ tel que $J \subset O$ et $M \subset A$. Un sous-contexte $(J, M, R \cap J \times M)$ de (O, A, R) est *compatible* lorsque pour tout concept (H, N) de (O, A, R), le couple $(J \cap H, M \cap N)$ est un concept de $(J, M, R \cap J \times M)$.

2.2.2 Relations flèches

Les sous-contextes flèche-fermés intervenant dans l'équivalence sont basés sur les relations flèches entre les concepts sup et inf-irréductibles du treillis. Soit $(J_L, M_L, \leq)$ le contexte réduit

	b	c	d	f	g	j
2	×	×	×	×	×	↕
3	×	↕	×	↑	×	×
5	×	×	×	↕	↕	○
6	×	×	↕	×	↓	○
9	↕	×	○	×	○	○

Figure 2: Le contexte réduit du treillis de la figure 1 avec les relations flèches.

du treillis $(L, \leq)$. Les relations flèches (Crawley & Dilworth, 1973; Grätzer, 1978) forment une partition de la relation $\not\leq$ en considérant le prédécesseur immédiat j^- d'un sup-irréductible j et le successeur immédiat m^+ d'un inf-irréductible m :

- $j \updownarrow m$ si $j \not\leq m$, $j \leq m^+$ et $j^- \leq m$.
- $j \uparrow m$ si $j \not\leq m$, $j \leq m^+$ et $j^- \not\leq m$.
- $j \downarrow m$ si $j \not\leq m$, $j \not\leq m^+$ et $j^- \leq m$.
- $j \circ m$ si $j \not\leq m$, $j \not\leq m^+$ et $j^- \not\leq m$.

Dans la figure 2, le contexte réduit est complété avec les quatre relations $\updownarrow$, $\uparrow$, $\downarrow$ et $\circ$ dans les cases vides qui correspondent toutes aux cas où $j \not\leq m$.

Par exemple avec $j = 5$ et $m = f$ deux noeuds respectivement sup-irréductibles et inf-irréductibles (voir figure 2), on a $j^- = 2$, $m^+ = c$ et $5 \updownarrow f$ puisque $5 \not\leq f$, $5 \leq c$ et $2 \leq f$.

2.2.3 Sous-contexte flèche-fermé

Un sous contexte $(J, M, R \cap J \times M)$ d'un contexte (O, A, R) est un *sous-contexte flèche-fermé* s'il vérifie les conditions suivantes :

$$\text{Si } j \uparrow m, m \in A \text{ et } j \in J \text{ alors } m \in M. \text{ Si} j \downarrow m, j \in O \text{ et } m \in M \text{ alors } j \in J$$

Le théorème 1 ci-dessous et l'algorithme 2 justifient l'introduction de la notion de sous-contexte flèche-fermé :

- Le théorème donne la sémantique : les sous-contextes flèche-fermés sont ceux qui sont compatibles avec la notion de concept.
- L'algorithme permet de les calculer. Autrement dit, la notion importante est celle de contexte compatible qui se calcule via celle de contexte flèche-fermé.

2.2.4 Théorème d'équivalence

Introduisons la première équivalence de ce papier dont nous avons besoin et dont la preuve se trouve dans (Ganter & Wille, 1999) :

Théorème 1
Soit $(J, M, J \times M \cap R)$ un sous-contexte de (O, A, R). Sont équivalentes :

- *Le sous-contexte* $(J, M, J \times M \cap R)$ *est compatible.*
- *Le sous-contexte* $(J, M, J \times M \cap R)$ *est flèche-fermé.*

2.3 Relations de congruence et treillis quotients

Dans cette section, nous introduisons l'équivalence entre les relations de congruence et les sous-contextes flèche-fermés.

2.3.1 Quotient

Une *relation d'équivalence* est une relation binaire R sur un ensemble E qui est réflexive, symétrique et transitive. La classe d'équivalence de $x \in E$ est $x_R = \{y \in E,\ xRy\}$.

L'ensemble des classes d'équivalences, appelé *ensemble quotient*, noté $E/R = \{x_R,\ x \in E\}$

2.3.2 Treillis quotient

Une relation de congruence Θ sur un treillis L est une relation d'équivalence telle que :

$$x_1 \Theta y_1 \text{ et } x_2 \Theta y_2 \Longrightarrow x_1 \wedge x_2 \Theta y_1 \wedge y_2 \text{ et } x_1 \vee x_2 \Theta y_1 \vee y_2$$

Les classes d'équivalence sont alors notées $x_\Theta, y_\Theta, \ldots$. La relation suivante, définie sur le quotient L/Θ, est une relation d'ordre :

$$x_\Theta \leq y_\Theta \Longleftrightarrow x \Theta (x \wedge y) \Longleftrightarrow (x \vee y) \Theta y$$

Muni de cette relation, L/Θ est un treillis, appelé treillis quotient. Un théorème tout à fait standard de l'algèbre, dont la preuve est omise, affirme que :

Théorème 2
La projection $L \to L/\Theta$ *est un morphisme de treillis surjectif.*

Dans la décomposition sous-directe qui est le point d'intérêt de ce papier, les treillis qui interviennent sont des treillis quotients. Ce théorème, par la propriété de surjectivité, assure que ces treillis quotient ne contiennent pas d'information supperflue, juste de l'information aggrégée.

2.3.3 Le second théorème d'équivalence

Nous avons désormais tous les éléments permettant de formuler la seconde équivalence dont la preuve peut-être trouvée dans (Ganter & Wille, 1999) :

Théorème 3
Etant donné un treillis L*, l'ensemble des relations de congruence dans* L *est bijection avec l'ensemble des sous-contextes flèche-fermés du contexte réduit de* L*.*

Comme expliqué précédemment, les contextes flèches fermés servent à calculer et les relations de congruences vont donner les treillis quotients qui interviennent dans la décomposition sous-directe.

2.4 Décompositions sous-directe

Dans cette section, nous introduisons l'équivalence entre les décompositions sous-directes et certaines familles de sous-contextes flèche-fermés.

2.4.1 Produit sous-directe

Définition 1
Un treillis sous-produit direct est un sous-treillis d'un treillis produit-direct $L_1 \times \cdots \times L_n$ tel que les projections sur chacun des facteurs soient surjectives. Les treillis $L_i, i \in \{1, \ldots, n\}$ sont des treillis facteurs. Une décomposition sous-directe d'un treillis L est la donnée d'un isomorphisme entre L et un treillis sous-produit direct, que l'on peut noter :

$$L \hookrightarrow L_1 \times \cdots \times L_n \twoheadrightarrow L_i$$

où la première flèche est le morphisme d'injection de L dans le produit directe des L_i et la seconde flèche est une projection surjective sur un des treillis facteurs.

Dans le meilleur des cas, le treillis L de départ est exactement isomorphe à un produit $L_1 \times \cdots \times L_n$. Dans ce cas, on obtient une décomposition directe et le treillis produit est exactement de la même taille que le treillis initial. Dans la pratique, partant d'un treillis quelconque, on n'obtient pas mieux qu'une décomposition sous-directe, c'est à dire que L est un sous-treillis de $L_1 \times \cdots \times L_n$ et le treillis produit est beaucoup plus gros. Cependant, la décomposition reste intéressante dès lors que l'on se concentre uniquement sur chacun des facteurs L_i qui sont eux petits.

D'autre part, cette propriété de recouvrement assure que toute l'information contenue dans le contexte initial se retrouve dans la décomposition finale; il n'y a pas de perte.

2.4.2 Troisième équivalence

La troisième et la plus importante des équivalences, dont la preuve se trouve dans (Ganter & Wille, 1999), fait un lien avec les familles couvrantes de sous-contextes flèche-fermés.

Proposition 1
Etant donné un contexte réduit (J, M, R), alors les décompositions sous-directes de son treillis de concepts L sont en bijections avec les familles de sous-contextes flèche-fermés $(J_j, M_j, J_j \times M_j \cap R)$ tels que $J = \cup J_j$ et $M = \cup M_j$.

La figure 3 donnent une telle décomposition du contexte réduit de la figure 2 du treillis de la figure 1. Dans ce cas particulier, on obtient une partition du contexte, mais d'une manière générale, il s'agit juste d'un recouvrement.

3 Notre contribution

Les résultats théoriques généraux sur la décomposition sous-directe sont connus depuis plusieurs années. Les nouveautés présentées dans ce papier résident essentiellement dans deux points :

- Un travail de synthèse : cette décomposition a été étudiée par de nombreux auteurs et repose sur beaucoup de notions théoriques; sans prétendre à l'exhaustivité, ce papier regroupe les points qui nous ont paru fondamentaux.
- Un nouveau point de vue sur cette décomposition ancienne est apporté; elle permet un accès à la fois local et structuré aux données, ce qui est important à une époque où les problèmes de volumétrie sont croissants.

3.1 Résultat principal

A partir des équivalences précédents présentes dans (Ganter & Wille, 1999), on peut déduire :

Corollaire 1
Etant donné un treillis L et son contexte réduit (J, M, R), sont en bijection :

1. *Les familles de sous-contextes flèche-fermés de (J, M, R) couvrant J et M,*
2. *Les familles de sous-contextes compatibles de (J, M, R) couvrant J et M,*
3. *Les familles $(\theta_i)_{i\in I}$ de relations de congruences de L telles que $\cap_{i\in I}\theta_i = \Delta$.* [1]
4. *L'ensemble des décompositions sous-directes de L et leurs treillis facteurs.*

Dans la suite, nous donnons la construction d'une décomposition sous-directe particulière et montrons un usage possible des treillis quotients.

Nous disposons donc au départ d'un contexte réduit pour lequel il n'est pas nécessaire de calculer intégralement son treillis de concepts. Le contexte est supposé grand de sorte que pour explorer ou construire les concepts, on souhaite accéder seulement à une partie des données, autrement dit des vues. La décomposition sous-directe permet de construire ces vues sous forme de sous-contextes compatibles. Ces derniers, par leur propriété de compatibilité, permettent de retrouver facilement les concepts du contexte initial. Ainsi, cette décomposition permet la répartition en sous-contextes des données initiales de manière à simplifier la navigation dans les concepts.

3.2 Calcul des facteurs irréductibles

Dans cette section, nous considérons les décompositions sous-directes d'un treillis L avec en entrée son contexte réduit (O, A, R). Avec le corrollaire 1, une décomposition sous-directe d'un treillis L peut être obtenue en calculant un ensemble de sous-contextes flèche-fermés de (O, A, R) qui couvrent O et A. Il y a évidemment plusieurs tels ensembles et donc plusieurs décompositions sous-directes. En particulier, la décompostion d'un treillis L en L lui-même, qui correspond à prendre un seul sous-contexte flèche-fermé : le contexte (O, A, R) en entier. Un algorithme de décomposition sous-directe a déjà été proposé (Funk *et al.*, 1995). Cependant, toutes les relations de congruence sont calculées et ensuite seulement des paires couvrantes de relations sont considérées. En conséquence, plusieurs décompositions peuvent être obtenues, et ces décompositions possèdent nécessairement seulement deux facteurs.

[1] $x\Delta y \iff x = y$

Dans cet article, nous nous concentrons sur une décomposition sous-directe en un nombre éventuellement grand de petits facteurs. Ces derniers étant irréductibles au sens suivant. Un treillis L est *irréductible* lorsqu'il apparait comme facteur de toute ses décompositions sous-directes, autrement dit, il n'admet pas de décomposition non-triviale. Une caractérisation des décompositions sous-directes en facteurs irréducitbles se trouve dans (Ganter & Wille, 1999) :

Proposition 2
Un treillis L est irréductible si et seulement si son contexte réduit est monogène.

Un contexte (O, A, R) est dit *monogène* lorsqu'il est obtenu par fermeture d'un contexte contenant un seul $j \in A$. Ainsi (O, A, R) est le plus petit contexte flèche-fermé contenant j. Nous pouvons donc déduire le résultat suivant :

Proposition 3
Soit L un treillis. On peut déduire de L un treillis produit direct $L_1 \times \cdots \times L_n$ tel que chaque treillis L_i est : le treillis de concept d'un sous-contexte monogène, irréductible et un treillis facteur d'un décomposition sous-directe.

Ce résultat peut être mis en oeuvre avec l'algorithme 1, polynomial en temps, qui permet de trouver les contextes des facteurs $L_1, \ldots, L_n$ d'une décomposition sous-directe, avec un contexte (O, A, R) en entrée. Les sous-contextes monogènes sont obtenus par fermeture de chaque $j \in J$, via l'algorithme 2; c'est à dire que les sous-contextes monogènes sont les plus petits sous-contextes flèche-fermé contenant un $j \in J$ donné. La décomposition sous-directe de L est alors obtenue en formant les treillis de concepts de ces sous-contextes.

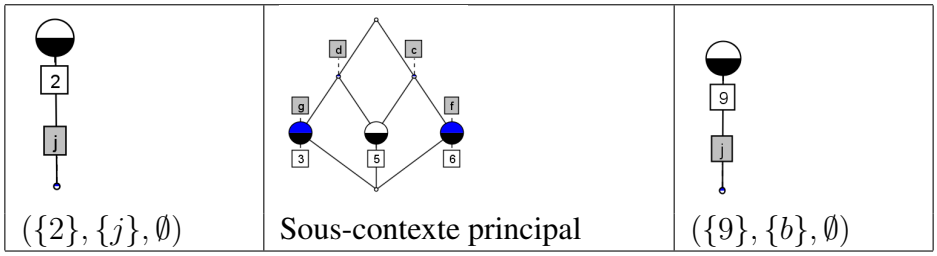
On peut remarquer que les fermetures sont calculées sur les sup-irréductibles, mais cela aurait pu être fait sur les inf-irréductibles.

Algorithme 1 : Décomposition_SousDirecte

Entrées : Un contexte (O, A, R)
Sorties : Liste $\mathcal{L}$ des sous-contextes (J_j, M_j, R_j) des facteurs irréductibles.

```
1 L ← ∅;
2 pour tous les j ∈ J faire
3     Calculer (J_j, M_j, R_j) = Flèche_Fermeture((j, ∅, ∅), (O, A, R)), sous-contexte
      monogène engendré par j;
4     si L ne contient pas de sous-context couvrant [a] (J_j, M_j, R_j) alors
5         ajouter (J_j, M_j, R_j) à L
6     si L contient un sous-contexte (J, M, R) couvert par (J_j, M_j, R_j) alors
7         effacer (J, M, R) de L
8     retourner L;
```

[a] Etant donnés deux sous-contextes $(J_1, M_1, J_1 \times M_1 \cap R)$ et $(J_2, M_2, J_2 \times M_2 \cap R)$ de (O, A, R), on dit que $(J_1, M_1, J_1 \times M_1 \cap R)$ *couvre* $(J_2, M_2, J_2 \times M_2 \cap R)$ lorsque $J_2 \subset J_1$ et $M_2 \subset M_1$.



$(\{2\},\{j\},\emptyset)$ Sous-contexte principal $(\{9\},\{b\},\emptyset)$

Figure 3: Les trois treillis facteurs de la décomposition, avec leur sous-contexte en légende

Algorithme 2 : Flèche_Fermeture

Entrées : Un sous-contexte $(\tilde{J},\tilde{M},\tilde{R})$ d'un contexte (J,M,R)
Sorties : La flèche-fermeture de $(\tilde{J},\tilde{M},\tilde{R})$

1 $J_c = \tilde{J}$; $M_c = \tilde{M}$;
2 $pred_J = 0$; $pred_M = 0$;
3 **tant que** $pred_M < card(M_c)$ *ou* $pred_J < card(J_c)$ **faire**
4 $\quad pred_J = \text{card}(J_c)$;
5 $\quad pred_M = \text{card}(M_c)$;
6 $\quad$ **pour tous les** $j \in J_c$ **faire**
7 $\quad\quad$ ajouter à M_c tous les $m \in M$ tels que $j \uparrow m$;
8 $\quad$ **forall the** $m \in M_c$ **do**
9 $\quad\quad$ ajouter à J_c tous les $j \in J$ tels que $j \downarrow m$;
10 $\quad$ Retourner $(J_c, M_c, R \cap J_c \times M_c)$

Considérons le contexte réduit de la Figure 2. Dans l'algorithme 1, pour chaque valeur de j, le sous-contextes monogènes contenant j est calculé puis éventuellement ajouté à $\mathcal{L}$ A la fin du processus, nous obtenons les trois treillis facteurs de la figure 3.

3.3 Morphisme injectif et FCA

La décomposition sous-directe d'un treillis L ayant pour facteurs $L_1,\ldots,L_n$ est pertinente car il existe un morphisme injectif de L dans le produit direct $L_1 \times \cdots \times L_n$. Cela signifie que toute l'information du treillis de départ se retrouve à l'identique dans le grand treillis produit. Ce morphisme est précisé par la bijection entre les sous-contextes compatibles et les relations de congruences donnée dans le corollaire 1 :

Proposition 4
Soit $(J,M,R\cap J\times M)$ un sous-contexte compatible d'un contexte (O,A,R), alors la relation $\Theta_{J,M}$ définie par : $(A_1,B_1)\Theta_{J,M}(A_2,B_2) \iff A_1\cap J = A_2\cap J \iff B_1\cap M = B_2\cap M$ est une relation de congruence, et son treillis quotient est isomorphe au treillis de concepts du sous-contexte $(J,M,R\cap J\times M)$.

Pour calculer ce morphisme, il suffit de parcourir les noeuds du treillis initial, les transformer en noeuds du treillis produit et de les marquer (par exemple avec un booléen). On obtient ainsi

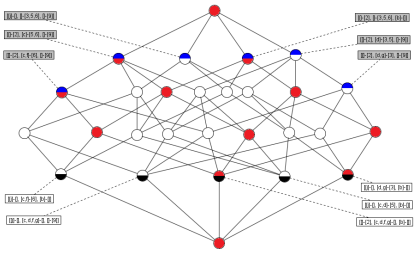


Figure 4: Le treillis produit dont les noeuds du treillis initial sont marqués en rouge

le morphisme injectif $L \hookrightarrow L_1 \times \cdots \times L_n$. Sur l'exemple traité, on récupère le marquage du gros treillis produit, illustré par la figure 4. Evidemment, cet algorithme n'a pas vocation à être utilisé pour des applications réelles utilisant de grands contextes puisque le treillis produit est beaucoup plus gros que le treillis original et l'un des objectifs principaux de la décomposition est de travailler avec des treillis plus petits. Cet algorithme est uniquement utile à des fins de tests ou d'illustrations.

Cependant cet algorithme peut être étendu pour un usage élémentaire en FCA. On ne calcule pas le gros treillis produit mais seulement les petits treillis facteurs $L_1, \ldots, L_n$, voire même seulement les sous-contextes irréductibles $C_1, \ldots, C_n$, qui sont interprétés comme des vues sur les données. La décomposition est alors utilisée de la manière suivante : au lieu de s'intéresser directement à des concepts de L, on s'intéresse à ces même concepts restreints aux vues $C_1, \ldots, C_n$. Les résultats théoriques s'interprètent alors comme suit : le morphisme injectif de L dans le produit $L_1 \times \cdots \times L_n$ nous indique qu'aucune information n'est perdue dans la décomposition et les morphismes surjectifs de projections nous indiquent qu'aucune information superflue n'est ajoutée.

4 Conclusion et perspectives

Dans ce papier, nous avons présenté un algorithme polynomial de décomposition d'un contexte réduit en sous-contextes tels que leurs treillis de concepts soient irréductibles. Cette décomposition est la conséquence directe des liens très forts, établis dans (Ganter & Wille, 1999), qui existent entre les facteurs de la décomposition, les relations de congruence, les sous-contextes flèche-fermés, et les sous-contextes compatibles.

Pour approfondir cette décomposition sous-directe, il serait intéressant de mener des expériences à plus grande échelle sur des données réelles. afin de mieux comprendre la sémantique cachée derrière les sous-contextes irréductibles. En particulier, les attributs qui interviennent dans plusieurs facteurs, autrement dit dans plusieurs vues sur les données, doivent avoir une sémantique forte qu'il est certainement important de comprendre. Il serait également utile de proposer à l'utilisateur de choisir de manière intéractive quelques facteurs de la décomposition et de mixer cette approche avec celle de (Funk *et al.*, 1995).

D'un point de vue théorique, nous pensons qu'il y a des liens forts entre les bases d'implications des treillis quotients et du treillis initial. A notre connaissance, cette problématique n'a jamais été étudiée et pourrait avoir des conséquences significatives d'un point de vue algorith-

mique. Ce problème a cependant été abordé par (Valtchev & Duquenne, 2008) dans le cadre d'une décomposition verticale d'un contexte en sous-contextes.

Comme l'étude empirique de (Snelting, 2005) montre que de nombreux contextes constitués de données réelles sont irréductibles, nous espérons pouvoir (i) identifier les cas de contextes nécessairement irréductibles, (ii) étudier, comparer, combiner à d'autres décompositions, par exemple en utilisant la congruence de Fratini (Duquenne, 2010), ou l'opération de doublement de convexes (Day, 1994; Nation, 1995; Geyer, 1994; Bertet & Caspard, 2002). Enfin, la construction du treillis à partir de ses quotients pourrait être obtenue en utilisant les principes d'optimisation utilisés dans l'opération relationnelle de jointure.

References

M. Barbut & B. Monjardet, Eds. (1970). *L'ordre et la classification*. Algèbre et combinatoire, tome II. Hachette.

Bertet K. & Caspard N. (2002). *Doubling convec sets in lattices: characterizations and recognition algorithms*. Rapport interne TR-LACL-2002-08, LACL (Laboratory of Algorithms, Complexity and Logic), University of Paris-Est (Paris 12).

Crawley P. & Dilworth R. (1973). *Algebraic theory of lattices*. Englewood Cliffs: Prentice Hall.

Day A. (1994). Congruence normality: The characterization of the doubling class of convex sets. *algebra universalis*, **31**(3), 397–406.

Demel J. (1982). Fast algorithms for finding a subdirect decomposition and interesting congruences of finite algebras. *Kybernetika (Prague)*, **18**(2), 121–130.

Duquenne V. (2010). Lattice drawings and morphisms. In *Formal Concept Analysis, 8th International Conference, ICFCA 2010, Agadir, Morocco, March 15-18, 2010. Proceedings*, p. 88–103.

Ferré S. (2014). *Reconciling Expressivity and Usability in Information Access from File Systems to the Semantic Web*. PhD thesis, Univeristy Rennes 1.

Freese R. (2008). Computing congruences efficiently. *Algebra universalis*, **59**(3-4), 337–343.

Funk P., Lewien A. & Snelting G. (1995). *Algorithms for concept lattice decomposition and their applications*. Rapport interne, TU Braunschweig.

Ganter B. & Wille R. (1999). *Formal concept analysis - mathematical foundations.* Springer.

Geyer W. (1994). The generalized doubling construction and formal concept analysis. *algebra universalis*, **32**(3), 341–367.

Grätzer G. (1978). *General lattice theory*. Basel: Birkhäuser-Verlag.

Mihók P. & Semanišin G. (2008). Unique factorization theorem and formal concept analysis. In S. Yahia, E. Nguifo & R. Belohlavek, Eds., *Concept Lattices and Their Applications*, volume 4923 of *Lecture Notes in Computer Science*, p. 232–239. Springer Berlin Heidelberg.

Nation J. (1995). Alan day's doubling construction. *algebra universalis*, **34**(1), 24–34.

Snelting G. (2005). Concept lattices in software analysis. In *Formal Concept Analysis, Foundations and Applications*, p. 272–287.

Valtchev P. & Duquenne V. (2008). On the merge of factor canonical bases. In *International Conference on Formal Concept Analysis ICFCA*, p. 182–198. Springer Berlin Heidelberg.

Visani M., Bertet K. & Ogier J.-M. (2011). Navigala: an Original Symbol Classifier Based on Navigation through a Galois Lattice. *International Journal on Pattern Recognition and Artificial Intelligence (IJPRAI)*.

Wille R. (1983). Subdirect decomposition of concept lattices. *Algebra Universalis*, **17**, 275–287.

Wille R. (1987). Subdirect product construction of concept lattices. *Discrete Mathematics*, **63**(2-3), 305–313.

C-SAKey : une approche de découverte de clés conditionnelles dans des données RDF

Nathalie Pernelle[1], Danai Symeonidou[2], Fatiha Saïs[1]

[1] LRI - UNIVERSITÉ PARIS SUD, Bâtiment Ada Lovelace, Orsay, France
prénom.nom@lri.fr

[2] Telecom Paris Tech, Paris, France
danai.symeonidou@telecom-paristech.fr

Résumé : L'exploitation des liens d'identité entre ressources RDF permet aux applications de combiner des données issues de différentes sources. Les approches permettant de lier des données sont largement fondées sur l'existence de clés éventuellement composites. Ces clés étant rarement disponibles, des approches récentes se sont intéressées à la découverte automatique de clés à partir de données RDF. Cependant, dans certains domaines, les classes de l'ontologie sont très générales et les clés valides pour tout l'ensemble d'instances d'une classe sont peu nombreuses. Aussi, dans l'approche C-SAKey, nous proposons de détecter des clés conditionnelles qui ne s'appliqueront qu'à un sous- ensemble des instances d'une classe. Nous avons réalisé une première expérimentation sur un jeu de données de l'INA qui montre que les clés découvertes par notre approche peuvent effectivement varier selon les conditions exprimées dans la clé.

Mots-clés : Intégration de données, Liens d'identité, Liage de données, Clés, RDF, OWL

1 Introduction

Les approches de liage de données permettent de générer des liens d'identité entre deux ressources RDF (voir Ferrara *et al.* (2011) pour un état de l'art). Pour la plupart, ces approches s'appuient sur l'utilisation de propriétés discriminantes (Hu *et al.* (2011); Saïs *et al.* (2009); Nikolov & Motta (2010); Al-Bakri *et al.* (2015)). Une clé est un ensemble de propriétés particulièrement discriminant puisque en théorie, les valeurs de ces propriétés permettent d'identifier un objet du monde réel sans ambiguïté. Ces clés peuvent être utilisées par un raisonneur pour inférer logiquement des liens d'identité ou pour construire des fonctions de similarité plus complexes prenant en compte des mesures de similarités élémentaires entre littéraux. Les clés peuvent être déclarées dans une ontologie représentée en OWL mais elles sont rarement disponibles. En effet, si certaines clés peuvent facilement être déclarées par un expert du domaine, telle que la propriété $isbn$ pour un livre, d'autres clés composites sont difficiles à déterminer, même pour un expert : dans quelle mesure un nom de famille, un prénom et une date de naissance suffisent-ils à identifier un chercheur ? Pourtant, les données RDF étant souvent incomplètes, plus les clés disponibles sont nombreuses, plus les liens d'identité générés grâce à ces clés par un outil de liage de données seront nombreux. Aussi, des approches récentes se sont intéressées à l'exploitation de sources de données RDF pour découvrir des clés automatiquement (Atencia *et al.* (2012); Pernelle *et al.* (2013); Soru *et al.* (2015); Symeonidou *et al.* (2014)).
Parfois, pour certains domaines, le nombre de clés découvertes par ces approches est limité, ou les clés sont trop complexes dans le sens où elles combinent un grand nombre de propriétés. Dans de telles applications, différentes stratégies peuvent être appliquées. Certaines approches s'intéressent aux ensembles de propriétés qui peuvent éventuellement comporter un certain nombre d'exceptions (Atencia *et al.* (2012); Symeonidou *et al.* (2014)). En effet, même

si un ensemble de propriétés n'est pas une clé, son utilisation peut permettre de découvrir de nombreux liens d'identité corrects. Ainsi, le téléphone est une propriété qui peut permettre de découvrir de nombreux liens d'identité pour des instances d'une classe $Restaurant$ même s'il existe un complexe hôtelier particulier pour lequel deux restaurants partagent le même numéro de téléphone. Atencia *et al.* (2012); Soru *et al.* (2015) découvrent des clés dont la sémantique ne s'appuie pas sur la sémantique OWL d'une clé qui sont cependant intéressantes quand les propriétés sont localement complètes. SAKey permet de découvrir des clés OWL, appelées *presque-clés*, comportant un nombre d'exceptions défini manuellement (Symeonidou *et al.* (2014)). Ces approches permettent de découvrir des clés valides dans des sous-ensembles d'instances des classes considérées mais ne permettent pas de caractériser ces sous-ensembles.
Dans cet article, nous présentons une approche appelée C-SAKey qui permet de découvrir des clés conditionnelles en exploitant des sources de données RDF. Une clé conditionnelle est une clé qui ne s'applique qu'à un sous-ensemble des instances d'une classe, sous-ensemble défini par des contraintes imposées sur les valeurs de certaines propriétés. Une première expérimentation sur un jeu de données de l'Institut National de l'Audiovisuel (INA) montre que les nouvelles clés découvertes peuvent effectivement varier selon les contraintes imposées sur les valeurs des propriétés.
En section 2, nous définissons quelques notions préliminaires. En section 3, nous présentons l'approche C-SAKey puis, en section 4, l'expérimentation que nous avons menée. Enfin, nous conclurons.

2 Préliminaires

Dans cette section, nous présentons tout d'abord le modèle de données puis l'approche SAKey sur laquelle s'appuie l'approche de découverte de clés conditionnelles C-SAKey.

Modèle de données

RDF (Resource Description Framework) est un modèle de données proposé par le W3C pour décrire des faits représentés sous forme de triplets $<subject, property, object>$. Une source de données RDF peut être associée à une ontologie qui va représenter le vocabulaire utilisé pour décrire les ressources. Une ontologie O peut être représentée par le tuple $(\mathcal{C}, \mathcal{P}, \mathcal{A})$ où $\mathcal{C}$ est l'ensemble des classes, $\mathcal{P}$ est l'ensemble des propriétés (relations et attributs) et $\mathcal{A}$ est un ensemble d'axiomes.

En OWL2 [1], il est possible de déclarer dans $\mathcal{A}$ qu'un ensemble de propriétés est une clé pour une expression de classe CE. Plus précisément, hasKey(CE ($ope_1, \ldots, ope_m$) ($dpe_1, \ldots, dpe_n$)) permet de déclarer que chaque instance de l'expression de classe CE est uniquement identifiée par les relations ou les relations inverses ope_i et les attributs dpe_j. Cela signifie qu'il n'existe pas de couple d'instances distinctes de la classe CE qui partagent des valeurs pour l'ensemble des relations et attributs de la clé. Par exemple hasKey($Researcher$ () ($firstName$, $lastName$)) permet d'exprimer que les attributs $firstName$ et $lastName$ forment une clé pour la classe $Researcher$.

1. http ://www.w3.org/TR/owl2-overview

Brève présentation de l'approche SAKey

SAKey est une approche efficace de découverte de clés dans des fichiers RDF (Symeonidou *et al.* (2014)). Afin de découvrir des ensembles de propriétés ayant un très fort pouvoir discriminant, et afin d'être capable de découvrir ces propriétés dans des fichiers RDF où des erreurs ou des duplicats peuvent exister, SAKey recherche des *n-presque clés* minimales : il s'agit des plus petits ensembles de propriétés pour lesquels il existe au plus n instances de la classe considérée qui partagent des valeurs pour cet ensemble de propriétés avec au moins une autre instance de la classe. Ces instances représentent les exceptions de la clé.

Dans l'ensemble de triplets RDF $D1$ présenté dans la figure 1, des exemples de clés découvertes pour la classe $Researcher$ seraient :

– 0-presque clés (pas d'exception) : {firstName, lastName}, {lastName, position} ...
– 2-presque clés (au plus deux exceptions) : {lastName}, {firstName}, ...

L'un des problèmes de la recherche de clés est de pouvoir passer à l'échelle afin de pouvoir apprendre des clés sur des volumes de données importants. Pour être efficace, SAKey recherche d'abord des *(n+1)-non-clés* maximales, des non clés présentant au moins n+1 redondances, et dérive ensuite les *n-presque-clés* minimales à partir de ces non clés. De plus, SAKey filtre les propriétés et leurs valeurs, et élague l'espace de recherche des non clés en s'appuyant sur certaines caractéristiques des données du jeu de données considéré, caractéristiques qui sont dynamiquement découvertes lors de l'application de SAKey.

Researcher(r1), $firstName(r1, "Fatiha"), lastName(r1, "Sais"),$
$worksIn(r1, "LRI"), position(r1," Assistant\ professor")$
Researcher(r2), $firstName(r2, "Nathalie"), lastName(r2, "Pernelle"),$
$worksIn(r2, "LRI"), position(r1," Assistant\ professor"),$
Researcher(r3), $firstName(r3, "Chantal"), lastName(r3, "Reynaud"),$
$worksIn(r3, "LRI"), position(r3, "Professor"),$
Researcher(r4), $firstName(r4, "Pierre"), lastName(r4, "Marquis"),$
$worksIn(r4, "CRIL"), position(r4," Professor")$
Researcher(r5), $firstName(r5, "Olivier"), lastName(r5, "Roussel"),$
$worksIn(r5, "CRIL"), position(r5, "Assistant\ professor")$
Researcher(r6), $firstName(r6, "Pierre"), lastName(r6, "Roussel"),$
$worksIn(r6, "CRIL"), position(r6, "Research\ engineer")$

FIGURE 1 – Exemple de source de données RDF ($D1$)

L'approche a été testée sur différents jeux de données et les résultats ont montré l'intérêt des *presque clés* pour le liage de données et l'efficacité de leur découverte. L'approche C-SAKey est une extension de cette méthode.

3 C-SAKey : une approche de découverte de clés conditionnelles

De façon à enrichir l'ensemble des clés susceptibles d'être découvertes, nous proposons une approche de découverte de *clés conditionnelles* appelée C-SAKey. Dans l'exemple D1, présenté en figure 1, nous pouvons observer que, sans exceptions, les propriétés $\{lastName, worksIn\}$ ne forment pas une clé pour la classe $Researcher$. Il existe en effet deux chercheurs dont le

nom de famille est $''Roussel''$ et qui travaillent dans le même laboratoire. Dans cet exemple, autoriser deux exceptions est suffisant pour découvrir que $\{lastName, worksIn\}$ peut être considéré comme une clé pour un grand nombre d'instances. Pourtant, dans une source de données de taille importante comportant de nombreux chercheurs, autoriser quelques exceptions ne sera peut-être pas suffisant. De plus, il existe peut-être des laboratoires pour lesquels le nom est suffisant pour identifier un chercheur. Il nous semble intéressant de pouvoir découvrir quels sont ces laboratoires. Dans cet exemple, la propriété $lastName$ est une clé pour les chercheurs travaillant au $''LRI''$, tandis qu'elle ne l'est pas pour les chercheurs du $''CRIL''$. Aussi, nous proposons de découvrir des clés valides pour des sous-ensembles de données caractérisés par une condition. Nous considérons des conditions pouvant être exprimées par une conjonction de propriétés pour lesquelles une valeur constante sera spécifiée. Plus précisément, si l'on considère une variable X représentant une instance de la classe c, et un ensemble de propriétés $Pcd = \{pcd_1, \ldots, pcd_m\}$ tel que $Pcd \subseteq \mathcal{P}$, une condition $cd(X)$ peut être exprimée par $pcd_1(X, v_1) \wedge \ldots \wedge pcd_m(X, v_m)$. Comme nous l'avons montré dans la section précédente, OWL2 permet de déclarer une clé pour une description de classe quelconque CE et donc pour une classe $c \sqcap cd$. Pour exprimer les conditions sur les valeurs des propriétés, les constructeurs *owl :DataHasValue*, que l'on notera $dhv(p, valeur)$, ou *owl :ObjectHasValue* devront être utilisés [2]. La sémantique d'une clé conditionnelle $((c \sqcap cd)\ (p_1, \ldots, p_n))$ (telle que l'ensemble $\{p_1, \ldots, p_n\}$ est disjoint de Pcd) peut alors être définie comme :

$$\forall X, \forall Y, \forall Z_1, \ldots, Z_n, \wedge c(X) \wedge c(Y) \wedge cd(X) \wedge cd(Y) \bigwedge_{i=1}^{n} (p_i(X, Z_i) \wedge p_i(Y, Z_i)) \Rightarrow X = Y$$

Pour une source de données RDF, nous considérons qu'une clé conditionnelle $((c \sqcap cd)\ (p_1, \ldots, p_n))$ est valide si la propriété suivante est vérifiée :

$$\forall X\ \forall Y\ ((X \neq Y) \wedge c(X) \wedge c(Y) \wedge cd(X) \wedge cd(Y)) \Rightarrow \nexists Z\ \nexists j\ (p_j(X, Z) \wedge p_j(X, Z)))$$

Seules les clés conditionnelles minimales sont intéressantes. Une clé $k1$ est plus générale qu'une clé $k2$ (noté $k1 \geq k2$) si la classe $c_1 \sqcap cd_1$ sur laquelle porte $k1$ subsume la classe $c_2 \sqcap cd_2$ de $k2$ et si l'ensemble des propriétés impliquées dans $k1$ est un sous-ensemble de l'ensemble des propriétés impliquées dans $k2$ (y compris les propriétés de la condition) :

$((c_1 \sqcap cd_1)(p_{11}, \ldots, p_{1n})) \geq ((c_2 \sqcap cd_2)(p_{21}, \ldots, p_{2m}))$

ssi $(c_2 \sqcap cd_2) \sqsubseteq (c_1 \sqcap cd_1)$ et $Pcd_1 \cup \{p_{11}, \ldots, p_{1n}\} \subseteq Pcd_2 \cup \{p_{21}, \ldots, p_{2m}\}$

En particulier, une clé définie pour une classe c de l'ontologie (Pcd = $\emptyset$) sera plus générale qu'une clé conditionnelle définie à partir de cette classe et pour le même ensemble de propriétés. On aura par exemple :

$(Researcher(lastName, firstName)) \geq ((Researcher \wedge dhv(worksIn, \textbf{\textit{LRI}})(lastName, firstName))$,

$(Person(lastName, worksIn)) \geq ((Researcher \wedge dhv(worksIn, \textbf{\textit{LRI}})(lastName))$

Une clé conditionnelle ki est minimale pour un jeu de données ssi $\nexists\ kj \neq ki$ tel que $kj \geq ki$.

Envisager de construire pour chaque classe de l'ontologie toutes les conditions, i.e., tous les ensembles de propriétés instanciées par leur ensemble de valeurs possibles, serait très peu efficace. De plus, de nombreuses clés conditionnelles seraient peu pertinentes car elles ne caractériseraient qu'un sous-ensemble de données de très petite taille. Une première approche consiste

2. voir http ://www.w3.org/TR/owl2-syntax/#Class_Expressions pour plus de détails.

à utiliser SAKey pour extraire les clés puis d'utiliser les non-clés maximales découvertes par SAKey pour élaguer l'espace de recherche des non clés conditionnelles. En effet, les propriétés impliquées dans une clé conditionnelle minimale sont incluses dans une non clé. Ensuite, soit :

- Un expert choisit un ensemble des propriétés (non clé) qui lui semblent pertinentes pour représenter des classes qui ne sont pas formalisées dans l'ontologie mais qui caractérisent des instances ayant potentiellement des propriétés instanciées différemment (exemples : regrouper les chercheurs par type de poste, regrouper les villes par régions de France etc.).
- Un pré-traitement sélectionne les propriétés des conditions et les constantes en fonction de la liste des non-clés et du support des classes restreintes par ces conditions candidates (i.e., en fonction du nombre d'instances des nouvelles classes).

Un pré-traitement sélectionne alors les descriptions RDF des instances qui contiennent les valeurs spécifiées dans les conditions pour ces propriétés. Les clés peuvent alors être recherchées en utilisant SAKey sur ce sous-ensemble d'instances. Les clés conditionnelles minimales peuvent ensuite être déduites à partir des non clés conditionnelles maximales en suivant un processus de dérivation similaire à celui introduit dans SAKey.

4 Expérimentations

Nous avons évalué notre approche sur un ensemble de données RDF fourni par l'Institut National de l'Audiovisuel (INA). Il s'agit de descriptions RDF de contenus audiovisuels (classe $Contenu$) et de personnes impliquées dans ces contenus. Les 44 779 instances de la classe $Contenu$ sont décrites par 82 propriétés. Par exemple, la propriété $ina{:}\,aPourTitreCollection$ représente le titre d'une émission, $ina{:}\,aPourTitrePropreIntegrale$ représente le titre d'un épisode particulier de l'émission. La propriété $ina{:}\,aPourGenre$ permet de représenter la catégorie d'un contenu audio-visuel (débat, sketch, série, ...). Sans exceptions, SAKey ne peut découvrir de clés dans ce corpus. Cela est dû à la présence de contenus ayant exactement les mêmes valeurs pour les 10 propriétés les plus instanciées que nous supposons être les plus pertinentes (fréquence>30%). Nous avons exécuté C-SAKey sur ces données en choisissant d'exploiter la propriété $ina{:}\,aPourGenre$ pour laquelle il existe 42 catégories de contenus : les conditions sont donc de la forme $ina{:}\,aPourGenre$=*"Sketch"*, $ina{:}\,aPourGenre$=*"Reportage"*, etc. Les clés découvertes sont présentées dans le tableau 1. Nous présentons au plus deux exemples de catégories pour chaque ensemble de clés trouvé. Par exemple, quand la condition $aPourGenre$=*"Sketch"* ou $aPourGenre$=*"Magazine"* est exploitée, la clé conditionnelle $\{ina{:}\,TitreCollection, ina{:}\,Participant, ina{:}\,TitrePropreIntegrale, ina{:}\,DateDiffusion\}$ est découverte.

Les résultats montrent que des clés conditionnelles peuvent être découvertes pour chacune des 42 catégories excepté pour la catégorie *Débat* qui contient les duplicats. De plus, nous pouvons observer que les clés conditionnelles peuvent varier en fonction de la catégorie même si certaines catégories partagent les mêmes clés. Une analyse qualitative des clés est bien sûr nécessaire. Pour cela, une évaluation par des experts de l'INA ainsi que l'utilisation des clés pour générer des liens d'identitéentre contenus pourraient être réalisées.

5 Conclusion

Nous avons présenté l'approche C-SAKey qui permet d'étendre une approche de découverte de clés pour découvrir des clés conditionnelles minimales, valides pour un sous-ensemble d'ins-

Condition/Valeur (ina: $aPourGenre$=...)	Clés conditionnelles découvertes
$Sketch$ $Magazine\ldots$	$\{\{ina: TitreCollection, ina: Participant,$ $ina: TitrePropreIntegrale, ina: DateDiffusion\}\}$
$Interview$ $Serie\ldots$	$\{\{ina: TitreCollection, ina: Participant, ina: TitrePropreIntegrale,$ $ina: Duree, ina: DateCreationNotice, ina: DateDiffusion\}\}$
$Chronique$ $Extrait$ $\ldots$	$\{\{ina: TitreCollection, ina: Participant, ina: TitrePropreIntegrale,$ $ina: Duree, ina: DateCreationNotice, ina: DateDiffusion\},$ $\{ina: TitreCollection, ina: Participant, ina: Theme,$ $ina: TitrePropreIntegrale, ina: Duree, ina: DateCreationNotice\}\}$
$Reportage$	$\{\{ina: TitreCollection, ina: Participant, ina: TitrePropreIntegrale,$ $ina: Duree, ina: DateCreationNotice\}\}$

TABLE 1 – Clés conditionnelles pour les 44 779 instances de la classe $Contenu$

tances de classe, sous-ensemble décrit par des conditions sur les valeurs de certaines propriétés. Une première expérimentation a été conduite sur des données réelles pour lesquelles aucune clé n'avait pu être découverte. Les résultats montrent la possibilité de découvrir des clés conditionnelles dans un tel contexte. D'autres expérimentations sont à mener et leurs résultats doivent être évalués.

Références

AL-BAKRI M., ATENCIA M., LALANDE S. & ROUSSET M.-C. (2015). Inferring same-as facts from linked data : an iterative import-by-query approach. In *Proceedings of AAAI 2015, to appear*.

ATENCIA M., DAVID J. & SCHARFFE F. (2012). Keys and pseudo-keys detection for web datasets cleansing and interlinking. In *EKAW*, p. 144–153.

FERRARA A., NIKOLOV A. & SCHARFFE F. (2011). Data linking for the semantic web. *Int. J. Semantic Web Inf. Syst.*, **7**(3), 46–76.

HU W., CHEN J. & QU Y. (2011). A self-training approach for resolving object coreference on the semantic web. In *WWW*, p. 87–96.

NIKOLOV A. & MOTTA E. (2010). Data linking : Capturing and utilising implicit schema-level relations. In *Proceedings of Linked Data on the Web workshop at 19th International World Wide Web Conference(WWW)'2010*.

PERNELLE N., SAÏS F. & SYMEONIDOU D. (2013). An automatic key discovery approach for data linking. *Journal of Web Semantics*, **23**, 16–30.

SAÏS F., PERNELLE N. & ROUSSET M.-C. (2009). Combining a logical and a numerical method for data reconciliation. *Journal on Data Semantics*, **12**, 66–94.

SORU T., MARX E. & NGONGA NGOMO A.-C. (2015). ROCKER – a refinement operator for key discovery. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*.

SYMEONIDOU D., ARMANT V., PERNELLE N. & SAÏS F. (2014). Sakey : Scalable almost key discovery in RDF data. In *13th International Semantic Web Conference (ISWC), Italy, 2014. Proceedings, Part I*, volume 8796 of *Lecture Notes in Computer Science*, p. 33–49 : Springer.

Eléments de réflexion sur l'utilisation de corpus pour la construction d'ontologies

Xavier Aimé[1]

INSERM UMRS 1142, LIMICS, F-75006, Paris, France ;
Sorbonne Universités, UPMC Univ. Paris 06, UMR_S 1142, LIMICS, F-75006, Paris, France ;
Université Paris 13, Sorbonne Paris Cité, LIMICS, (UMR_S 1142), F-93430, Villetaneuse, France.
xavier.aime@inserm.fr

Résumé : L'Ingénierie des Connaissances (IC) est née d'une inter-disciplinarité alliant informatique, intelligence artificielle, psychologie cognitive, ergonomie et sciences de gestion. Aujourd'hui, il semblerait que l'IC soit plus orientée vers la dimension technologique, notamment de par les problèmes d'hétérogénéité et de quantité massive de données qu'elle doit gérer. Un petit retour aux sources, vers les Sciences Humaines et Sociales, ne serait pas préjudiciable. De nos jours, il est communément fait appel aux corpus de textes pour la construction d'ontologies de domaine, corpus essentiellement considérés comme des ensembles de termes plus ou moins structurés. Or un texte est avant tout un acte de communication, un outil de transmission d'un message de la part d'un émetteur doté d'une intention à destination d'un récepteur. Nous proposons dans cet article quelques éléments de réflexions à prendre en compte pour la constitution du corpus et l'extraction de termes en nous fondons notamment sur l'analyse de contenu utilisé en psychologie.

Mots-clés : Ontologies, Corpus, Psychologie sociale, Analyse de contenu.

1 Introduction

L'Ingénierie des Connaissances (IC), depuis plus de vingt-cinq ans, vise à fournir des modèles de connaissances pour des systèmes d'aide à la décision, pour faciliter l'interopérabilité entre systèmes d'informations ou encore pour augmenter les possibilités de recherche d'information. A la fin des années 1980, les chercheurs de ce domaine de l'Intelligence Artificielle souhaitaient bâtir des systèmes qui étaient censés reproduire la dimension *dynamique* des experts [1] : la résolution de problèmes et la prise de décisions. Pour cela, il fallait en premier lieu être en mesure de reproduire la dimension *statique* (et souvent implicite) de ces experts : leurs connaissances. Des approches pluri-disciplinaires ont été développées à cette fin, associant psychologues (cognitifs et sociaux), ergonomes et informaticiens. On peut citer parmi elles KOD (Vogel, 1988) ou encore KADS (Wielinga *et al.*, 1992).

Dans les années 1990, cette approche a cependant dû être abandonnée pour plusieurs raisons. La première d'entre elles est que l'expert ne peut être considéré comme l'unique « dépositaire d'un système conceptuel qu'il suffirait de mettre au jour » (Bourigault *et al.*, 2004). Les chercheurs se sont ainsi rendu compte qu'il était impossible de construire un duplicata numérique d'un expert. Non seulement, cette ambition est très couteuse en termes de temps et de finances,

1. Pour Aimé (2014), un individu est considéré comme expert s'il est en mesure de respecter les trois conditions suivantes : (1) la compétence, (2) la référence et (3) la légitimité. La compétence, ou expertise, est pour un individu la détention d'une information utile au groupe dont ses membres ne disposent pas. La référence est pour un individu le pouvoir d'incarner la norme (mais aussi les valeurs, les croyances...) du groupe. Enfin, la légitimité pour un individu est le pouvoir qui lui est donné (par accord social, consensus, règlement, etc.) pour diriger/orienter les conduites des autres membres du groupe.

mais en plus cela ne fonctionne pas. Une autre raison tient au fait qu'un expert, dans une situation donnée, prend rarement une décision seule. La dimension sociale et pragmatique de cette prise de décision, et de la détention de la connaissance d'un domaine, n'était en effet jusque-là pas véritablement prise en compte. Il ne s'agit désormais plus de modéliser la connaissance d'un seul individu ayant le statut d'expert, mais celle d'un groupe d'individus qui partagent un ensemble de connaissances, et ce au moyen de méthodes comme Comon-KADS (Schreiber *et al.*, 1994).

La fin des années 1990 et le début des années 2000 voient l'avènement des ontologies computationnelles dont l'objectif est de modéliser et de formaliser – pour un domaine donné – des connaissances consensuelles. Les méthodes de construction d'ontologies proposées sont fondées principalement sur trois points : (1) développement logiciel (par exemple, Methontology (Fernández-López *et al.*, 1997)), (2) diversification toujours plus grande des sources de connaissances et (3) réutilisabilité des ressources produites. « Les documents et les corpus sont désormais reconnus comme sources principales de connaissances, qu'il y a lieu d'organiser en systèmes utiles» (Prié, 2000). L'ère du *Personal Computer* est révolu. Désormais le monde est connecté, le web est participatif. Les sites internet ne sont plus figés et la seule propriété des informaticiens. Tout le monde est à la fois producteur et consommateur, juge et parti (Ganascia, 2009). Chacun est désormais en mesure de produire du contenu et de se prétendre expert. Les sources de connaissances explosent véritablement en quantité (pas forcément en qualité). De ce fait, la question – pour l'ingénieur des connaissances – n'est plus dans le *comment je prends* mais dans *qu'est-ce que je prends*, dans la sélection, dans le filtrage des ressources qui ne seront plus seulement celles de l'expert. Pour Smith (2012), « le travail des ontologues est une réponse au fait que les nombreuses organisations industrielles, gouvernementales et scientifiques, dont les activités reposent sur l'utilisation des ordinateurs, doivent faire de plus en plus face à des difficultés découlant de leur nécessité de combiner des données provenant de plusieurs sources hétérogènes. »

Durant la décennie passée, et face à la croissance quasi exponentielle de la quantité de documents disponibles sur le Web, des réflexions ont été menées sur l'utilisation de documents textuels dans l'extraction de connaissances afin de construire des ontologies. L'idée se fonde sur « une vision unificatrice de la connaissance : le monde de la connaissance est découpé en domaines stables, dont chacun est équivalent à un réseau fixe de concepts, les termes étant les représentants linguistiques de ces concepts » (Bourigault *et al.*, 2004). Le risque est cependant de tomber dans le même piège des systèmes experts, à savoir considérer que toute la connaissance est exprimée dans le corpus de textes. Une place grandissante est laissée désormais aux méthodes issues du Traitement de la Langue Naturelle (TAL) et aux outils statistiques pour pouvoir gérer des corpus dont la taille est toujours de plus en plus grande. De plus, avec l'avènement du multimédia, les communications écrites comme orales doivent être prises en compte. Parallèlement, on peut constater, au fil des ans, que les recherches en IC sont de moins en moins pluridisciplinaires et deviennent de plus en plus informatiques. Elles s'intéressent de plus en plus à la gestion de la quantité de ces ressources au détriment de la gestion de leur qualité, pour des raisons évidentes de coût des projets, de nécessité de temps de traitement, etc. Plus que jamais, face à cette profusion de ressources et d'experts divers et variés, il est primordial de se poser la question de la qualité des documents et de leur utilisation pour construire des ontologies : est-ce que tout est dans le texte ? peut-on faire confiance aux textes ?

Cet article poursuit plusieurs objectifs. Le premier objectif est tout d'abord de poser quelques

définitions de la notion même de document et des ontologies, notamment sous l'angle de la psychologie sociale (cf. section 2). Le deuxième objectif est de présenter brièvement les grands points de méthodes de construction d'ontologies à partir de textes (cf. section 3). Le troisième objectif est de souligner la difficulté de construire un tel corpus à cet usage (cf. section 4). Enfin le dernier objectif est d'étudier et de proposer quelques pistes pour une meilleure prise en compte de certains critères lors de la construction d'ontologies à partir de textes (cf. section 5).

2 Définitions

L'une des vocations des ontologies est de permettre à un groupe d'individus de tenter d'avoir un consensus sur la signification des termes et des concepts qu'ils emploient. Les propos de cet article s'articulant autour de deux concepts, (1) le document et (2) l'ontologie, il semble pertinent d'essayer dans un premier temps d'en poser quelques bases de définitions.

2.1 Le document

De nombreux domaines s'intéressent au concept de document. De l'Ingénierie Documentaire à la Philosophie, en passant par l'IC et la Psychologie, les définitions sont nombreuses mais semblent néanmoins former un certain consensus. Ethymologiquement, le terme « document » vient du latin *docere* qui signifie enseigner. Selon l'Encyclopédie (1^{ere} édition, Volume 5, 1751), un document réfère à tous titres, pièces et autres preuves qui peuvent donner quelque connaissance d'une chose. Dans le même esprit, l'*International Institute for Intellectual Cooperation* définit un document comme étant toute base de connaissance, fixée matériellement, susceptible d'être utilisée pour consultation, étude ou preuve (par exemple un manuscrit, un imprimé, une représentation graphique ou figurée, ou encore un objet de collection). Plus synthétique, Buckland (1997) résume le document à toute expression de la pensée humaine ; ce que complète Ranganathan (1963), pour qui le document est synonyme d'une micro pensée incarnée apte à la manipulation physique, le transport à travers l'espace, et la conservation dans le temps. Pour Briet (1951), un document est « tout indice concret ou symbolique, conservé ou enregistré, aux fins de représenter, de reconstituer ou de prouver un phénomène physique ou intellectuel ». Selon Prié (2000), un document est « une trace de l'activité humaine, créée par un auteur et mise à disposition de lecteurs. [...] (il) a été créé dans un certain contexte de production, en vue d'un certain contexte de réception, il appartient également à un genre, et réfère de façon implicite ou non à d'autres documents, qui en tant qu'inter-texte en prescrivent également l'interprétation ». Bachimont & Crozat (2004b) abordent le document suivant trois axes : (1) la forme, (2) le signe et (3) le medium. « Le document comme forme renvoie au fait que le document est une forme physique perceptible dont la matérialité physique se prête à l'instrumentation technique. [...] Le document comme signe renvoie aux problèmes de manipulation, lecture et interprétation du contenu, en prenant en compte la double face matérielle et intelligible du signe. [...] Le document comme medium thématise le document comme objet social, objet de négociation et de transaction culturelle et économique. Il constitue un milieu d'échange entre des individus et des groupes qui s'articule et s'ancre dans la vie sociale ». (Bachimont & Crozat, 2004b)

En résumé, le document serait à la fois vue (1) selon une dimension cognitive comme une *mémoire persistante* avec une intention de preuve, et (2) selon une dimension sociale comme un *objet de communication*, que ce soit communication d'une connaissance ou transmission d'un

message doté d'une intentionnalité de la part d'un émetteur à destination d'un (ou plusieurs) récepteur(s) dans un contexte donné.

2.2 Les ontologies

Outre son aspect informatique et son origine philosophique, une ontologie computationnelle peut être considérée comme un objet de la *psychologie sociale*, comme une sorte de *représentation sociale* (dans son approche structurale) formalisée. Jodelet (1989) définit les représentations sociales comme « un ensemble de connaissances / croyances correspondant à un système d'interprétation du réel construit conjointement par un groupe afin de gérer la réalité. » Cette définition est très proche de celle de Gruber (1995) concernant les ontologies computationnelles. Selon cet auteur, ces connaissances / croyances partagées (en règle général de sens commun) facilitent la communication interindividuelle et limitent les conflits. Elles ont un impact sur le plan individuel de par le fait qu'elles définissent l'identité de l'individu comme membre du groupe, mais également son mode de pensée par la sémantique de chaque concept qu'elle définit consensuellement pour le groupe (norme). Ce côté social des ontologies est saillant tant dans la phase de leur construction (question du consensus) que dans celle de leur utilisation (question de l'appropriation), et plus globalement dans la question de la norme qu'elles constituent. Selon Stewart & Fraïssé (2009), les représentations sociales modélisent « ce que les gens pensent connaître et sont persuadés de savoir à propos d'objets, de situations, de groupes donnés. Ces connaissances de sens commun que sont les représentations sociales permettent alors de saisir la signification d'un objet ou d'une situation. Mais cette signification n'est pas inhérente à l'objet de représentation. C'est une réalité construite, appropriée par un individu ou un groupe et intégrée dans son système de valeurs dépendant de son histoire et du contexte social qui l'environne. »

Une ontologie reflète un aspect contextuel d'une catégorisation. Par exemple, modéliser le fait que l'eau *est un* liquide transparent est correct à nos conditions normales de température et de pression atmosphérique. Cette modélisation diffère si nous changeons l'un de ses paramètres, en dessous de zéro degré l'eau *est un* solide blanc (ou transparent). Une ontologie reflète également un aspect social d'une catégorisation. Par exemple, modéliser le fait qu'un chien *est un* animal de compagnie non comestible est correct pour la plupart d'entre nous. Mais cette modélisation diffère dans certains pays d'Asie où un chien *est un* animal comestible.

C'est à ce niveau que l'ontologie, pour un domaine, un endogroupe [2] et un contexte donnés, semble être le plus à même de modéliser une représentation sociale. Si on se réfère au modèle structuré de Abric (1987), d'un point de vue structurel (et intensionnel), le noyau central regrouperait les concepts de haut niveau contenus dans l'ontologie de haut niveau et dans l'ontologie de domaine. Le système périphérique contiendrait essentiellement les concepts de bas niveaux. D'un point de vue extensionnel (les instances), il s'agit principalement d'instances des concepts de bas niveau et donc rattachables au système périphérique. Enfin d'un point de vue expressionnel, il est envisageable de considérer les termes les plus typiques pour dénoter chaque concept comme membre du noyau central du fait de leur aspect consensuel et leur stabilité dans le temps au niveau du groupe.

2. Ce terme est issu de la théorie de l'identité en psychologie sociale. Il s'agit d'un groupe d'individus partageant un ensemble de valeurs ou d'intérêts.

Voyons maintenant quelques éléments de méthodologies pour construire une ontologie à partir d'un corpus.

3 Quelques éléments de méthodologie

Le processus de construction d'une ontologie à partir de corpus de textes est assez simple et se compose (très) schématiquement de quatre grandes étapes.

La première étape, primordiale pour la qualité de l'ontologie, est la *constitution du corpus*. « Le corpus se constitue de documents produits dans le contexte où le problème à résoudre se pose. Ce sont par exemple des documentations techniques, des ouvrages de références, des documents de travail, des manuels propres au domaine ou à l'industrie concernée, ou bien encore la transcription d'interviews menées avec des spécialistes » (Bachimont, 2000). Ce corpus doit être suffisamment large pour couvrir tout le domaine, et consensuel pour répondre à l'objectif d'une ontologie qui est – par définition – la formalisation d'une conceptualisation consensuelle.

La deuxième étape réside dans la *sélection des candidat termes* au moyen d'outils comme SYNTEX (Bourigault *et al.*, 2005) ou BIOTEX (Lossio-Ventura *et al.*, 2014). Ces syntagmes nominaux sont associés au moins à une notion de fréquence. Plusieurs stratégies peuvent dès lors être adoptées. Par exemple, on peut ne prendre que les candidats termes de fréquence élevée ou décider que certains candidats termes de fréquence faible ont aussi une importance dans le domaine. Selon Bourigault *et al.* (2004), « sachant qu'il ne pourra analyser tous les candidats termes extraits du corpus, l'analyste doit adopter une stratégie optimale qui, étant donné le temps qu'il consacre à l'analyse terminologique et le type de la ressource à construire, lui garantit que, parmi les candidats qui auront échappé à son analyse, la proportion de ceux qui auraient pu être pertinents est faible. » On voit ici que le principal critère de sélection réside sur la notion de fréquence (quand elle est élevée), ou sur une pertinence supposée par l'ontologue et validée par un expert. La distribution statistique ne pouvant tenir lieu à elle seule de sémantique, il est nécessaire à ce stade de passer par une étape de normalisation sémantique dans un paradigme donné comme par exemple l'un de ceux proposés par Rastier *et al.* (1997) : (1) le paradigme *référentiel* où chaque terme est lié à l'objet qu'il représente, (2) le paradigme *psychologique* où chaque terme est associé à une image mentale (une représentation psychologique construite et stockée en mémoire sémantique) ou (3) le paradigme *différentiel* où chaque terme est associé aux termes voisins suivant le principe de communauté (avec le père et les frères) et le principe de différence (avec le père et les frères) – paradigme choisi et développé en IC par Bachimont (2000).

La troisième étape va consister en (1) des regroupements de candidats termes synonymes dénotant un même concept, et (2) en une *structuration hiérarchique de ces concepts*. Ces étapes peuvent se réaliser au moyen de l'outil TERMINAE (Aussenac-Gilles *et al.*, 2008), dont on soulignera au passage la possibilité de visualiser le contexte des candidats termes.

La quatrième étape réside dans la *validation* de la conceptualisation obtenue au moyen d'outils collaboratifs tels WebProtégé (Tudorache *et al.*, 2013) ou, d'un point de vue plus formel, par des outils tels OOPS ! (Poveda-Villalón *et al.*, 2014).

4 De la difficulté de choisir un corpus

Pas de bon corpus, pas de bonne ontologie, pas de bons résultats. « La constitution d'un corpus est très délicate de manière générale car le corpus conditionne largement le type et la nature des traitements que l'on peut effectuer sans que l'on ait forcément loisir de choisir le type de données le plus adéquat. Le choix d'un corpus introduit des biais sans qu'il soit toujours loisible de les apprécier » (Bachimont, 2000). S'il n'existe pas a priori de règles établies, Bourigault *et al.* (2004) énumèrent néanmoins quelques conditions à respecter. Il y a, en premier lieu, la question de la *disponibilité des ressources*, disponibilité que l'on peut décliner suivant trois critères : (1) critère de *connaissance* (on ne peut y répondre sans les experts du domaine), (2) critère *légal* (cela nécessite parfois de se plier à certaines contraintes administrative comme l'anonymisation préalable ou une déclaration CNIL), et (3) critère *social* (cela nécessite aussi de savoir qui donne l'accès à la connaissance, d'où une bonne connaissance de l'écosystème, et parfois le respect des voies hiérarchiques et des contraintes politiques). En deuxième lieu, il « convient de s'assurer auprès des spécialistes que les textes choisis ont un *statut suffisamment consensuel* pour éviter toute remise en cause ultérieure de la part de spécialistes ou d'utilisateurs ». Enfin, en troisième lieu, le corpus doit avoir une taille convenable pour couvrir assez largement le domaine étudié, mais aussi pour pouvoir être utilisé manuellement par l'ontologue pour vérification. La question sociale est ici tout aussi importante que la question du contenu du corpus. Le contexte de production est également à prendre en compte tant pour la sélection des documents que pour la sélection des termes (un auteur n'utilise pas toujours les mots désignant exactement ce dont il veut parler). On peut dès lors regretter que la dimension *acte de communication* de ces documents soit rarement abordée en IC. La question de l'objet d'utilisation de l'ontologie est également primordiale. Il doit en effet y avoir adéquation entre le contenu de l'ontologie construite et ce à quoi elle va servir. Dans le cadre de la réflexion menée dans cet article, ce contenu est obtenu principalement à partir d'un corpus de textes. Or, contrairement aux apparences, ce corpus n'est pas toujours porteur des bonnes, ou nécessaires, connaissances pour élaborer l'ontologie. A titre d'illustration, prenons trois domaines : (1) la Psychiatrie, (2) la Mode et l'Habillement et (3) le Droit.

4.1 Domaine de la Psychiatrie

Le domaine de la Psychiatrie comporte de multiples approches théoriques (psychanalytique, systémique, humaniste, cognitiviste, béhavioriste, biomédicale) plus ou moins (in)compatibles, ce qui engendre de nombreuses conséquences tant sur le plan de la conceptualisation (choix des documents pour parler d'une même pathologie) que celui de l'analyse sociale préalable (choix des auteurs, puis des experts pour valider l'ontologie construite). Entre l'approche psychanalytique (et ses nombreux courants) et l'approche biomédicale, les points de vue sont très différents, si ce n'est inconciliables dans une même ontologie. Une fois l'approche théorique choisie, il reste à déterminer les documents en tenant compte du contexte de production. Afin de construire un corpus de référence pour construire une ontologie de la psychiatrie pour un service donné, nous avons questionné au préalable les praticiens pour savoir si de tels documents étaient utilisables en l'état. Nous avons été confrontés à plusieurs cas de figures qui illustrent assez bien la complexité de la tâche. Le contexte de production est – dans ce cas précis – un praticien qui doit écrire un document décrivant un patient atteint d'une pathologie mentale. Ce

document, un acte de communication, est destiné à un autre praticien qui connait le domaine, à la famille du patient qui n'appartient pas au corps médical, et au patient qui souffre d'une pathologie pouvant altérer son jugement et dont l'état peut être aggravé à la lecture du dit document. En conséquence, certains praticiens nous ont révélé ne rien écrire, la transmission à leur confrère se faisant uniquement par oral. D'autres nous disent que le vocabulaire est très adapté à la situation, voir édulcoré. Par exemple, des praticiens ne parlent jamais de *relations sexuelles*, préférant l'usage du syntagme *relations intimes*. De même, ils préfèrent parler de *personnalité complexe* plutôt que de détailler certains éléments de la pathologie. Les termes utilisés sont alors plus à prendre comme des codes dont seuls les praticiens connaissent la véritable signification. Enfin, une grande partie révèle qu'il y a beaucoup d'implicite. Si nous nous tournons maintenant vers les grandes classifications utilisées en psychiatrie, se pose le problème du consensus car – ici peut-être plus que dans les autres domaines de la médecine – ces classifications sont porteuses d'idéologies. Les détracteurs du manuel diagnostique et statistique des troubles mentaux (DSM) soulignent le fait qu'aujourd'hui le problème n'est pas tant dans la classification (qui sera toujours critiquable) que dans les critères diagnostiques qui seraient largement influencés par l'industrie pharmaceutique et déplacerait suivant leur bon vouloir le seuil d'atteinte de telle ou telle pathologie (nécessitant de facto tel ou tel traitement). En résumé, il est assez difficile dans ce domaine de constituer un corpus le plus consensuel possible. Il n'est pas de bonne solution, tout juste la moins mauvaise.

4.2 Domaine de la Mode et de l'Habillement

Depuis l'apparition, au Moyen-Âge, des premiers mots inventés pour désigner un vêtement dans les langues romane et germanique, à l'origine du français et de l'anglais modernes, le vocabulaire de l'habillement a évolué, voire muté, et n'a cessé de se développer en cohérence avec la propre évolution de la garde-robe occidentale. Le nombre de vêtements d'allure distincte en usage il y a plus de deux mille ans, de même que celui des termes spécifiques les désignant, ne dépasse pas la cinquantaine. En 2012, ce chiffre s'élève à plus de cinq cents.

Dans le cadre de nos travaux pour la création de VETIVOC, une ressource termino-ontologie du domaine du Textile, de la Mode et de l'Habillement (Aimé *et al.*, 2014), nous sommes amenés à étudier les vocabulaires utilisés dans les différents champs couverts par ce domaine. Plusieurs ressources à caractère consensuel sont disponibles. La grande majorité des publications dédiées à la mode (95 %) sont positionnées soit sur le créneau dit « beaux livres » (autrement dit, des ouvrages privilégiant l'aspect iconographique à l'aspect textuel), soit sur le créneau littéraire avec des textes historiques, sociologiques ou hagiographiques. Les 5 % des publications restantes sont des dictionnaires spécialisés qui peuvent être répartis en deux groupes : (1) les dictionnaires spécialisés proposant une approche généraliste de la mode et (2) les lexiques exclusivement consacrés aux vêtements ou aux accessoires, avec une approche sélective. Cependant, il est un autre type de ressources – consensuelles – à considérer car bien plus consultés que les ouvrages spécialisés évoqués plus haut : les magazines et les catalogues commerciaux. Véritables prescripteurs, on pourrait penser qu'une telle responsabilité mériterait une rigueur accrue en ce qui concerne les termes employés pour désigner les vêtements et les accessoires. Pourtant, face à l'ampleur, la complexité du vocabulaire de la mode, et le nombre restreint d'outils qui le formalisent, les erreurs et les imprécisions sont fréquentes dans la presse comme dans les catalogues commerciaux.

A titre d'illustrations, prenons l'exemple d'un article de la revue *FashionMag* [3] en date du 11 janvier 2015 sur la pré-collection de Kenzo. Les créations y sont entre autres décrits par les matières utilisées. On y note, d'une part, l'usage mélangé de termes en français et en anglais (alors qu'il existe leur correspondance en français). Mais on y repère, d'autre part, un mauvais usage de ces termes. Ainsi, il est notifié la présence de « shearling », i.e. de mouton retourné. Or la collection est en fausse fourrure ; il ne peut s'agir de vrai mouton retourné. L'emploi de ce terme (1) n'informe donc pas le lecteur avec clarté si son anglais est insuffisant et (2) il induit en plus erreur le lecteur anglophone en lui laissant supposer qu'il s'agit de vrai fourrure. Balteiro (2014) a analysé cette même tendance dans la presse spécialisée espagnole. L'auteur constate que le côté « tendance » de ces anglicismes y est très souvent favorisé en dépit de larges glissements sémantiques.

4.3 Domaine du Droit

Une autre illustration de la difficulté de choisir un corpus est fournie par le domaine juridique. La disponibilité des sources juridiques n'est aujourd'hui plus vraiment un problème, puisque l'accessibilité de la loi est un objectif de valeur constitutionnelle et internet en favorise aisément la mise à disposition physique des textes juridiques. N'apparaît pas non plus problématique la taille convenable de ce corpus, car il y a assez de textes juridiques pour construire une ontologie digne d'intérêt. Au contraire, il peut être nécessaire de limiter la taille du corpus pour assurer une certaine cohérence à l'ensemble.

En revanche, le statut suffisamment consensuel des textes juridiques choisis peut être un sujet de débat. En effet, on ne peut appréhender le domaine juridique de manière unitaire. D'importantes distinctions sont à opérer et elles font souvent l'objet de débats. On peut ainsi d'abord classer le domaine juridique selon les diverses grandes branches du droit : droit administratif, droit pénal, droit civil, droit international, droit européen, etc. On peut également subdiviser ce domaine en fonction de l'objet qu'il entend régir : on distingue ainsi le droit des libertés publiques, le droit des affaires, le droit de la fonction publique, le droit fiscal, le droit de l'urbanisme, le droit de l'environnement, le droit de la famille, le droit des collectivités territoriales, le droit du travail, etc. Ces deux grands critères de classification peuvent se croiser : on trouve ainsi le droit pénal des affaires ou le droit européen de l'environnement. Chaque sous-domaine juridique correspond à une communauté d'expertise distincte. D'un sous-domaine à un autre, les mêmes termes juridiques peuvent ainsi avoir des nuances et des implications différentes. Il est donc important de construire un corpus en tenant compte de ces spécificités et en remettant les termes dans le contexte juridique adéquat.

Un autre critère de choix du corpus peut être la portée juridique des textes. On peut décider de construire un corpus contenant seulement les textes juridiquement contraignants, tels que les règlements européens, les lois et les règlements (décrets, arrêtés) en droit national. Ce sont eux qui créent des droits et des devoirs pour les individus, les entreprises et toute autre personne morale. On écartera ainsi les circulaires, les instructions, les décisions jurisprudentielles, ainsi que la doctrine juridique (articles de revues spécialisées, manuels et ouvrages). L'orientation de ces choix dépendra essentiellement de l'objectif poursuivi par l'ontologie. En d'autres termes, il faudra se poser la question de savoir à qui s'adressent le corpus et l'ontologie. S'ils visent

3. http ://www.fashionmag.com

un public spécialisé de juristes, juges et avocats, le fait de ne pas prendre en compte les textes d'orientation, telles que les circulaires et les instructions, les jurisprudences et les articles et ouvrages de doctrine, engendrera un corpus amputé et sera loin de remplir sa fonction d'aide à la décision. S'ils visent un public de non juristes qui cherche à trouver des éléments pour l'aider à prendre une décision en vue de résoudre un problème opérationnel, il peut être intéressant, pour ne pas noyer les utilisateurs, de limiter le corpus aux textes juridiquement contraignants en écartant l'insertion directe des autres composantes d'un bon corpus juridique (circulaires, instructions, jurisprudences, doctrine). Toutefois, ces dernières ne devraient pas être purement et simplement supprimées mais remplacées par des textes de vulgarisation permettant d'éclairer les textes juridiques contraignants dont la seule lecture n'est pas suffisante pour aider à la décision.

5 Discussion

La démarche de construction d'une ontologie à partir d'un corpus de textes s'apparente en de nombreux points au processus d'analyse de contenu. Ce processus a été développé il y a plusieurs décennies en Psychologie (elle s'apparente également à l'herméneutique ancienne, dans une tradition aristotélicienne). Il s'agit d'un « ensemble de techniques d'analyse des communications visant, par des procédures systématiques et objectives de description du contenu des énoncés, à obtenir des indicateurs (quantitatifs ou non) permettant l'inférence de connaissances relatives aux conditions de production/réception (variables inférées) de ces énoncés » (Bardin, 1977). Cette procédure, de type structuraliste, comporte deux étapes successives : (1) l'*inventaire des termes*, et (2) leur *classification*. Ainsi, l'analyse de contenu vise à décrire objectivement, systématiquement et quantitativement le contenu du corpus étudié. Cette méthode – telle que présentée par l'auteur – a pour objectif de construire une catégorisation de manière exhaustive à partir des termes utilisés dans le corpus et ainsi fournir à l'analyste (ou l'ontologue) un réseau sémantique reflétant le contenu du corpus étudié. Cette méthode se fonde sur l'hypothèse de Whorf quant à l'interdépendance de la langue et de la pensée (Whorf *et al.*, 2012). Pour l'auteur, la catégorisation en analyse de contenu « a pour objectif premier de fournir par condensation une représentation simplifiée des données brutes. » Sur cette représentation, l'analyste va opérer des inférences, avec comme hypothèse forte qu'il n'introduit pas de biais. Deux possibilités sont envisagées : (1) une *catégorisation a priori* où l'analyste élabore sa catégorisation à partir des termes recueillis, ou (2) une *catégorisation a posteriori* où l'analyste dispose d'une classification qu'il va peupler avec de nouvelles catégories ou avec des termes extraits. Bardin (1977) énumère cinq critères à respecter pour obtenir une bonne catégorisation : (1) l'*exclusion mutuelle* où un terme ne peut appartenir qu'à une seule catégorie (éviter au grand maximum les ambiguïtés) ; (2) l'*homogénéité* où « un même principe doit gouverner l'organisation des catégories » ; (3) la *pertinence* où les catégories construites doivent répondre à la problématique retenue ; (4) l'*objectivité* et la *fidélité* où si on a plusieurs analystes, on doit avoir la même grille de lecture, et les variables traitées doivent être clairement définies dès le départ ; et (5) la *productivité* où les catégories créées doivent permettre une inférence riche, des hypothèses nouvelles et représenter des données fiables.

Dans ce cadre, de nombreuses réflexions ont été menées sur le choix des termes et le sens à y associer. En effet, le sens donné à un terme, le type de message sous-tendu, et donc la validité du terme utilisé dépendent du contexte, lequel peut être exprimé en termes de sphères tels que pro-

posés par Searle et Austin dans la classification des actes de discours. Par exemple, Chabrol & Bromberg (1999) propose une grille de lecture permettant l'identification d'une cinquantaine de types d'actes de parole, regroupés en cinq sphères dont une sphère informationnelle regroupant tout ce qui est à propos des objets du monde (elle sert à construire un environnement manifeste ; c'est assez typiquement la sphère des connaissances).

Un texte est un acte de communication dont l'objet est à la fois (1) de transmettre une information à autrui, et (2) d'agir sur autrui. Pour Bachimont & Crozat (2004b), un document est un « objet matériel qui se déploie dans la temporalité d'une lecture et qui donne lieu à une interaction avec le lecteur ». Considérer une ontologie construite uniquement à partir de textes revient à (re-)définir la notion d'ontologie comme étant « le reflet d'une des façons dont la connaissance peut être perçue et dite » (Aussenac-Gilles & Sörgel, 2005). Selon Grice (1957), « the meaning (in general) of a sign needs to be explained in terms of what users of the sign do (or should) mean. » Cela a deux conséquences : (1) si, par définition, une ontologie est un objet formalisant une conceptualisation consensuelle, alors il doit également y avoir consensus au préalable sur ce corpus et donc sur le discours des auteurs, et (2) lorsqu'il construit une ontologie à partir d'un corpus de textes, l'ontologue se trouve influencé non seulement par sa propre subjectivité, par l'objectif de son analyse, ses options théoriques, les enjeux du projet, les besoins d'exhaustivité, mais aussi par les auteurs du corpus sur lequel il travaille.

Un autre point est également à prendre en compte : celui de la confusion entre la dimension syntaxique (et statistique) et la dimension sémantique. Selon Roche (2007), « les structures conceptuelles construites à partir de textes ne sont pas des ontologies au sens d'une conceptualisation d'un domaine au-delà de tout discours. Elles relèvent de la sémantique lexicale : il n'y a pas de concepts dans un texte, mais uniquement des usages linguistiques de ces concepts. La construction d'ontologies à partir de textes repose sur un ensemble d'hypothèses fortes. [...] La première est de dire que les experts peuvent traduire leurs connaissances ontologiques du domaine dans des textes et que ces derniers constituent un monde plus ou moins clos. La deuxième considère le processus de rétro-ingénierie comme possible, basée sur le fait que certaines catégories de mots et que certaines relations linguistiques traduisent un usage en langue de concepts et de relations conceptuelles. La troisième hypothèse postule que les structures lexicale et conceptuelle sont relativement isomorphes. Enfin, que la validation par les experts suffit à ériger la structure conceptuelle comme une ontologie du domaine. » L'ensemble de ces hypothèses font que l'ontologie ainsi construite ne repose finalement que sur le décret du statut conceptuel de constats syntaxiques fondés sur des fréquences. Mais construire une ontologie en dehors de ce cadre statistique est difficilement automatisable et donc fortement chronophage. On comprend dès lors pourquoi cette solution n'est pas aujourd'hui majoritairement prisée.

6 Conclusion

L'IC est née d'une inter-disciplinarité alliant informatique, intelligence artificielle, psychologie cognitive, ergonomie et sciences de gestion. Aujourd'hui, il semblerait que l'IC soit plus orientée vers la dimension technologique, notamment de par les problèmes d'hétérogénéité et de quantité massive de données qu'elle doit gérer.

Dans ses premières heures, l'IC – avec ses systèmes experts – était dans un idéal de biomimétisme de l'expert où la machine allait reproduire le fonctionnement de son cerveau. Nous assistons aujourd'hui à un véritable changement de paradigme en IC. Il ne s'agit plus de repro-

duire virtuellement et fidèlement, sous la forme d'une boite noire autonome, le fonctionnement d'un expert avec l'aide d'autres disciplines des Sciences Humaines. Il s'agit de fournir un outil informatique donnant de manières appropriée et intelligente des informations à un utilisateur qui est confronté à un problème, et qui vont l'aider à résoudre ce problème en lui apportant des connaissances supplémentaires adéquates. Les systèmes ne sont plus alors décisionnels, mais qualifiés d'*aide à la décision*.

Il nous semble donc qu'un petit retour aux sources ne serait pas préjudiciable. Nous n'inventons rien, nous ne faisons que rappeler certaines évidences que les ontologues d'aujourd'hui ne voient peut-être plus. Dans la lignée de Vygostky, il nous faut ré-aborder les connaissances comme étant enracinées dans le social et considérer les documents au-delà de leur aspect mémoriel. Une ontologie crée à partir d'un corpus de textes bien choisi et dans lequel les termes auront été extraits d'un point de vue pragmatique, puis utilisé en adéquation avec son milieu, offre d'énormes avantages que ce soit en termes d'appropriation (dans le sens de « faire sien le contenu et de l'intégrer comme une part de soi » (Bachimont & Crozat, 2004a)) ou en termes de représentation sociale. Pour connaître sa réutilisabilité, il est également nécessaire de lui adjoindre en méta-données un certain nombre d'informations sociales telles que les auteurs du corpus utilisé (et évaluer leur statut d'autorité ainsi que la consensualité du corpus), le contexte de production du corpus, quel groupe d'individus est destinataire du projet, dans quelle sphère ont été choisi les termes...

Remerciements

Je remercie Rossella Pintus et Sophie George pour leur éclairage respectifs sur les domaines du Droit et de la Mode/Habillement.

Références

ABRIC J. (1987). *Coopération, Compétition et Représentation Sociale*. Fribourg, Suisse : Delval.

AIMÉ X. (2014). Pour une approche écologique des ontologies computationnelles. *Intellectica – Dossier spécial "Philosophie du Web et Ingénierie des Connaissances"*, **1**(61), 189–210.

AIMÉ X., GEORGE S. & HORNUNG J. (2014). VETIVOC, une Ressource termino-ontologique modulaire du domaine du textile, de la mode et de l'habillement. *Revue d'Intelligence Artificielle*, **6**, 689–728.

AUSSENAC-GILLES N., DESPRES S. & SZULMAN S. (2008). The terminae method and platform for ontology engineering from texts. In *Proceedings of the 2008 Conference on Ontology Learning and Population : Bridging the Gap Between Text and Knowledge*, p. 199–223, Amsterdam, The Netherlands, The Netherlands : IOS Press.

AUSSENAC-GILLES N. & SÖRGEL D. (2005). Text Analysis for Ontology and Terminology Engineering. *Applied Ontology, IOS Press*, **1**(1), 35–46.

BACHIMONT B. (2000). Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. In J. CHARLET, M. ZACKLAD, G. KASSEL & D. BOURIGAULT, Eds., *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, p. 305–323. Paris, France : Eyrolles.

BACHIMONT B. & CROZAT S. (2004a). Instrumentation numérique des documents : pour une séparation fonds/forme. *Information-Interaction-Intelligence*, **4**(1), 95–104.

BACHIMONT B. & CROZAT S. (2004b). Réinterroger les structures documentaires : de la numérisation à l'informatisation. *Information–Interaction–Intelligence*, **4**(1), 59–74.

BALTEIRO I. (2014). The influence of English on Spanish Fashion Terminology : -ING forms. *Journal of English for Specific Purposes at tertiary level*, **2**(2), 156–173.

BARDIN L. (1977). *L'analyse de contenu*. Paris, France : PUF.

BOURIGAULT D., AUSSENAC-GILLES N. & CHARLET J. (2004). Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas. *Revue d'Intelligence Artificielle*, **18**(4), 97–110.

BOURIGAULT D., FABRE C., FRÉROT C., JACQUES M. & OZDOWSKA S. (2005). Syntex, analyseur syntaxique de corpus. In *TALN 2005, Dourdan, 6-10 juin*.

BRIET S. (1951). *Qu'est-ce que Ia documentation*. Editions Documentaires Industrielles et Techniques.

BUCKLAND M.-K. (1997). What is a "document" ? *JASIS*, **48**(9), 804–809.

CHABROL C. & BROMBERG M. (1999). Préalables à une classification des actes de parole. *Psychologie française*, **44**(4), 291–306.

FERNÁNDEZ-LÓPEZ M., GÓMEZ-PÉREZ A. & JURISTO N. (1997). METHONTOLOGY : From ontological art towards ontological engineering. In *Proceedings of the The Fourteenth National Conference on Artificial Intelligence (AAAI'97), Workshop on ontological engineering*, p. 33–40, Stanford, USA.

GANASCIA J. (2009). *Voir et pouvoir : qui nous surveille ? Un essai sur la sousveillance et la surveillance à l'ère de l'infosphère*. Paris, France : Editions du Pommier.

GRICE H.-P. (1957). Meaning. *Philosophical Review*, (66), 377–388.

GRUBER T. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, **43**(5/6), 907–928.

JODELET D. (1989). *Les représentations sociales*. Paris, France : PUF.

LOSSIO-VENTURA J., JONQUET C., ROCHE M. & TEISSEIRE M. (2014). BIOTEX : A system for Biomedical Terminology Extraction, Ranking, and Validation. In *Proceedings of the 13th International Semantic Web Conference (ISWC'14). Trento, Italy*.

POVEDA-VILLALÓN M., GÓMEZ-PÉREZ A. & SUÁREZ-FIGUEROA M. (2014). OOPS !(OntOlogy Pitfall Scanner !) : An On-line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, **10**(2), 7–34.

PRIÉ Y. (2000). Sur la piste de l'indexation conceptuelle de documents, une approche par l'annotation. *Document Numérique*, **4**(162), 11–35.

RANGANATHAN S. (1963). *Documentation and its facets*. London, UK : Asia Publishing House.

RASTIER F., CAVAZZA M. & ABEILLÉ A. (1997). *Sémantique pour l'analyse*. Paris, France : Masson.

ROCHE C. (2007). Dire n'est pas concevoir. In 18es *Journées Francophones d'Ingénierie des Connaissances (IC'2007)*, p. 157–168, Toulouse, France : Cépaduès. ISBN 978-2-85428-773-8.

SCHREIBER G., WIELINGA B., AKKERMANS H., VAN DE VELDE W. & ANJEWIERDEN A. (1994). CML : The CommonKADS conceptual modelling language. In *A future for Knowledge Acquisition*, p. 1–25. Springer.

SMITH B. (2012). How to do Things with Documents. *Rivista di Estetica*.

STEWART I. & FRAÏSSÉ C. (2009). *La pensée sociale*, chapter Les schèmes cognitifs de base : un modèle pour étudier les représentations sociales, p. 99–119. Eres : Paris, France.

TUDORACHE T., NYULAS C., NOY N. & MUSEN. M. (2013). WebProtégé : A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic web*, **4**(1), 89–99.

VOGEL C. (1988). *Génie cognitif*. Paris, France : Masson (Sciences Cognitives).

WHORF B.-L., J.-B.CARROLL, LEVINSON S.-C. & LEE P. (2012). *Language, thought, and reality : Selected writings of Benjamin Lee Whorf*. Mit Press.

WIELINGA B., SCHREIBER A. & BREUKER J. (1992). KADS : A modelling approach to knowledge engineering. *Knowledge Acquisition*, **4**(1), 5–53.

Complémentarité de personnes partageant des propriétés dans les Réseaux Sociaux

Michel Plantié[1], Michel Crampes[1]

LABORATOIRE LGI2P, Ecole des Mines, Parc Georges Besse, 30035 Nîmes
michel.{plantie,crampes}@mines-ales.fr,
http:social-networks.mines-ales.fr

Résumé : Nos travaux précédents portaient sur la détection unifiée de communautés dans les réseaux de personnes représentés sous la forme de graphes généralement bipartis : réseaux sociaux, communautés d'acteurs, etc. Dans cet article nous tentons de répondre à des problèmes de complémentarité qui se posent dès que l'on souhaite associer des personnes dans le but de remplir au mieux un objectif. Nous définissons donc la notion de complémentarité entre les sommets d'un graphe biparti. Nous utilisons pour cela les notions d'entropie et d'information mutuelle. Nous montrons l'utilité d'une telle démarche et l'intérêt de l'approche par une expérimentation sur des exemples bien connus.

Mots-clés : Détection de Communautés, Communautés recouvrantes, Equilibre de Nash, Complémentarité

1 Introduction

Avec le développement d'Internet et l'importance prise par les réseaux sociaux, l'étude de ces derniers fait l'objet de nombreux travaux de recherche. Une des voies est la détection de communautés où le principe général consiste à regrouper les individus de telle manière que les liens qu'ils entretiennent avec ceux présents à l'intérieur de leur communauté soient plus nombreux ou plus forts que les liens qu'ils entretiennent avec ceux présents dans les autres communautés. Les champs d'application dépassent les réseaux sociaux puisqu'on en trouve aussi en biochimie, en phylogénie, dans l'analyse des écosystèmes, ou encore plus récemment en neurologie. Beaucoup d'algorithmes aux approches très variées ont été proposés. Les plus importants sont rapportés dans (Papadopoulos *et al.*, 2011; Yang *et al.*, 2010; Porter *et al.*, 2009) et de manière plus détaillée dans (Fortunato, 2009). En grande majorité ils portent sur la détection de communautés non recouvrantes (on parle aussi de partitionnées) dans des graphes unipartis. Dans ce type de graphes tous les noeuds peuvent éventuellement être reliés. Cependant de nombreuses relations sociales sont médiatisées par des propriétés communes entre individus. Il n'y a pas de relations directes entre individus ou entre propriétés mais directement des relations entre individus et propriétés. On parle alors de graphes multimodaux, le cas le plus fréquent étant le graphe bimodal (ou biparti).

Dans cet article nous répondons à des problèmes de complémentarité qui est la suite naturelle des problématiques suggérées par la décomposition en communautés. En effet une communauté est d'autant plus viable qu'elle obéit à des règles sémantiques et pragmatiques pilotées par les réalités quotidiennes des individus. Une règle possible et fréquemment appliquée dans la réalité est la complémentarité qui contribue à la stabilité d'un groupe. Si les individus sont en concurrence dans un groupe, alors ce groupe peut devenir instable. Par ailleurs un groupe dont les membres collaborent en fonction de leurs compétences variées à l'obtention d'un but commun est souvent recherché. C'est le cas pour toute entreprise, équipe sportive, etc. Il convient donc

de s'intéresser dans un premier temps à définir ce que nous entendons par complémentarité dans un groupe social. Nous laissons de côté pour l'instant la notion de communauté pour mieux approfondir théoriquement et expérimentalement la complémentarité d'un groupe qui peut être un réseau social ou une communauté plus réduite. Nous définissons la complémentarité puis cherchons à l'appliquer sur des réseaux sociaux. L'article est structuré ainsi : après un état de l'art sur la complémentarité, nous définissons les différentes notions nécessaires à l'établissement d'une mesure de complémentarité, puis nous définissons également une méthode pour obtenir une stabilité dans un réseau social en se basant sur la complémentarité. Enfin nous montrons des applications sur un réseau connu afin d'illustrer les concepts et les résultats obtenus.

2 État de l'art

Les travaux liés à la complémentarité ont été publiés principalement dans les domaines de l'économie, de l'innovation et du management mais aussi en mathématiques. Globalement le principe est que deux entités sont complémentaires si leur apport au système est plus grand quand ils sont présents ensemble dans le système que leur apport séparé. Dans ce contexte, (Cassiman & Veugelers, 2006) donnent une définition formelle de la complémentarité entre activités liées à l'innovation. Dans le domaine de l'innovation, elle est étudiée pour comprendre sous quelles conditions des activités liées à l'innovation peuvent être complémentaires. Il est donc primordial d'identifier des variables qui affectent la complémentarité. La définition formelle de la complémentarité de (Cassiman & Veugelers, 2006) entre activités complémentaires liées à l'innovation liste les variables qui l'impactent : force d'exportation, force d'innovation, information publique disponible, propriété intellectuelle, etc. Dans leur approche, la complémentarité est définie par rapport à une performance accrue (corrélation positive). Leur conception de la complémentarité est définie selon la formule suivante :

$$Complémentarité(a, b) : \Pi(1, 1) - \Pi(0, 1) > \Pi(1, 0) - \Pi(0, 0) \quad (1)$$

où Π est la fonction d'utilité ou de performance du système, a et b sont deux éléments du système. $\Pi(1, 0)$ signifie que a est présent et b absent. L'utilité mesure la différence entre Π lorsque a est là et Π lorsque a est absent dans les deux cas en présence de b par rapport aux mêmes situations lorsque b est absent.

La complémentarité d'information sur un sujet a été abordée dans la communauté Recherche d'Information (RI). L'approche proposée dans (Ma *et al.*, 2006) repose sur des mots-clés structurés. Les auteurs envisagent la recherche d'information complémentaire en particulier en croisant plusieurs média. Les mathématiques ont étudié abondamment la complémentarité et de nombreux travaux comme (Topkis, 1998) définissent la complémentarité comme une fonction mesurant l'effet combiné de deux acteurs simultanés sur un système. Dans la suite nous utiliserons cette notion de complémentarité de (Cassiman & Veugelers, 2006) en utilisant le concept d'entropie et d'information mutuelle que nous développerons plus loin.

La complémentarité entre les personnes est pressentie par les activités collaboratives et interactionnelles développées lors de participation à des réseaux et/ou à des projets. Elle est aussi visible dans la co-édition de documents où les apports de chacun des auteurs peuvent être spécifiés (voir figure 1).

La complémentarité d'information proposée dans (Ma *et al.*, 2006) définit une structure de graphe contenant des sujets et des contenus : topic structure. On peut cependant redouter qu'en

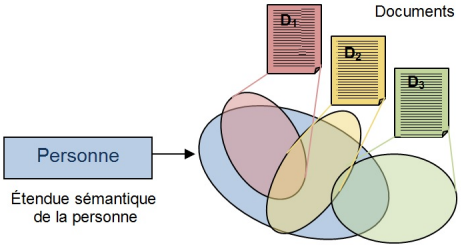


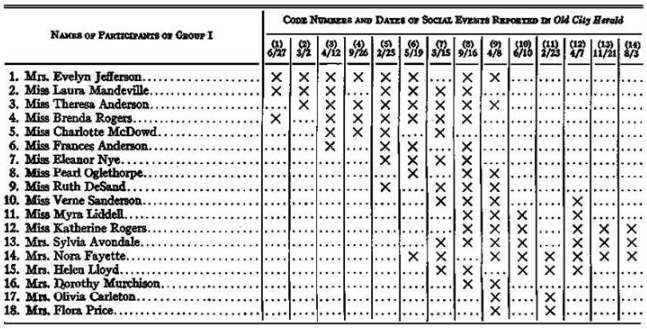
FIGURE 1 – Complémentarité de documents et de personnes.

se basant uniquement sur les termes, l'homonymie dégrade la pertinence des résultats. Les auteurs ambitionnent, dans les perspectives de (Ma *et al.*, 2006) d'utiliser un modèle sémantique (ontologie). Dans sa théorie des hypergraphes C Berge (Berge, 1987) évoque la notion de Transversal : un transversal d'un hypergraphe $H = (E_1, ..., E_m)$ sur un ensemble X est un ensemble $T \subseteq X$ qui a une intersection non vide avec chaque hyperarête $T \cap E_i \neq 0, (i = 1, 2, ..., m)$. Les transversaux d'un hypergraphe ne sont pas uniques et l'on peut également parler de transversal minimal. La notion de transversal se rapproche de la complémentarité, si l'on considère des graphes bipartis qui sont équivalents à des hypergraphes. La complémentarité dans ce type de structure consiste en particulier à trouver les éléments du premier sous ensemble de sommets du graphe qui couvrent le maximum de connexions avec les sommets du deuxième sous-ensemble de sommets. La complémentarité se rapproche donc de la problématique de la recherche du transversal minimum dans un hypergraphe. La recherche de transversaux minimum a fait l'objet de nombreux travaux (Kavvadias & Stavropoulos, 2005; Khachiyan *et al.*, 2005; Eiter & Gottlob, 1995). Dans notre cas la complémentarité bien que proche du problème de la recherche de transversaux minimum s'en distingue du fait que nous rajoutons un critère supplémentaire d'utilité. En effet nous souhaitons associer des éléments complémentaires au regard de leur capacité à couvrir l'ensemble des problématiques du système étudié mais aussi à maximiser une fonction d'utilité.

Dans la théorie des ensembles ce thème de la complémentarité a été abondamment étudié et le point de départ est le fait qu'un sous-ensemble S_a d'un ensemble S est complémentaire à un autre sous-ensemble S_b si il est constitué de la différence de l'ensemble S moins le sous-ensemble S_b. Cette définition fondamentale a ensuite été étendue et de nombreux travaux mathématiques comme (Topkis, 1998) cité précédemment élargissent la définition de la complémentarité pour se rapprocher de celle de Cassiman.

En logique floue, la notion de mesure floue ou capacité se rapproche de la notion de complémentarité que l'on peut interpréter come le poids de la coalition d'éléments. Si le poids de deux éléments est positif alors leur coalition a un apport positif pour le système. Le poids peut être une mesure floue (caractérisée uniquement par sa monotonie).

dans (Jelassi *et al.*, 2014) les auteurs ont abordé de façon différente la notion de complémentarité. Ils considèrent un graphe monoparti déjà décomposé en communautés non-recouvrantes. Ils introduisent alors la notion de multi-membre : un ensemble de sommets de taille minimum avec au moins un sommet dans chaque communauté et si possible avec plusieurs sommets dans les communautés de plus grande taille.



Names of Participants of Group I	Code Numbers and Dates of Social Events Reported in *Old City Herald*													
	(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/8	(10) 6/10	(11) 2/23	(12) 4/7	(13) 11/21	(14) 6/3
1. Mrs. Evelyn Jefferson	×	×	×	×	×	×	.	×	×	.	.	.	.	.
2. Miss Laura Mandeville	×	×	×	.	×	×	×	×	.	.	.	.	.	.
3. Miss Theresa Anderson	.	×	×	×	×	×	×	×	×	.	.	.	.	.
4. Miss Brenda Rogers	×	.	×	×	×	×	×	×	.	.	.	.	.	.
5. Miss Charlotte McDowd	.	.	×	×	×	.	×	.	.	.	.	.	.	.
6. Miss Frances Anderson	.	.	×	.	×	×	.	×	.	.	.	.	.	.
7. Miss Eleanor Nye	.	.	.	.	×	×	×	×	.	.	.	.	.	.
8. Miss Pearl Oglethorpe	.	.	.	.	.	×	.	×	×	.	.	.	.	.
9. Miss Ruth DeSand	.	.	.	.	×	.	×	×	×	.	.	.	.	.
10. Miss Verne Sanderson	.	.	.	.	.	.	×	×	×	.	.	×	.	.
11. Miss Myra Liddell	.	.	.	.	.	.	.	×	×	×	.	×	.	.
12. Miss Katherine Rogers	.	.	.	.	.	.	.	×	×	×	.	×	×	×
13. Mrs. Sylvia Avondale	.	.	.	.	.	.	×	×	×	×	.	×	×	×
14. Mrs. Nora Fayette	.	.	.	.	.	×	×	.	×	×	×	×	×	×
15. Mrs. Helen Lloyd	.	.	.	.	.	.	×	×	.	×	×	×	.	.
16. Mrs. Dorothy Murchison	.	.	.	.	.	.	.	×	×	.	.	.	.	.
17. Mrs. Olivia Carleton	.	.	.	.	.	.	.	.	×	.	×	.	.	.
18. Mrs. Flora Price	.	.	.	.	.	.	.	.	×	.	×	.	.	.

FIGURE 2 – Matrice des personnes du réseau WE

3 Définition et formules

3.1 Graphe biparti

Soit un réseau social de personnes partageant des propriétés. Nous pouvons considérer ce réseau comme un système S représenté par un graphe biparti $G = (Pe, Pr, E)$, avec Pe un sous-ensemble de n_1 personnes, Pr un sous-ensemble de n_2 propriétés, E un ensemble de m arêtes entre les personnes de Pe et les propriétés de Pr, ayant comme matrice d'adjacence : $A(i, j)$, i représentant l'indice des personnes Pe et j l'indice des propriétés Pr. Nous définissons un système S comme l'ensemble des propriétés Pr_j partagées par les personnes Pe_i. Une extension de ce type de graphe est d'ajouter des poids sur les relations entre personnes et propriétés qui représentent l'importance d'une propriété pour une personne. Bien que plus riche sémantiquement parlant, nous traiterons ce cas dans un article futur.

Un petit exemple de cette configuration est le graphe biparti intensément étudié (voir la meta-analyse de (Freeman, 2003)) "Southern Women" (SW) ou "'Women Events"' (WE). Il consiste en un relevé de la participation différenciée de 18 dames à 14 évènements sociaux (voir figure 2). Chaque colonne représente un événement de type "Tea party" et chaque ligne représente les dames y ayant participé. Les propriétés sont donc les événements représentés en colonnes.

3.2 Entropie d'un système

L'entropie d'un système représente le degré d'information que comporte ce système. Elle se calcule généralement comme suit : si un système comporte n éléments, l'entropie (voir (Yang & Petersen, 1997)) est la quantité : $H(S) = \frac{-\sum_{i=1}^{n} P(i) \times ln(P(i))}{ln(n)}$. Afin d'obtenir des propriétés intéressantes de la fonction d'entropie, le facteur de normalisation $ln(n)$ est utilisé. Dans ce cas l'entropie varie entre 0 et 1 (voir figure 3) et est maximum quand les probabilités des n éléments sont à 0,5 soit équirépartis dans le système.

La signification de l'entropie est donc qu'elle permet de représenter la quantité d'information fournie par le système, plus les éléments sont équirépartis dans le système, plus il faut d'information pour les représenter. Si chaque élément est par contre de probabilité soit 0 soit 1 c'est à dire parfaitement défini sans incertitude, alors le système a besoin de moins d'information pour le représenter, ce qui correspond à une valeur d'entropie plus faible. Dans le cas d'un réseau social représenté par un graphe biparti l'entropie d'un système se calcule sur l'ensemble des personnes partageant des propriétés. Chaque propriété a une probabilité d'apparition dans

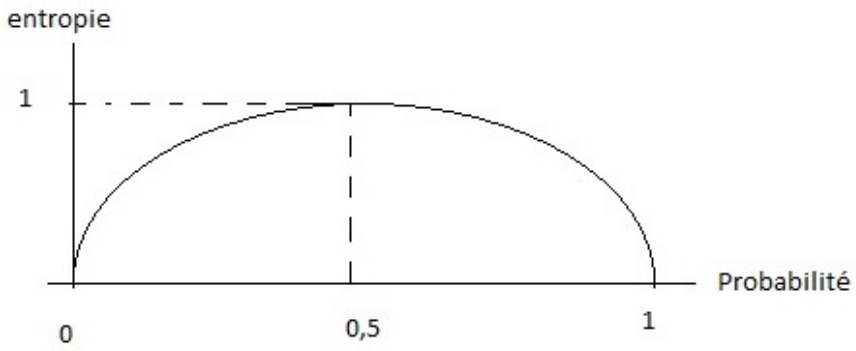


FIGURE 3 – courbe entropie : $-\sum n \times ln(n)$

le système, c'est à dire qu'elle est associée par une arête à une ou plusieurs personnes. Le système défini ici est donc : **un ensemble de personnes partageant des propriétés**. Dans notre exemple, une dame qui participera à un nombre élevé d'événements aura une entropie plus forte.

L'entropie se formule donc comme suit : $H(S) = \frac{-\sum_{j=1}^{n_2} P(Pr_j) \times ln(P(Pr_j))}{ln(n_2)}$, où la probabilité $P(Pr_j)$ est définie par $P(Pr_j) = \frac{\sum_{i=1}^{n_1} A(i,j)}{m}$. $P(Pr_j)$ est donc la probabilité d'apparition de la propriété Pr_j pour toutes les personnes du réseau dans le système S.

3.3 Entropie conditionnelle

L'entropie conditionnelle représente la quantité d'information associée à un élément du système. Dans le cas du réseau social, nous mesurons avec cette entropie la quantité d'information portée par une personne, en tenant compte de la probabilité de présence d'une personne :

$H(S/Pe_j) = \frac{-\sum_{i=1}^{n_2} P(Pr_i/Pe_j) \times ln(P(Pr_i/Pe_j))}{ln(n_2)}$

ou de non-présence d'une personne :

$H(S/\overline{Pe_j}) = \frac{-\sum_{i=1}^{n_2} P(Pr_i/\overline{Pe_j}) \times ln(P(Pr_i/\overline{Pe_j}))}{ln(n_2)}$

Dans notre exemple, c'est l'entropie de l'ensemble des événements connaissant la présence ou l'absence d'une dame aux événements.

3.4 Gain d'information

Le gain d'information toujours selon (Yang & Petersen, 1997) représente la quantité d'information spécifique différentielle apportée par un élément, dans notre cas une personne du réseau social. Selon (Yang & Petersen, 1997) c'est la différence des entropies du système et des entropies conditionnelles pour une personne Pe_j et son complémentaire $\overline{Pe_j}$:

$$G(Pe_j) = H(S) - P(Pe_j) \times H(S/Pe_j) - P(\overline{Pe_j}) \times H(S/\overline{Pe_j}) \quad (2)$$

Si un système doté d'une personne a une entropie plus faible que le système tout seul, c'est le signe que la personne en question diminue l'entropie, et donc apporte de l'information. Plus les entropies conditionnelles de la personne Pe_j sont faibles plus le gain d'information est

élevé, signifiant le fait que la personne Pe_j contribue fortement à l'apport d'information pour le système ou dit autrement fait baisser l'incertitude du système.

Dans notre exemple, c'est le gain apporté par la connaissance de la participation d'une dame aux événements.

4 Complémentarité

4.1 Notre définition de la complémentarité

Une problématique souvent évoquée concerne l'utilité des individus dans un groupe. Une des questions se pose alors : les personnes formant un groupe ont-elles un intérêt à être associées au même groupe. Un critère de cet intérêt est la capacité des individus à être complémentaires au sein d'un groupe au sens des propriétés partagées. Dans notre étude, les graphes bipartis sont très bien adaptés pour répondre à ce type de question. La complémentarité a pour but de trouver des éléments complémentaires dans un système. Le principe directeur dans notre cas est donc de trouver les éléments d'un graphe biparti ou réseau social (les lignes de la matrice d'incidence) qui couvre l'ensemble des propriétés (les colonnes de la matrice d'incidence) avec le minimum d'éléments de ce graphe

Dans notre exemple, c'est la mesure d'utilité apportée par la présence simultanée de deux dames aux "tea parties" comparativement à leur présence individuelle.

Dans un graphe biparti, les personnes partagent des propriétés. Ces propriétés sont par exemple : l'appartenance à une photo, la coécriture d'un article, l'appartenance à une thématique, un concept associé à une personne, un "<j'aime"> de facebook, un "<suivre"> sur twitter, etc. Ainsi dans le cas de concepts ou thématiques partagés par exemple, il est intéressant de connaître le nombre minimum de personnes couvrant l'ensemble des thématiques.

Nous adoptons la notion de complémentarité de Cassiman qui permet mieux que les autres visions de la complémentarité de représenter l'apport supplémentaire de l'association de personnes, ce qui est l'essence même d'un réseau social. en y incorporant le concept d'entropie et d'information mutuelle. La notion de multi-membre telle qu'évoquée dans l'état de l'art ne prend pas en compte la notion d'utilité. La complémentarité entre deux personnes, conformément à la formule 1, est donc basée sur le fait suivant : la présence simultanée de deux personnes apporte une plus grande utilité ("fonction" d'utilité Π) au système que chacun des deux éléments séparément.

La fonction d'utilité Π, c.f. formule 1, peut être définie de différentes façons, et dans notre cas nous utiliserons le concept d'entropie et d'information mutuelle.

4.2 Le gain d'information comme fonction d'utilité

Nous utiliserons la mesure d'entropie et ses dérivés pour représenter l'utilité car c'est la mesure d'information apportée par un élément dans un ensemble de classe d'éléments (Yang & Petersen, 1997), et elle est très pertinente dans le contexte du partage d'information et de propriétés que porte un graphe biparti. Dans notre cas nous pouvons considérer la fonction d'utilité comme le gain d'information défini plus haut. Plus le gain d'information apporté par deux éléments au système est important, plus ces deux éléments en question sont utiles ensemble pour

le système. La complémentarité se traduit donc par :

$$complémentarité(Pe_i, Pe_j) = G(Pe_i, Pe_j) - G(\overline{Pe_i}, Pe_j) - (G(Pe_i, \overline{Pe_j}) - G(\overline{Pe_i}, \overline{Pe_j})), \quad (3)$$

si l'on développe cette formule cela donne :

$$\begin{aligned}
&complémentarité(Pe_i, Pe_j) = \\
&\tfrac{1}{ln(n_2)} \times [H(S) - P(Pe_i \wedge Pe_j) \times H(S/Pe_i \wedge Pe_j) - P(\overline{Pe_i \wedge Pe_j}) \times H(S/\overline{Pe_i \wedge Pe_j}) \\
&-H(S) + P(\overline{Pe_i} \wedge Pe_j) \times H(S/\overline{Pe_i} \wedge Pe_j) + P(\overline{\overline{Pe_i} \wedge Pe_j}) \times H(S/\overline{\overline{Pe_i} \wedge Pe_j}) - H(S) \\
&+P(Pe_i \wedge \overline{Pe_j}) \times H(S/Pe_i \wedge \overline{Pe_j}) + P(\overline{Pe_i \wedge \overline{Pe_j}}) \times H(S/\overline{Pe_i \wedge \overline{Pe_j}}) \\
&+H(S) - P(\overline{Pe_i} \wedge \overline{Pe_j}) \times H(S/\overline{Pe_i} \wedge \overline{Pe_j}) - P(\overline{\overline{Pe_i} \wedge \overline{Pe_j}}) \times H(S/\overline{\overline{Pe_i} \wedge \overline{Pe_j}})]
\end{aligned}$$

soit après simplification :

$$\begin{aligned}
&complémentarité(Pe_i, Pe_j) = \tfrac{1}{ln(n_2)} \times [-P(Pe_i \wedge Pe_j) \times H(S/Pe_i \wedge Pe_j) - P(\overline{Pe_i} \vee \overline{Pe_j}) \times \\
&H(S/\overline{Pe_i} \vee \overline{Pe_j}) + P(\overline{Pe_i} \wedge Pe_j) \times H(S/\overline{Pe_i} \wedge Pe_j) + P(Pe_i \vee \overline{Pe_j}) \times H(S/Pe_i \vee \overline{Pe_j}) \\
&+P(Pe_i \wedge \overline{Pe_j}) \times H(S/Pe_i \wedge \overline{Pe_j}) + P(\overline{Pe_i} \vee Pe_j) \times \\
&H(S/\overline{Pe_i} \vee Pe_j) - P(\overline{Pe_i \vee Pe_j}) \times H(S/\overline{Pe_i \vee Pe_j}) - P(Pe_i \vee Pe_j) \times H(S/Pe_i \vee Pe_j)]
\end{aligned}$$

ou sous une autre forme :

$$\begin{aligned}
&complémentarité(Pe_i, Pe_j) = \\
&\tfrac{1}{ln(n_2)} \times [-P(Pe_i \wedge Pe_j) \times H(S/Pe_i \wedge Pe_j) - P(\overline{Pe_i \wedge Pe_j}) \times \\
&H(S/\overline{Pe_i \wedge Pe_j}) + P(\overline{Pe_i} \wedge Pe_j) \times H(S/\overline{Pe_i} \wedge Pe_j) + P(\overline{\overline{Pe_i} \wedge Pe_j}) \times H(S/\overline{\overline{Pe_i} \wedge Pe_j}) \\
&+P(Pe_i \wedge \overline{Pe_j}) \times H(S/Pe_i \wedge \overline{Pe_j}) + P(\overline{Pe_i \wedge \overline{Pe_j}}) \times \\
&H(S/\overline{Pe_i \wedge \overline{Pe_j}}) - P(\overline{Pe_i} \wedge \overline{Pe_j}) \times H(S/\overline{Pe_i} \wedge \overline{Pe_j}) - P(\overline{\overline{Pe_i} \wedge \overline{Pe_j}}) \times H(S/\overline{\overline{Pe_i} \wedge \overline{Pe_j}})]
\end{aligned}$$

le but sera donc d'évaluer si cette complémentarité est positive, qui signifie que les deux éléments Pe_i et Pe_j ont intérêt à être associés dans le système.

4.3 Algorithme de la complémentarité

Ainsi nous cherchons à trouver l'ensemble minimum des personnes qui couvrent au mieux l'ensemble des propriétés du système. Pour cela, nous effectuons plusieurs étapes :

1. Calculer l'entropie du système
2. Chercher la personne Pe_{max} ayant le maximum de gain d'information (formule 2). Si plusieurs sommets présentent la même valeur maximum, alors prendre la personne offrant le maximum de couverture des événements
3. Chercher la personne Pe_{comp_max} ayant la complémentarité la plus forte avec Pe_{max} (formule 3)
4. **condition d'arrêt** : est-ce que toutes les propriétés ont été couvertes au moins une fois ?
   si oui : arrêt,
   si non : continuer au point suivant

personnes	Gain	personnes	Gain
1	**0,2590131**	10	0,09085268
2	0,21040042	11	0,09085268
3	**0,2590131**	12	0,16715123
4	0,21040042	13	0,21040042
5	0,09085268	14	**0,2590131**
6	0,09085268	15	0,12768425
7	0,09085268	16	0,02141921
8	0,05572812	17	0,02141921
9	0,09085268	18	0,02141921

FIGURE 4 – Mesures du gain d'information pour les personnes du réseau WE

5. Chercher la personne ayant la complémentarité la plus forte avec Pe_{comp_max} et Pe_{max} qui s'exprime ainsi :
   $complémentarité(Pe_i, Pe_{comp\_max} \wedge Pe_{max}) = G(Pe_i, Pe_{comp\_max} \wedge Pe_{max})$
   $-G(\overline{Pe_i}, Pe_{comp\_max} \wedge Pe_{max}) - (G(Pe_i, \overline{Pe_j}) + G(\overline{Pe_i}, \overline{Pe_{comp\_max} \wedge Pe_{max}}))$,
6. continuer sur tous les sommets jusqu'à ce que la complémentarité de tous les sommets du graphe soit négative
7. si l'ensemble des sommets a été parcouru et toutes les complémentarités sont positives, s'arrêter sinon reprendre au 4.

l'ensemble des sommets ainsi identifiés constitue le groupe de sommets présentant une complémentarité maximale pour l'ensemble du graphe. Dans le cas où l'on s'arrête avant d'avoir couvert l'ensemble complet des personnes, la couverture (connexion avec toutes les propriétés) est totale. Dans le second cas la couverture est partielle.

4.4 Complexité du calcul de complémentarité

La complexité est de $O(m \times n^2)$, n étant le nombre de personnes, m étant le nombre de propriétés. En effet chaque calcul d'entropie s'effectue sur l'ensemble des propriétés, et le calcul de complémentarité requiert au maximum $\frac{n \times (n-1)}{2}$ opérations.

5 Expérimentation

Nous utilisons un graphe biparti intensément étudié (voir la meta-analyse de (Freeman, 2003)) "Southern Women" (SW) ou "'Women Events"' (WE). Il consiste en un relevé de la participation différenciée de 18 dames à 14 évènements sociaux (voir figure 2).

5.1 Mesure d'information mutuelle et de complémentarité sur le graphe biparti WE

Le premier calcul est selon les étapes 1 et 2 de la liste du paragraphe 4.3 celui du gain d'information de chaque sommet ou personne selon la formule 2. la figure 4 montre les valeurs pour chaque sommet. Ce calcul donne trois sommets avec une entropie maximum, nous choisissons le premier de ces sommets, le sommet 1, qui offre une couverture maximale des évènements.

Ensuite selon toujours la liste du paragraphe 4.3 nous calculons la complémentarité de chaque sommet/personne avec le sommet 1 que nous montrons dans la figure 5.1. Ce calcul

personnes	complémentarité	personnes	complémentarité
		10	0,032024786
2	-0,241284557	11	0,095461542
3	-0,249998904	12	0,198456905
4	-0,175434381	13	**0,267897921**
5	-0,126538143	14	0,206555642
6	-0,172270819	15	0,090862253
7	-0,113088838	16	-0,092933968
8	-0,146138666	17	0,016246178
9	-0,051317974	18	0,162265216

personnes	complémentarité	personnes	complémentarité
		10	-0,205307041
2	-0,056074702	11	-0,171822694
3	-0,126066811	12	-0,153876567
4	-0,074081708	13	
5	-0,011852276	14	**0,187685538**
6	-0,094459499	15	-0,142052887
7	-0,137395416	16	-0,202092756
8	-0,207179604	17	-0,098646124
9	-0,205307041	18	0,166238366

FIGURE 5 – Mesures de complémentarité avec la personne 1 puis 1 et 13 du réseau WE

donne le sommet 13 comportant la complémentarité maximum avec le sommet 1. On constate à ce point que la majorité des sommets comportent une complémentarité négative, ce qui signifie que l'on a atteint un quasi équilibre en choisissant les sommets 1 et 13 qui représentent donc la grande majorité des sommets du graphe en terme de représentativité.

Puis selon toujours la liste du paragraphe 4.3 nous calculons la complémentarité de chaque sommet/personne avec les sommet 1 et 13 et que nous montrons dans la figure 5.1. Ce calcul donne le sommet 14 comportant la complémentarité maximum avec les sommets 1 et 13. On constate à ce point que tous les événements du graphe sont couverts, c'est à dire la condition d'arrêt 4 de l'algorithme est vérifiée. Nous obtenons donc un premier résultat, la couverture complète par trois personnes sur le graphe des Women Events. Les trois personnes sont considérées comme un des meilleurs arrangements de couverture complémentaire pour ce graphe.

5.2 Mesure de complémentarité sur d'autres jeux de données plus importants

Nous avons expérimenté notre méthode sur le graphe du club de karaté Zachary (1977) un jeu de données bien connu dans l'analyse de graphes. Ce graphe représente les relations d'affinités entre les personnes d'un club de karaté. C'est un graphe monoparti, cependant nous pouvons le considérer comme un graphe biparti, dans lequel les personnes partages des propriétés (les propriétés étant dans ce cas les personnes liées par des liens d'amitié). Nous obtenons des résultats similaires au graphes des Women Events. La couverture totale est obtenue avec 25 personnes sur 34.

Nous avons appliqué les mêmes procédures à un autre jeu de données plus conséquent : un graphe biparti de partage de 700 photos entre environ 274 personnes provenant d'un compte Facebook. Les photos jouent ici le rôle de propriétés partagées et les personnes le même rôle que dans le graphe Women Events. C'est un graphe biparti. Sur ce jeu de données le nombre d'itérations nécessaires pour obtenir la couverture complète est égal au nombre de personnes. Après analyse du jeu de données, on constate qu'aucune combinaison des personnes inférieure au nombre total ne permet d'obtenir une couverture complète. Les photos considérées correspondent à des périodes différentes dans le temps et donc les personnes appartiennent souvent à des groupes disjoints. En effet le jeu de photos est effectué avec des groupes de personnes pratiquement disjoints correspondant aux différentes périodes de vie du propriétaire du compte facebook. Nous pouvons en déduire que pour obtenir une complémentarité avec un nombre de personnes réduit, les personnes doivent partager un minimum les propriétés considérées. Ce qui n'est pas le cas dans ce jeu de données. Notre méthode met donc en lumière sur ce jeu de données la disparité des personnes et leur difficulté à devenir complémentaires.

Nota : ces travaux ont été réalisés grâce au concours d'un groupe de trois élèves de niveau Master de l'Ecole des Mines d'Alès.

6 Conclusion

Dans les réseaux sociaux trouver des groupes de personnes stables, peut s'avérer intéressant pour différents objectifs. Après avoir exploré les différentes voies, nous utilisons la notion de complémentarité pour exprimer cette stabilité. Après avoir défini la notion de complémentarité, nous montrons comment utiliser la notion d'Information Mutuelle pour exprimer cette complémentarité. Nous montrons à travers plusieurs jeux de données que cette mesure est prometteuse pour évaluer la stabilité d'un ensemble de personnes socialement complémentaires.

Références

BERGE C. (1987). Hypergraphes, Combinatoires des ensembles finis. *Gauthier-Villars*.

CASSIMAN B. & VEUGELERS R. (2006). In Search of Complementarity in Innovation Strategy : Internal R&D and External Knowledge Acquisition. *Management Science*, **52**(1), 68–82.

EITER T. & GOTTLOB G. (1995). Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, **24**, 1278–1304.

FORTUNATO S. (2009). Community detection in graphs. *Physics Reports*, **486**(3-5), 103.

FREEMAN L. C. (2003). Finding social groups : A meta-analysis of the southern women data. In *Dynamic Social Network Modeling and Analysis. The National Academies*, p. 39—-97. Press.

JELASSI M. N., LARGERON C. & YAHIA S. B. (2014). Efficient unveiling of multi-members in a social network. *Journal of Systems and Software*, **94**(0), 30 – 38.

KAVVADIAS J. & STAVROPOULOS E. (2005). An efficient algorithm for the transversal hypergraph generation. *Journal of Graph Algorithms and Applications*, **9**, 239–264.

KHACHIYAN L., BOROS E., ELBASSIONI K. & GURVICH V. (2005). Computing and combinatorics. *Lecture Notes in Computer Science*, **3595**, 767–776.

MA Q., NADAMOTO A. & TANAKA K. (2006). Complementary information retrieval for cross-media news content. *Inf. Syst.*, **31**(7), 659–678.

PAPADOPOULOS S., KOMPATSIARIS Y., VAKALI A. & SPYRIDONOS P. (2011). Community detection in Social Media. *Data Mining and Knowledge Discovery*, **1**(June), 1–40.

PORTER M. A., ONNELA J.-P. & MUCHA P. J. (2009). Communities in Networks.

TOPKIS D. (1998). *Supermodularity and Complementarity*. Frontiers of Economic Research. Princeton University Press.

YANG B., LIU D., LIU J. & FURHT B. (2010). *Discovering communities from Social Networks : Methodologies and Applications*. Boston, MA : Springer US.

YANG Y. & PETERSEN J. O. (1997). A comparative Study on Feature Selection in Text Categorisation. In *Fourteenth International Conference on Machine Learning, ICML'97*.

ZACHARY W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, **33**(4), 452–473.

Quelques enseignements tirés de l'application de la Competence-based Knowledge Space Theory aux Serious Games

Naïma El-Kechaï[1], Javier Melero[1], Jean-Marc Labat[1]

[1] LIP6, Université Pierre et Marie Curie, 75005 Paris,
{naima.el-kechai, javier.melero-gallardo, jean-marc.labat}@lip6.fr

Résumé : L'approche fondée sur la Competence-based Knowledge Space Theory (CbKST) est une approche prometteuse qui permet d'adapter le parcours d'apprentissage d'un apprenant dans un Serious Game (SG) en fonction de l'état courant (supposé) de ses compétences. Elle s'appuie sur la description du modèle du domaine ciblé par le SG pour pouvoir générer la *competence structure*. Néanmoins, définir un tel modèle est une tâche coûteuse pour les experts en termes d'efforts et de temps. Une alternative permettant de surmonter ce problème est d'inférer le modèle du domaine implicite ciblé par le SG en considérant la Q-Matrice. Dans cet article, nous relatons les principaux résultats de l'application de CbKST dans trois SG en considérant (a) le modèle du domaine défini par l'expert et (b) le modèle du domaine inféré par la Q-Matrice. De ces résultats, nous tirons des enseignements fort intéressants quant à l'application de ces deux approches pour l'adaptation des SG.

Mots-clés : Competence-based Knowledge Space Theory, Adaptation de serious games, Modèle de domaine, Q-Matrice, Formalismes de représentation de connaissances, Traitements et raisonnements sur les connaissances, Évaluation de modèle de connaissances.

1 Introduction

Depuis plusieurs années, les Serious Games (SG) se démocratisent et suscitent un engouement certain de la part des enseignants, des chercheurs et des entreprises ; en témoigne la multitude des travaux, colloques et rencontres dédiés à ce thème. Les SG sont des applications informatiques qui utilisent des ressorts ludiques tels que les challenges, la compétition, les récompenses pour catalyser la curiosité et l'attention des apprenants et ainsi faciliter leur apprentissage (Dondlinger, 2007). À l'instar des autres environnements d'apprentissage, l'adaptation des SG est considérée comme une question clé tant des disparités existent entre les apprenants en termes de connaissances, compétences, préférences, motivation, etc. Pour ce faire, il est nécessaire de réaliser le suivi de l'apprenant : collecter les informations le concernant et les analyser ; construire et mettre à jour son modèle. En fonction de ce dernier et selon l'objectif recherché, l'adaptation consiste alors à proposer à l'apprenant des activités qui correspondent à l'état actuel de ses connaissances, compétences ou préférences (Shute & Zapata-Rivera, 2012).

L'adaptation dans les SG est basée sur des règles qui consistent à suggérer des activités (niveaux, études de cas, etc.) avec suffisamment de challenge sans pour autant frustrer l'apprenant par la complexité de la tâche à réaliser (Göbel et al., 2010). L'idée est de le garder dans le flow (Csíkszentmihályi, 1991), c'est-à-dire complètement immergé dans l'expérience de jeu à travers le maintien d'un juste équilibre entre le plaisir de jouer et le challenge constitué par l'acquisition de connaissances et compétences.

Une des approches prometteuses permettant de réaliser une telle adaptation est fondée sur la Competence-based Knowledge Space Theory (CbKST) (Augustin et al., 2013 ; Heller et al., 2005 ; Kickmeier-Rust et al., 2008 ; Kopeinik et al., 2012 ; Peirce et al., 2008). CbKST

permet de structurer un domaine d'apprentissage en un ensemble de compétences que l'on cherche à faire acquérir à l'apprenant. Ces compétences sont reliées entre elles par des relations de précédence. Dans ces travaux, une évaluation non invasive des compétences de l'apprenant est réalisée sans interrompre le flux dans le jeu (Kopeinik et al., 2012). Autrement dit, le niveau courant (supposé) des compétences de l'apprenant est inféré en fonction des actions qu'il réalise dans le jeu. La prochaine activité est alors proposée en fonction de ce niveau courant, en tenant compte de la relation de précédence qui existe entre les compétences.

Pour utiliser CbKST il est nécessaire de disposer du modèle du domaine, c'est-à-dire le modèle de compétences, qui sous-tend le SG afin de garantir une cohérence dans les activités proposées et donc dans les parcours d'apprentissages proposés à l'apprenant. Néanmoins, la construction ou la définition d'un tel modèle est une tâche complexe et coûteuse en termes d'efforts et de temps. Cette complexité est notamment due à la difficulté à définir les dépendances qui existent entre les compétences d'un domaine donné (Falmagne et al., 2006). Dans cet article, nous proposons une autre approche, moins complexe, pour inférer les liens qui existent entre les compétences. Il s'agit d'utiliser la Q-Matrice (Tatsuoka, 1983) qui représente l'indexation des niveaux du SG par les compétences qu'ils visent.

Dans cet article, nous analysons les problèmes qui apparaissent quand nous appliquons CbKST pour créer des parcours d'apprentissage. Plus précisément, nous présentons les principaux résultats obtenus quand nous appliquons les deux approches pour générer la *competence structure* : l'une fondée sur le modèle du domaine défini par un expert ; et l'autre fondée sur la Q-Matrice. L'objectif principal est d'analyser ces deux approches en mettant en exergue les forces et les faiblesses de chacune d'elles.

Nous précisons que dans ce travail exploratoire, bien que notre approche porte sur les SG, nous considérons de manière prioritaire les compétences pédagogiques dans la génération des structures de compétences.

Dans la section 2, nous décrivons les principaux concepts de *CbKST*. Dans la section 3, nous décrivons les deux approches proposées pour générer la *competence structure.* Dans la section 4, nous appliquons ces approches sur trois SG différents. Dans la section 5, nous présentons les principaux résultats. Nous terminons enfin par une discussion en indiquant les travaux en cours et les orientations futures de notre travail.

2 La Competence-based Knowledge Space Theory (CbKST)

La Competence-based Knowledge Space Theory (CbKST) (Augustin et al., 2013 ; Heller et al., 2005) est une extension, orientée compétences, de la Knowledge Space Theory (KST) (Doignon & Falmagne, 1985, 1999 ; Falmagne et al., 2006). CbKST permet de structurer un domaine de compétences en utilisant trois concepts clés : la *relation de précédence*, le *competence state*, et la *competence structure*.

Une ***relation de précédence*** (« a » ≤ « b ») indique que la compétence « a » est un prérequis pour acquérir la compétence « b ». Inversement, si l'apprenant maitrise la compétence « b », cela suppose qu'il maitrise également la compétence « a ». Ces relations de précédence peuvent être représentées par un diagramme de Hasse comme illustré par la figure 1.

Considérant les *relations de précédence* qui existent entre les différentes compétences, les ***competence state (CS)*** sont dérivés. Ils représentent différentes combinaisons possibles et admissibles de compétences simples. Toutes les combinaisons ne sont pas admissibles. Par exemple, compte tenu de la *relation de précédence* qui existe entre les compétences de la figure 1, {a, c} ne peut pas être considéré comme un *CS* admissible car pour travailler la compétence « c », il est nécessaire de travailler préalablement la compétence « b ».

La ***competence structure*** représente l'ensemble des *CS* admissibles en tenant compte de la *relation de précédence* dans un domaine donné. Par exemple, la partie droite de la figure 1 représente la *competence structure* déduite du domaine illustré dans la partie gauche. Le *CS* le plus bas représente l'état naïf (l'apprenant ne maitrise aucune compétence) et le CS le plus

haut représente l'ensemble des compétences du domaine (l'apprenant maitrise toutes les compétences).

Un chemin ou un parcours d'apprentissage représente alors un chemin possible dans la *competence structure*, c'est-à-dire entre les *cs*. L'adaptation des parcours d'apprentissage consiste alors à proposer à l'apprenant une activité dont les compétences se trouvent dans le prochain *cs* en fonction du *cs* courant dans lequel il se trouve.

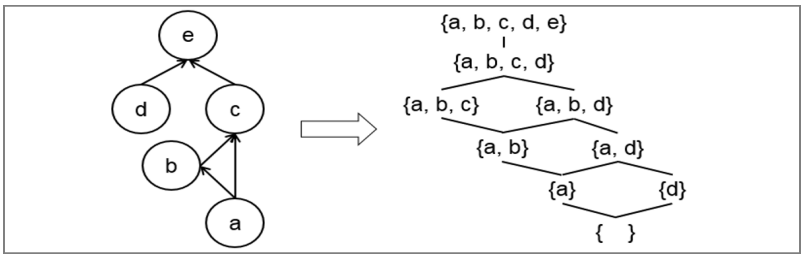


FIGURE 1 – Exemple de diagramme de Hasse illustrant les *relations de précédence* entre compétences d'un domaine donné (partie gauche) et la *competence structure* correspondante (partie droite).

3 Deux approches pour générer la *competence structure*

CbKST est considérée comme une approche pertinente pour adapter des environnements d'apprentissage, dont des SG (Augustin et al., 2013 ; Kickmeier-Rust et al., 2008 ; Kopeinik et al., 2012 ; Peirce et al., 2008). Cependant, définir les dépendances (liens) entre les compétences du domaine d'apprentissage ciblé est la tâche la plus coûteuse en termes d'efforts et de temps (Falmagne et al., 2006).

Afin de surmonter ce problème, nous proposons une autre approche qui permet d'inférer le modèle du domaine qui est implicite dans le SG. En effet, les niveaux d'un SG peuvent être décrits par les objectifs pédagogiques ou les compétences qu'ils permettent de travailler. Pour inférer le modèle du domaine implicite ciblé par le SG, nous proposons de nous appuyer sur une telle description, la Q-Matrice (Tatsuoka, 1983) qui représente l'indexation des niveaux par les compétences qu'ils visent. Ainsi, nous pourrons déduire les différents *CS*.

3.1 Générer la *competence structure* en utilisant le modèle du domaine

La première approche consiste à générer la *competence structure* d'un SG en utilisant le modèle du domaine d'apprentissage ciblé par le SG tel qu'il est défini par un expert.

Avant d'aller plus loin, nous précisons que le travail que nous présentons ici fait partie d'un travail plus étendu réalisé dans le cadre d'un projet FUI (fonds unique interministériel) Play Serious (« Play Serious Project », 2013). La finalité de ce projet est de développer des outils pour faciliter la conception et la réalisation de SG. Dans ce contexte, nous avons proposé une modélisation du domaine qui s'appuie sur trois liens qui sont définis ci-après :

1 Le lien de « précédence » entre les compétences « a » et « b » indique qu'il est conseillé ou préférable d'acquérir la compétence « a » avant d'acquérir la compétence « b »,

2 Le lien de « prérequis » entre les compétences « a » et « b » indique que la compétence « a » est indispensable pour acquérir la compétence « b »,

3 Le lien de « composition » indique qu'une compétence est composée de deux ou plusieurs sous-compétences de niveau de granularité inférieur.

Pour générer la *competence structure* en considérant ces trois liens, nous procédons ainsi :

- les liens « précédence » et « prérequis » sont considérés comme des *relations de précédence* (au sens de CbKST). Pour le lien de « composition », nous considérons uniquement les sous-compétences puisqu'elles correspondent à des compétences opérationnelles et correspondent donc davantage aux tâches à réaliser dans un niveau du SG. Concrètement, nous convertissons les liens de « composition » comme des *relations de précédence* comme illustré dans la figure 2. Si la compétence « a » est composée des compétences « b » et « c ». Alors le lien de précédence entre « a » et « d » est transformé par deux *relations de précédence* : une de « b » vers « d » et l'autre de « c » vers « d ».

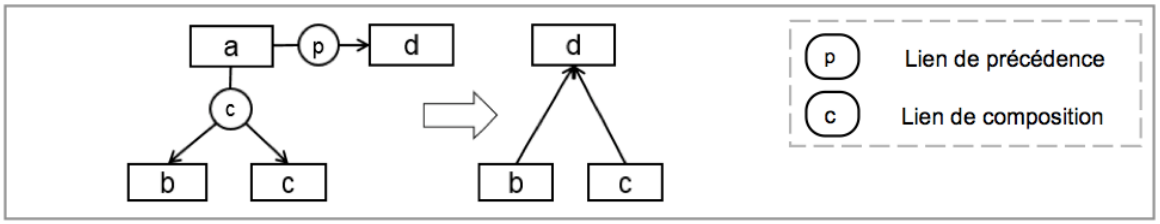


FIGURE 2 – Exemple de transformation d'un extrait d'un modèle de domaine en *relations de précédence*.

- De la nouvelle version du domaine dans lequel il n'y a que des *relations de précédence*, nous dérivons tous les *CS* en appliquant l'approche classique de CbKST.
- Enfin, chaque niveau du SG est relié au *CS* qui lui correspond. Autrement dit, l'ensemble des compétences visées dans un niveau du jeu forme un *CS* donné.

3.2 Générer la *competence structure* en utilisant la Q-Matrice

Dans les cas où le modèle du domaine d'apprentissage n'est pas disponible (le domaine est difficile à modéliser, il n'y a pas d'expert pour le faire, etc.), alors nous proposons de nous appuyer sur la Q-Matrice. Pour cela, il est nécessaire de disposer des informations sur la liste des compétences qui sont visées par chaque niveau du SG. De cette Q-Matrice, nous procédons ainsi :

- D'abord, nous identifions les différents *CS* en considérant que pour un niveau donné, le *CS* correspondant est celui formé par l'ensemble des compétences qu'il vise ;
- Ensuite, nous créons des liens entre les différents *CS* obtenus pour générer la *competence structure*. Il s'agit de relier chaque *CS* à son *CS successeur*. Autrement dit, un *CS* qui contient n éléments est relié à un autre *CS* qui devient son successeur si ce dernier contient exactement les mêmes n éléments plus un élément supplémentaire.

Dans la section qui suit, nous appliquons les deux approches sur trois SG différents.

4 L'analyse de l'application de CbKST sur trois SG

4.1 Blockly : Maze

Blockly : Maze[1] est un SG développé par Google, destiné à l'apprentissage de la programmation par briques (type Scratch). Il est composé de 10 niveaux. Le but de l'apprenant est de programmer la trajectoire d'un avatar automatique dans un labyrinthe au moyen de blocs d'instructions de type et de nombre éventuellement limités. Aucun modèle de

[1] https://blockly-demo.appspot.com/static/apps/maze/index.html

domaine explicite ni de Q-Matrice n'est disponible pour ce SG. En tant qu'enseignants en informatique, nous avons analysé le SG et nous proposons un modèle de domaine en nous appuyant sur une liste d'objectifs d'apprentissage des fondamentaux de la programmation proposée par IEEE et ACM (IEEE & ACM, 2001). La figure 3 illustre le modèle du domaine que nous proposons.

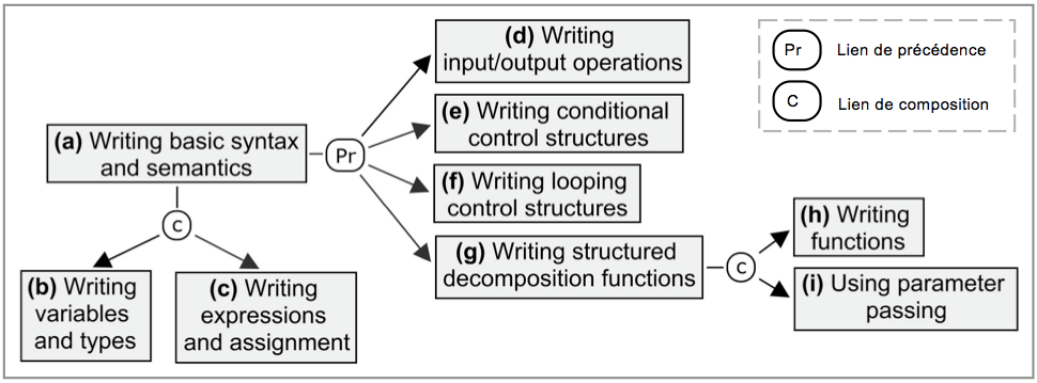


FIGURE 3 – Le modèle de domaine pour le SG *Blockly : Maze*.

Nous avons également indexé chaque niveau de *Blockly : Maze* par les compétences qui sont mises en œuvre en regardant les actions nécessaires à sa résolution et aux blocs de programmation disponibles dans ce niveau. Le tableau 1 illustre la Q-Matrice résultante (la croix indique que la compétence est travaillée dans le niveau considéré).

Après cela, nous avons comparé les *competence structure* générées par les deux approches : celle en partant du modèle de domaine (figure 4) générée comme expliqué dans la section 3.1 ; et celle en partant de la Q-Matrice (figure 5).

TABLE 1 – *La Q-Matrice de Blockly : Maze.*

	[a]		[e]	[f]	CS (*competence states*)
	[b]	[c]			
Niveau 1		x			[c]
Niveau 2	x	x			[b, c]
Niveau 3	x	x		x	[b, c, e]
Niveau 4	x	x		x	[b, c, e]
Niveau 5	x	x		x	[b, c, e]
Niveau 6	x	x	x	x	[b, c, d, e]
Niveau 7	x	x	x	x	[b, c, d, e]
Niveau 8	x	x	x	x	[b, c, d, e]
Niveau 9	x	x	x	x	[b, c, d, e]
Niveau 10	x	x	x	x	[b, c, d, e]

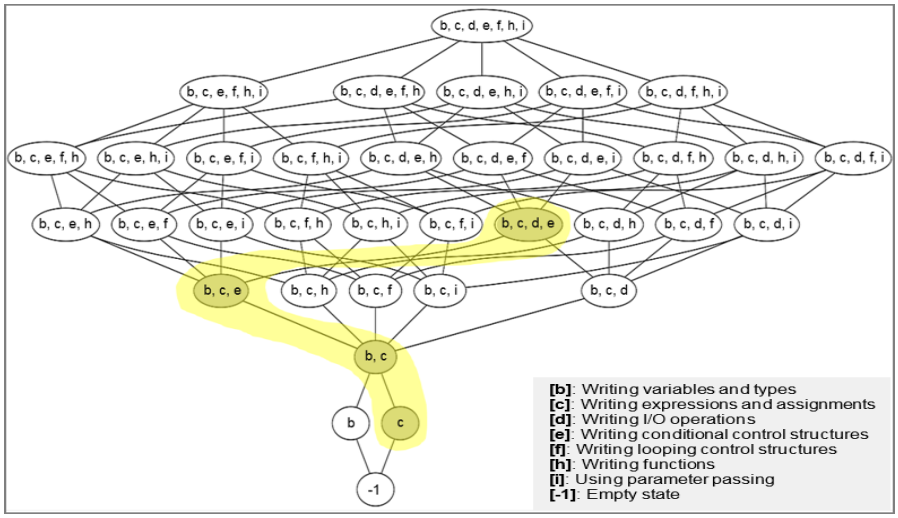


FIGURE 4 – La *competence structure* de *Blockly : Maze* générée par le modèle de domaine.

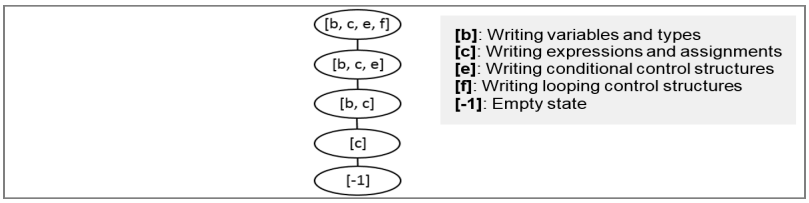


FIGURE 5 – La *competence structure* de *Blockly : Maze* générée par la Q-Matrice.

Avant de décrire le prochain SG, nous mettons en évidence quelques observations au regard des *competence structure* générées :

- Observation 1 (Ob. 1.1) : dans la *competence structure* générée par le modèle de domaine défini par un expert, le nombre de *CS* ayant des niveaux correspondants est très faible, uniquement 4 *CS* sur 35- [c] [b, c] [b, c, e] [b,c,d,e]- (représentés en gris sur la figure 4) ;
- Observation 2 (Ob. 1.2) : la *competence structure* générée par la Q-matrice est linéaire ;
- Observation 3 (Ob. 1.3) : en comparant les deux *competence structure*, nous remarquons que la seconde (de la Q-Matrice) représente un sous-graphe de la première (représentée par un chemin dans la figure 4).

4.2 Les Cristaux d'Éhère

Les Cristaux d'Éhère[2] est un prototype de SG destiné à des élèves de collège pour apprendre la physique des changements d'état de l'eau. Il est composé de 11 niveaux. Dans chaque niveau, l'objectif de l'apprenant est de résoudre une énigme en mettant en jeu les compétences liées aux changements d'état de l'eau.

Le modèle du domaine associé à ce SG a été construit par un enseignant du secondaire, expert du domaine (figure 6). Par ailleurs, cet expert a également indexé les différents niveaux du SG par les compétences qu'ils permettent de travailler. Le tableau 2 représente un extrait de la matrice d'indexation (Q-matrice).

[2] http://seriousgames.lip6.fr/Cristaux_Ehere

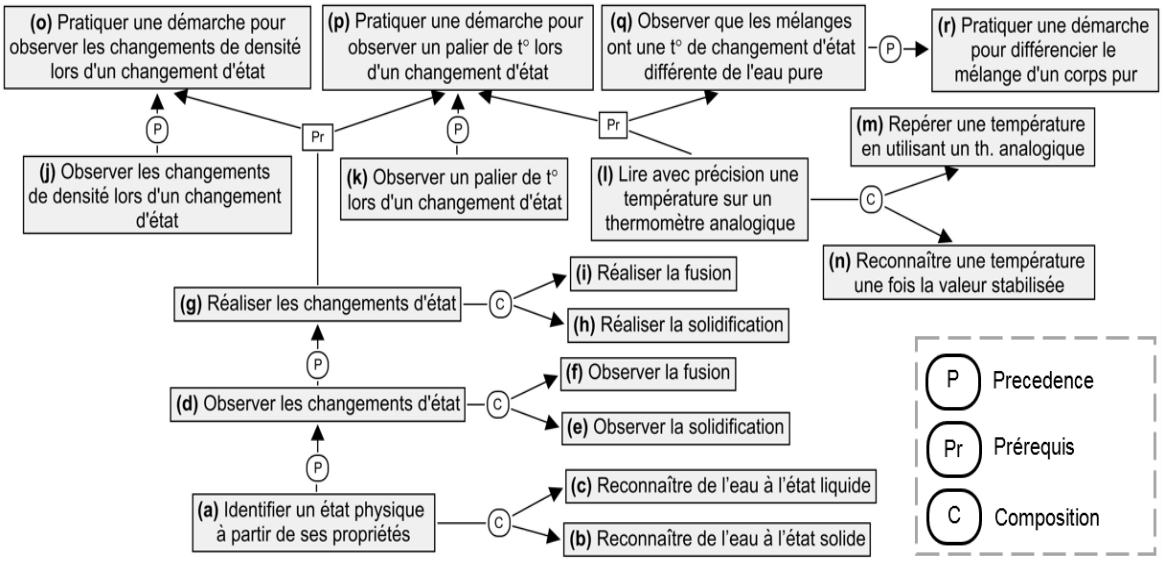


FIGURE 6 –Le modèle du domaine pour le SG *Les Cristaux d'Éhère*.

Table 2 – *La Q-Matrice de Les Cristaux d'Éhère.*

Niveaux	[a]		[d]		[g]		[p]	[l]		[r]	*Competence state*
	[c]	[b]	[e]	[f]	[i]	[h]		[m]	[n]		
Niveau 1								x			[m]
Niveau 2		x						x		x	[b, m, r]
Niveau 3			x	x			x	x	x		[e, f, p, m, n]
Niveau 4		x		x				x	x		[b, f, m, n]
Niveau 5			x					x	x		[e, m, n]
Niveau 6		x			x			x			[b, i, m]
Niveau 7				x	x			x	x		[f, i, m, n]
Niveau 8	x		x			x					[c, e, h]
Niveau 9	x		x					x	x		[c, e, m, n]
Niveau 10								x	x		[m, n]
Niveau 11								x	x		[m, n]

Comme pour le premier SG, nous avons généré les *competence structure* par les deux approches, du modèle de domaine (figure 7) et de la Q-Matrice (figure 8).

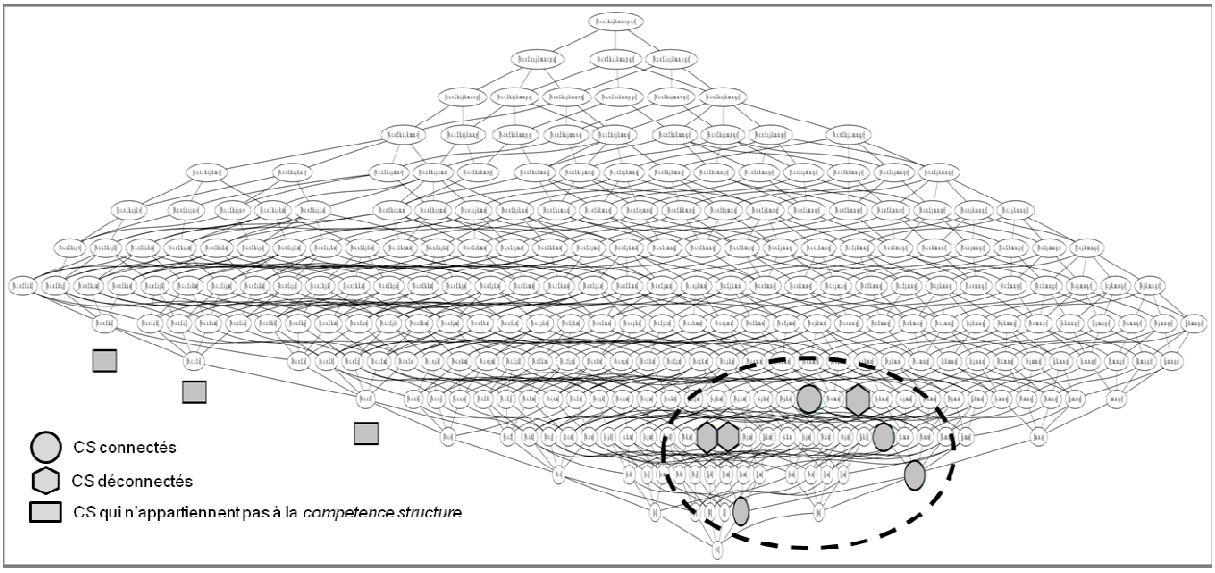


FIGURE 7 –La *competence structure* de *Les Cristaux d'Éhère* générée par le modèle de domaine (disponible à http://javiermelero.es/Cristaux_IC15.png).

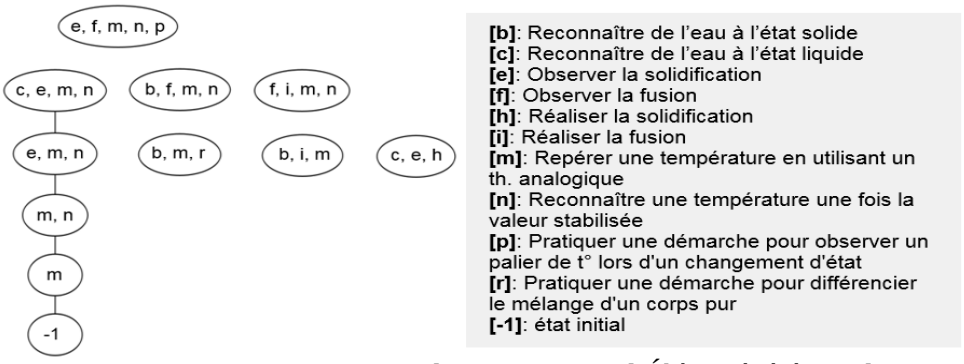


FIGURE 8 –La *competence structure* de *Les Cristaux d'Éhère* générée par la Q-Matrice.

Pour ce SG, nous mettons également en évidence quelques observations :

- Observation 1 (Ob. 2.1) : dans la *competence structure* générée par le modèle de domaine défini par un expert, le nombre de *CS* ayant des niveaux correspondants est très faible, uniquement 7 *CS* sur 266 (représentés en gris dans l'ellipse en pointillés sur la figure 7) ;
- Observation 2 (Ob. 2.2) : la *competence structure* générée par la Q-matrice n'est pas complètement connectée. De plus, les *CS* qui sont connectés, le sont de manière linéaire ;
- Observation 3 (Ob. 2.3) : curieusement, certains *CS* (carrés gris dans la figure 7) construits par la Q-Matrice ne correspondent à aucun *CS* de la *competence structure* générée par le modèle de domaine (ex. le *CS* [e,f,p,m,n] associé au niveau 3. Cette observation soulève la question de l'incohérence entre les deux approches. Y-a-t-il des défauts dans le modèle de domaine construit par l'expert ? Y-a-t-il des erreurs dans la Q-Matrice, c'est-à-dire dans l'indexation des niveaux par les compétences ?
- Observation 4 (Ob. 2.4) : l'ensemble des *CS* construits par la Q-Matrice semble correspondre à une branche ou sous-graphe de la *competence structure* générée par le modèle de domaine (ellipse en pointillés sur la figure 7).

4.3 Refraction

Refraction[3] est un SG dont la visée pédagogique est de consolider les acquis sur l'addition et la multiplication de fractions. Il est composé de 61 niveaux regroupés en 7 mondes. Le but du jeu est d'alimenter des vaisseaux spatiaux avec des lasers en évitant des obstacles. Chaque laser a sa puissance et les vaisseaux nécessitent des valeurs différentes. Il est alors nécessaire de découper ces lasers ou de les combiner pour atteindre la valeur demandée grâce à des éléments. Ce qui revient à additionner, multiplier ou réduire au même dénominateur (si nécessaire) des fractions.

Pour ce SG, nous nous sommes inspirés du travail fait autour d'un framework d'analyse de SG proposé par (Bernard & Yessad, 2014). Dans ce travail, une indexation des compétences mathématiques a été proposée pour plusieurs SG dont *Refraction*. La liste des compétences travaillées dans les niveaux, que nous avons adaptées, est donnée par la figure 9 et un extrait de l'indexation est donné dans le tableau 3.

TABLE 3 – *Extrait de la Q-Matrice de Refraction (5ème monde).*
La matrice complète est disponible à http://javiermelero.es/Refraction_IC15.pdf

monde 5	[a]	[b]	[c]	[d]	[e]	[f]	[g]	[h]	[i]	[j]	[k]	[l]	*CSs*
Niveau 5.1					x								[e]
Niveau 5.2					x								[e]
Niveau 5.3	x		x		x	x							[a, c, e, f]
Niveau 5.4					x								[e]
Niveau 5.5					x	x			x	x			[e, f, i, j]
Niveau 5.6					x	x			x	x	x		[e, f, i, j, k]
Niveau 5.7	x	x	x		x	x		x					[a, b, c, e, f, h]
Niveau 5.8	x	x			x	x			x	x			[a, b, e, f, i, j]
Niveau 5.9	x	x	x	x	x	x		x					[a, b, c, d, e, f, h]
Niveau 5.10	x	x	x		x	x			x	x	x		[a, b, c, e, f, i, j, k]

La figure 9 illustre la *competence structure* résultante en utilisant la seconde approche, c'est-à-dire générée par la Q-Matrice. Nous obtenons 23 *CS* différents. Nous rappelons qu'un *CS* correspond à l'ensemble des compétences travaillées dans un niveau donné.

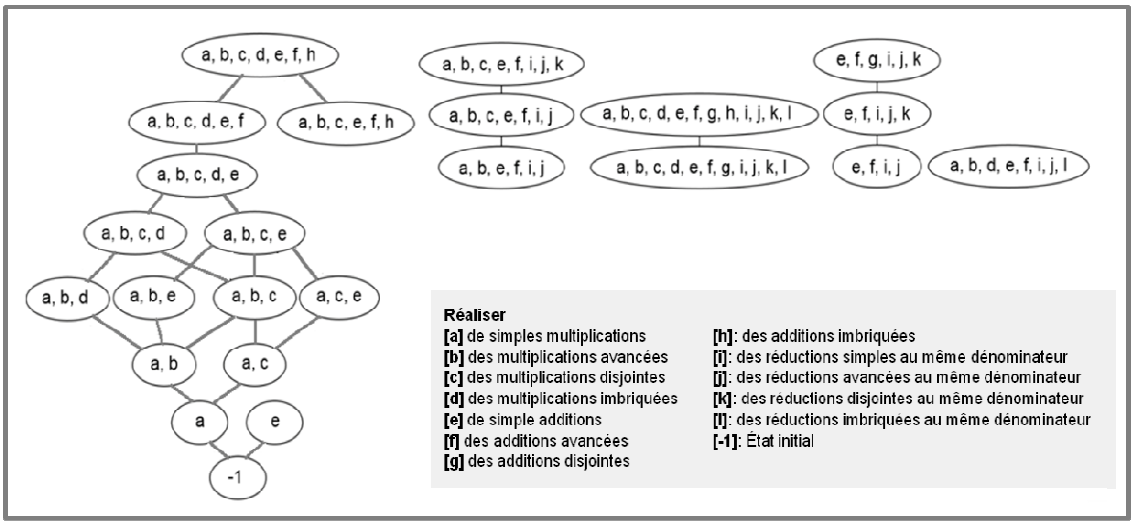


FIGURE 9 –La *competence structure* de *Refraction* générée par la Q-Matrice.

[3] http://games.cs.washington.edu/refraction/refraction.html

Pour ce SG, nous mettons en évidence les observations suivantes :

- Observation 1 (Ob. 3.1) : la *competence structure* générée par la Q-matrice n'est pas complètement connectée puisque nous retrouvons des *CS* isolés. Par conséquent, l'adaptation ne peut être mise en œuvre à un certain niveau de la *competence structure* ;
- Observation 2 (Ob. 3.2) : contrairement aux deux autres SG, le sous-graphe connecté n'est pas linéaire.

De l'étude des trois SG, nous tirons plusieurs enseignements qui font l'objet de la section suivante.

5 Les enseignements tirés de l'application de CbKST

Les principales leçons tirées des *competence structure* générées par les deux approches sont les suivantes :

- La *competence structure* générée par le modèle du domaine peut être utilisée comme un moyen de valider la conception des niveaux du SG. En effet, en partant de la Q-Matrice, il est possible de constater si tous les niveaux correspondent au modèle du domaine (Ob. 1.3) ou non (Ob. 2.3).
- Les *competence structure* générées par les Q-Matrices semblent plus simples que celles générées par le modèle du domaine. Quand un SG contient peu de niveaux, la *competence structure* tend à être une séquence linéaire de *CS* (Ob. 1.2, Ob. 2.2). Néanmoins, quand il existe plusieurs niveaux dans le SG, cette *competence structure* est plus intéressante dans le sens où elle correspond à un graphe dans lequel plusieurs parcours d'apprentissage peuvent être envisagés (Ob. 3.2). Par conséquent, l'adaptation peut être mise en œuvre.
- Les *competence structure* dérivées du modèle du domaine sont plus complètes puisqu'elles couvrent l'ensemble du domaine. À contrario, celles générées par les Q-Matrices sont plus étroitement liées à la conception des niveaux du SG, et de ce fait couvrent seulement un sous-ensemble du domaine. Par conséquent, la *competence structure* générée par le modèle du domaine peut servir de base de conception et d'analyse pour plusieurs SG qui ciblent ledit domaine. En revanche, celle générée par la Q-Matrice doit être « instanciée » pour chaque SG.
- Tous les *CS* d'une *competence structure* générée par la Q-Matrice contiennent des niveaux auxquels ils sont rattachés puisque les *CS* sont construits à partir de l'indexation des niveaux. À l'inverse, une *competence structure* générée par le modèle du domaine contient plusieurs *CS* qui ne correspondent à aucun niveau (Ob. 1.1 et Ob. 2.1). Cela suggère que plusieurs niveaux peuvent être conçus de manière à ce que chaque *CS* puisse être associé à, au moins, un niveau. Cela étant, un SG n'a pas forcément vocation à (ou la prétention de) couvrir l'ensemble d'un domaine donné (donc tous les *CSS* possibles). Il est d'usage qu'un SG couvre plus un sous-ensemble particulier d'un modèle de domaine (Ob. 2.4).
- Les compétences doivent être décrites à un bon niveau de granularité. Un niveau trop fin facilite le suivi et l'évaluation de l'apprenant car les compétences de bas niveau correspondent à un niveau opérationnel, c'est-à-dire proche des actions de l'apprenant. En revanche, cette finesse se traduit par une *competence structure* dense (voir figure 7), ce qui rend le mécanisme d'adaptation plus complexe à mettre en œuvre.
- Il y a des « trous » entre les *CS* dans la *competence structure* générée par la Q-Matrice. En effet, relier un *CS* contenant n éléments (c'est-à-dire n compétences) à un *CS* contenant exactement les mêmes n éléments plus un autre n'est pas toujours possible en partant de la Q-Matrice (voir figure 8). Autrement dit, il n'y a pas un continuum entre les niveaux existants du SG. Ceci peut constituer une piste de réflexion pour les

concepteurs du SG pour les amener à penser d'autres niveaux de manière à ce que la progression dans l'acquisition des compétences soit plus graduelle.
- Par ailleurs, certains *CS* obtenus par la Q-Matrice ne correspondent à aucun *CS* de la *competence structure* obtenue par le modèle de domaine (Ob 2.3). Cela met en évidence des problèmes de modélisation du domaine ou des problèmes d'indexation.

Ces enseignements constituent des pistes de réflexion sur tous les aspects à considérer quand on veut appliquer une approche fondée sur la CbKST pour adapter des SG au profil de l'apprenant. Plusieurs perspectives de recherche émergent de ces enseignements.

6 Discussion, travaux en cours et orientations futures

Dans le travail présenté dans cet article, nous considérons deux moyens pour modéliser le modèle du domaine qui est ciblé par le SG, un explicite construit par l'expert et l'autre implicite inféré par la Q-Matrice. Ces deux moyens peuvent être complémentaires. Inférer le modèle (implicite) par la Q-Matrice peut faciliter le travail des experts quand les liens entre les compétences sont difficiles à établir.

Pour utiliser CbKST comme approche pour l'adaptation des parcours d'apprentissage dans les SG, il est nécessaire de prendre en compte deux considérations. D'une part, lorsque le SG s'appuie sur un modèle du domaine explicite défini par l'expert, les niveaux doivent être conçus de manière à ce qu'ils correspondent à des *CS* admissibles (significatifs). D'autre part, si aucun modèle explicite du domaine n'est disponible pour le SG, les niveaux devraient être conçus de manière à ce que les compétences soient travaillées progressivement. Autrement, plusieurs « trous » peuvent exister et par conséquent, l'adaptation serait difficilement applicable. Dans ce cas, il est nécessaire de proposer une solution pour créer des *CS* connectés entre eux. Une solution possible est de créer autant de nouveaux *CS* que nécessaire pour relier les *CS* isolés.

Comme nous l'avons montré, les *competence structure* des SG générées par le modèle du domaine sont plus complètes que celles générées par les Q-Matrices. Néanmoins, pour un domaine donné, le nombre de *CS* peut dépasser plusieurs centaines de milliers. Ceci peut constituer un inconvénient s'il faut créer un niveau pour chaque *CS*. Une solution possible impliquerait de concevoir des SG qui ciblent, à chaque fois, un sous-ensemble de *CS* (une partie de la *competence structure* générée par le modèle du domaine). En outre, les experts et les concepteurs du SG doivent trouver le bon niveau de granularité pour décrire les compétences du domaine de sorte à rendre possible la mise en œuvre concomitante du suivi et de l'adaptation.

Pour concevoir un SG qui s'adapte à l'apprenant, il est nécessaire de varier les niveaux permettant de travailler divers ensembles de compétences, permettant d'obtenir plusieurs parcours d'apprentissage. En effet, comme nous l'avons constaté, quand il y a peu de niveaux, la *competence structure* tend à être linéaire, par conséquent l'adaptation ne peut être réalisée.

Nous avons vu qu'il peut y avoir des incohérences entre un modèle de domaine proposé par un expert par rapport à l'indexation dans la Q-Matrice. La *competence structure* générée par le modèle du domaine pourrait constituer ainsi un outil pertinent au stade de la conception pour créer de nouveaux niveaux qui comblent les lacunes ou les « trous » dans les parcours d'apprentissage et/ou pour vérifier si le SG ne présente pas de défauts de conception au niveau pédagogique du terme.

Les pistes de recherche que nous explorons actuellement sont nombreuses. Parmi celles-ci, nous étudions les possibilités (a) de considérer le modèle du domaine réalisé par l'expert comme un outil de validation pour la conception des niveaux du SG ; (b) de comparer les scénarios faits par des enseignants à ceux générés automatiquement par la *competence structure*.

Remerciements. Nous tenons à remercier la région Île-de-France et le ministère français de l'économie et des finances pour leur soutien au projet FUI Play Serious. Nous tenons à remercier également Bertrand Marne pour sa relecture et ses conseils avisés.

Références

Augustin, T., Hockemeyer, C., Kickmeier-Rust, M. D., Podbregar, P., Suck, R., & Albert, D. (2013). The simplified updating rule in the formalization of digital educational games. *Journal of Computational Science*, *4*(4), 293‑303. doi :10.1016/j.jocs.2012.08.020

Bernard, N., & Yessad, A. (2014). Framework Multidimensionnel d'Analyse de Niveaux de Jeux Sérieux. Dans *TICE 2014* (pp. 13‑24). Béziers, France. Repéré à http://ticeconf.org/fr/fr/images/actes-SCI-provisoires.pdf

Csíkszentmihályi, M. (1991). *Flow : the psychology of optimal experience*. New York : HarperPerennial.

Doignon, J.-P., & Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, *23*(2), 175‑196. doi :10.1016/S0020-7373(85)80031-6

Doignon, J.-P., & Falmagne, J.-C. (1999). *Knowledge Spaces*. (S.l.) : Springer Berlin. Repéré à http://www.springer.com/computer/ai/book/978-3-540-64501-6

Dondlinger. (2007). *Journal of Applied Educational Technology*, *4*(1), 21‑31.

Falmagne, J.-C., Cosyn, E., Doignon, J.-P., & Thiéry, N. (2006). The Assessment of Knowledge, in Theory and in Practice. Dans R. Missaoui & J. Schmidt (Éds), *Formal Concept Analysis* (pp. 61‑79). (S.l.) : Springer Berlin Heidelberg. Repéré à http://link.springer.com/chapter/10.1007/11671404_4

Göbel, S., Wendel, V., Ritter, C., & Steinmetz, R. (2010). Personalized, Adaptive Digital Educational Games Using Narrative Game-Based Learning Objects. Dans X. Zhang, S. Zhong, Z. Pan, K. Wong, & R. Yun (Éds), *Entertainment for Education. Digital Techniques and Systems* (pp. 438‑445). (S.l.) : Springer Berlin Heidelberg. Repéré à http://link.springer.com/chapter/10.1007/978-3-642-14533-9_45

Heller, J., Mayer, B., & Albert, D. (2005). Competence-based Knowledge Structures for Personalised Learning. Dans *1st International ELeGI Conference on Advanced Technology for Enhanced Learning*. Vico Equense-Naples, Italy. Repéré à http://telearn.archives-ouvertes.fr/hal-00190482/

IEEE, & ACM. (2001). *Computing Curricula 2001, Computer Science*. Repéré à http://www.acm.org/education/curric_vols/cc2001.pdf

Kickmeier-Rust, M. D., Göbel, S., & Albert, D. (2008). 80Days: Melding Adaptive Educational Technology and Adaptive and Interactive Storytelling in Digital Educational Games. Dans *International Workshop on Story-Telling and Educational Games*. Maastricht.

Kopeinik, S., Nussbaumer, A., Bedek, M., & Albert, D. (2012). Using CbKST for Learning Path Recommendation in Game-based Learning. Dans *20th International Conference on Computers in Education* (pp. 26‑30).

Peirce, N., Conlan, O., & Wade, V. (2008). Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences (pp. 28‑35). Communication présentée au 2nd IEEE International Conference on Digital Games and Intelligent Toys Based Education. doi :10.1109/DIGITEL.2008.30

Play Serious Project. (2013). *Site officiel*. Repéré à http://www.playserious.fr/

Shute, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. *Adaptive technologies for training and education*, 7–27.

Tatsuoka, K. K. (1983). Rule Space: An Approach for Dealing with Misconceptions Based on Item Response Theory. *Journal of Educational Measurement*, *20*(4), 345‑354. doi :10.1111/j.1745-3984.1983.tb00212.x

MEMORAe : Plateforme web pour supporter l'annotation collaborative

Ala Atrash, Marie-Hélène Abel

SORBONNE UNIVERSITÉS, UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE
UMR CNRS 7253 , laboratoire HEUDIASYC
Centre de recherche Royallieu
CS 60 319, 60 203 Compiègne cedex - France
{ala-aldin.atrash, marie-helene.abel}@utc.fr

1 Contexte

Selon nous, une annotation est "la transcription d'une idée concernant un sujet au sens large appelé cible (document, partie de document, personne, etc.) exprimée dans un corps (contenu de l'annotation) au format et au support désirés". Lorsque l'annotation est partagée, c'est-à-dire destinée à différents lecteurs, elle est dite collaborative. Aujourd'hui de nombreuses plateformes offrent la possibilité de créer des annotations collaboratives, cependant elles présentent toutes la même limite : l'annotation créée devient partie intégrante de la cible Su *et al.* (2010). Le seul moyen d'accéder à une annotation consiste à accéder en premier à la cible (document pour lui-même ou pour la partie le composant identifiée comme cible). Il n'est donc pas possible de rechercher une annotation pour elle-même pour ensuite accéder à sa cible. Afin de dépasser cette limite, nous avons choisi de considérer l'annotation comme une ressource à part entière Atrash *et al.* (2014).

2 La plateforme Web MEMORAe

MEMORAe est une plateforme web [1] qui exploite la puissance des nouvelles technologies support à la collaboration (technologies web 2.0, etc.) Atrash *et al.* (2015). La plateforme est basée sur le modèle sémantique MEMORAe-core 2. Ce modèle est développé en OWL (Web Ontology Language) et exploite des standards du web sémantique (foaf [2], sioc [3], bibo [4], oa [5]). Dans ce modèle, les utilisateurs appartiennent à un ou plusieurs groupes. A chaque groupe est associé un espace de partage dans lequel sont accessibles des ressources. Les ressources sont indexées par un ou plusieurs concepts d'une ontologie définissant un référentiel métier (ontologie d'application). Les concepts sont présentés aux utilisateurs sous la forme d'une carte sémantique. Les utilisateurs peuvent naviguer au sein de cette carte. Lorsqu'ils veulent accéder aux

1. Vidéo disponible : http://memorae.hds.utc.fr/api/documents/MEMORAe.avi
2. Friend of a friend : http ://www.foaf-project.org/
3. Semantically-Interlinked Online Communities : http ://rdfs.org/sioc/spec/
4. Bibliographic Ontology Specification : http ://bibliontology.com/
5. Open Annotation : http ://www.w3.org/ns/oa

ressources indexées par un concept, il suffit de sélectionner ce concept dans la carte et d'ouvrir les espaces de partage auxquels ils ont accès. Les ressources visibles dans ces espaces seront celles indexées par le concept sélectionné. Les espaces de partage peuvent être visualisés en parallèle ce qui facilite le transfert de connaissance d'un groupe à un autre en rendant l'accès d'une ressource d'un espace à un autre par un simple glisser-déposer. Les utilisateurs de la plateforme peuvent annoter tout type de ressources y compris des documents, des notes, d'autres annotations, etc. Il est également possible d'annoter les concepts de la carte sémantique. Puisqu'une annotation est une ressource à part entière, nous proposons deux façons de les accéder. La première est classique puisqu'il s'agit de passer par l'ouverture de sa cible. La deuxième consiste à ouvrir directement l'annotation à partir d'un espaces de partage. En effet, l'annotation, étant une ressource, est indexée. Elle est donc accessible par son ou ses index comme toute ressource. Une fois ouverte, il est possible de voir son contenu mais également d'accéder à sa cible.

3 Discussion

Un test de la plateforme a eu lieu au département de génie informatique de l'Université de Technologie de Compiègne auprès des étudiants suivant un cours sur les techniques de modélisation, de capitalisation et de gestion des connaissances. Le test a duré quatre mois (octobre 2014 à janvier 2015). Les étudiants devaient effectuer de manière collaborative une veille technologique sur un sujet particulier. Quatre groupes de dix ou douze membres ont été constitués. Chaque groupe a été amené à construire une ontologie délimitant le périmètre de l'étude. La construction de cette ontologie faisait partie intégrante du processus d'apprentissage. Les étudiants l'ont ensuite utilisée pour produire la carte sémantique au sein de la plateforme MEMORAe et commencer à capitaliser et échanger autour de cette dernière (partage de documents, liens web, notes, annotations, etc.). A la fin du travail de veille, les étudiants ont rempli anonymement un questionnaire sur leur ressenti. 80 % des étudiants ont indiqué qu'ils trouvaient intéressant de considérer les annotations comme ressource à part entière. Nous poursuivons actuellement le développement de fonctionnalités autour de notre modèle d'annotation : annotation multi-cible et annotation multi-auteurs.

Références

ATRASH A., ABEL M.-H. & MOULIN C. (2014). Supporting organizational learning with collaborative annotation. In *International Conference on Knowledge Management and Information Sharing*, p. 237–244.

ATRASH A., ABEL M.-H., MOULIN C., DARÈNE N., HUET F. & BRUAUX S. (2015). Note-taking as a main feature in a social networking platform for small and medium sized enterprises. *Computers in Human Behavior*.

SU A. Y., YANG S. J., HWANG W.-Y. & ZHANG J. (2010). A web 2.0-based collaborative annotation system for enhancing knowledge sharing in collaborative learning environments. *Computers & Education*, **55**(2), 752–766.

Un outil d'extraction interactive de connaissances à partir de traces : Transmute

Pierre-Loup Barazzutti[1], Amélie Cordier[1], Béatrice Fuchs[2].

[1] LIRIS CNRS, UMR 5202, Université Lyon 1,

[2] LIRIS CNRS, UMR 5202, Université Lyon 3, France.

Résumé : TRANSMUTE est un outil d'assistance à l'extraction de connaissances à partir de données (ECD), appliqué aux traces. TRANSMUTE affiche les résultats de la fouille dans la trace analysée pour mieux évaluer leur pertinence et par les interactions avec l'analyste, met à jour dynamiquement les résultats. TRANSMUTE a été développé sous forme d'une plate-forme web et s'appuie sur le module de visualisation SAMOTRACES ainsi que sur le module d'ECD DISKIT. **Mots-clés** : traces, découverte de connaissances, interactions

TRANSMUTE est un outil interactif d'extraction de connaissances à partir de données (ECD), appliqué aux traces. Il a pour but d'illustrer l'interactivité dans le processus d'ECD en vue d'apporter une assistance à toutes les étapes. Nous appliquons l'ECD aux traces. Une trace est constituée d'un ensemble d'éléments observés temporellement situés appelés des *obsels*. Elle est associée à un modèle de trace décrivant les types d'obsels, leurs attributs et leurs relations avec d'autres types d'obsels. Les traces sont manipulées par des *transformations* qui produisent de nouvelles traces appelées *traces transformées*. Il existe différents types de transformations, parmi lesquelles la *réécriture* qui crée une trace d'un plus haut niveau d'abstraction où des séquences de la trace initiale sont remplacées par par de nouveaux types d'obsels. La réécriture se situe au cœur du dispositif mis en place dans Transmute.

L'architecture de TRANSMUTE s'articule autour de plusieurs modules. TRANSMUTE s'appuie d'abord sur le framework Samotraces [1] qui permet de personnaliser l'affichage des éléments observés en fonction de leurs caractéristiques comme leurs types, leurs attributs, *etc.* Samotraces utilise le gestionnaire de traces kTBS [2] qui assure les manipulations élémentaires de trace. Le processus d'ECD est mis en œuvre dans le module DISKIT. La fouille utilise un algorithme de fouille de séquence, DMT4SP [3] qui recherche des épisodes séquentiels fréquents en fonction des contraintes spécifiées par l'analyste. Actuellement, la fouille est limitée à une seule trace à la fois. La démonstration porte sur la phase de post-traitement.

La figure 1 montre l'interface de TRANSMUTE qui comporte plusieurs parties : (1) la trace en cours d'analyse, (2) la trace transformée en cours de construction où sont affichés les motifs, (3) un rappel du modèle de trace, (4) les motifs produits par la fouille. La fouille produit un grand nombre de motifs caractérisés par une forte redondance combinatoire. L'analyste doit traiter ces motifs et choisir ceux qui font sens au vu de son expertise du domaine. Les motifs sont enrichis à l'aide d'indicateurs d'intérêt permettant de les trier et ainsi mettre en avant

1. https://github.com/bmathern/samotraces.js
2. kernel for Trace Based System, http://tbs-platform.org/tbs/doku.php
3. Data Mining Techniques For Sequence Processing
http://liris.cnrs.fr/~crigotti/dmt4sp.html

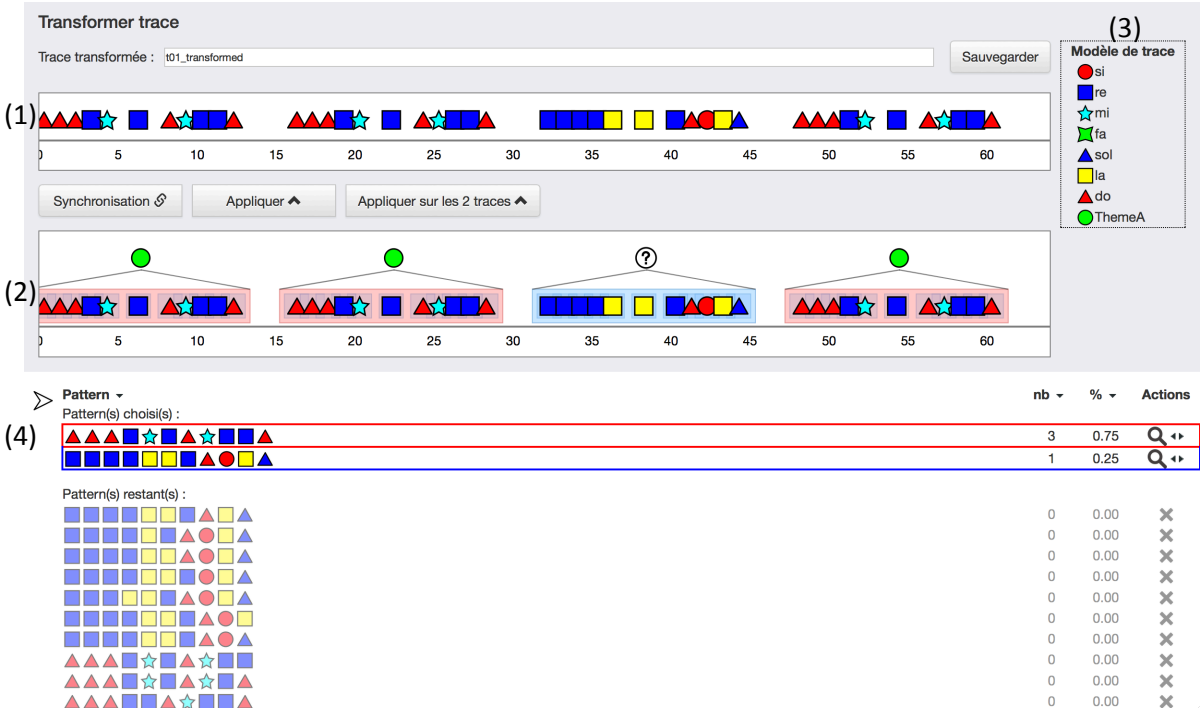


FIGURE 1 – L'interface principale de TRANSMUTE

les motifs potentiellement les plus intéressants. À chaque sélection d'un motif, l'analyste peut voir l'impact de son choix sur la trace mais aussi sur la liste des motifs. Il peut voir et accéder à toutes les occurrences du motif sélectionné dans la trace afin d'estimer la pertinence de son choix. La liste des motifs est alors mise à jour dynamiquement. Toutes les occurrences de motifs ayant au moins un obsel en commun avec les occurrences du motif sélectionné sont identifiés et éliminés. Les indicateurs associés aux motifs sont alors recalculés et les motifs ne satisfaisant plus les contraintes (notamment le support) sont éliminés à leur tour : ils sont estompés dans l'interface, et la liste des motifs restants est triée en conséquence. Ce processus permet de diminuer graduellement le nombre de résultats et ainsi de faciliter le prochain choix de l'analyste en l'aidant à se focaliser sur d'autres motifs. Pour un motif sélectionné donné, l'analyste peut créer un nouveau type d'obsel qui se substituera à chaque occurrence du motif sélectionné dans la trace transformée. Lorsque l'analyste a sélectionné tous les motifs qui l'intéressent et choisi les types d'obsels correspondants, il peut confirmer les changements et déclencher le processus de création d'une nouvelle trace transformée. Ceci a pour conséquence de supprimer définitivement tous les motifs précédemment éliminés. L'analyste peut alors continuer soit à partir des motifs restants, soit en relançant l'analyse avec d'autres contraintes sur la même trace ou sur la trace transformée nouvellement créée.

TRANSMUTE, initialement conçu pour la visualisation interactive et la transformation de traces, a été enrichi dans ce travail avec un processus d'ECD permettant la saisie des paramètres et le lancement de la fouille et l'affichage des résultats afin de construire des traces transformées. Actuellement l'assistance est limitée au post-traitement, nous prévoyons de l'étendre au prétraitement pour aider l'analyste à préparer les données et ajuster les paramètres de la fouille.

PEW : un outil d'aide à la conception d'ontologies par l'exploration des mondes possibles

Sébastien Ferré

IRISA/Université de Rennes 1
Campus de Beaulieu, 35042 Rennes cedex
ferre@irisa.fr

Mots-clés : Web sémantique, ontologies, OWL, conception, fossé syntaxe/sémantique.

La conception d'ontologie pose un certain nombre de difficultés. Une de ces difficultés est le fossé entre syntaxe et sémantique, c'est-à-dire entre la forme de surface de l'ontologies (axiomes) et ce qu'elle rend nécessaire/possible/impossible (modèles). Ce fossé entraîne des divergences entre l'intention du concepteur et sa modélisation : inférences inattendues, absence d'inférences attendues, voire incohérences. PEW a d'abord été développé pour aider un concepteur à compléter une ontologie existante avec des contraintes négatives (par exemple, séparation de classes), qui sont souvent omises et une cause fréquente d'absence d'inférences attendues (Ferré & Rudolph, 2012).

Récemment, nous avons étendu PEW en terme d'expressivité et de fonctionnalités pour permettre également la conception d'ontologies *de novo* (Ferré, 2015). PEW offre ainsi une nouvelle méthodologie de conception d'ontologie. Celle-ci reste à valider et à comparer expérimentalement, mais elle présente des propriétés intéressantes et originales. Tout d'abord, PEW empêche par construction la production d'axiomes qui rendrait l'ontologie incohérente. Deuxièmement, PEW présente à l'utilisateur un *feedback* permanent sur les inférences (faits nécessaires) et les absences d'inférence (faits possibles) à partir de la description d'une situation. Troisièmement, l'utilisateur n'a pas besoin de maîtriser une syntaxe et est guidé pas à pas dans la construction de ces descriptions. Enfin, l'utilisateur ne définit pas d'axiomes, mais indique simplement dans l'interface les faits possibles qui ne devraient pas l'être.

PEW [1] est implémenté en OCaml et utilise le raisonneur HermiT via OWL API. Il a été dérivé de Sewelis dont il réutilise l'interface et l'interaction homme-machine (Ferré & Hermann, 2012). La figure 1 présente une capture d'écran de PEW appliqué à l'ontologie des pizzas [2]. Dans cette capture d'écran, la situation courante est celle d'une pizza sans *topping*, laquelle est donc rendue possible par l'ontologie. L'arbre à droite liste les faits possible pour une telle pizza : être un pays ou pas, nécessairement avoir une base comme ingrédient, etc. À partir de là, l'utilisateur peut rendre impossible pour une pizza sans *topping* d'être un pays ou bien le fait même d'exister.

Le langage des descriptions recouvre une grande partie des expressions de classe OWL : classes atomiques et nominales, restrictions existentielles, propriétés inverses, classe $\top$, intersection, union et complément. Les axiomes sont dérivés des descriptions de situation, selon les

1. http://www.irisa.fr/LIS/softwares/pew/
2. http://www.co-ode.org/ontologies/pizza/pizza.owl

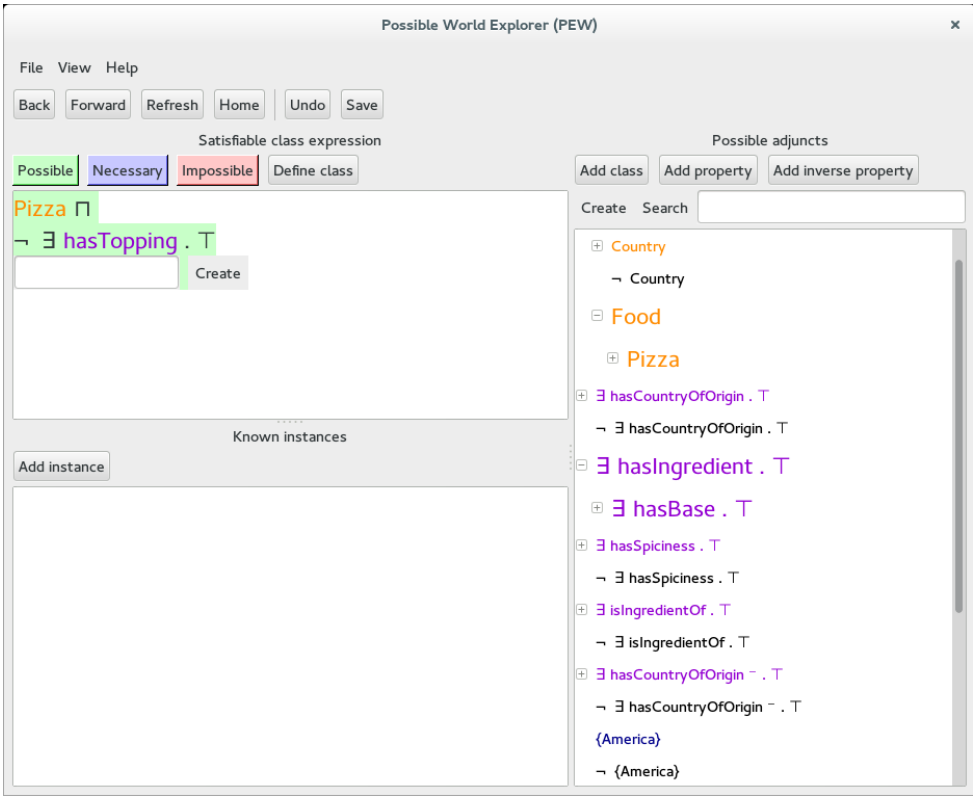


FIGURE 1 – Capture d'écran de PEW explorant les pizzas sans *topping*.

actions de l'utilisateur. Tous les types axiomes OWL sont couverts sauf les axiomes de propriétés (ex., sous-propriété de, transitivité). L'utilisateur peut introduire de nouvelles classes et propriétés et ajouter des individus comme instances de la description courante. Il peut rendre impossible ou nécessaire tout ou partie de cette description et de l'arbre de faits possibles. Enfin, il peut rendre un ensemble de faits possibles mutuellement disjoints. Dans l'état actuel, l'utilisateur ne peut pas rendre à nouveau possible une situation devenue impossible, mais une fonction *Undo* permet de supprimer les derniers axiomes émis.

La démonstration prend l'ontologie de pizzas comme exemple pour montrer les possibilités offertes par PEW. Dans un premier temps, PEW est utilisé pour révéler les incomplétudes de l'ontologie existante et les combler. Dans un deuxième temps, PEW est utilisé pour reconstruire (une partie de) cette ontologie *de novo*.

Références

FERRÉ S. (2015). Conception interactive d'ontologies par élimination de mondes possibles. In M.-H. ABEL, Ed., *Journées francophones d'Ingénierie des Connaissances*. À paraître.

FERRÉ S. & HERMANN A. (2012). Reconciling faceted search and query languages for the Semantic Web. *Int. J. Metadata, Semantics and Ontologies*, **7**(1), 37–54.

FERRÉ S. & RUDOLPH S. (2012). Advocatus diaboli - exploratory enrichment of ontologies with negative constraints. In A. TEN TEIJE ET AL., Ed., *Int. Conf. Knowledge Engineering and Knowledge Management (EKAW)*, LNAI 7603, p. 42–56 : Springer.

LinkedVocabularyEditor: une extension MediaWiki pour l'édition collaborative et la publication de vocabulaires liés

Maxime Lefrançois[1], Antoine Zimmermann[1]

École Nationale Supérieure des Mines, FAYOL-ENSMSE, Laboratoire Hubert Curien, F-42023 Saint-Étienne, France
{maxime.lefrancois,antoine.zimmermann}@emse.fr

Résumé : Nous présentons le prototype d'une extension pour MediaWiki qui permet l'édition collaborative d'ontologies par les experts de domaines, ainsi que la publication des ontologies conformément aux bonnes pratiques du W3C.
Mots-clés : Ontologies, Vocabulaires liés, Edition collaborative, MediaWiki.

Introduction. Nous présentons LINKEDVOCABULARYEDITOR [1], le prototype d'une extension pour MediaWiki qui en transforme une instance en une plateforme (i) d'édition collaborative d'ontologies par les experts de domaines, et (ii) de publication des ontologies conformément aux bonnes pratiques du W3C.

Besoins identifiés. Plus précisément, le but de LINKEDVOCABULARYEDITOR est de répondre à l'ensemble des besoins suivants :

— l'outil doit offrir aux experts du domaine des interfaces de visualisation et d'édition simplifiées d'une ontologies et de ses différents modules ;
— l'outil doit permettre aux ingénieurs de connaissances de surveiller l'évolution du vocabulaire : les notifier des changements, permettre le contrôle de versions ;
— l'outil doit parallèlement servir de plateforme de publication des ontologies, suivant les bonnes pratiques énoncées par W3C ;
— l'outil doit s'intégrer dans un outil de travail collaboratif open-source courant, afin d'encourager une communauté de développeurs à l'améliorer.

Les outils existants ne répondent pas à l'ensemble de ces besoins. Par exemple, Protégé et sa version web (Noy *et al.*, 2001; Tudorache *et al.*, 2008) ciblent les ingénieurs des connaissances. La plateforme Neologism (Basca *et al.*, 2008) ne permet pas la gestion de versions. Semantic MediaWiki (Krötzsch *et al.*, 2006) et WikiData (Vrandečić, 2012) se focalisent respectivement sur l'annotation des pages et sur l'édition de données structurées, plutôt que sur l'édition et la publication des vocabulaires qui structurent ces annotations ou ces données.

Principe de l'extension. L'extension LINKEDVOCABULARYEDITOR définit un espace de nom [2] spécial nommé *Resource:*, synchronisé avec un triple store ARC2 [3]. Chaque page y représente la description d'une ressource, comme suit : si le préfixe *seas:* représente `http://`

1. Le code de LinkedVocabularyEditor est disponible sur GitHub - `https://github.com/maximelefrancois86/LinkedVocabularyEditor` ; une instance est testable en ligne avec le compte : login : **IC2015**, mot de passe : **lve** - `http://wiki-lve.maxime-lefrancois.info`,
2. Espace de nom MediaWiki - `http://fr.wikipedia.org/wiki/Aide:Espace_de_noms`
3. ARC2 - PHP&MySQL-based triplestore - `https://github.com/semsol/arc2/`

`purl.org/NET/seas#`, alors la page *Resource:Seas:Site* décrit la ressource `http://purl.org/NET/seas#Site`. Lorsqu'on accède à cette page, une négociation de contenu est opérée, et le wiki sert une représentation HTML, RDF/XML, turtle, ou N-triples suivant l'option HTTP Accept de la requête.

En interne, le contenu d'une page est une représentation en JSON-LD de la ressource décrite, dans chaque graphe nommé du triple store où cette ressource apparaît. Le contenu de la page HTML est généré non seulement à partir de ce JSON-LD, mais également à l'aide d'appels au triple store. L'interface d'édition de la page utilise principalement *angularjs* [4], et des appels au Endpoint SPARQL de ARC2 pour la recherche de ressources dans le wiki.

Utilisation. Lorsqu'un expert du domaine crée une nouvelle page ou édite une page existante, un formulaire simple lui est proposé, et la validation de ce formulaire met à jour le triple store. l'extension permet déjà d'éditer des ontologies avec le profile OWL LD (Glimm *et al.*, 2012). De son côté, l'ingénieur des connaissances bénéficie des fonctionnalités de notification, modération, et de contrôle de version, déjà implémentées dans MediaWiki. De plus, il a accès à deux pages spéciales. L'une lui permet d'éditer les espaces de noms RDF (pour définir l'espace de nom d'un module d'ontologie par exemple), l'autre lui permet d'importer un vocabulaire existant (elle lui permet d'exporter le vocabulaire, de l'éditer à la main en dehors du wiki, puis de l'importer à nouveau).

Limitations actuelles et évolutions possibles. LINKEDVOCABULARYEDITOR implémente un petit ensemble de fonctionnalités essentielles, qui forme une base intéressante pour le développement et l'évaluation de nouvelles techniques de visualisation et d'édition collaborative d'ontologies. D'autre fonctionnalités manquent, comme le renommage ou la suppression des ressources. Par ailleurs, certaines fonctionnalités de MediaWiki méritent d'être utilisées, comme les hiérarchies de catégories pour représenter les hiérarchies des classes et de propriétés. Enfin, chaque page contient seulement les triplets qui ont la ressource représentée comme sujet. Ce choix temporaire supprime les inter-dépendances entre contenus de pages, et des questions d'ingénierie et de recherche non triviales se poseront lorsque l'on supprimera cette limitation.

Remerciements. Ce travail a été développée dans le cadre du projet ITEA2 14002 SEAS - Semantic Energy Aware Systems - `https://itea3.org/project/seas.html`

Références

BASCA C., CORLOSQUET S. & CYGANIAK R. (2008). Neologism : Easy vocabulary publishing. In *4th Workshop on Scripting for the Semantic Web*.

GLIMM B., HOGAN A., KRÖTZSCH M. & POLLERES A. (2012). OWL : Yet to arrive on the Web of Data ? *LDOW'2012*.

KRÖTZSCH M., VRANDEČIĆ D. & VÖLKEL M. (2006). Semantic mediawiki. In *The Semantic Web-ISWC 2006*, p. 935–942 : Springer Berlin Heidelberg.

NOY N., SINTEK M. & DECKER S. (2001). Creating semantic web contents with protege-2000. *IEEE intelligent systems*, **16**(2), 60–71.

TUDORACHE T., VENDETTI J. & NOY N. (2008). Web-Protege : A Lightweight OWL Ontology Editor for the Web. *OWLED'2008*, **432**.

VRANDEČIĆ D. (2012). Wikidata : A new platform for collaborative data collection. In *WWW'2012*, p. 1063–1064 : ACM.

4. Angular.js - `https://angularjs.org/`

PERSOREC : un système personnalisé de recommandations pour les *folksonomies* basé sur les concepts quadratiques

Mohamed Nader Jelassi[1,2,3], Sadok Ben Yahia[1] et Engelbert Mephu Nguifo[2,3]

[1] UNIVERSITÉ DE TUNIS EL-MANAR, FACULTÉ DES SCIENCES DE TUNIS, LIPAH,2092, TUNIS, TUNISIE
[2] CLERMONT UNIVERSITÉ, UNIVERSITÉ BLAISE PASCAL, LIMOS, BP 10448, F-63000 CLERMONT FERRAND
[3] CNRS, UMR 6158, LIMOS, F-63171 AUBIÈRE, FRANCE, nader.jelassi@isima.fr
sadok.benyahia@fst.rnu.tn,engelbert.mephu_nguifo@univ-bpclermont.fr

Mots-clés : folksonomies, personnalisation, recommandation, concepts quadratiques.

1 Introduction et Motivations

Une *folksonomie* désigne un système de classification collaborative par les internautes (Mika (2007)). Les *folksonomies* ont à tenir compte des besoins de ses utilisateurs lors de la recommandation de tags ou de ressources. Cela a incité les chercheurs à proposer des systèmes de recommandation personnalisés afin de suggérer les tags et ressources les plus appropriés aux utilisateurs et de répondre aux besoins de chaque utilisateur. En effet, le domaine de personnalisation tente de fournir des solutions afin d'aider les utilisateurs à partager les bons tags et les bonnes ressources parmi le très grand nombre de données dans les *folksonomies*. De plus, la personnalisation tente d'aider les utilisateurs à aborder le problème de surcharge d'information (Das *et al.* (2012)). Et pour réussir ou tenter de répondre au mieux aux attentes de chaque utilisateur de la *folksonomie*, il est utile d'avoir plus d'informations sur lui. Pour atteindre cet objectif, nous considérons une nouvelle dimension dans une *folksonomie*, classiquement composée de trois dimensions (utilisateurs, tags et ressources), qui contient des informations supplémentaires sur les utilisateurs et nous proposons une approche de regroupement des utilisateurs aux intérêts équivalents sous forme de structure appelées concepts quadratiques (Jelassi *et al.* (2013)). Un concept quadratique illustre une conceptualisation partagée dans la *folksonomie*. Par exemple, un concept peut être : *"Jack et Kate qui sont agés entre* 18 *et* 25 *ans ont utilisé les tags 'action' et 'aventure', parmi d'autres, pour annoter des films comme 'Indiana Jones' et 'Star Wars'"*.

2 PERSOREC : un système personnalisé de recommandations pour les *folksonomies*

Le pseudo code de PERSOREC est disponible dans le papier (Jelassi *et al.* (2013)). PERSOREC prend un ensemble de quadri-concepts $\mathcal{QC}$ comme entrée ainsi qu'un utilisateur cible u avec son profil P et (optionellement) une ressource r (à annoter). PERSOREC donne en sortie trois ensembles : un ensemble d'utilisateurs proposés, un ensemble de tags suggérés et un ensemble de ressources recommandées. À l'algorithme original, nous avons ajouté une mesure de score afin de classifier les recommandations par ordre d'importance. La mesure (notée rec_score) correspondant à un profil P est définie comme suit :

$$rec\_score(r_i, P) = \frac{|u_i|}{|UU|} / \exists t_i \exists r_i \exists p_i \in P, (u_i, t_i, r_i, p_i) \in \mathcal{QC} \quad (1)$$

La mesure rec_score d'une ressource r_i correspondant à un profil P est le nombre d'utilisateurs uniques ayant le même profil P (ou au moins une information de profil $p_i \in P$) et ayant partagé la même ressource r_i, divisé par le nombre total d'utilisateurs uniques dans l'ensemble des quadri-concepts (noté UU). Par exemple, si une ressource r_1 a été partagée par 7 utilisateurs différents parmi un ensemble total de 67 utilisateurs uniques, son score sera égal à $0,104$.

FIGURE 1 – Un snapshot du siteweb de PERSOREC pour le jeu de données MOVIELENS. (**gauche**) le profil de l'utilisateur *Yasmine*, ses films partagés et ses amis ; (**milieu**) les recommandations de films pour l'utiliateur *Yasmine* ; (**droite**) la liste d'amis proposés à *Yasmine*.

La démonstration de notre système personnalisé de recommandation PERSOREC [1] démontre à travers deux jeux de données, *i.e.,* MOVIELENS (*http ://movielens.umn.edu/*) et BOOKCROSSING(*http ://www.bookcrossing.com/*), le processus de recommandation pour un utilisateur donné. Par ailleurs, il est à noter que notre système est générique, *i.e.*, peut être appliqué à n'importe quel jeu de données ayant la structure quadratique (utilisateur, tag, ressource, profil).

Références

DAS M., THIRUMURUGANATHAN S., AMER-YAHIA S., DAS G. & YU C. (2012). Who tags what ? an analysis framework. *In Proceedings of PVLDB*, **5**(11), 1567–1578.

JELASSI M. N., BEN YAHIA S. & MEPHU NGUIFO E. (2013). A personalized recommender system based on users' information in folksonomies. In *Proc. of the 22nd International Conference on World Wide Web companion*, WWW '13 Companion, p. 1215–1224.

MIKA P. (2007). Ontologies are us : A unified model of social networks and semantics. *Journal of Web Semantics.*, **5**(1), 5–15.

1. une vidéo est disponible sur ce lien : *https ://plus.google.com/u/0/113676282606125158455/posts/CE4DB2uuqU9 ?pid=6125446455288662610&oid=113676282606125158455*

[illegible]

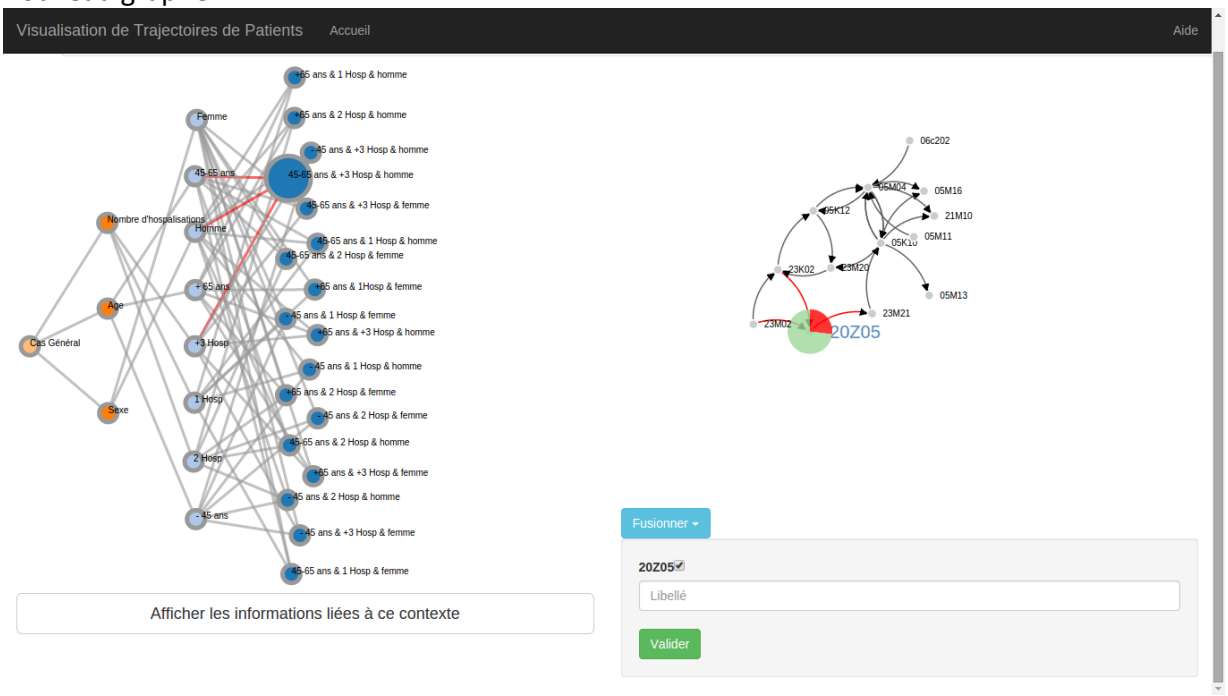
Jessica Pinaire[1,2,3], Soumaya Ben Alouane[1], Jérôme Azé[1], Sandra Bringay[1,4], Paul Landais[2,3], Arnaud Sallaberry[1,4]
[1]LIRMM, UMR 5506, Université Montpellier, France
[2]Equipe d'accueil 24-15, Institut Universitaire de Recherche Clinique, Université Montpellier, Montpellier, France
[3]CHU, Département d'information médicale, Nîmes, France
[4]AMIS, Université Paul Valéry Montpellier

Mots clés : Fouille de données, Trajectoires de patients, Base PMSI, Motifs séquentiels contextuels, Visualisation

Introduction. La collecte des données hospitalières dans le cadre du PMSI (Programme de Médicalisation du Système d'Information) génère sur le plan national des bases de données de l'ordre de 25 millions d'enregistrements (séjours) par an. Ces données recueillies à des fins économiques, peuvent *a posteriori*, servir à des fins d'analyse et de recherche, pour examiner des questions médicales et épidémiologiques. L'objectif de cette démonstration est de présenter une visualisation interactive des trajectoires de patients construites à partir des évènements chronologiques fréquents ou des pathologies associées communes à des Groupes Homogènes de Malades (GHM). Les enjeux associés à ce type d'approche sont importants pour prédire l'évolution et les coûts associés à l'évolution de la santé des populations (notamment dans le cas des maladies chroniques).

Méthode d'extraction des motifs contextuels.
Considérons une base de données décrivant les différents actes médicaux (symbolisés par des lettres) effectués au cours du temps par des professionnels de santé pour des patients dans un service. Les activités peuvent correspondre dans la vie réelle à : petit-déjeuner, prise de température, bain, nouvelle perfusion, kinésithérapie, etc. Ces données sont séquentielles car elles présentent des évènements (les actes) disposés suivant un ordre (le temps). Par exemple, pour un patient P1, les actes *a* et *d* sont réalisés ensemble entre 8h et 9h puis l'acte *b* entre 10h et 11h. En examinant l'ensemble des données, nous pouvons fixer un seuil minimum (le support minimum par exemple égal à 50%) et ne rechercher que les motifs fréquents, et ainsi constater que le motif « *a* suivi plus tard par *b* » noté <(*a*)(*b*)>, vérifié par plus de 70% des patients, est alors fréquent. Cet exemple considère la base comme un ensemble indivisible pour la recherche des motifs. Pourtant, les circonstances liées aux données impliquent l'existence de sous-ensembles de données rassemblant des propriétés similaires. Pour notre cas d'étude, par exemple, nous pouvons intégrer des informations supplémentaires qui associent à chaque patient son âge (jeune ou âgé) et son genre (homme ou femme). Ces informations contextuelles peuvent avoir une influence non négligeable sur ce qui se produit dans les données et l'extraction de motifs doit rendre cette influence perceptible pour l'utilisateur afin de lui offrir une vue contextualisée des données. Nous extrayons alors des motifs spécifiques à une population (e.g. fréquent chez les jeunes et non fréquents chez les âgés) ou généraux (fréquents dans l'ensemble de la population). Pour extraire les motifs utilisés en entrée de l'outil de visualisation, nous avons utilisé l'algorithme décrit dans (Rabatel 2014).

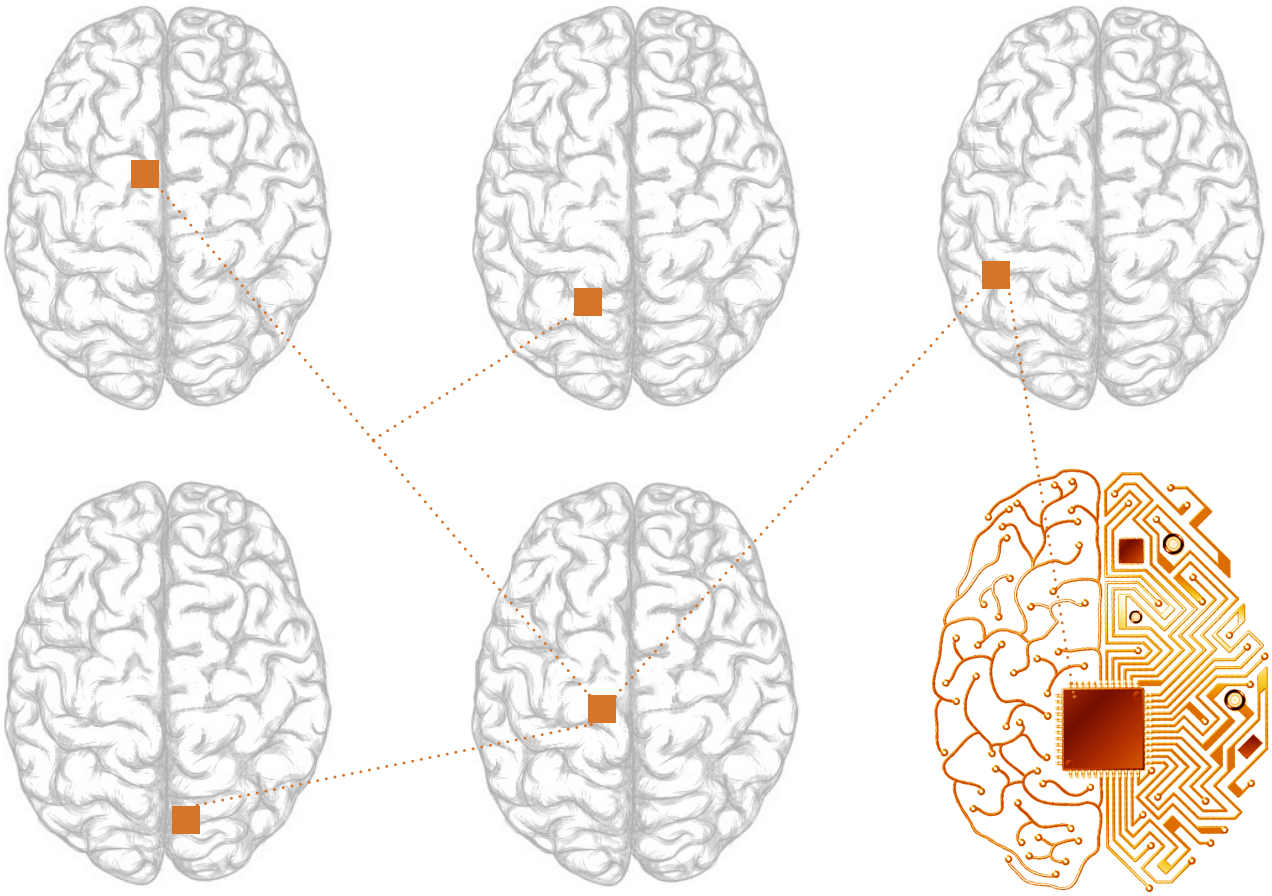
Description de l'interface.
Sur la partie gauche de l'écran, nous visualisons la hiérarchie des contextes. Chaque rond correspond à un contexte (par exemple, les hommes entre 20 et 40 ans). Le contexte le plus à gauche est le plus général et les contextes les plus à droite sont les plus spécifiques.
Lorsque l'on choisit un contexte, on fait apparaître sur la partie droite de l'écran, un graphe correspondant à l'agrégation des motifs séquentiels contextuels associés à un contexte. Les points sont des GHM, les arcs ordonnent les GHM. La taille des arcs est proportionnelle au nombre de patients concernés. Lorsque l'on choisi un GHM, on peut faire apparaître le libellé du GHM, un graphique circulaire correspond au nombre de personnes décédées et d'autres informations paramétrées par les professionnels de santé. Ces derniers peuvent interagir avec l'interface, pour fusionner des GHM qu'ils considèrent comme similaires puis générer le nouveau graphe.



Conclusion
Nous proposons dans cet article un ensemble de méthodes implantées au sein d'une interface dédiée aux professionnels de santé non informaticiens mais experts des données. Une perspective à court terme est d'exploiter l'approche présentée pour prédire automatiquement pour un patient ou un ensemble de patients les possibles évènements à venir.

Références
Julien Rabatel, Sandra Bringay, Pascal Poncelet: Mining Representative Frequent Patterns in a Hierarchy of Contexts. IDA 2014, pp. 239-250.

Touati, M., Cherif Rahal, M., Quantin, C., Le Teuff, G., Limam, M., Afonso, F., and Bataglia, G. (2006). Analyse de trajectoires hospitalières de patients atteints d'un infarctus du myocarde. (Lille), pp. 51–66.



PFIA 2015

http://pfia2015.inria.fr

Plate-forme Intelligence Artificielle

Rennes du 29 juin au 3 juillet 2015