Actes des 31es journées francophones d'Ingénierie des Connaissances

Sébastien Ferré

AfIA

Association française
pour l'Intelligence Artificielle

IC

Journées francophones d'Ingénierie des Connaissances

PFIA 2020

Table des matières

Éditorial

Les journées francophones d'Ingénierie des Connaissances (IC) sont organisées chaque année depuis 1997, d'abord sous l'égide du Gracq (Groupe de Recherche en Acquisition des Connaissances) puis sous celle du collège SIC (Science de l'Ingénierie des Connaissances) de l'AFIA. Cette année encore, IC est hébergée par la plateforme PFIA, avec plusieurs autre conférences francophones dans le domaine de l'intelligence artificielle.

L'ingénierie des connaissances peut être vue comme la partie de l'Intelligence Artificielle se préoccupant des connaissances selon les points de vue de la représentation, l'acquisition et l'intégration dans des environnements numériques. Sa finalité est la production de méthodes et outils « intelligents », capables d'aider l'humain dans ses activités et ses prises de décisions.

La conférence Ingénierie des Connaissances réunit la communauté francophone et est un lieu d'échanges et de réflexions, de présentation et de confrontation des théories, pratiques, méthodes et outils. Cette communauté doit désormais prendre en compte l'essor des algorithmes d'apprentissage et leurs retombées sur les pratiques individuelles et collectives, tout en conservant l'humain au centre des systèmes de données et connaissances. Ces approches centrées utilisateur constituent d'ailleurs le thème d'IC 2020 avec comme objectif de mettre l'ingénierie des connaissances au service des détenteurs de données.

Pour cette édition 2020 de la conférence, nous avons eu l'honneur de recevoir Oscar Corcho – professeur à l'Université Polytechnique de Madrid – qui a donné une conférence invitée intitulée "Experiences in Setting up Ontology Governance Models with Multiple Stakeholders". Concernant les contributions scientifiques, 23 articles ont été soumis. Au total 15 articles ont été acceptés et constituent le contenu de ces actes. Ces articles sont de de plusieurs types. 7 articles longs présentent des contributions originales dans les thèmes de la conférence. 4 articles sont des versions françaises d'articles déjà publiés dans de bonnes conférences internationales et contribuent ainsi à leur diffusion et à leur discussion dans la communauté francophone. Enfin, 4 articles sont des papiers courts, 3 posters et 1 démonstration, présentant soit des travaux en cours soit des outils pour l'ingénierie des connaissances.

Cette édition 2020 aura été marquée par la pandémie de Covid-19 et la virtualisation de la plateforme PFIA. Malgré cela, grâce à l'implication des auteurs, du comité de programme et du comité de pilotage IC, nous avons pu collecter dans ces actes des articles scientifiques de très haute qualité et couvrant des sujets très variés. Je remercie chaleureusement tous ces acteurs pour cela !

Sébastien Ferré

Comité de programme

Président

— Sébastien Ferré – IRISA, Univ. Rennes 1

Membres

— Marie-Hélène Abel – Sorbonne Universités, UTC
— Xavier Aimé – Cogsonomy / LIMICS UMRS 1142 Inserm
— Yamine Ait Ameur – IRIT/INPT-ENSEEIHT
— Bruno Bachimont – Sorbonne Université
— Jean-Paul Barthès – UTC
— Aurélien Bénel – Université de technologie de Troyes
— Nacéra Bennacer Seghouani – LRI CentraleSupélec
— Bertrand Braunschweig – INRIA
— Nathalie Bricon-Souf – IRIT Université Paul Sabatier Toulouse
— Sandra Bringay – LIRMM
— Patrice Buche – INRA
— Davide Buscaldi – LIPN, Université Paris 13, Sorbonne Paris Cité
— Sylvie Calabretto – LIRIS
— Gaoussou Camara – Université Alioune Diop de Bambey – Sénégal
— Pierre-Antoine Champin – LIRIS, Université Claude Bernard Lyon1
— Jean Charlet – AP-HP & INSERM UMRS 1142
— Olivier Corby – INRIA
— Sylvie Despres – Laboratoire d'Informatique Médicale et de BIOinformatique (LIM&BIO)
— Jean-Pierre Evain – EBU
— Gilles Falquet – University of Geneva
— Catherine Faron Zucker – Université Nice Sophia Antipolis
— Cécile Favre – ERIC – Université Lyon 2
— Béatrice Fuchs – LIRIS, université de Lyon
— Frédéric Fürst – MIS – Université de Picardie – Jules Verne
— Alban Gaignard – CNRS
— Jean-Gabriel Ganascia – Pierre and Marie Curie University – LIP6
— Serge Garlatti – IMT Atlantique
— Alain Giboin – INRIA
— Ollivier Haemmerlé – IRIT, Univ. Toulouse le Mirail
— Mounira Harzallah – LS2N
— Nathalie Hernandez – IRIT
— Liliana Ibanescu – UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay
— Sébastien Iksal – LIUM – Le Mans Université
— Antoine Isaac – Europeana & VU University Amsterdam
— Clement Jonquet – University of Montpellier – LIRMM
— Mouna Kamel – IRIT – Université Paul Sabatier – Toulouse
— Gilles Kassel – University of Picardie Jules Verne
— Pascale Kuntz – Laboratoire d'Informatique de Nantes Atlantique
— Michel Leclère – LIRMM (CNRS – UM2)
— Marie Lefèvre – LIRIS, Université Claude Bernard Lyon 1
— Maxime Lefrançois – MINES Saint-Etienne
— Dominique Lenne – Heudiasyc, Université de Technologie de Compiègne
— Cédric Lopez – Emvista
— Pascal Molli – University of Nantes – LS2N
— Alexandre Monnin – Origens Medialab
— Isabelle Mougenot – Université Montpellier EspaceDev
— Fleur Mougin – ERIAS, INSERM U1219 – Université de Bordeaux
— Amedeo Napoli – LORIA Nancy (CNRS – Inria – Université de Lorraine)
— Jérôme Nobécourt – LIMICS
— Nathalie Pernelle – LRI – Universit Paris Sud

— Yannick Prié – LINA – University of Nantes
— Cédric Pruski – Luxembourg Institute of Science and Technology
— Sylvie Ranwez – LGI2P / Ecole des mines d'Alès
— Chantal Reynaud – LRI, Univ. Paris-Sud, CNRS, Université Paris-Saclay
— Catherine Roussey – Irstea Clermont-Ferrand Center
— Fatiha Saïs – LRI (Paris Sud University & CNRS 8623), Paris Saclay University
— Pascal Salembier – Université de Troyes
— Karim Sehaba – LIRIS – Université Lumière Lyon 2
— Konstantin Todorov – LIRMM / University of Montpellier
— Raphaël Troncy – EURECOM
— Haifa Zargayouna – University Paris 13

MASBPL : Une méthode d'évaluation multicritère pour guider le choix d'un langage de modélisation des processus métier sensibles

Mariam Ben Hassen, Mohamed Turki, Faïez Gargouri

Université de Sfax, Institut Supérieur d'Informatique et de Multimédia de Sfax, Laboratoire de recherche MIRACL, P.O. Box 242, 3021 Sfax, Tunisie

mariem.benhassen@isims.usf.tn, mohamed.turki@isims.usf.tn, faiez.gargouri@isims.usf.tn

Résumé : Ce travail de recherche présente MASBPL, une méthode d'aide à l'évaluation et à la sélection d'un langage approprié pour la modélisation des processus métier sensibles (Sensitive Business Processes : SBP) dans une perspective de gestion des connaissances cruciales. Cette approche est basée sur l'aide multicritère à la décision. Elle est composée de quatre phases. Dans la première phase, un ensemble de « langages de modélisation des SBP de référence » est identifié. Dans la deuxième phase, ces langages sont analysés de façon approfondie. Dans la troisième phase, une famille cohérente de critères d'évaluation est construite. La quatrième phase consiste à évaluer, sur la base de ces critères, la capacité représentative des langages pour supporter parfaitement un SBP. Nous avons choisi BPMN 2.0.2 qui représente, aujourd'hui, le standard de modélisation des processus métier le plus développé et le mieux approprié pour enrichir la modélisation des SBP. MASBPL est le résultat de plusieurs expérimentations d'analyse et de modélisation des SBP dans le domaine médical dans le cadre de l'Association de Sauvegarde des Handicapés Moteurs de Sfax-Tunisie.

Mots-clés : Gestion des connaissances, modélisation des processus métier, processus métier sensible, langages de modélisation des processus, ontologies noyaux de domaine, aide multicritère à la décision.

1 Introduction

Afin d'améliorer leurs performances, les organisations modernes sont de plus en plus conscientes de la nécessité d'identifier, préserver et gérer les connaissances cruciales (individuelles et collectives) mobilisées et produites par leurs SBP. Il va falloir, ainsi, identifier, spécifier, modéliser et analyser ces processus, afin d'améliorer la gestion des connaissances de l'organisation. En effet, un SBP est un type particulier de processus métier centré sur les connaissances, les informations et les données. Il a ses propres caractéristiques qui le distinguent des processus classiques (structurés et conventionnels). Il comprend un nombre élevé d'activités critiques (individuelles et/ou collectives), mobilisant des connaissances cruciales, sur lesquelles il faut capitaliser. Également, il contient des activités à forte intensité de connaissances qui valorisent l'acquisition, la dissémination, le partage, la conversion et la création des connaissances individuelles et collectives (tacites et explicites). Ainsi, il mobilise une grande diversité de sources de connaissances consignant une masse très importante de connaissances hétérogènes. Son exécution implique la collaboration et l'interaction de nombreux participants (internes et/ou externes à l'organisation) et dépend fortement des connaissances tacites et stratégiques des experts ayant des niveaux d'expertise et de compétences hétérogènes. Ce type de processus peut être semi-structuré, structuré ou non structuré, possédant un degré élevé de complexité, de flexibilité et de dynamisme. Par ailleurs, le degré de contribution de ce processus pour atteindre les objectifs stratégiques de l'organisation et le coût de sa réalisation sont très importants (Ben Hassen et al., 2018). Ainsi, ces différentes caractéristiques de SBP rendent sa modélisation et sa représentation graphique beaucoup plus difficiles. Toutefois, le choix du meilleur langage pour la représentation des

SBP est une opération complexe vu, d'une part, la disponibilité de divers et de multitude de langages de modélisation avec des classifications et des objectifs différents et, d'autre part, l'absence d'une approche rigoureuse pour l'aide à l'évaluation des différentes classifications de langages afin de sélectionner le mieux approprié pour la modélisation des SBP.

D'ailleurs, les différents approches et langages de modélisation contemporains existants dans le domaine de modélisation des processus métier (BPM) ainsi que dans le domaine de gestion des connaissances (KM), tels que, eEPC (extended Event-driven Process Chains), RAD (Role Activity Diagram), UML AD (UML 2.0 Activity Diagrams), BPMN 2.0.2 (Business Process Modeling Notation), CMMN 1.1 (Case Management Model and Notation), BPKM (Business Process Knowledge Method), PROMOTE, GPO-WM, KMDL (Knowledge Modeling and Description Language), NKIP (Notation for Knowledge–Intensive Processes), etc. présentent des déficiences concernant leur capacité à représenter parfaitement et explicitement les particularités de modélisation de SBP (Ben Hassen et al., 2017 ; 2019). Cela nous conduit à développer des modèles de SBP ambigus et incompréhensibles.

Dans les perspectives de pallier les différentes limitations détectées et répondre de manière adéquate aux nouvelles exigences de modélisation des SBP, la spécification d'une conceptualisation rigoureuse, commune et consensuelle de SBP avec une notation de modélisation appropriée qui intègre tous les enjeux appropriés au couplage de BPM-KM dans les modèles de SBP sont d'une importance primordiale. Un tel langage devrait supporter explicitement toutes les perspectives de modélisation pertinentes de SBP, *i.e.,* les dimensions fonctionnelle, organisationnelle, comportementale, informationnelle, intentionnelle et connaissance, afin de développer des modèles de représentation graphiques complets et expressifs.

Ainsi, nous proposons, au premier abord, une spécification conceptuelle rigoureuse pour ce type de BP, organisée dans une ontologie noyau de SBP, que nous appelons « COSBP : Core ontology of Sensitive Business Process » (Cf. Figure 1). Cette ontologie est sémantiquement riche et bien fondée sur des ontologies noyaux de domaine (*e.g.,* (Kassel, 2005), (Bottazzi and Ferrario, 2009), (Kassel, 2010), (Kassel et al., 2012), (Turki et al., 2016), (Ghrab et al., 2017)). Leurs concepts sont définis à partir des concepts et des relations issus par la spécialisation de l'ontologie fondatrice DOLCE (Masolo et al., 2004). COSBP offre un référentiel de concepts et relations sémantiquement riches classés par catégorie de six modules ontologiques relatifs aux six dimensions de modélisation des SBP (Cf. Table2).

L'objectif de ce papier consiste à présenter la méthode multicritère d'aide à l'évaluation et à la sélection d'un langage de modélisation des SBP, baptisée MASBPL (Multicriteria Approach for Selecting a Sensitive Business Process Modeling Language). Notre démarche est, essentiellement, justifiée par le fait qu'aucun des langages existants ne supporte toutes les exigences de modélisation des SBP. L'analyse multicritère d'aide à la décision est alors la démarche la mieux appropriée pour leur évaluation et leur comparaison. Notre objectif consiste à guider et justifier le choix du « meilleur » langage de modélisation des SBP, afin d'améliorer l'identification et la gestion des connaissances cruciales. Soulignons que cette méthode a été initialement proposée dans un travail de recherche antérieur (Ben Hassen et al., 2019). Sur la base de plusieurs nouvelles exigences, la méthode a été révisée en ce qui concerne, notamment, son objectif, l'utilisation d'une ontologie noyau pour la conceptualisation du domaine de modélisation de SBP, l'apport de l'aide multicritère à la décision pour la construction d'une famille cohérente de critères pour l'évaluation des langages de modélisation des SBP, la nouvelle classification des langages de modélisation contemporains, ainsi que les critères d'évaluation de ces langages.

La méthode MASBPL est le résultat de plusieurs expérimentations d'analyse et de modélisation des SBP dans le domaine médical. Ces expérimentations ont été menées dans le cadre de l'Association de Sauvegarde des Handicapés Moteurs de Sfax-Tunisie (ASHMS) pour consolider et valider la méthode.

Le reste de ce papier est organisé comme suit : la section 2 expose les objectifs de la méthode MASBPL. La section 3 présente les quatre phases de la méthode proposée. La section 4 illustre l'applicabilité pratique du langage de modélisation de SBP sélectionné, en s'appuyant sur une étude de cas réelle dans le domaine médical. Nous clôturons ce papier par une conclusion en donnant les perspectives de ce travail de recherche.

2 Objectif de la méthode

MASBPL est une méthode d'évaluation multicritère pour l'aide à la sélection d'un langage approprié de modélisation des SBP dans une perspective d'identification et de gestion des connaissances (cruciales). Cette méthode vise, en premier lieu, à étendre, améliorer et consolider les différentes méthodes de repérage des connaissances centrées sur l'analyse des processus proposées par notre équipe de recherche, à savoir la méthode d'aide à l'identification des connaissances cruciales (Saad et al., 2009), la méthode de cartographie des connaissances (Ghrab and Saad, 2016), ainsi que la méthode d'aide à l'identification des processus d'organisation sensibles (SOPIM) (Turki et al., 2014). En effet, ces différentes méthodes visent à identifier, modéliser et analyser les processus sensibles, afin d'identifier les connaissances cruciales. Cependant, l'activité critique de « modélisation des processus métier sensibles » n'est ni explicitée ni étudiée en profondeur. Suite à l'analyse de ces différentes méthodes, nous notons, particulièrement, les deux limitations suivantes : (i) absence d'une caractérisation fine et une spécification rigoureuse de la notion de SBP qui intègre, notamment, la « dimension connaissance » *(e.g.,* les connaissances individuelles et collectives, les connaissances tacites et explicites, les sources de connaissances, les différentes possibilités de conversion de connaissances, etc.), et (ii) absence des langages de BPM expressifs qui supportent de manière adéquate les dimensions de modélisation des SBP. Ainsi, notre méthode vise à optimiser la phase de modélisation des SBP qui est nécessaire pour leur évaluation, leur analyse et leur amélioration, afin de repérer les connaissances cruciales. Ceci permet de réaliser des gains au niveau de la qualité des connaissances à capitaliser, le délai d'application de ces méthodes ainsi que le coût de leur mise en œuvre.

Par ailleurs, il existe une diversité et une multitude de langages de BPM avec différentes classifications proposées dans la littérature. À mesure que les modèles de SBP deviennent plus complexes, la sélection d'un langage approprié pour la modélisation des SBP est relativement critique. Ce langage devrait couvrir parfaitement tous les aspects et les exigences de modélisation fonctionnel, organisationnel, comportemental, informationnel, intentionnel et connaissance. Dans ce contexte, de nombreuses études existent proposant des méthodes et des frameworks d'évaluation et de comparaison des langages de BPM, en se basant sur des critères et dimensions d'évaluation spécifiques. Nous citons principalement : le framework d'évaluation des langages selon des méta-modèles de BP génériques (*e.g.,* (Heidari et al., 2013), (Adamo et al., 2017)) ; le framework de la qualité sémiotique (*e.g.,* (Nysetvold and Krogstie, 2005), (Scanavachi Moreira Campos and de Almeida, 2016)) ; le framework/l'ontologie BWW (Bunge-Wand-Weber) (Wand and Weber 1993) (*e.g.,* (Recker et al., 2009), (Aldin and de Cesare, 2011), (Heidari et al., 2013)) ; les patrons de workflow (Kahina and Nurcan, 2008) ; couverture des vues/perspectives de modélisation (*e.g.,* (Mili et al., 2010), (Sultanow et al., 2012)) ; représentation des aspects de KM (*e.g.,* (Businska et al., 2013), (Di Ciccio et al., 2015), (Sigmanek and Lantow, 2015)) ; représentation des interactions dans un BP collaboratif (Canché et al., 2019), ainsi que d'autres frameworks qui adoptent à la fois divers critères d'évaluation (*e.g.,* la compréhensibilité, l'expressivité, la disponibilité des outils, la gestion des exceptions, la flexibilité, l'extensibilité, la modularité, le dynamisme, l'adaptabilité) (*e.g.,* (Malekan and Afsarmanesh, 2013), (Di Ciccio et al., 2015), (Abdel-Fattah et al., 2017), (Pankowska, 2019)). Cependant, peu d'entre ces études offrent réellement une solution adéquate pour sélectionner un langage approprié ((Kahina and Nurcan, 2008), (Scanavachi Moreira Campos and de Almeida, 2016)). En revanche, nous notons l'absence de : (i) une étude comparative systématique des différents langages basée des critères d'évaluation des langages rigoureuses supportant parfaitement la modélisation des SBP. La plupart des frameworks d'évaluation proposés ne se focalisent que sur certains éléments de modélisation ou certains aspects de langages de BPM et ne supportent pas systématiquement les nouvelles exigences de modélisation des SBP (*e.g.,* la représentation des connaissances et des interactions); (ii) un langage approprié supportant les caractéristiques de SBP; (iii) une démarche d'évaluation rigoureuse qui tient compte de toutes les dimensions sémantiques aidant au choix du meilleur langage de modélisation des SBP dans un contexte de KM. Ainsi, notre méthode MASBPL vise à surpasser, en second lieu, ces

limitations. Étant donné qu'aucun des langages existants ne supporte toutes les exigences de modélisation des SBP (Ben Hassen et al., 2019), l'analyse multicritère d'aide à la décision s'impose comme meilleure démarche de résolution de ce type de problème.

3 Présentation de la méthode MASBPL

La méthode MASBPL est basée sur l'approche d'aide multicritère à la décision (Roy and Bouyssou, 1993). Elle est composée de quatre phases successives, afin de sélectionner le plus approprié pour la représentation des SBP.

3.1 Phase 1 : Définition d'un ensemble de « langages de modélisation des SBP de référence »

L'intérêt croissant pour le BPM a abouti à l'apparition d'une diversité de langages de modélisation (Ben Hassen et al., 2019). En revanche, les différentes classifications proposées ne répondent pas à nos besoins de modélisation des SBP. Ainsi, en nous basant sur les différents aspects et exigences de modélisation des SBP (y compris les aspects de KM), nous proposons deux grandes catégories de langages susceptibles de modéliser ce type particulier de BP, qui ne possèdent pas le même niveau d'expressivité. La première classe concerne les « langages de modélisation des BP/Workflows conventionnels (à usage général) » (*e.g.,* ARIS eEPC (Wagner and Klueckmann, 2006 ; Scheer, 2013), RAD (Weidong and Weihui, 2008), UML 2.0 AD (OMG, 2011), BPMN 2.0.2 (OMG, 2013), CMMN 1.1 (OMG, 2016), DMN 1.3 (OMG, 2019), etc.). La deuxième concerne les « langages de modélisation des connaissances orientés processus » (*e.g.,* PROMOTE (Woitsch and Karagiannis 2005), GPO-WM (Heisig 2006), KMDL (Arbeitsbericht, 2009), Oliveira (Oliveira, 2009), NKIP (Netto, 2013), etc.).

Compte tenu du nombre important des langages de modélisation proposés dans la littérature, il est difficile de les analyser et les évaluer en totalité. Par conséquent, il est nécessaire de définir un échantillon d'apprentissage qui inclut un nombre adéquat d'exemples représentatifs afin de sélectionner le langage le mieux approprié pour la modélisation des SBP. Pour se conformer à la terminologie utilisée en aide multicritère à la décision (Greco et al., 2001), nous appelons cet ensemble d'apprentissage, ensemble de « langages de modélisation des SBP de référence ». Le choix et la construction de cet ensemble s'effectue avec les parties prenantes et les décideurs impliqués dans le processus d'évaluation et de choix du meilleur langage pour les SBP. La liste des acteurs impliqués dans ce processus de décision inclut : l'analyste (un modélisateur, un ingénieur en système d'information ou un expert du domaine de BPM-KM), le professionnel de santé (un médecin ou un paramédical de l'ASHMS (*e.g.,* le neuropédiatre, le néonatologue, etc.), le chef de projet considéré par le processus de décision) et le comité de pilotage (Ben Hassen et al., 2019). Dans cette phase, nous avons sélectionné les langages les plus fréquemment étudiées et utilisées, à la fois, dans la littérature scientifique et dans les scénarios pratiques. Les « langages de modélisation des SBP de référence » retenus pour cette étude sont : UML 2.0 AD, RAD, eEPC, BPMN 2.0.2 (langages orientés BPM), PROMOTE, KMDL 4.0 et Oliveira (langages orientés modélisation des connaissances). Une description détaillée de ces langages est présentée dans (Ben Hassen et al., 2019).

3.2 Phase 2 : Analyse en profondeur des « langages de modélisation des SBP de références »

Au terme de cette phase, nous analysons en profondeur les « langages de modélisation de SBP de référence » retenus en étudiant leurs sémantiques, leurs syntaxes concrètes et leurs syntaxes abstraites ainsi que leurs aspects pragmatiques, afin de comprendre leurs capacités d'interprétation et de modélisation des nouvelles exigences de modélisation des SBP. Nous caractérisons les différents langages en termes de : contexte et objectif de modélisation ; constructions de modélisation, représentation du domaine de SBP et ses dimensions de modélisation ; méta-modèle (syntaxe abstraite et syntaxe concrète) ; réutilisation de concepts ; complexité structurelle ; pouvoir descriptif ; facilité d'utilisation ; accessibilité ; diffusion ; adaptabilité ; dynamisme ; extensibilité ; coût de conception, etc. (Ben Hassen et al., 2019).

3.3 Phase 3 : Construction d'une famille cohérente de critères d'évaluation des langages de modélisation des SBP de référence

Afin de construire une famille cohérente de critères d'évaluation des langages de modélisation, nous avons adopté une approche mixte à la fois ascendante et descendante. Nous rappelons que la différence entre les deux approches réside dans la façon dont les critères sont construits (Roy and Bouyssou, 1993).

Approche ascendante − tout d'abord, nous avons analysé les principaux travaux de référence du domaine de BPM-KM portant sur l'évaluation des langages de modélisation selon différents critères et selon divers courants de recherche (Cf. section 2). Cette analyse nous a permis de dégager une liste d'indicateurs (ou conséquences) relatifs aux caractéristiques des différents langages qui ont été pour nous une source d'inspiration pour la construction des critères d'évaluation. Ensuite, nous avons ressenti les limites des représentations graphiques des différentes dimensions proposées pour la modélisation des SBP (*i.e.,* la dimension connaissance, la dimension fonctionnelle, la dimension organisationnelle, la dimension informationnelle, la dimension comportementale et la dimension intentionnelle) ; c'est pourquoi nous avons procédé par une analyse ontologique du concept « Processus métier sensible ». Cette analyse nous a conduit à construire une ontologie noyau des processus métier sensibles « COSBP » (Cf. Figure 1). Cette ontologie présente des définitions rigoureuses des concepts se rapportant au domaine des SBP suivant plusieurs dimensions. Ces définitions et dimensions nous ont été très utiles lors de la construction de l'ensemble des critères d'évaluation. Finalement, nous avons profité d'un terrain d'application très riche à l'ASHMS. Nous avons organisé trois séances de brainstorming avec les professionnels de santé de l'association, à la suite desquelles, nous avons validé la liste définitive des critères (qui sont issus des différentes études réalisées dans la littérature (Cf. section 2)). Une famille de douze critères a été construite et approuvée (Cf. Table 1).

Approche descendante − nous nous sommes inspirés des trois axes du triangle sémiotique de Le Moigne (1990). Le modèle sémiotique a été sélectionné afin de construire une famille cohérente de critères selon trois points de vue (ou aspects) : (1) le point de vue syntaxique qui correspond à ce qu'est l'objet (signe) (la manifestation de l'objet), (2) le point de vue sémantique qui correspond à ce que l'objet signifie (la signification de l'objet) et (3) le point de vue pragmatique qui correspond à ce que l'objet devient (le sens et le contexte d'utilisation de l'objet). Ces différents points de vue correspondent à ce qui suit :

- **Le point de vue syntaxique** − consiste à déterminer des critères liés aux aspects purement structurels du langage (*i.e.,* éléments de modélisation et leurs relations, symboles, forme, règles de création de modèles) sans considérer le sens, dans le but d'étudier son pouvoir d'expression. Ces critères incluent : (g_1) Expressivité/Étendue des concepts ; (g_2) Notation Standard ; (g_3) Modularité ; et (g_4) Complexité structurelle.
- **Le point de vue sémantique** − concerne la signification (interprétation) intrinsèque des éléments de modélisation du langage qui est bien évidemment distincte de sa syntaxe. En d'autres termes, cette dimension indique la correspondance du langage (éléments syntaxiques abstraites ou concrètes) au domaine qu'il représente (le domaine de modélisation des SBP), dans le but d'étudier son appropriabilité et son exhaustivité pour remplir les exigences et les objectifs de modélisation de SBP. Les critères liés à cet aspect sémantique sont : (g_5) Complétude ontologique ; (g_6) Compréhensibilité et facilité d'utilisation ; et (g_7) Niveau de détails/Accessibilité.
- **Le point de vue pragmatique** − consiste à déterminer des critères liés à l'aspect dynamique du langage (son évolution). Ces critères nécessitent la prise en compte des informations concernant l'utilisation, la durée d'usage, l'environnement ou le contexte du langage dans le but d'étudier sa richesse et sa pertinence. Les critères relevant à cette dimension sont : (g_8) Flexibilité ; (g_9) Extensibilité ; (g_{10}) Niveau d'adoption ; (g_{11}) Utilisateurs cibles ; et (g_{12}) Disponibilité des outils supports.

La Table 1 présente une synthèse de la liste des critères d'évaluation des langages de modélisation des SBP. Nous avons associé à chaque critère (primitif ou composite) une échelle qualitative ordinale.

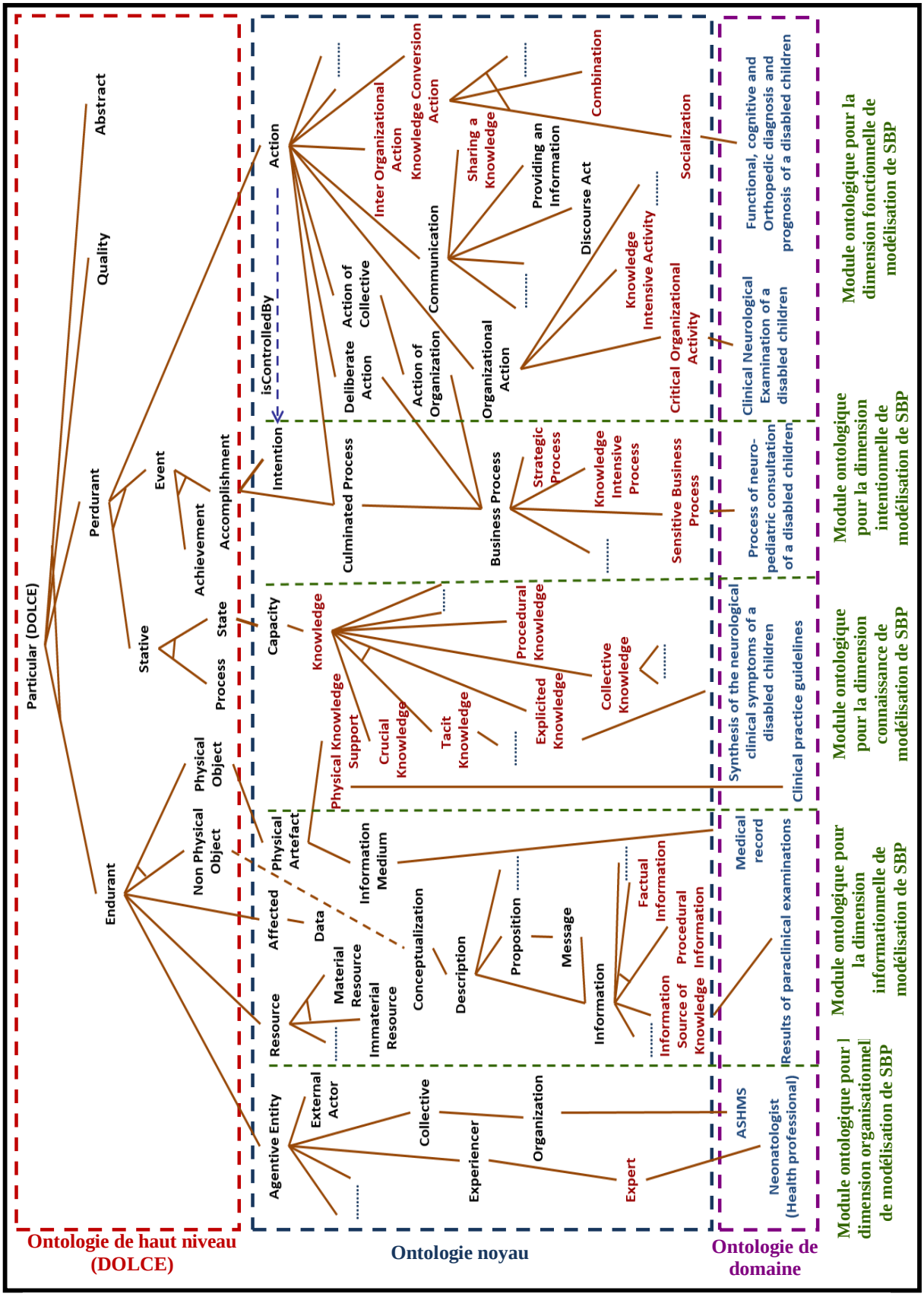


FIGURE 1 – Vue d'ensemble des modules ontologiques de COSBP

TABLE 1 – Liste des critères d'évaluation des langages de modélisation des SBP)

	Critère	Description	Codage des
	Classification des critères selon le point de vue syntaxique		
Critères simples/primitives	g_1 : Expressivité/ Étendue des concepts	Ce critère évalue, d'une part, la puissance expressive du langage pour décrire les différentes constructions du modèle de SBP. Ceci dépend du nombre d'éléments du langage et la richesse sémantique de leur bibliothèque de symboles. D'autre part, il évalue la large portée des concepts du langage, *i.e.,* l'étendue de son utilisation (*e.g.,* la documentation, la simulation et/ou l'exécution des processus).	Large/importante = 3 (si le nombre de constructions est supérieur à 40) ; Moyenne = 2 ; Faible/Limitée = 1
	g_2 : Notation Standard	Ce critère indique si le langage a été normalisé et approuvé formellement par un consortium puissant (*e.g.,* OMG). Une notation standard dispose d'un méta-modèle valide et détaillé qui explicite rigoureusement leur syntaxe abstraite	Standard=1; Non=0
	g_3 : Modularité/ Granularité	Ce critère évalue le niveau de décomposition et de structuration des représentations de SBP. Il peut être mesuré en considérant le support des processus abstraits et des sous-processus ainsi que le nombre de diagrammes requis pour une vision complète du modèle de SBP.	Élevée (plusieurs niveaux) = 3 ; Moyenne = 2 ; Faible/Limitée = 1
	g_4 : Complexité structurelle	Ce critère évalue la difficulté de modéliser et d'analyser un modèle de SBP. La prise en compte des différentes activités de SBP (*e.g.,* les activités critiques, les actions de conversion de connaissances), du contrôle des flux avancé (*e.g.,* traitement des exceptions, les flux de connaissances), l'interaction et la forte intensité des connaissances échangées entre les participants et les activités, l'évolution constante et la nature dynamique de SBP sont les principaux facteurs contribuant à la complexité des langages de BPM et des modèles de SBP.	Très complexe= 4 ; Complexe =3 ; Moyennement complexe =2 ; Faiblement complexe =1
	Classification des critères selon le point de vue sémantique		
Critères composites	g_5 : Complétude ontologique	Ce critère évalue l'expressivité et la pertinence du langage à décrire et couvrir complètement et parfaitement, d'une manière non ambiguë, tous les concepts ontologiques de COSBP relatifs aux différentes dimensions de modélisation de SBP (*i.e.,* les dimensions fonctionnelle, organisationnelle, comportementale, informationnelle, intentionnelle et connaissance).	Élevée = 4 ; Modérée = 3 ; Faible = 2 ; Très faible = 1 ; Pas de couverture = 0
	g_6 : Compréhensibilité et facilité d'utilisation	Ce critère évalue, d'une part, la compréhensibilité du langage de modélisation et les différentes expertises nécessaires pour le maîtriser et, d'autre part, la facilité d'utilisation du langage par plusieurs communautés (universitaires ou praticiens). La « facilité de compréhension des caractéristiques introduites dans un modèle », la « capacité de présenter des concepts sans erreur » et l'« étiquetage moins ambigu » constituent les principes fondamentaux pour la compréhensibilité des langages de BPM (Ben Hassen et al., 2019).	Très compréhensible = 4 ; Compréhensible = 3 ; Moyennement compréhensible = 2 ; Faiblement compréhensible=1
	g_7 : Niveau de détails / Accessibilité	Ce critère évalue le degré de détail et d'accessibilité du langage de modélisation, qui est documenté dans des formats et des normes spécifiques. Une documentation détaillée des langages ainsi qu'une représentation de haut niveau des BP/SBP (des modèles détaillés) améliorent le niveau de compréhension pour les concepteurs, les développeurs et les utilisateurs finaux.	Très élevé= 4 ; Élevé =3 ; Moyen=2 ; Faible=1
	Classification des critères selon le point de vue pragmatique		
Critères primitives	g_8 : Flexibilité/ Adaptabilité	Ce critère évalue la capacité d'un langage à supporter, d'une part, des changements (prévisibles et non prévisibles) d'une partie de BP (qui se produisent, par exemple, suite à une modification du schéma d'exécution de BP), en gardant les autres parties stables et, d'autre part, des adaptations à des situations spécifiques et des exceptions.	Flexible = 2 ; Moyennement flexible = 1 ; Faiblement flexible/Non flexible=0
	g_9 : Extensibilité	Ce critère évalue la capacité d'un langage d'être extensible (*i.e.,* possibilité de réaliser des ajouts pour des raisons d'évolutivité) afin de supporter les exigences spécifiques de modélisation des SBP (en intégrant de nouveaux concepts et symboles spécifiques au domaine).	Extensible = 2 ; Moyennement extensible = 1 ; Non extensible = 0
	g_{10} : Niveau d'adoption	Ce critère évalue la diffusion/dissémination du langage de BPM dans des communautés différents (universitaires et/ou praticiens) ayant des intérêts différents concernant le BPM	Élevé/large =3 ; Moyen/modéré =2 ; Faible/limité =1
	g_{11} : Utilisateurs cibles	Ce critère correspond aux principaux bénéficiaires du langage de modélisation. Les principaux utilisateurs sont parmi les membres du milieux universitaire ainsi que de l'industrie.	Tous les utilisateurs de diverses communautés =3 ; Certains utilisateurs =2 ; Communauté spécifique=1
	g_{12} : Disponibilité des outils	Ce critère correspond à l'accessibilité des utilisateurs à divers outils supportant le langage de BPM, ainsi qu'à la transformation en un autre format afin de tirer profit de l'échange de modèles entre les outils.	Élevée=3 (supérieur à 30) ; Moyenne=2 ; Faible=1 ; Aucune=0

3.4 Phase 4 : Évaluation des langages de référence selon les critères et sélection du meilleur langage de modélisation des SBP

L'objectif de cette phase consiste à évaluer d'une manière approfondie et rigoureuse l'expressivité et la pertinence des sept langages de modélisation étudiés (*i.e.,* UML AD, RAD, eEPC, BPMN, PROMOTE, Oliveira and KMDL) selon les différentes critères définies (*i.e.*, syntaxiques, sémantiques et pragmatiques), afin d'en choisir le mieux approprié pour les SBP.

3.4.1 Évaluation des langages de modélisation des SBP de référence sur les critères

Suite à des séances de brainstorming avec les décideurs, nous avons évalué les particularités et les spécifications de chaque langage (en se basant sur des guides d'entretien spécifiques), afin de déterminer leur conformité aux exigences de modélisation identifiées. Cette évaluation (subjective) est basée, d'une part, sur la couverture des différentes dimensions sémantiques de SBP en prenant la conceptualisation définie par l'ontologie COSBP comme une base de référence (Cf. Table 2) et, d'autre part, sur un ensemble spécifique d'indicateurs clés de performance supportant la modélisation des SBP (notation standard, compréhensibilité et facilité d'utilisation, complexité structurelle, flexibilité, extensibilité, niveau d'adoption, disponibilité des outils, etc.) (Cf. Table 1).

Faute de place, nous ne pouvons pas présenter, toutefois dans ce papier, tous les résultats de l'analyse comparative des différents aspects syntaxiques, sémantiques et pragmatiques des langages de modélisation de SBP de références (qui sont approuvés par les décideurs).

Dans la suite, nous nous limitons à la représentation des résultats de l'évaluation des langages selon le critère pivot *complétude ontologique* de notre framework d'évaluation.

3.4.1.1. Évaluation des aspects sémantiques des langages de modélisation des SBP : le critère « complétude ontologique »

Au terme de cette étape, nous avons analysé la syntaxe et la sémantique des « langages de modélisation des SBP de référence » pour comprendre leurs capacités à couvrir complètement tous les concepts ontologiques invariants de SBP définis dans l'ontologie COSBP (Core ontology of Sensitive Business Process) et qui sont relatifs aux dimensions de caractérisation et de modélisation des SBP (Cf. Figure 1) afin de construire des modèles compréhensibles, complets et rigoureux. Cette ontologie noyau (qui est basée sur DOLCE) définit une spécification conceptuelle explicite, formelle, cohérente et rigoureuse des six dimensions de modélisation des SBP sous forme de six classes de modules ontologiques (MO) (*i.e.,* MO relatif à la dimension fonctionnelle, MO relatif à la dimension organisationnelle, MO relatif à la dimension comportementale, MO relatif à la dimension informationnelle, MO relatif à la dimension intentionnelle et MO relatif à la dimension connaissance), en se basant sur une définition rigoureuse des concepts mis en jeu. Ces dimensions fondamentales du domaine de BPM-KM sont cruciales pour une conceptualisation rigoureuse et complète des SBP, permettant ainsi de bâtir des représentations riches et expressives de ces processus. COSBP fournit une sémantique commune et consensuelle des éléments de modélisation/des modèles de SBP. Dans cette étude, nous exploitons la richesse sémantique de COSBP comme étant un benchmark de référence (un méta-modèle de base) pour évaluer la pertinence des langages de BPM et des langages de modélisation des connaissances pour fournir des représentations explicites des différents concepts ontologiques relatifs aux six aspects de SBP.

La Table 2 présente une description du critère d'évaluation « complétude ontologique ». Ce critère composite est composé de six sous-critères (qui sont à leur tour composites) qui sont relatifs à la couverture des six dimensions de modélisation des SBP. Ainsi, le critère « degré de couverture d'une dimension de SBP » est évalué sur une échelle à 4 niveaux :

- *Couverture élevée* : si le langage comporte un nombre considérable de symboles pour représenter les différents concepts ontologiques de la dimension considérée (échelon 4).
- *Couverture modérée* : si le langage comporte un nombre satisfaisant de symboles pour représenter les concepts associés à la dimension considérée (échelon 3).
- *Couverture faible* : si le langage comporte un nombre faible de constructions graphiques pour représenter les concepts de la dimension considérée (échelon 2).

TABLE 2 – Description du critère d'évaluation « complétude ontologique » (critère composite)

	Critère (composite)	Description	Concepts de COSBP
Complétude ontologique/ Couverture des six dimensions de modélisation des SBP conformément à COSBP	Couverture (des concepts) de la dimension fonctionnelle	Ce critère correspond à la description des aspects structurels, des aspects de collaboration et d'interaction, ainsi que les aspects relatifs à la conversion et la création des connaissances de SBP	Deliberate Action, Individual Action, Action of Collective, Action of Organization, Organizational Informal Group, Inter Organizational Action, Organizational Action, Organizational Individual Action, Task, Organizational Unit Action, Organizational Sub Process, Communication, Receiving An Information, Providing an Information, Exchanging Knowledge, Sharing Knowledge, Disseminating Knowledge, Discourse Act, Organizational Critical Activity, Organizational Intensive Activity, Organizational Collaborative Activity, Knowledge Conversion Action, Socialization, Internalization, Externalization, Combination, Explicitation
	Couverture de la dimension organisationnelle	Ce critère correspond à la description des différents types d'entités agentives opérants dans la réalisation des activités de SBP	Agentive Entity, Collective, Organization, Informal Group, Organization Unit, Human, Agent, Experiencer, Expert, Internal Actor, External Actor
	Couverture de la dimension comportementale	Ce critère correspond à la description des aspects dynamiques de BP/SBP (les concepts qui affectent, déclenchent et contrôlent l'évolution des flux (*e.g.*, les flux de contrôle, d'information, de connaissance, etc.) durant la réalisation de SBP	Flow Element, Control Flow, Connecting Object, Association Flow, Sequence Flow, Communication Link, Message Flow, Knowledge Flow, Knowledge Association Flow, Data Flow, Data Association Flow, Information Flow, Information Flow, Conditional Control Flow, Non-Conditional Control Flow, Control Node, Exclusive Decision, Inclusion Decision, Parallel Decision, Complex Decision, Exclusive Event Based Decision, Parallel Event Based Decision, etc.
	Couverture de la dimension informationnelle	Ce critère correspond à la description des informations, des données (et leurs sources) et des conversations qui sont échangées entre les différentes entités agentives de SBP, ainsi que les entrées, les sorties, les événements, les artefacts, les ressources mobilisées et produites par les différentes activités	Information, Proposition, Instructional Description, Crucial Information, Information Source of Knowledge, Explicitable Information, Explicited Information, Factual Information, Procedural Information, Individual Information, Collective Information, Organizational Information, Internal Information, External Information, Information Medium, Physical Support, Data, Data Objet, Data Store, Message, Discourse, Informal Exchange, Artefact, Physical Artefact, Non Physical Artefact, Artefact Of Communication, Input, Output, Resource, Material Resource, Immaterial Resource, Human Resource, Event, Contingency
	Couverture de la dimension intentionnelle	Ce critère correspond à la description des principales caractéristiques de BP/SBP ainsi que des éléments contextuels impliqués dans les activités de SBP (*e.g.*, les différentes typologies de SBP, les intentions distales qui planifient, contrôlent et réalisent les actions de SBP, les objectifs à atteindre, les résultats fournis, les clients, etc.)	Intention, Objective, Distal Intention, Collective Intention, Collective Distal Intention, Organizational Distal Intention, Objective, Individual Objective, Collective Objective, Organizational Objective, Strategic Objective, Operational Objective, Culminated Process, First Level Process, Sensitive Business Process, Knowledge Intensive Process, Internal Process, Inter Functional Process, Intra Functional Process, External Process, Partial External Process, Core Process, Management Process, Support Process, Strategic Process, Operational Process, Repetitive Process, Project, Client
	Couverture de la dimension connaissance	Ce critère correspond à la description des différentes typologies de connaissances qui sont mobilisées et créées par les activités de SBP, les différentes sources de connaissances ainsi que les flux de connaissances	Knowledge, Belief State, Tacit Knowledge, Explicitable Knowledge, Explicited Knowledge, Individual Knowledge, Collective Knowledge, Organizational Knowledge, Knowledge of Organization, Organizational Unit Knowledge, Organizational Informal Group Knowledge, Propositional Knowledge, Procedural Knowledge, Strategic Knowledge, Operational Knowledge, Internal Knowledge, External Knowledge, Intra Functional Knowledge, Inter Functional Knowledge, Crucial Knowledge, Potentially Crucial Knowledge, No Crucial Knowledge, Physical Knowledge Support

- *Couverture très faible* : si le langage comporte un nombre très faible de constructions graphiques pour représenter les concepts de la dimension considérée (échelon 1).
- *Pas de couverture* : si le langage ne dispose pas de syntaxes concrètes spécifiques pour la représentation des concepts de la dimension considérée (échelon 0) (Cf. Table 5).

Chaque concept de SBP relatif à chaque dimension représente un critère primitif, qui est évalué sur une échelle à 3 niveaux de 0 à 2 : un concept de COSBP est complétement supporté (noté 2), est partiellement supporté (noté 1) et n'est pas supporté (noté 0) (Cf. Table 3).

Tous les langages examinés sont évalués et comparés selon leurs aptitudes à couvrir de manière adéquate chaque concept ontologique invariant de SBP associé à chacune de ces six dimensions. Par exemple, la Table 3 illustre les résultats de l'évaluation, en mesurant la complétude sémantique de chaque langage de modélisation pour décrire et couvrir la dimension connaissance de modélisation de SBP. Faute de place, nous ne pouvons pas présenter tous les résultats de l'analyse comparative des différents langages obtenus du point de vue du critère « complétude ontologique ». Pour une présentation détaillée de tous ces résultats, le lecteur peut se référer à nos travaux de recherche (Ben Hassen et al. 2019).

TABLE 3 – Matrice d'évaluation du degré de couverture des différents aspects de la dimension connaissance de modélisation de SBP

Concepts de COSBP	**Langages de modélisation des SBP de référence**						
	UMLAD	**RAD**	**ARIS eEPC**	**BPMN 2.0.2**	**PROMOTE**	**Oliveira**	**KMDL**
Knowledge	0	0	2 (Knowledge category, Documented knowledge Object)	0	2 (Knowledge Object/Resource model)	2	2 (Knowledge Object
Tacit Knowledge	0	0	1 (Knowledge category)	0	1	1	2 (Knowledge Object)
Explicitable Knowledge	0	0	1	0	1	1	1
Explicited Knowledge	0	0	1 (Documented Knowledge)	1 (Message, Text Annotation)	1 (Knowledge structure model)	1	1(Information Object)
Individual Knowledge	0	0	0	0	1	1	1
Collective Knowledge	0	0	0	0	1	1	1
Knowledge of Organization	0	0	0	0	0	0	0
Organizational Knowledge	0	0	0	0	0	1	1
Organizational Unit Knowledge	0	0	0	0	0	0	0
Organizational Informal Group Knowledge	0	0	0	0	0	1	1
Belief State	0	0	0	0	0	0	0
Propositional Knowledge	0	0	0	0	0	0	0
Procedural Knowledge	0	0	0	0	0	0	0
Strategic Knowledge	0	0	0	0	0	0	0
Operational Knowledge	0	0	0	0	0	0	0
Internal Knowledge	0	0	1	1	1	1	1
External Knowledge	0	0	0	1	0	0	0
Intra Functional Knowledge	0	0	0	0	0	0	0
Inter Functional Knowledge	0	0	0	0	0	0	0
Crucial Knowledge	0	0	0	0	0	0	0
Potentially Crucial Knowledge	0	0	0	0	0	0	0
No Crucial Knowledge	0	0	0	0	0	0	0
Physical Knowledge Support	1(Data Store Node)	1	1 (Documented knowledge Object/Document)	1 (Data Object, Data Store /Information Item	1	0	0
Couverture de la dimension connaissance	***1/48***	***1/48***	***7/48***	***4/48***	***9/48***	***11/48***	***12/48***

La Table 4 ci-contre mesure l'accumulation/le score d'évaluation des différents langages selon la couverture de chaque dimension de modélisation de SBP, en montrant leur capacité potentielle pour représenter tous les concepts ontologiques associés (conformément à COSBP).

TABLE 4 – Taux de couverture de chaque dimension de SBP par langage

Dimensions de modélisation des SBP (conformément à l'ontologie COSBP)	**Langages de modélisation des SBP de référence**						
	UML AD	**RAD**	**eEPC**	**BPMN 2.0.2**	**PROMOTE**	**Oliveira**	**KMDL**
Dimension Fonctionnelle	16/42	15/42	14/42	22/42	10/42	16/42	18/42
Dimension Organisationnelle	8/18	8/18	8/18	12/18	7/18	7/18	7/18
Dimension Comportementale	8/22	6/22	14/22	16/22	7/22	10/22	10/22
Dimension Informationnelle	13/56	10/56	15/56	32/56	10/56	8/56	12/56
Dimension Intentionnelle	10/34	7/34	13/34	13/34	8/34	10/34	11/34
Dimension Connaissance	1/48	1/48	7/48	4/48	9/48	11/48	12/48

3.4.1.2. Construction de la table de performance

L'objectif de cette étape consiste à construire une table de performance (matrice de décision) contenant les évaluations des aspects syntaxiques, sémantiques et pragmatiques des langages de modélisation des SBP de référence selon les critères considérés, puis choisir le mieux approprié pour améliorer l'identification et la localisation des connaissances. Cette table contient : (i) l'évaluation des degrés de complétude ontologique des langages selon la couverture des six dimensions de SBP (des exigences de modélisation fonctionnelles) conformément aux résultats d'évaluation obtenus au terme de l'étape précédente (Cf. Table 4). Ces dimensions représentent les critères {g_1, g_2, g_3, g_4, g_5, g_6}. Les résultats sont présentés sur une échelle cardinale de 0 à 4 ; (ii) les performances des langages selon les autres critères (des exigences de modélisation non fonctionnelles/des indicateurs clés de performance) {g_7, g_8, g_9, g_{10}, g_{11}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16}, et g_{17}} (Cf. Table 5). Ces résultats d'évaluation sont approuvés collectivement par tous les décideurs impliqués dans le processus de décision d'évaluation et du choix d'un langage approprié pour les SBP.

TABLE 5 – Table de performance

Critères d'évaluation	**Langages de modélisation des SBP de référence**						
	UML AD	**RAD**	**eEPC**	**BPMN 2.0**	**PROMOTE**	**Oliveira**	**KMDL**
g_1: Dimension Fonctionnelle	2	2	2	3	1	2	2
g_2: Dimension Organisationnelle	2	2	2	3	2	2	2
g_3 : Dimension Comportementale	2	2	3	4	2	2	2
g_4: Dimension Informationnelle	1	1	2	3	1	1	1
g_5: Dimension Intentionnelle	2	1	2	2	1	2	2
g_6: Dimension Connaissance	1	1	2	1	2	2	2
g_7 : Compréhensibilité et facilité d'utilisation	2	3	2	4	1	1	1
g_8: Expressivité/Étendue des concepts	2	1	2	3	1	1	2
g_9: Modularité/ Représentation de la granularité	3	1	2	3	3	2	2
g_{10}: Notation Standard	1	0	0	1	0	0	0
g_{12}: Flexibilité/Adaptabilité	1	1	1	2	0	0	0
g_{13}: Extensibilité	2	0	1	2	0	0	0
g_{14}: Niveau de détails / Accessibilité	2	1	2	3	1	1	2
g_{15}: Niveau d'adoption	2	1	1	3	1	1	1
g_{16}: Complexité structurelle	1	1	1	3	1	1	2
g_{17}: Utilisateurs cibles	2	2	1	3	1	1	1
g_{18}: Disponibilité des outils supports	3	1	1	3	1	0	1

Tout compte fait des résultats de l'évaluation, les dimensions fonctionnelle, organisationnelle, comportementale, informationnelle et intentionnelle sont mieux supportées dans la catégorie des langages de BPM conventionnels (UML AD, eEPC, BPMN) (mais avec des capacités de représentation plus ou moins faibles). En revanche, BPM est un défi pour les langages de modélisation des connaissances (PROMOTE, l'approche d'Oliveira et KMDL) qui se concentrent davantage sur la conversion des connaissances. Ces notations ont des capacités limitées pour modéliser de manière adéquate la perspective de processus et la logique /les flux de contrôle de BP, si on les compare à BPMN et eEPC. De même, elles sont inappropriées pour représenter parfaitement les aspects pertinents relatifs à la dimension connaissance (*e.g.,* la distinction explicite entre les données, les informations et les connaissances (et leurs flux) qui sont mobilisées par les différentes activités, les différents types de connaissances (*e.g.,* les aspects individuelle/collective, tacite/explicite, etc.), les différentes sources de connaissances, les possibilités de création et de partage des connaissances, etc.).

En résumé, aucun des langages ne peut couvrir complètement toutes les dimensions pertinentes de SBP conformément à l'ontologie COSBP. De la liste des langages sélectionnés, BPMN 2.0.2 semble être la notation la plus prometteuse qui satisfait le critère de complétude ontologique, offrant la couverture la plus large des concepts de COSBP (orientés modélisation métier), quoiqu'elle soit trop faible dans la modélisation de la dimension connaissance.

3.4.2 Sélection d'un langage approprié pour la modélisation des SBP

Dans une approche multicritère d'aide à la décision, l'attribution des poids aux différents critères est l'une des étapes les plus importantes du processus. En effet, les poids nous montrent l'importance de l'utilisation d'un critère. Leurs valeurs augmentent conformément à l'importance des critères (Cf. Table 6).

TABLE 6 – Importance des critères pour la modélisation des SBP

Critère d'évaluation	**Importance des critères**	**Critère d'évaluation**	**Importance des critères**
Dimension Fonctionnelle	++	Notation Standard	++
Dimension Organisationnelle	++	Flexibilité	+
Dimension Comportementale	+	Extensibilité	++
Dimension Informationnelle	++	Niveau de détails	+
Dimension Intentionnelle	+	Niveau d'adoption	-
Dimension Connaissance	++	Complexité structurelle	+
Compréhensibilité et facilité d'utilisation	+	Utilisateurs cibles	--
Expressivité/Étendue des concepts	+	Disponibilité des outils	++

Pour pouvoir donner des poids différents aux critères en fonction des exigences de modélisation des SBP, nous avons évalué, d'abord, l'importance des critères d'évaluation qui diffère d'une exigence à une autre. La Table 6 décrit notre évaluation de cette pertinence pour tous les critères considérés dans notre approche. Cette importance est représentée sur une échelle à quatre valeurs allant du « critère très important (++) » au « critère peu/faiblement (--) », avec deux valeurs intermédiaires « critère moyennement important (-) » et « critère important (+) ». Notons que cette évaluation est indépendante des langages de modélisation (Ben Hassen et al., 2019). Ensuite, nous avons déterminé et attribué les poids aux différents critères proposés en se basant sur les différentes caractéristiques de SBP. Concrètement, ces poids sont définis, en attribuant à chacun des critères qualitatifs (très important, important, peu/moyennement important, faiblement important) un poids compris entre zéro et un (Cf. Table 7).

TABLE 7 – Poids des critères

Critères qualitatifs	Très important	Important	Moyennement important	Faiblement important
Poids	0,5	0,35	0,1	0,05

En fonction des résultats d'évaluation obtenus au terme des étapes précédentes, la Figure 2 classifie les différents langages selon leur aptitude à modéliser un SBP du plus adapté au moins adapté, afin de faciliter le choix du meilleur langage. Comme nous pouvons le lire sur cette figure, le standard BPMN est le langage le plus approprié qui répond à la majorité des

critères (*e.g.*, compréhensibilité, expressivité, disponibilité des outils, flexibilité, extensibilité). Il offre une bonne variété de fonctionnalités pour supporter la modélisation des SBP.

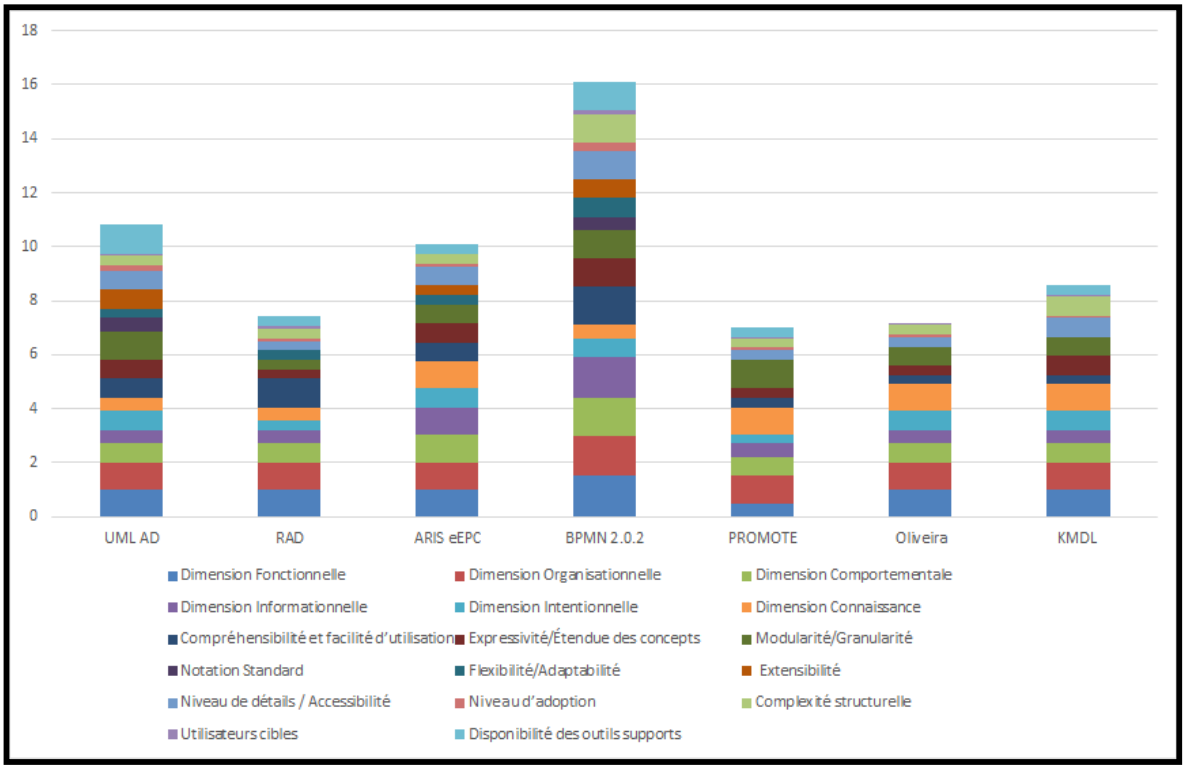


FIGURE 2 – Classement des langages selon leur aptitude à supporter la modélisation de SBP

A l'issue de cette étude, nous avons tiré profit de la richesse de la spécification BPMN 2.0 (OMG, 2013) pour la représentation des SBP afin d'améliorer l'identification et la gestion des connaissances. Cette notation fournit une panoplie d'avantages (Ben Hassen et al., 2019) :

- BPMN est un standard de BPM en pleine expansion qui est largement utilisé dans les deux communautés (universitaires ou praticiens). Par ailleurs, la définition du langage est basée sur un méta-modèle bien défini et valide (basé sur MOF). Le méta-modèle BPMN 2.0 comprend 151 méta-classes et 200 méta-associations (OMG, 2013).
- BPMN est une notation simple, facile à utiliser, aisément compréhensible, maîtrisée, communicable et accessible par tous les acteurs de l'organisation.
- BPMN est l'un des langages de BPM les plus récents qui est basé ainsi sur l'expérience des langages antérieurs, qui le rend ontologiquement l'un des langages les plus complets.
- BPMN est une notation très expressive et riche en concepts. Ses éléments de modélisation sont rigoureusement définis permettant d'avoir une couverture adéquate des différents aspects de BP/SBP (les dimensions fonctionnelle, comportementale, organisationnelle et informationnelle). Notamment, un modèle de processus d'Orchestration comprend environ 100 constructions différentes, comprenant entre autres, 5 types de sous-processus, 9 types de tâches, 6 types de passerelles, 3 marqueurs d'activités, 5 types de données, 3 types de flux de séquence et 51 types d'événements.
- BPMN supporte différents types de diagrammes (Collaboration, Conversation et Chorégraphie) permettant la modélisation des BPs complexes et interactifs.
- BPMN est actuellement la notation la plus utilisée par les praticiens de BPM et des acteurs majeurs de l'industrie IT (IBM, SAP, Oracle) (un taux de préférence > 70%).
- BPMN est supportée par plusieurs outils de modélisation puissants (plus de 80 outils diagrammatiques disponibles) (*e.g.*, Eclipse BPMN2 Modeler, BIZAGI, Aris Express, Bonitasoft) permettant son succès grandissant et son utilisation pratique échelle.

– Outre ses capacités de modélisation, BPMN possède l'avantage d'être extensible. Elle admet une approche d'extension rigoureuse, afin de satisfaire des domaines spécifiques et pallier tous ses insuffisances (notamment, la dimension connaissance de SBP).

Cependant, malgré ses capacités de représentation expressives, BPMN reste incomplète pour supporter explicitement et de manière adéquate les dimensions et les concepts centraux de SBP (conformément à COSBP) (*e.g.*, `Crucial Knowledge`, `Collective Tacit Knowledge`, `Explicited Knowledge`, `Critical Activity`, `Knowledge Intensive Activity`, `Knowledge Conversion Action`, `Expert`, etc.). Or, pour combler ces insuffisances et aboutir à une telle représentation expressive des SBP, nous proposons une extension rigoureuse de la notation BPMN 2.0.2, baptisée « BPMN4SBP » conformément au mécanisme d'extension de cette spécification. Cette extension doit enrichir et améliorer, d'une part, les dimensions fonctionnelle, organisationnelle, informationnelle et intentionnelle, et d'autre part, intégrer les nouveaux aspects de la dimension connaissance des SBP, afin de développer des modèles complets, non ambigus et compréhensibles. Par ailleurs, nous avons implémenté les extensions apportées en développant le plug-in Eclipse « BPMN4SBP-Modeler » (Ben Hassen et al., 2019).

4 Étude de cas : Processus relatif à la prise en charge médicale précoce des enfants IMC

Dans cette section, nous appliquons les concepts de SBP étendus (conformément à l'extension BPMN4SBP proposée) sur un cas réel dans le domaine médical. Nous visons évaluer l'applicabilité et la pertinence de la notation BPMN étendu pour bâtir une représentation graphique compréhensible, adéquate et expressive des SBP.

Le cadre applicatif de ce travail de recherche a été mené chez l'Association de Sauvegarde des Handicapés Moteurs de Sfax-Tunisie (ASHMS). Nous nous intéressons, particulièrement, au processus global de soins relatif à la prise en charge précoce des enfants atteints d'une Infirmité Motrice Cérébrale (IMC). Il s'agit d'un processus sensible, fortement dynamique, flexible, complexe et à haute intensité de connaissances médicales (tacites et explicitées). Il est composé de plusieurs sous-processus qui sont réalisés individuellement ou collectivement par les professionnels de santé de divers domaines métiers (sous forme d'examens médicaux et paramédicaux ainsi que des évaluations), tels que : le processus relatif à la prise en charge neuro-pédiatrique, le processus relatif à l'évaluation initiale, le processus relatif à la prise en charge kinésithérapique, etc.). Une analyse approfondie du processus de prise en charge médicale des enfants IMC est présentée dans (Ben Hassen et al., 2017 ; 2019).

Dans un tel contexte, le défi consiste à mener à bien la localisation, l'identification, le partage, l'exploitation et la gestion des différents types et modalités de connaissances médicales cruciales au bon escient, au bon moment et au bon endroit, afin d'améliorer la qualité de soins médicaux, réduire leurs coûts et prendre des décisions thérapeutiques efficaces.

Comme exemple, la Figure 3 illustre un extrait d'un modèle de représentation de SBP (simplifié) relatif au « Processus de l'évaluation (neuro-développementale) initiale d'un enfant IMC» à l'aide de l'outil BPMN4SBP-Modeler. Ce modèle est enrichi des différentes extensions que nous avons apporté à la notation BPMN 2.0.2, notamment, la dimension fonctionnelle (*e.g.*, `Critical Organizational Activity`, `Collective Action`, `Knowledge Conversion Action`), la dimension connaissance (*e.g.*, `Crucial Knowledge`, `Collective Tacit Knowledge`, `Explicited Knowledge`, `Physical Knowledge Support`), la dimension organisationnelle (*e.g.*, `Expert`, `Collective`), la dimension informationnelle (*e.g.*, `Information Source of Knowledge`, `Information Medium`), la dimension comportementale (*e.g.*, `Knowledge Flow`) et la dimension intentionnelle (*e.g.*, `Objective`, `Distal Intention`). L'élaboration de ce modèle est basée sur la validation de la communauté médicale impliquée dans la réalisation de ce SBP (le neuro-pédiatre, le néonatologue et l'orthopédiste).

Dans ce modèle de SBP, l'`Organizational Sub Process` «Évaluation de la maturation neuro-développementale d'un enfant atteint d'une IMC» `isCarriedOutBy` l'`Organization Unit` «Service Néonatologique» qui `hasForAgent` un groupe

d'Experts (le neuropédiatre, le néonatologue et l'orthopédiste) qui représentent des External Actors pour l'ASHMS. Cette action est contrôlée par (isControlledBy) une Distal Intention «Pronostic fonctionnel et orthopédique de l'enfant IMC» ayant comme contenu (hasForContent) l'Objective «Améliorer la qualité de la prise en charge précoce de l'enfant handicapé (prévention des risques d'anomalies neurologiques) ».

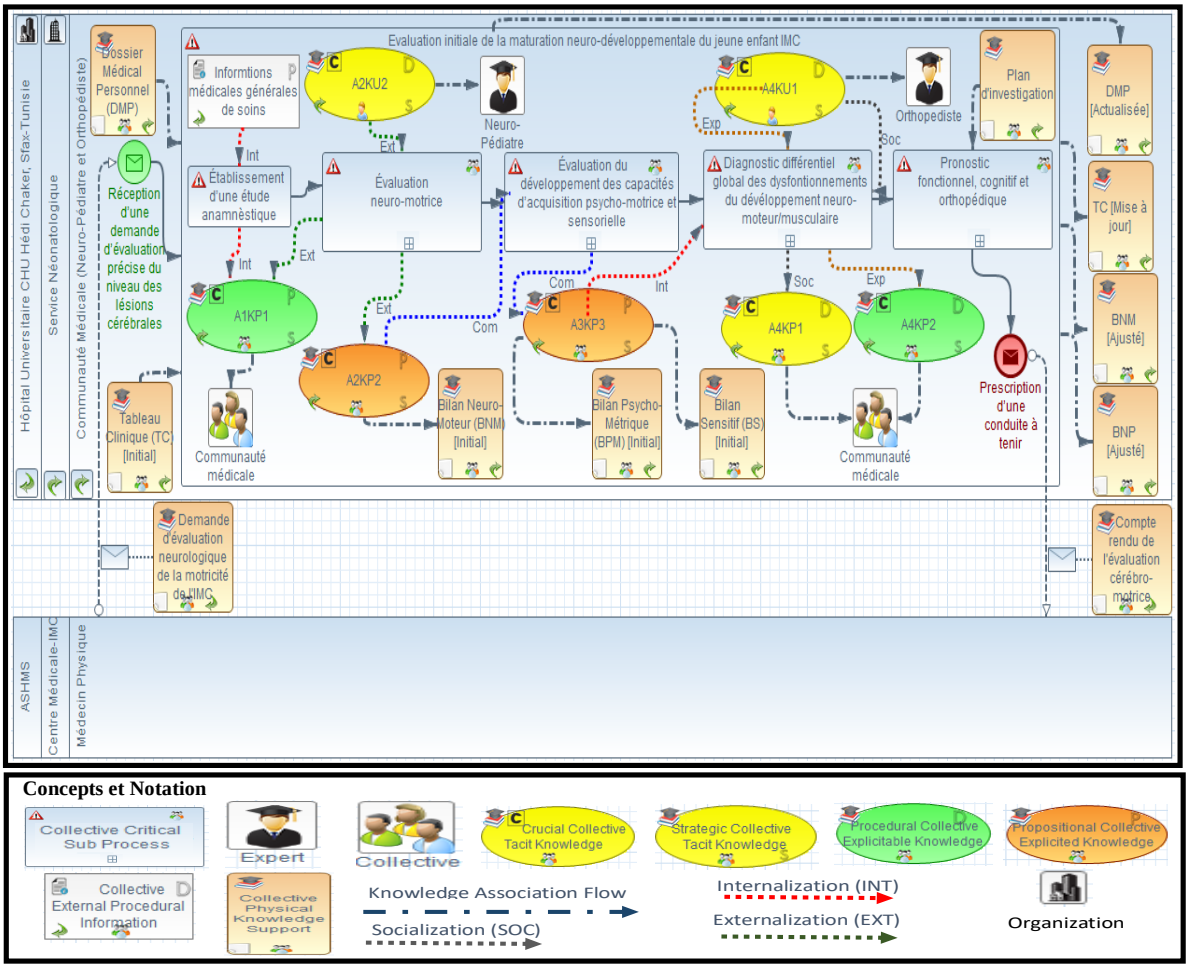


FIGURE 3 – Un extrait de modèle de représentation graphique d'un SBP relatif à l'évaluation neurodéveloppementale initiale d'un enfant IMC avec BPMN4SBP- Modeler

Par exemple, la connaissance (Knowledge) A_3K_{p3} « Synthèse de l'évaluation du développement neuro et psycho-cognitifs et neuro-sensorielles de l'enfant IMC » est produite par (isAResultOf) la Critical Organizational Activity A_3 « Évaluation du développement des capacités d'acquisition psycho-motrice et sensorielle ». A_3K_{p3} est une Collective Explicited Knowledge et elle est qualifiée à la fois comme étant une Crucial Knowledge, une Procedural Knowledge et une Startegic Knowledge. Elle est mémorisée (isBorneBy) dans les Collective Physical Knowledge Supports suivants : le bilan neuro-moteur (BNM), le bilan psycho-métrique (BPM) et le bilan sensitif (BS). Ces bilans sont consignés dans le dossier médical personnel et dans le tableau clinique (TC) du jeune enfant IMC. Ces supports physiques de connaissances sont localisés en interne dans le service néonatologique au sein du CHU Hédi Chaker. A_3K_{p3} est mobilisée par (isUsedBy) par la Critical Organizational Activity A_4 « Diagnostic différentiel global des dysfonctionnements du développement neuro-moteur et neuro-musculaire ». Il s'agit d'une activité à forte intensité de connaissances. Ainsi, A_4 est une Knowledge Conversion Action durant laquelle les différents professionnels de

santé interagissent, partagent et échangent leurs expertises et leurs compétences techniques pour prendre des bonnes décisions et atteindre des objectifs collectifs. Durant l'action de conversion des connaissances `Socialization`, l'`Individual Tacit Knowledge` «A_4K_{u1}» relative au « Jugement intuitif sur l'évolution possible des maladies neuro-musculaires » qui est détenue par l'orthopédiste est convertie et transmise en une nouvelle `Collective Tacit Knowledge` «A_4K_{p1}» qui est détenue par la communauté médicale à travers les discussions constructives sur l'évolution de l'état de santé de l'enfant.

5 Conclusion

Ce travail de recherche apporte une solution au problème de représentation graphique des SBP. Nous avons proposé une méthode d'aide à l'évaluation et la sélection d'un langage approprié pour la modélisation des SBP, baptisée MASBPL. Cette approche est basée sur l'aide multicritère à la décision. L'originalité de notre contribution réside dans la définition d'un cadre d'évaluation générique, multi-dimensionnel, systématique, rigoureux et exhaustif qui tient compte des différentes exigences de modélisation des SBP (*i.e.,* les exigences fonctionnelles et les exigences non fonctionnelles) et qui peut être appliqué à d'autre domaines de modélisation et d'exécution des différents types de BP.

La méthode MASBPL est composée de quatre phases successives : la première consiste à identifier un ensemble de « langages de modélisation des SBP de référence » (UML AD, RAD, eEPC, BPMN 2.0.2, PROMOTE, KMDL 2.2 et Oliveira). Dans la deuxième phase, nous avons analysé en profondeur ces langages. Dans la troisième phase, nous avons proposé un modèle sémiotique (Le Moigne, 1990) permettant de construire une famille cohérente de critères d'évaluation classée selon trois points de vue : les aspects syntaxiques, sémantiques et pragmatiques des différents langages de modélisation. Dans la quatrième phase, nous avons évalué rigoureusement les capacités de représentation des différents langages selon les différentes critères/dimensions définis, afin d'en choisir le mieux approprié pour les SBP. Ainsi, les résultats de l'application de notre méthode ont prouvé que le standard BPMN est le mieux approprié pour enrichir la modélisation des SBP. En revanche, nous avons proposé une extension de BPMN « BPMN4SBP » pour la spécification graphique multi-dimensionnel des SBP. Finalement, nous avons évalué l'applicabilité pratique de BPMN étendu pour bâtir des représentations expressives des SBP sur un cas réel dans le domaine médical.

Différents axes de réflexion s'ouvrent pour compléter le travail réalisé. En premier lieu, nous prévoyons d'appliquer une (des) méthode(s) d'analyse multicritère d'aide à la décision (*e.g.*, DRSA, ELECTRE TRI, ELECTRE III) (Roy et al., 1993) comme étant une phase supplémentaire de notre méthode MASBPL. L'application de ces méthodes doit permettre d'effectuer systématiquement la classification des langages des meilleurs aux moins bons selon leur pertinence pour la modélisation des SBP et, d'autre part, de valider rigoureusement notre choix de BPMN 2.0.2. En second lieu, nous envisageons compléter l'implémentation de tous les concepts définis dans les modules ontologiques de COSBP (qui sont relatifs aux six dimensions de modélisation des SBP). En dernier lieu, nous envisageons améliorer et valider notre approche dans d'autres contextes. Ce travail de recherche a révélé leurs intérêts dans le contexte d'un vrai projet dans le domaine médical. Nous visons que l'approche proposée soit applicable à d'autres scénarios réels et complexes dans un maximum de situations d'identification des connaissances nécessitant une opération de capitalisation. Nous citons, notamment, la prise en charge des situations de crises dans le contexte actuel de pandémie mondiale.

Références

Abdel-Fattah, M. A., Khedr, E., & Aldeen, Y. N. (2017). An Evaluation Framework for Business Process Modeling Techniques. I. Journal of Computer Science and Information Security (IJCSIS), 15(5), 382-392.

Aldin, L., & de Cesare, S. (2011). A literature review on business process modelling: new frontiers of reusability. *Enterprise Information Systems*, *5*(3), 359-383.

Arbeitsbericht, (umfangreicheBeschreibung) (2009) KMDL®v2.2. http://www.kmdl.de

Ben Hassen, M., Turki M., Gargouri, F. (2017). Sensitive Business Processes Representation: A Multi-Dimensional Comparative Analysis of Business Process Modeling Formalisms». In Shishkov. B. (eds) Business Modeling and Software Design (BMSD). LNBP, Vol.257, pp. 83-118, Springer.

Ben Hassen, M., Turki M., Gargouri, F. (2018). Sensitive Business Processes: Characteristics, Representation and Evaluation of Modeling Approaches. International Journal of Strategic Information Technology and Applications (IJSITA), 9(1), pp.41-77.

Ben Hassen, M., Turki M., Gargouri, F. (2019). A Multi-criteria Evaluation Approach for Selecting a Sensitive Business Process Modeling Language for Knowledge Management. Journal on Data Semantics, 8 (3), pp. 157-202.

Businska, L., Supulniece, I., &Kirikova, M. (2013). On data, information, and knowledge representation in business process models. In *Information Systems Development* (pp. 613-627). Springer, New York, NY.

Canché, M., Ochoa, S. F., Perovich, D., & Gutierrez, F. J. (2019). Analysis of notations for modeling user interaction scenarios in ubiquitous collaborative systems. *Journal of Ambient Intelligence and Humanized Computing*, 1-13.

Di Ciccio, C., Marrella, A., & Russo, A. (2015). Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. *Journal on Data Semantics*, *4*(1), 29-57.

Ghrab, S., Saad, I., Kassel, G., Gargouri, F. (2017). A Core Ontology of Know-How and Knowing-That for improving knowledge sharing and decision making in the digital age. J. of Decision Systems, 26(2), 138-151.

Greco S, Matarazzo B, Slowinski R (2001) Rough sets theory for multicriteria decision analysis. Eur J OperRes 129(1):1–47.

Gronau, N., Korf, R. and Müller, C. (2005). KMDL Capturing, Analysing and Improving Knowledge-Intensive Business Processes. Journal of Universal Computer Science, vol. 11, no. 4, pp. 452-472.

Heidari, F., Loucopoulos, P., Brazier, F., &Barjis, J. (2013). A meta-meta-model for seven business process modeling languages. In 2013 IEEE 15th Conference on Business Informatics (pp. 216-221).

Kahina, B., & Nurcan, S. (2008). Guider le choix d'un formalisme de modélisation de processus : Démarche multicritère basée sur les patrons. Actes du XXVIe congrès INFORSID Fontainebleau, pp.149-166.

Kassel G. (2005). Intégration de l'ontologie de haut niveau DOLCE dans la méthode OntoSpec, http://hal.ccsd.cnrs.fr/ccsd-00012203.

Kassel, G. (2010). A formal ontology of artefacts. *Applied Ontology*, *5*(3-4), 223-246.

Kassel, G., Turki, M., Saad, I., Gargouri, F (2012). From collective actions to actions of organizations: an ontological analysis. Symposium Understanding and Modelling Collective Phenomena (UMoCop), England.

Le Moigne J-L : *La modélisation des systèmes complexes*, Dunod, Paris, 1990

Malekan, H. S., Afsarmanesh, H. (2013). Overview of business process modeling languages supporting enterprise collaboration. In *International Symposium on Business Modeling and Software Design*. p. 24-45.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Schneider, L. (2003). *The WonderWeb Library of Foundational Ontologies and the DOLCE ontology*". WonderWeb Deliverable 18, Final Report (version 1.0

Netto, J.M, Franca, J. B. S., Baião, A. Santoro, F. M. (2013). A notation for Knowledge-Intensive Processes. The 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 190-195.

OMG, 2011. UML-Unified Modeling Language (2011) V2.4.1. Object Management Group. http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF

OMG, 2013. Business Process Modeling and Notation (BPMN). Version 2.0.2, 2013. http://www.omg.org/spec/BPMN/2.0.2/pdf

OMG, 2016. Case Management Model and Notation (CMMN). Version 1.1. http://www.omg.org/spec/CMMN/1.1

Oliveira FF (2009) Ontology collaboration and its applications.MSc dissertation, Programa de Pós-GraduaçãoemInformática, Universidade Federal do Espírito Santo, Vitória, Brazil

Pankowska, M. (2019). Business Models in CMMN, DMN and ArchiMate language. *Procedia Computer Science*, *164*, 11-18.

Roy B., Bouyssou D. (1993) Aide multicritère à la décision : méthodes et cas. Economica, Paris

Saad, I., Grundstein M., Sabroux, C., Une méthode d'aide à l'identification des connaissances cruciales pour l'entreprise. Revue Systèmes d'Information et Management (SIM), 14 (3),43–78, 2009.

Scanavachi Moreira Campos, A. C., & de Almeida, A. T. (2016). Multicriteria framework for selecting a process modelling language. *Enterprise Information Systems*, *10*(1), 17-32.

Sigmanek, C., & Lantow, B. (2015). A Survey on Modelling Knowledge-intensive Business Processes from the Perspective of Knowledge Management. In *KMIS* (pp. 325-332).

Turki M, Kassel G, Saad I, Gargouri F (2016) A core ontology of business processes based on DOLCE. J Data Semantic 5(3) :165–177.

Wagner K., & Klueckmann, J (2006). Business Process Design as the Basis for Compliance Management, Enterprise Architecture and Business Rules. In AGILITY by ARIS Business Process Management : 117–127.

Weidong, Z. and Weihui, D. (2008). Integrated Modeling of Business Processes and Knowledge Flow Based on RAD. In IEEE International Symposium on Knowledge Acquisition and Modeling, China, 49-53.

Immersion de divisions territoriales et description de leur évolution dans le Web sémantique

Camille Bernard[1], Marlène Villanova-Oliver[1], Jérôme Gensel[1], Philippe Genoud[1], Hy Dao[2]

[1] UNIV. GRENOBLE ALPES, CNRS, GRENOBLE INP, LIG
camille.bernard@univ-grenoble-alpes.fr,
marlene.villanova-oliver@univ-grenoble-alpes.fr,
jerome.gensel@univ-grenoble-alpes.fr, philippe.genoud@univ-grenoble-alpes.fr

[2] UNIV. DE GENÈVE, DÉPARTEMENT DE GÉOGRAPHIE ET ENVIRONNEMENT
hy.dao@unige.ch

Résumé :
Partout dans le monde, les découpages géographiques, dont les territoires font l'objet, sont soumis à des modifications de noms, d'affiliations, de frontières, etc. Ces changements sont un obstacle à la comparabilité des données statistiques (socio-économiques, sanitaires, environnementales...) mesurées sur ces territoires sur de longues périodes. Afin d'apporter une solution conceptuelle et opérationnelle à cette problématique, nous proposons un framework, nommé *Theseus*, qui s'appuie sur les technologies du Web sémantique pour décrire l'évolution des découpages géographiques dans le Web des données ouvertes et liées (*Linked Open Data* (LOD) en anglais). Notre approche couvre la modélisation des zones géographiques et de leurs changements au cours du temps, mais aussi la détection et la description automatique des changements, ainsi que l'exploitation de ces descriptions dans le LOD Cloud au moyen de requêtes SPARQL.

Mots-clés : Web Sémantique, Ontologies, Évolution, Données géographiques.

1 Introduction

Depuis la mise en place de directives officielles visant à l'ouverture des données créées par les institutions publiques un peu partout dans le monde, le volume du Web des données ouvertes et liées augmente continuellement. En particulier, le secteur public, les organismes gouvernementaux et notamment les instituts officiels de statistique et de cartographie déversent de plus en plus de contenus. Ces institutions publient des statistiques géo-codées à travers des découpages géographiques permettant aux responsables politiques d'accéder à des analyses fines du territoire dont ils ont la charge. Les découpages géographiques sur lesquels reposent ces données statistiques sont construits pour les besoins de la statistique, mais dérivent généralement de structures électorales ou administratives. On parle dans ce contexte de *Nomenclatures Statistiques Territoriales* (acronyme TSN en anglais). Les TSN codifient les unités géographiques qui, sur plusieurs niveaux d'imbrication (par exemple, en France les niveaux régional, départemental, communal, etc.), composent un territoire observé.

Une TSN fournit ainsi un catalogue d'Unités Territoriales (UT, également appelées domaines statistiques), souvent présentés selon une hiérarchie qui se reflète dans les codes numériques ou alphabétiques qui leur sont attribués. Une TSN est utilisée pour compiler des données statistiques sur un ou plusieurs territoire(s) étudié(s), à un ou plusieurs niveau(x) de division géographique. Les TSN sont utilisées par les instituts statistiques (locaux, régionaux, nationaux, internationaux) ou d'autres organisations pour établir et publier des statistiques comparables et intégrées dans l'espace.

Partout dans le monde, les divisions territoriales sont soumises à des modifications de noms, d'affiliations, de frontières, etc. Par exemple en 2016, en France, les régions administratives ont fusionné en grandes régions. Ces changements territoriaux sont très fréquents, d'autant plus que le maillage géographique est fin. Or, ces changements sont un obstacle à la comparabilité de données statistiques (socio-économiques, sanitaires, environnementales...)

ayant été agrégées spatialement par régions géographiques (des données ponctuelles agrégées par régions administratives, par exemple). Ce problème de comparabilité, bien connu sous le nom de *problème d'agrégation spatiale* (*Modifiable Areal Unit Problem* en anglais) (Openshaw, 1984) décrit le phénomène selon lequel des données collectées dans différents découpages géographiques (ou différentes versions d'un même découpage) ne sont pas comparables en raison des différences potentielles entre les zones géographiques observées.

Face à ces problèmes de changements territoriaux pouvant entraîner des ruptures dans les séries chronologiques, les instituts statistiques choisissent de fournir une estimation des données passées calculées pour les toutes dernières versions des découpages géographiques ou de s'affranchir des frontières territoriales, en créant des cartes de chaleur par exemple. Les traces des changements territoriaux (changements des frontières des régions, changements de noms de communes, etc.) sont alors comme effacées, ignorées, bien qu'en général, de tels changements aient un sens et soient capitaux pour comprendre un territoire. Il est alors crucial de conserver et d'enrichir ces informations relatives aux changements territoriaux, par devoir de mémoire, mais aussi pour fiabiliser les traitements des données relatives à ces territoires. D'une part, l'objectif est de fournir les connaissances permettant une lecture du territoire fidèle à la réalité du terrain et assurant une certaine traçabilité des décisions politiques. D'autre part, il s'agit d'éviter toute erreur d'analyse des données par exemple lorsque les zones géographiques auxquelles elles se rapportent changent alors même qu'elles conservent leur identifiant dans les jeux de données statistiques (dans les TSN, des unités de même code peuvent ne plus couvrir tout à fait le même espace territorial). Ainsi, un enjeu est de structurer la connaissance permettant de comprendre l'organisation et l'évolution des territoires, dans l'optique de fournir à différents acteurs (statisticiens, chercheurs, décideurs ou encore citoyens) des descriptions riches aidant, par suite, à une meilleure exploitation et une meilleure compréhension des données statistiques géo-codées.

Afin d'apporter une solution conceptuelle et opérationnelle à cette problématique, nous proposons un framework, nommé *Theseus*, qui s'appuie sur les technologies du Web sémantique pour automatiser la représentation de découpages géographiques et de leurs évolutions au cours du temps, sous forme de données ouvertes et liées (*Linked Open Data* en anglais (LOD)). Nous adoptons les technologies du Web sémantique afin de bénéficier de la base de connaissances distribuées que constitue le LOD Web (ou LOD Cloud) nous permettant d'enrichir les informations sur les changements territoriaux avec des métadonnées et d'autres ressources disponibles dans le Web pouvant contribuer à expliquer les changements (par exemple, des événements historiques). Ces technologies garantissent également l'interopérabilité syntaxique et sémantique entre des systèmes échangeant des TSN produites par les agences statistiques. Le framework Theseus se compose d'un ensemble de modules permettant la gestion du cycle de vie des TSN dans le LOD : de la modélisation des zones géographiques et de leurs changements au cours du temps, à la détection automatique des changements via un algorithme d'appariement de versions de TSN (Bernard *et al.*, 2018a), jusqu'à l'exploitation de ces descriptions dans le LOD Cloud au moyen de requêtes SPARQL. L'ensemble des modules logiciels est articulé autour de deux ontologies nommées TSN Ontology (http://purl.org/net/tsn) et TSN-Change Ontology (http://purl.org/net/tsnchange) (Bernard *et al.*, 2018b; Bernard, Camille *et al.*, 2018), conçues pour une description spatiale et temporelle non ambiguë des structures géographiques et de de leurs modifications au cours du temps. Ces deux ontologies s'appuient sur des ontologies standards telles que l'ontologie OWL-Time (pour la dimension temporelle des données) et l'ontologie GeoSPARQL (pour la représentation de la dimension spatiale des données). Les graphes de connaissances générés par Theseus améliorent la compréhension des dynamiques territoriales, en fournissant aux décideurs politiques, aux techniciens, aux chercheurs et au grand public des descriptions sémantiques fines des changements territoriaux, exploitables pour des analyses fiables et traçables.

Cet article présente l'ensemble du travail que nous avons réalisé pour la publication de TSN évolutives dans le LOD : ensemble qui prend la forme d'un framework jamais présenté jusqu'alors. Cet objectif soulève des défis qui s'inscrivent à différentes étapes du cycle de vie établi par (Villazón-Terrazas *et al.*, 2011) pour la publication des données liées gouvernementales. Dans une première partie, nous présentons ces défis et montrons en quoi des approches existantes contribuent à les relever en partie, sans toutefois y répondre pleinement.

Notre approche, qui comble les lacunes de travaux existants, est incarnée par le framework Theseus dont nous exposons les grands principes. Nous justifions des choix de modélisation retenus pour les ontologies TSN et TSN-Change qui offrent une solution inédite pour la représentation des versions de nomenclatures et la description des changements. Nous montrons comment ces ontologies ont été utilisées pour produire des graphes de connaissances sur les évolutions de trois nomenclatures territoriales officielles. Ceci illustre l'applicabilité et la généricité de notre approche pour verser sous forme de graphes RDF des divisions territoriales évolutives et la description de leurs changements dans le LOD. Enfin, nous présentons quelques unes des requêtes que nous avons prédéfinies pour interroger ces graphes de connaissances avant de conclure et de donner les perspectives envisagées pour ce travail.

2 Problématique et défis visés

2.1 Défi 1 : représenter les divisions territoriales évolutives de façon générique et interopérable

De nombreux instituts statistiques dans le monde publient désormais leurs statistiques sous forme de LOD (par exemple, les données statistiques italiennes [1], les données statistiques écossaises [2], les données statistiques du ministère britannique des communautés et des collectivités locales [3], les données statistiques aragonaises [4] ou les données statistiques japonaises [5]). Quand il s'agit d'associer une composante géographique à leurs données, ces instituts créent souvent leur propre ontologie pour la description des zones. Un état de l'art sur ces différentes initiatives est présenté dans (Bernard, 2019). Nous pouvons citer par exemple, la *Territorio Ontology* [6] de l'Institut national italien de statistique, la *Geography Ontology* [7] du gouvernement écossais, l'ontologie de l'*Ordnance Survey* d'Irlande [8] (Debruyne *et al.*, 2017). Ceci se traduit par une prolifération contre-productive de vocabulaires non alignés. L'initiative la plus approfondie est celle de l'*Office for National Statistics* (ONS) du Royaume-Uni qui propose des vocabulaires [9] pour représenter les zones géographiques dans le contexte de la publication de données statistiques. Dans cette approche, on peut noter un certain niveau d'abstraction sur les termes utilisés pour décrire les UT, ce qui permet de décrire d'autres UT que celles du Royaume-Uni. Néanmoins aucun de ces vocabulaires ne permet la description de n'importe quelle structure TSN et des niveaux qui la composent. Il est nécessaire d'ajouter de nouveaux concepts à l'ontologie Geography [10] si l'on veut décrire de nouveaux territoires et de nouvelles hiérarchies territoriales que ceux énumérés. Aucune des initiatives étudiées n'offre un niveau d'abstraction suffisant pour décrire sémantiquement n'importe quelle structure hiérarchique de type TSN. Du côté des recommandations du W3C, l'ontologie [11] RDF Data Cube (QB) est largement utilisée pour décrire des statistiques dans le LOD. Basée sur la norme ISO SDMX, elle assure une certaine interopérabilité pour l'échange et le partage de données statistiques et de métadonnées entre les organisations (Cyganiak & Reynolds, 2014). En particulier, l'extension de QB pour les composants spatio-temporels (QB4ST) [12] (Atkin-

1. http://datiopen.istat.it/index.php?language=eng
2. http://statistics.gov.scot/
3. http://opendatacommunities.org/data
4. http://opendata.aragon.es/
5. http://data.e-stat.go.jp/lodw/en
6. http://datiopen.istat.it/odi/ontologia/territorio/
7. http://statistics.gov.scot/vocabularies/
8. Ireland's Administrative boundaries ontology :http://ontologies.geohive.ie/osi#
9. http://statistics.data.gov.uk/vocabularies
10. http://statistics.data.gov.uk/graph/ontology/geography
11. http://purl.org/linked-data/cube#
12. http://www.w3.org/ns/qb4st/

son, 2017) fournit des termes canoniques pour définir la dimension spatiale et la dimension temporelle des données. Deux concepts fournissent un moyen d'exprimer des entités imbriquées (par exemple, les pays contenant des unités administratives).

Le vocabulaire QB4OLAP [13], quant à lui, permet la représentation de données multidimensionnelles, y compris, pour la dimension spatiale, la description de la hiérarchie des niveaux géographiques (Etcheverry & Vaisman, 2012). Cependant, en ce qui concerne la dimension temporelle des données, il s'agit d'une approche discrète qui ne représente pas les processus d'évolution des éléments dans le temps, ni les changements et événements sous-jacents. Dans (Plumejeaud *et al.*, 2011), nous avons proposé un modèle de base de données relationnelle offrant un niveau d'abstraction suffisant pour décrire sémantiquement toute TSN. Ce modèle constitue un socle pertinent pour notre objectif mais doit être adapté en vue d'une immersion dans le Web sémantique. Par suite, visant le Web des Données Ouvertes et Liées, il doit être possible de référencer chaque UT avec précision. Ceci prend tout son sens dans le contexte de représentation de la connaissance qui est le nôtre, des connaissances relative aux évolutions des territoires. Ici, la notion d'UT doit être rapprochée de celle de *version d'UT*, elle-même liée à la notion de version de TSN. Cela nécessite de créer des identifiants uniques (URI dans le LOD) pour chaque ressource que constitue chaque représentation de ce qu'a été une UT au cours de sa vie (*i.e.*, chaque version d'UT). Ainsi, dans une perspective applicative, nous visons un modèle permettant à des producteurs de données de déclarer précisément pour chaque observation ou mesure quelle est exactement la version de l'UT concernée, et à quelle division géographique (hiérarchie) elle appartient (*i.e.*, version de TSN).

2.2 Défi 2 : représenter finement les changements territoriaux

En ce qui concerne la nature évolutive des zones géographiques couvertes par les observations et les TSN auxquelles ces zones évolutives appartiennent, aucune des ontologies décrites ci-dessus ne fournit un vocabulaire suffisamment générique et riche pour décrire la façon dont n'importe quelle TSN, mais aussi les niveaux territoriaux et les UT qui la composent, évoluent au fil du temps. Pourtant, représenter finement le changement territorial vise à en comprendre les raisons et à garantir le transfert fiable de données statistiques d'une version de TSN à l'autre. Il manque donc un modèle qui puisse aider à savoir comment un territoire a évolué dans le temps, un pré-requis essentiel à une interprétation et une compréhension plus justes des valeurs statistiques observées ou mesurées sur un territoire évolutif. La plupart des contributions des agences limitent leur description à des changements d'attributs isolés et ne proposent aucun vocabulaire pour décrire et regrouper les changements qui interviennent lors d'un même événement.

Parmi les initiatives déjà évoquées, nous revenons ici sur deux d'entre elles qui traitent du problème de l'évolution dans le temps :

— le Royaume-Uni (ONS) propose un vocabulaire [14] pour représenter l'évolution des zones géographiques dans le cadre de la publication de données statistiques. Néanmoins, aucun des concepts de l'ontologie des changements de frontières de l'ONS ne permet de nommer les changements qui ont un impact sur plusieurs UT en même temps. Les concepts décrivant les événements de changement sont peu nombreux et se limitent aux termes *Recoding change*, *Boundary change*.

— Les vocabulaires du Bureau des statistiques du Japon [15] représentent les zones géographiques et leur évolution, y compris la description des changements de zone géographique, dans le contexte de la publication de données statistiques. Cette proposition se limite à la description des changements dans les municipalités, par un simple littéral.

Ainsi, les changements intervenant dans les structures des TSN sont rarement décrits et lorsqu'ils le sont, les descriptions sont faites UT par UT, sans lien entre les changements. Il est donc difficile d'identifier chacun des composants de la TSN (territoires, niveaux et UT)

13. http://purl.org/qb4olap/cubes

14. http://statistics.data.gov.uk/def/boundary-change

15. http://data.e-stat.go.jp/lod/sac/, http://data.e-stat.go.jp/lod/terms/sacs#, http://data.e-stat.go.jp/lod/sace/

qui change. Il manque également un support pour décrire des changements qui se propagent de niveau en niveau (par exemple, le changement des limites d'une UT peut également avoir un impact sur les limites de ses sous-UT). Compte tenu de la nature imbriquée des éléments d'une TSN, cet aspect est un élément favorisant une compréhension plus globale d'une évolution territoriale observée selon une approche *top-down* ou *bottom-up*.

La littérature sur la représentation de l'évolution territoriale propose des solutions pour l'expression de lien de filiation (Del Mondo *et al.*, 2010; Harbelot *et al.*, 2013) entre unités géographiques et la caractérisation du changement (à travers des typologies (Claramunt & Thériault, 1995; Del Mondo *et al.*, 2010; Plumejeaud *et al.*, 2011; Del Mondo *et al.*, 2013) et des structures de représentations dédiées (Kauppinen *et al.*, 2008; Lopez-Pellicer *et al.*, 2012; Lacasta *et al.*, 2014)). Ces travaux ont inspiré notre contribution visant à offrir un cadre conceptuel pertinent pour une représentation sémantiquement riche de l'évolution. Les modèles ontologiques que nous proposons répondent ainsi aux défis 1 et 2 que nous venons de présenter. Ils sont décrits dans (Bernard, Camille *et al.*, 2018; Bernard, 2019) et brièvement rappelés dans la Section 4.

2.3 Défi 3 : automatiser la description des changements territoriaux

Dans (Lacasta *et al.*, 2014), les auteurs proposent un processus semi-automatique pour créer et alimenter le modèle représentant de l'évolution d'unités administratives correspondant à des Domaines Juridictionnels proposés dans (Lopez-Pellicer *et al.*, 2012). Le processus repose sur une approche qui permet de définir les caractéristiques des sources de données à intégrer afin de gérer les Domaines Juridictionnels de n'importe quel pays. Toutefois, pour peupler le modèle ontologique, il est nécessaire d'utiliser en entrée un dictionnaire énumérant tous les changements individuels. Cette contrainte est également observée dans le travail de (Kauppinen *et al.*, 2008). Ceci constitue un inconvénient majeur pour les systèmes d'information statistique actuels, car l'établissement manuel de la liste des changements observés dans toute une nomenclature est coûteuse en temps.

D'autres approches telles que (Plumejeaud, 2011) et (Harbelot *et al.*, 2015) consistent à développer des algorithmes et programmes afin de *détecter* les changements entre deux versions, au lieu de les retranscrire à partir d'une liste préétablie. Détecter un changement entre deux UT puis le décrire sémantiquement nécessite de traiter la question de l'identité des UT. D'une part, pour qu'un changement soit décrété, il faut définir quelles sont les variations à observer pour conclure à un changement. Ensuite, selon ses variations, une seconde décision doit être prise concernant l'impact de ce changement. En effet, un changement peut être considéré comme suffisamment important en regard de l'identité d'une UT pour que soit questionnée la continuité de l'existence-même de cette UT. Par exemple, une UT, dont le nom ou la géométrie ou les deux sont modifiés, reste-elle la même UT après ces changements ? Il n'y a pas de réponse unique à cette question, car la définition de ce qui constitue l'identité d'une zone géographique varie d'une nomenclature à une autre. Il s'agit donc d'adopter une approche configurable pour répondre aux différents besoins. Des questions en termes de performance des algorithmes sont aussi à considérer puisque, s'agissant de détecter des changements dans des nomenclatures territoriales, les calculs vont impliquer des géométries à comparer.

A partir de ces trois défis, nous avons fait le constat qu'un système paramétrable était à développer pour que des utilisateurs tels que des agences de statistique puissent facilement, et en fonction des caractéristiques des nomenclatures qu'elles gèrent, créer de nouvelles versions de leur TSN, documenter leurs changements, déverser cette connaissance dans le Web LOD et l'exploiter (référencement, requêtes). Dans ce but, nous avons conçu et développé un framework, nommé Theseus présenté dans la partie suivante.

3 Le framework Theseus

3.1 Vue d'ensemble

Theseus [16] est un framework conçu et développé pour gérer l'ensemble du cycle de vie des TSN : de la modélisation des données à leur exploitation dans le LOD. Lors de la conception de ce framework, nous suivons les recommandations du W3C pour la publication de données liées (Hyland *et al.*, 2014) et l'approche de (Villazón-Terrazas *et al.*, 2011) pour la publication de données gouvernementales sous la forme de données liées. La Figure 1 illustre les cas d'utilisation auxquels Theseus permet de répondre (Bernard, 2019).

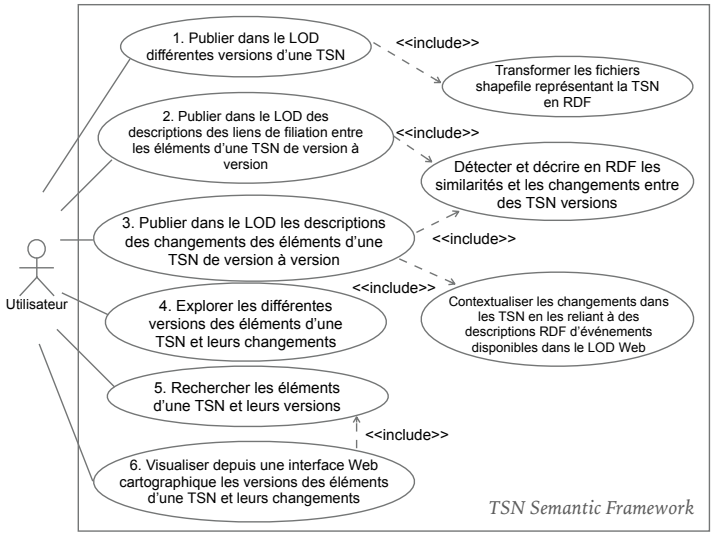


FIGURE *1 – Cas d'utilisation du framework Theseus.*

Ce framework automatise la publication dans le LOD de toute TSN et de ses différentes versions, la détection des changements qui impactent leurs unités et la publication de ces changements dans le LOD. Il encapsule deux ontologies que nous avons conçues afin de décrire les TSN et leurs évolutions dans le temps : les ontologies TSN-Ontology et TSN-Change Ontology (voir Section 4). Il encapsule également une implémentation, appelée *TSN Semantic Matching Algorithm* (Bernard *et al.*, 2018a), de l'algorithme de (Plumejeaud *et al.*, 2011) qui détecte et décrit les similitudes et changements entre deux versions consécutives d'une TSN. L'algorithme original a été modifié pour produire des descriptions RDF d'une TSN et de ses changements dans le temps, sur la base des concepts définis dans l'ontologie TSN-Change.

Ce framework sémantique est, à notre connaissance, le premier à assurer la gestion de versions de structures de type TSN. De plus, bien plus que de simplement lier les éléments d'une TSN à travers les versions, Theseus fournit aux utilisateurs des descriptions sémantiques riches des changements des UT au fil du temps.

3.2 Pré-requis et portée

Le Framework Theseus prend en entrée plusieurs fichiers géospatiaux (fichiers au format shapefile ESRI [17]), un pour chaque version de TSN, et les transforme en graphes RDF.

16. Ce nom a été choisi en référence à la question philosophique identitaire soulevée par le bateau de Thésée reconstruit entièrement au fil des ans, ses planches ayant été cassées et remplacées les unes après les autres, soulevant alors une question relative à l'identité de ce bateau, préservée ou non alors que le bateau est été entièrement reconstruit ?

17. https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

Concernant la description des changements, qui opère en partie sur la géométrie des UT, une condition préalable pour des résultats optimaux est que toutes les géométries soient de même niveau de généralisation et dans le même référentiel spatial. Cela a en effet un impact sur la pertinence des résultats comme discuté dans (Bernard, 2019).

Dans cet article, nous montrons comment le framework permet de décrire les changements de nature géographique et liés à une nomenclature. Il s'agit donc de changements portant par exemple sur la géométrie, la surface, l'emplacement d'une capitale, l'organisation de la structure spatiale, les informations toponymiques, etc. Plus largement, il est possible de faire évoluer le framework pour permettre la détection de changements liés à d'autres informations, qualifiées de thématiques, qui seraient associées à une UT (telles que son nombre d'habitants, ou tout autre mesure la concernant).

Comme évoqué précédemment la question identitaire est centrale pour qualifier finement le changement. Il est possible de paramétrer au sein du framework une définition de ce qui constitue l'identité des UT de la TSN traitée. Cette définition apparaît sous la forme d'une liste d'attributs d'UT pondérés. Cette définition est établie par un expert de la nomenclature qui détermine d'abord quels attributs doivent être considérés lors de la comparaison des UT de version à version et quelle importance ont ces attributs dans la définition de l'identité des UT, via l'attribution de poids à chacun des attributs de la liste. Ainsi, il est par exemple possible de définir l'identité d'une UT comme étant constituée à 40% de son code dans la nomenclature, à 40% de sa géométrie, à 10% de son nom et à 10% de son unité englobante.

Cette souplesse répond à l'hétérogénéité avec laquelle est définie l'identité dans les nomenclatures. Dans certaines TSN, le nom d'une UT est primordial et si ce nom change, l'UT n'est plus considérée comme la même UT, bien que sa géométrie soit restée la même. Au contraire, dans une autre TSN, bien que le nom ait changé, l'UT est considérée comme "la même UT" car ses frontières sont inchangées. La liste définissant l'identité des UT dans la TSN est un paramètre en entrée d'une fonction de calcul de similarité, implémentée au sein du framework pour calculer automatiquement si l'identité d'une UT est préservée dans la version suivante de la nomenclature.

Deux UT sont considérées comme similaires si la fonction renvoie un score de correspondance supérieur à un seuil global, fixé par l'expert de la nomenclature qui estime le taux de variation admis, permettant de conclure que l'UT persiste malgré ces variations. Cela se traduit alors par un lien de filiation de type *Continuation* entre les deux UT : l'identité de l'UT est conservée dans la version suivante de la nomenclature.

3.3 Architecture globale du framework

Le Framework Theseus est composé de plusieurs modules organisés en quatre niveaux fonctionnels comme illustré par la Figure 2. Les niveaux correspondent aux étapes de la procédure établie par (Villazón-Terrazas *et al.*, 2011) pour la publication de données gouvernementales dans le LOD Cloud :

1. Spécifier et modéliser : au cœur même du Framework Theseus se trouve son modèle de données, composé des ontologies TSN et TSN-Change. Nous avons spécifié ce modèle selon la méthodologie pour la conception d'ontologies exposée dans (Bachimont *et al.*, 2002) : un corpus de TSN et d'ontologies existantes (présenté dans (Bernard, 2019)) nous a aidés à déterminer les concepts et le vocabulaire à utiliser pour être proche des désignations utilisées notamment par les agences produisant des statistiques géocodées. Le modèle ontologique TSN/TSN-Change fournit aux modules du framework une représentation formelle des TSN et de leur évolution dans le temps. Il est brièvement mis en regard de l'état de l'art et expliqué dans la Section 4 .
2. Générer : pour la génération de données RDF à partir des ontologies TSN et TSN-Change, plusieurs modules logiciels sont utilisés. Le logiciel *Geotriples* vise à transformer les fichiers géospatiaux (shapefile) TSN en triplets RDF, en utilisant les concepts définis par l'ontologie TSN. Les modules *TSN Change Detector* et *TSN Change annotator*, que nous avons développés pour implémenter l'algorithme TSN Semantic Matching, détectent les changements dans les géométries et autres attributs, et décrivent

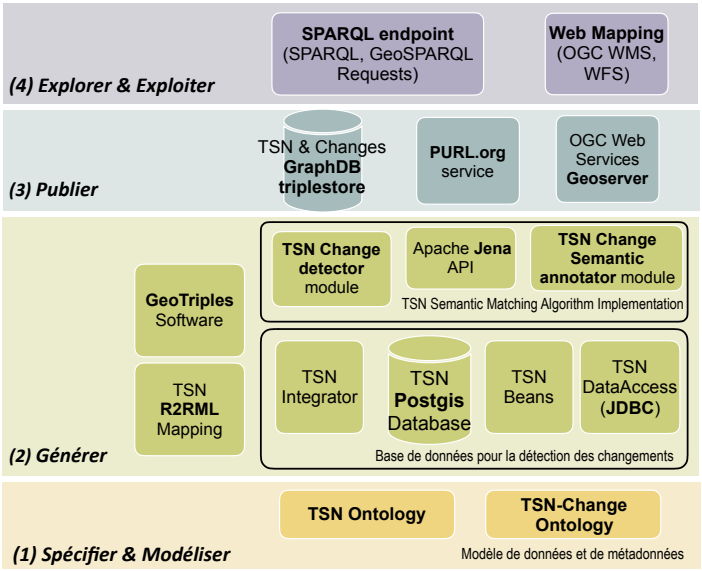


FIGURE 2 – *Les modules du framework Theseus.*

ces changements en RDF via les concepts de l'ontologie TSN-Change. Ce sont notamment ces modules qui ont permis de produire les connaissances relatives à trois nomenclatures territoriales comme expliqué dans la Section 5.

3. Publier : à cette étape, trois logiciels existants sont utilisés : le service purl.org (pour fournir un URI persistant aux données) ; le triplestore GraphDB (pour publier les données RDF créées) ; le logiciel GeoServer (pour publier les géométries des UT liées à leur représentation LOD).
4. Explorer et exploiter : afin de fournir aux utilisateurs des outils pour explorer et exploiter les données publiées, nous avons mis en place divers moteurs de recherche, en utilisant des mécanismes Web standard : un point d'accès SPARQL pour effectuer des requêtes SPARQL et GeoSPARQL sur les données RDF (disponible depuis l'URL http://steamerlod.imag.fr/sparql) ; une interface utilisateur (UI) pour visualiser et interroger les données géospatiales, via des requêtes OGC standard aux Web services WMS et WFS générés par une instanciation du logiciel GeoServer se connectant à la base de données TSN PostGIS de notre framework.

4 Représenter des nomenclatures territoriales et leurs évolutions

4.1 Etat de l'art

La représentation de la dynamique des territoires nécessite de modéliser les entités et les relations spatiales, spatio-temporelles et de filiation entre elles (Del Mondo *et al.*, 2010). La modélisation des entités spatio-temporelles, dont on souhaite capturer l'évolution, repose sur des fondamentaux (voir (Bernard, 2019), Chapitre 3) que nous rappelons brièvement car ils constituent un cadre utile à la compréhension de l'approche décrite dans cet article.

Parler d'évolution requiert de déterminer ce qui fait qu'une entité est toujours ou non la même, dans le temps, donc ce qui constitue son identité. La question de l'identité des entités spatio-temporelles est discutée dans les travaux de (Hornsby & Egenhofer, 2000; Harbelot *et al.*, 2013).

Par ailleurs, des typologies de processus spatio-temporels ((Claramunt & Thériault, 1995; Del Mondo *et al.*, 2010; Plumejeaud *et al.*, 2011; Del Mondo *et al.*, 2013)) introduisent des termes pour la description de changements observables d'entités (comme par exemple, une

fusion d'unités pour en créer une nouvelle, ou à l'inverse une *scission*, donnant lieu à de nouvelles entités à partir d'une autre). Ces travaux définissent également une sémantique pour caractériser les liens de filiation entre entités, selon qu'il s'agit de relation de *Continuation* (une entité subit un changement mais continue d'être cette même entité) ou de *Dérivation* (la modification d'une entité entraine la perte de son identité, ce n'est alors plus la même entité suite au changement).

Du côté des ontologies de haut niveau, l'ontologie BFO (Basic Formal Ontology (Grenon & Smith, 2004)) offre un cadre théorique pour la représentation d'entités et de leurs évolutions. BFO se structure en deux sous-ontologies exploitées à ces fins : les ontologies pour *continuants* aussi appelées *SNAP* permettant de représenter des entités qui ont une existence continue et une capacité à supporter (persister de façon identique) à travers le temps même en subissant différentes sortes de changements (par exemple, un personne, la planète Terre) ; et les ontologies pour *occurrents* aussi appelées *SPAN* permettant de décrire des processus ou des événements (par exemple, un sourire, le passage d'une tempête de pluie sur une forêt) (Grenon & Smith, 2004).

Les concepts introduits brièvement ci-dessus se retrouvent dans plusieurs modèles de la littérature visant à représenter l'évolution dans le temps d'unités territoriales. Différentes approches s'inscrivent dans le Web sémantique comme nous le faisons. Elles traitent d'espaces géographiques particuliers comme de parcelles de couverture terrestre (Harbelot *et al.*, 2013, 2015), de régions historiques (Kauppinen & Hyvönen, 2007; Kauppinen *et al.*, 2008), ou encore de domaines juridictionnels ou régions administratives (López-Pellicer *et al.*, 2008; Lopez-Pellicer *et al.*, 2012; Lacasta *et al.*, 2014)). Dans (Bernard, 2019) un état de l'art de ces différentes approches ontologiques est proposé. Il analyse notamment les travaux dont s'est nourrie notre proposition d'un système basé sur les deux ontologies TSN et TSN-Change. Nous rappelons dans la section suivantes leurs grand principes.

4.2 Les Ontologies TSN et TSN-Change

Notre proposition s'inspire tout d'abord des systèmes de gestion de version utilisés dans le développement logiciel et de la façon dont ces systèmes gèrent les changements (Redmond *et al.*, 2008; Völkel & Groza, 2012).

Notre modèle ontologique est, au sens de (Grenon & Smith, 2004), une Trans-Ontologie SNAP-SPAN. Elle décrit les entités spatiales et leur structure, et dépeint leur vie (ou leur histoire) dans le temps, en gérant à cet effet respectivement les vues SNAP et SPAN.

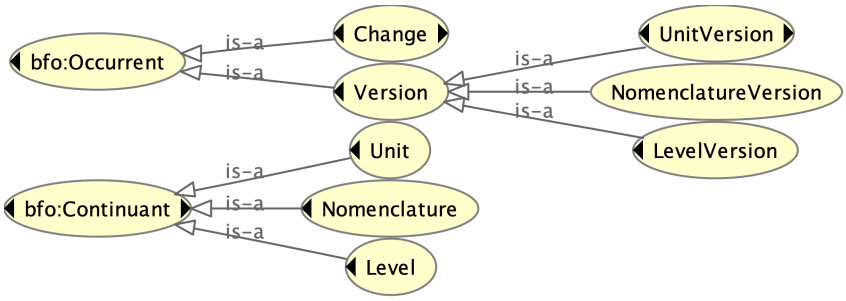


FIGURE *3 – Principaux concepts du Modèle Ontologique TSN hérités de l'ontologie BFO (Concepts Occurrents et Continuants de (Grenon & Smith, 2004)).*

D'un côté, une TSN et ses composants sont des entités continues (voir Figure 3) : les concepts de l'ontologie TSN *Unit*, *Nomenclature*, *Level* héritent du concept *Continuant* de BFO. De l'autre, tous les composants versionnés d'une TSN et les changements qu'ils subissent sont des entités occurrentes (qui dépendent des objets continus précédemment cités dont elles sont des versions). Ensemble, les composants versionnés et les changements dépeignent la vie des UT au fil du temps. Ces tranches de vie reposent sur des constructions

4D conformément à l'approche perdurantiste de l'ontologie pour les *fluents* (Welty *et al.*, 2006) aussi exploités dans (Harbelot *et al.*, 2015) et dont nous reprenons les principes (voir Figure 4, *cadre 1*). Cependant, là où les ontologies pour les *fluents* utilisent le terme "tranche de temps" *Slice*, nous préférons utiliser le terme "version", pour être aussi proches que possible du vocabulaire des statisticiens qui constituent des utilisateurs cibles (Figure 4, *cadre 3 présentant notre approche*).

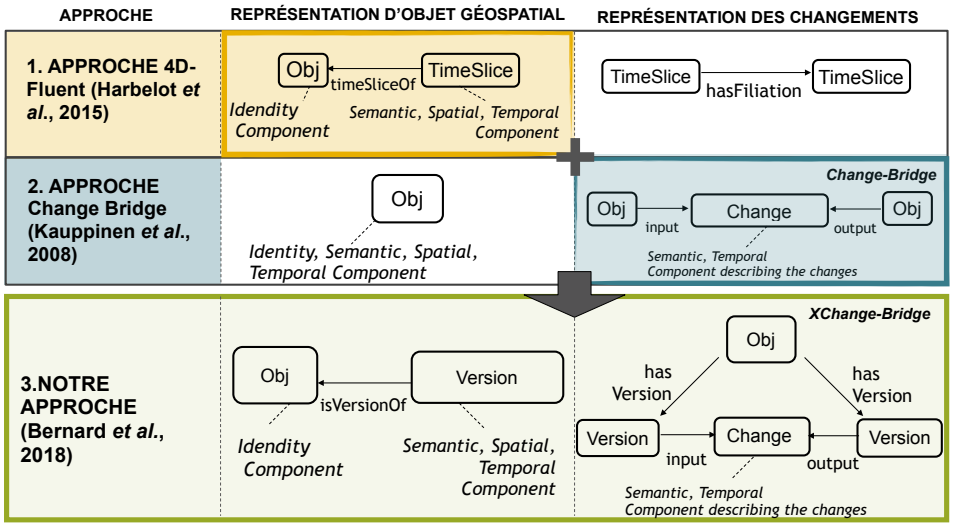


FIGURE *4 – Combinaison d'approches à l'origine du modèle ontologique TSN/TSN-Change*

Nous adoptons également l'approche Change Bridges de (Kauppinen *et al.*, 2008), pour gérer l'union des versions successives (Figure 4, *cadre 2*). Au sein de ces ponts de changement (que nous appelons *XChange-Bridges*, *X* pour *eXtended*), nous décrivons les différences entre deux versions et caractérisons la nature des changements territoriaux, en utilisant une typologie des changements basée sur les typologies de (Claramunt & Thériault, 1995) et (Plumejeaud *et al.*, 2011) que nous étendons afin de décrire des changements de frontières des territoires (par exemple, l'élargissement de l'UE). La structure d'un changement territorial XChange-Bridges est présentée dans la figure 5. Comme (Plumejeaud *et al.*, 2011), nous considérons qu'un changement est rarement isolé et indépendant des autres changements qui se produisent simultanément au sein des autres unités à l'intérieur d'une zone donnée. Sur la base de cette observation, nous créons des graphes de changement territorial à plusieurs niveaux qui décrivent et relient des changements concomitants qui ont un impact sur plusieurs niveaux de la TSN. Ainsi, nous fournissons aux analystes une représentation détaillée d'un changement territorial dans une TSN, via les informations suivantes :

— la connaissance des versions d'UT impactées. Le prédicat *tsnchange :input* (propriété inverse *tsnchange :before*) pointe vers un *TSNComponent* (*LevelVersion* et / ou *UnitVersion*) qui change ; le prédicat *tsn-change :output* pointe vers un *TSNComponent* créé ou modifié après l'événement de changement.
— des liens pour découvrir les conséquences d'un changement. Par exemple, un changement de frontières d'une UT sera relié aux changements affectant ses sous-UT. Le prédicat *tsnchange :lowerChange* (propriété inverse *tsnchange :upperChange*) permet de relier des changements affectant l'élément courant à des changements subis par des sous-élément à l'élément courant.
— une représentation détaillée des causes de ce changement en le reliant à des ressources dans le LOD cloud (événements historiques par exemple). Le prédicat *isCausedBy* indique les raisons contextuelles du changement (à condition que ces informations soient disponibles dans le LOD, dans DBpedia par exemple). Nous recommandons l'utilisation de l'ontologie Linking Open Descriptions of Events (LODE) (Shaw *et al.*,

2009) pour la représentation des événements (historiques) causant les changements des UT.

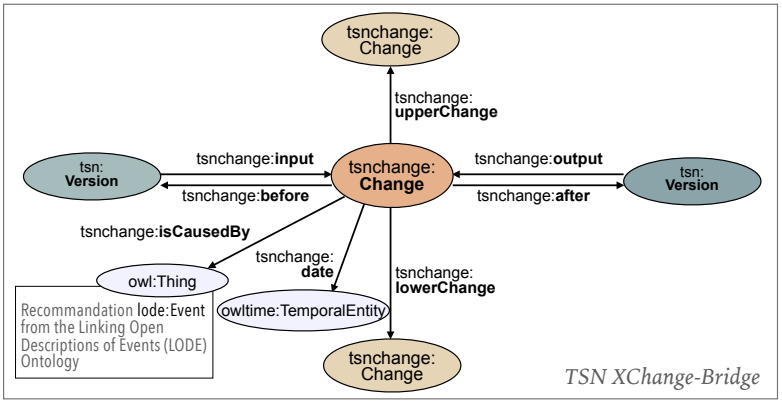


FIGURE *5 – Proposition de structure X-ChangeBridge pour la description de changement territoriaux dans des TSN.*

Pour la représentation des propriétés spatiales et temporelles des éléments d'une TSN (par ex., la géométrie des UT), les ontologies TSN et TSN-Change s'appuient sur les ontologies GeoSPARQL (Perry & Herring, 2012) de l'OGC et OWL Time (Cox & Little, 2017) du W3C. Les ontologies TSN et TSN-Change constituent une approche innovante (combinant l'approche Change-Bridge et l'approche des ontologies de fluents) pour la représentation de changements territoriaux dans le LOD via des graphes RDF de changements territoriaux multi-niveau.

5 Production de connaissances reversées dans le Web des Données

Certains jeux de données correspondant à des TSN sont publiés dans le Web LOD au format RDF, tels que, par exemple des versions de la nomenclature européenne NUTS. Néanmoins, ces jeux de données sont, soit des versions isolées, ne reflétant pas les différents stades de l'évolution de la nomenclature (le NUTS-RDF Geovocab [18], ou encore une instanciation de l'ontologie Eionet RAMON [19]), soit, lorsqu'ils décrivent des changements, ces descriptions ne sont que très partielles (Linked NUTS (Correndo & Shadbolt, 2013)).

Nous avons testé notre approche en l'appliquant à la production de connaissances pour trois TSN :

— La Nomenclature européenne des Unités Territoriales Statistiques (NUTS) (versions 1999, 2003, 2006 et 2010) de l'Institut statistique européen d'Eurostat disponibles dans le Web sous forme de fichiers vectoriels (shapefile ESRI) contenant la liste des UT de chaque version avec leurs attributs (code, nom, niveau, géométrie, ...).
— La nomenclature australienne *Australian Statistical Geography Standard (ASGS)*, élaborée par le Bureau australien des statistiques, qui comprend sept divisions imbriquées du territoire australien, dans les versions 2011 et 2016.
— La nomenclature des unités administratives suisses (SAU) de l'Office fédéral de la statistique, qui décrivent les cantons, districts et communes de Suisse en 2017 et 2018.

Après avoir instancié l'ontologie TSN pour chacune des versions de nomenclatures retenues, nous avons exécuté l'algorithme TSN Semantic Matching Algorithm pour ces différents jeux de données (et donc instancié l'ontologie TSN-Change).

18. http://nuts.geovocab.org/

19. http://rdfdata.eionet.europa.eu/ramon/nuts.rdf http://rdfdata.eionet.europa.eu/ramon/nuts2008.rdf http://rdfdata.eionet.europa.eu/ramon/nuts2003.rdf

En ce qui concerne la question de l'identité, ces trois nomenclatures utilisent des définitions similaires. Nous avons néanmoins montré dans (Bernard, 2019), chapitre 10) l'impact du choix des attributs constituant l'identité d'une UT et des poids associés sur la détection et la caractérisation des changements en simulant des définitions d'identité différentes.

Du point de vue des territoires décrits, les deux jeux de données NUTS et SAU, bien qu'ils soient tous deux basés en Europe, diffèrent par la taille des UT qu'ils contiennent. Les plus grandes UT de la NUTS sont les États membres de l'UE, tandis que les plus grandes UT de SAU sont les cantons, ce qui équivaut aux plus petites régions de la NUTS (niveau 3). Les plus petites unités de SAU sont des communes, des unités beaucoup plus petites et plus nombreuses que dans la NUTS. Nous avons fait le choix du troisième jeu de données australien ASGS parce qu'il couvre une région du monde très différente, un vaste territoire dont les géométries des UT sont composées de 4 000 sommets en moyenne (contre une moyenne de 28 sommets dans la NUTS).

Ces nomenclatures nous ont permis d'évaluer la capacité de notre système Theseus à générer des descriptions de filiations et de changements pour des jeux de données de taille importante et de tester son aptitude à calculer, notamment, des distances surfaciques entre deux géométries très précises (donc composées de nombreux sommets) ou de niveaux de généralisation différents (voir (Bernard, 2019), chapitre 10 pour une discussion sur les performances de l'algorithme TSN). Notre programme a ainsi détecté et décrit les similarités et changements entre les versions des trois nomenclatures. Ces connaissances nouvelles ont été versées dans le LOD. Le triplestore Theseus compte 156 162 triplets pour les 4 versions traitées de la NUTS (http://purl.org/steamer/nuts), 76 504 pour les 2 versions de SAU (http://purl.org/steamer/sau), et 89 974 triplets pour les 2 versions de la nomenclature australienne ASGS (http://purl.org/steamer/asgs).

A titre d'exemple, la Figure 6 donne le nombre de changements détectés entre les deux versions 2017 et 2018 de la nomenclature SAU. Notre algorithme détecte automatiquement tous les changements recensés dans la liste officielle qui présente 33 "mutations" (numérotées de 3580 à 3622), incluant chacune des changements tels que des fusions d'unités, des modifications de code ou de noms, etc. Pour chacun de ces changements, des descriptions sémantiquement plus riches (grâce à notre typologie des changements) et plus complètes (grâce au chainage multi-niveau des changements) sont fournies ce qui explique les 361 changements décrits. Nous avons confronté nos résultats à la liste officielle des changements recensés [20] et créé un catalogue confrontant nos résultats aux descriptions textuelles présentes dans la liste officielle [21].

Matching of the versions	SAU 2017 - 2018
Number of Feature Change	336
Number of GeometryChange	65
Number of NameChange	0
Number of IdentifierChange	30
Number of SubUnitChange	19
Number of SuperUnitChange	7
Number of Structure Change	25
Number of Split	0
Number of Merge	14
Number of Redistribution	1
Number of IdentificationRestructuration	6
Total Number of Change	361

FIGURE *6 – Nombre de changements détectés entre les deux versions 2017 et 2018 de SAU*

20. https://www.bfs.admin.ch/bfs/fr/home/statistiques/catalogues-banques-donnees/publications.assetdetail.4123244.html

21. http://purl.org/steamer/tsndoc/resources/sau_2017_2018_tsn_change_descriptions.pdf

Les graphes RDF créés constituent des catalogues d'UT et de leurs évolutions. Ils améliorent la compréhension des dynamiques territoriales, en fournissant aux statisticiens des descriptions pour comprendre les motivations et l'impact du redécoupage, et les moyens de référencer précisément les territoires pour lesquels ils produisent des données.

6 Exploitation des graphes créés et génération de nouvelles connaissances

Les graphes RDF créés par notre framework, basés sur le modèle ontologique TSN/TSN-Change, réutilisant lui-même GeoSPARQL et OWL Time, fournissent une représentation des relations spatiales, temporelles et de filiation entre les éléments d'une TSN. Ainsi, en utilisant le module SPARQL endpoint de notre framework http://steamerlod.imag.fr/, les utilisateurs peuvent interroger ces trois types de relations.

Nous avons, à titre d'exemples de ce qu'il est possible d'obtenir comme information, défini un ensemble de requêtes SPARQL permettant l'exploration des graphes produits pour les jeux de données NUTS, SAU et ASGS http://purl.org//steamer/tsndoc/resources/tsn_sparql_requests.pdf. Nous illustrons simplement ici par un exemple la capacité offerte par notre approche à produire de la connaissance.

Ainsi, la requête présentée par la figure 7 permet de reconstruire la ligne de vie d'une unité territoriale (ici l'unité ayant pour code ES63 dans la nomenclature NUTS) en parcourant les 4 versions successives de cette nomenclature que nous avons instanciées et pour lesquelles nous avons généré les liens de filiation et documenté les changements. La Figure 8 montre une représentation graphique automatiquement générée à partir du graphe résultat de la requête. Les nœuds rouges de la ligne supérieure représentent les versions successives de l'UT de code ES63 dans la NUTS. Le nœud étiqueté `nuts:V2003_L2_ES64`, représenté en rouge également, est un nœud en sortie du changement de type *extraction*, selon notre terminologie, qui a eu lieu entre les versions 1999 et 2003 de la NUTS. Ce changement a donné lieu à des modifications de la géométrie et du nom de l'UT ES63. Comme le montre cette vue, il s'agit du seul changement qui a affecté l'UT ES63 au cours de sa vie depuis 1999, dénotant une certaine stabilité territoriale pour cette unité (au moins jusqu'à la version NUTS 2010).

```
1   PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2   PREFIX tsnchange: <http://purl.org/net/tsnchange#>
3   PREFIX tsn: <http://purl.org/net/tsn#>
4   CONSTRUCT { ?TU_input tsnchange:inputUnitVersion ?change ;
5               tsnchange:hasNextVersion ?TU_output  .}
6   FROM <http://purl.org/steamer/nuts> WHERE{{
7       ?TU_input tsnchange:inputUnitVersion ?change .
8           ?change   tsnchange:unitVersionAfter ?TU_output .
9           ?TU_input tsn:hasIdentifier "ES63".}
10  UNION {?TU_input tsnchange:hasNextVersion ?TU_output .
11      ?TU_input tsn:hasIdentifier "ES63".}}
```

FIGURE *7 – Requête SPARQL retournant la ligne de vie d'une unité territoriale, ici l'unité ES63 telle que codifié dans la NUTS*

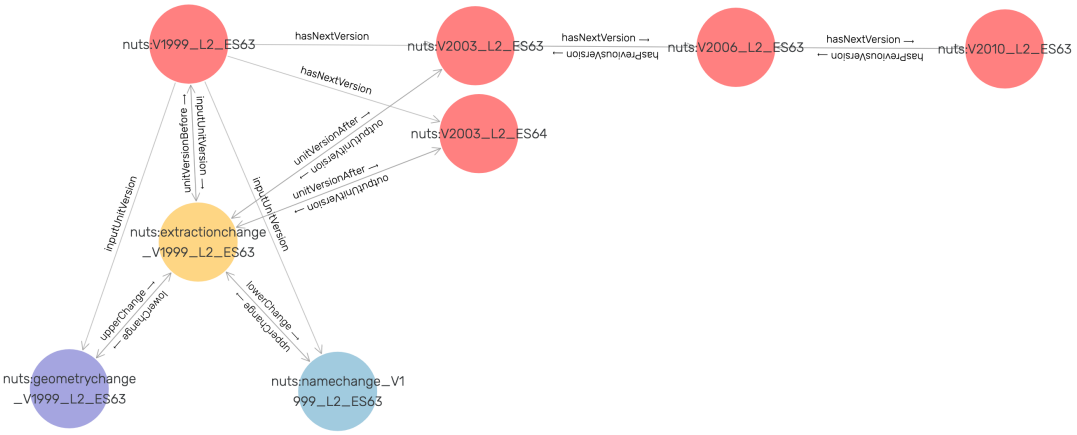


FIGURE 8 – *Visualisation de la ligne de vie de l'UT ES63 (résultat de la requête précédente), visible aussi à http://purl.org/steamer/nuts_ES63_lifeline*

Cette capacité à reconstituer la ligne de vie de chaque UT dans le temps par une lecture horizontale des graphes, en suivant la même entité à travers le temps, à travers les versions, est complétée par la capacité à parcourir les graphes selon une lecture verticale : nous avons créé d'autres requêtes qui fournissent une représentation de la propagation d'un changement à travers les niveaux de la nomenclature. Un exemple de requête est donné dans (Bernard, 2019), page 172), et le résultat associé visible depuis http://purl.org/steamer/nuts_V1999_ES6_change_graph.

En exploitant les données disponibles dans le LOD, il est possible d'enrichir les graphes générés, par exemple en recherchant dans d'autres graphes de connaissances tels que DBPedia ou Wikidata (le versant LOD de Wikipedia) des informations complémentaires (telles que le nom des gouvernants, le nombre d'habitants, voire (et surtout) une explication des changements...) sur les unités territoriales à partir de leur nom. En outre, on constate, dans ces graphes de données encyclopédiques, la quasi absence de représentation ontologique des changements territoriaux au cours du temps. En effet, s'il existe bien une classe DBPedia pour la représentation d'événements (http://dbpedia.org/ontology/Event), les événements décrits sont essentiellement des événements de type bataille militaire ou événement sportif (Hienert & Luciano, 2015). De fait, lorsque les changements subis par des régions sont décrits, ils ne le sont qu'à travers des champs de texte libre (tels que `dbo:abstract`), donc non structurés et non directement exploitables via des requêtes SPARQL. Ceci milite pour un référencement croisé des graphes de connaissances dédiées à l'évolution des unités territoriales statistiques proposés par notre approche, et des graphes de connaissances plus générales et grand public tels que ceux accessibles via DBPedia.

7 Conclusion et Perspectives

Dans cet article nous avons présenté une solution conceptuelle basée sur deux ontologies (TSN et TSN-Change), et logicielle (le framework Theseus, incluant notamment un algorithme, nommé TSN Semantic Matching Algorithm) pour représenter, dans le Web Sémantique, des connaissances relatives aux évolutions des territoires décrits par des nomenclatures territoriales statistiques. Notre approche est suffisamment générique pour couvrir tous les cas de structure hiérarchique de type TSN. En utilisant notre approche, les agences statistiques peuvent créer des unités spatiales déversables dans le LOD sous la forme de ressources référençables. Cela permet l'association précise de leurs données statistiques aux unités du territoires concernés qui sont désormais disponibles dans toutes les versions de la nomenclature. Des liens de filiation sont établis entre versions et nœuds de changements, contribuant à une meilleure compréhension des dynamiques territoriales. Le modèle intègre enfin le moyen de décrire et relier des changements qui affectent plusieurs niveaux de la TSN pour en donner une vision plus précise et plus complète, selon une approche ascendante ou descendante.

Nos descriptions des changements territoriaux sont générées automatiquement grâce à l'algorithme TSN *Semantic Matching Algorithm*, capable de s'adapter aux spécificités des TSN décrites, notamment sur la question de l'identité des unités territoriales.

Parmi les perspectives de ce travail, nous prévoyons d'étudier l'applicabilité de la démarche à d'autres types de nomenclatures utilisées notamment dans la description de zones médico-sociales, de zones trans-frontalières, de zones d'emploi, de zones urbaines. Un des points à explorer concerne la question de la modélisation de hiérarchies différentes des TSN car elles peuvent être non couvrantes (certaines unités peuvent avoir une ou plusieurs unités supérieures à un niveau territorial qui n'est pas immédiatement supérieur), non strictes (une UT peut avoir plusieurs UT englobantes), non hiérarchiques, etc.

Du côté de la restitution des connaissances, nous travaillons à la création d'un outil de visualisation des changements. Des outils logiciels spécialisés dans la gestion de version tels que *GeoGIG* ou le projet *GitHub Inc.* fournissent un moyen de visualiser les différences de géométries, UT par UT (Negretti, 2015; Boundless, 2014b,a). Néanmoins, ils ne permettent pas de regrouper des changements qui affectent plusieurs objets géographiques en même temps (par exemple, une fusion de deux UT), ou d'adjoindre une sémantique donnant le contexte et la nature de ce changement territorial. Notre objectif est de doter le framework Theseus d'un module de géovisualisation des changements intégrant les apports et originalités de l'approche de représentation des connaissances territoriales évolutives basée sur TSN et TSN-Change.

Références

ATKINSON R. (2017). QB4ST : RDF Data Cube extensions for spatio-temporal components.

BACHIMONT B., ISAAC A. & TRONCY R. (2002). Semantic commitment for designing ontologies : a proposal. In *International Conference on Knowledge Engineering and Knowledge Management*, p. 114–121 : Springer.

BERNARD C. (2019). *Immersing evolving geographic divisions in the semantic Web*. PhD thesis, Université Grenoble Alpes.

BERNARD C., PLUMEJEAUD-PERREAU C., VILLANOVA-OLIVER M., GENSEL J. & DAO H. (2018a). An ontology-based algorithm for managing the evolution of multi-level territorial partitions. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '18, p. 456–459, New York, NY, USA : ACM.

BERNARD C., VILLANOVA-OLIVER M., GENSEL J. & DAO H. (2018b). Modeling changes in territorial partitions over time : Ontologies tsn and tsn-change. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, p. 866–875 : ACM.

BERNARD, CAMILLE, VILLANOVA-OLIVER, MARLÈNE, GENSEL, JÉRÔME & DAO, HY (2018). Ontologies pour représenter l´évolution des découpages territoriaux statistiques. *Rev. Int. Geomat.*, **28**(4), 409–437.

BOUNDLESS (2014a). GeoGit in Action : Distributed Versioning Geospatial Data - Boundless.

BOUNDLESS (2014b). Introducing Versio : Version Control for Spatial Data - Boundless.

CLARAMUNT C. & THÉRIAULT M. (1995). Managing Time in GIS An Event-Oriented Approach. In C. J. VAN RIJSBERGEN, J. CLIFFORD & A. TUZHILIN, Eds., *Recent Advances in Temporal Databases*, p. 23–42. London : Springer London. DOI : 10.1007/978-1-4471-3033-8_2.

CORRENDO G. & SHADBOLT N. (2013). Linked nomenclature of territorial units for statistics. *Semantic Web*, **4**(3), 251–256.

COX S. & LITTLE C. (2017). Time Ontology in OWL - W3C Recommendation 19 October 2017.

CYGANIAK R. & REYNOLDS D. (2014). The RDF Data Cube Vocabulary.

DEBRUYNE C., MEEHAN A., CLINTON É., MCNERNEY L., NAUTIYAL A., LAVIN P. & O'SULLIVAN D. (2017). Ireland's authoritative geospatial linked data. In *International Semantic Web Conference*, p. 66–74 : Springer.

DEL MONDO G., RODRÍGUEZ M., CLARAMUNT C., BRAVO L. & THIBAUD R. (2013). Modeling consistency of spatio-temporal graphs. *Data & Knowledge Engineering*, **84**, 59–80.

DEL MONDO G., STELL J. G., CLARAMUNT C. & THIBAUD R. (2010). A graph model for spatio-temporal evolution. *J. UCS*, **16**(11), 1452–1477.

ETCHEVERRY L. & VAISMAN A. A. (2012). QB4OLAP : a new vocabulary for OLAP cubes on the semantic web. *Proceedings of COLD*.

GRENON P. & SMITH B. (2004). SNAP and SPAN : Towards Dynamic Spatial Ontology. *Spatial Cognition & Computation*, **4**(1), 69–104.

HARBELOT B., ARENAS H. & CRUZ C. (2013). Continuum : A spatiotemporal data model to represent and qualify filiation relationships. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on GeoStreaming*, p. 76–85 : ACM.

HARBELOT B., ARENAS H. & CRUZ C. (2015). LC3 : A spatio-temporal and semantic model for knowledge discovery from geospatial datasets. *Web Semantics : Science, Services and Agents on the World Wide Web*, **35**, 3–24.

HIENERT D. & LUCIANO F. (2015). Extraction of Historical Events from Wikipedia. In E. SIMPERL, B. NORTON, D. MLADENIC, E. DELLA VALLE, I. FUNDULAKI, A. PASSANT & R. TRONCY, Eds., *The Semantic Web : ESWC 2012 Satellite Events*, volume 7540, p. 16–28. Berlin, Heidelberg : Springer Berlin Heidelberg.

HORNSBY K. & EGENHOFER M. J. (2000). Identity-based change : a foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science*, **14**(3), 207–224.

HYLAND B., ATEMEZING G. A. & VILLAZÓN-TERRAZAS B. (2014). Best Practices for Publishing Linked Data.

KAUPPINEN T. & HYVÖNEN E. (2007). Modeling and reasoning about changes in ontology time series. In *Ontologies*, p. 319–338. Springer.

KAUPPINEN T., VÄÄTÄINEN J. & HYVÖNEN E. (2008). *Creating and using geospatial ontology time series in a semantic cultural heritage portal*. Springer.

LACASTA J., LOPEZ-PELLICER F. J., FLORCZYK A., ZARAZAGA-SORIA F. J. & NOGUERAS-ISO J. (2014). Population of a spatio-temporal knowledge base for jurisdictional domains. *International Journal of Geographical Information Science*, **28**(9), 1964–1987.

LÓPEZ-PELLICER F. J., FLORCZYK A. J., LACASTA J., ZARAZAGA-SORIA F. J. & MURO-MEDRANO P. R. (2008). Administrative units, an ontological perspective. In *Advances in Conceptual Modeling–Challenges and Opportunities*, p. 354–363. Springer.

LOPEZ-PELLICER F. J., LACASTA J., FLORCZYK A., NOGUERAS-ISO J. & ZARAZAGA-SORIA F. J. (2012). An ontology for the representation of spatiotemporal jurisdictional domains in information retrieval systems. *International Journal of Geographical Information Science*, **26**(4), 579–597.

NEGRETTI M. (2015). Operation-based revision control for geospatial data sets. p.1̃5.

OPENSHAW S. (1984). The modifiable areal unit problem. *Concepts and Techniques in Modern Geography*, **38**.

PERRY M. & HERRING J. (2012). OGC GeoSPARQL - A Geographic Query Language for RDF Data. p.7̃5.

PLUMEJEAUD C. (2011). *Modèles et méthodes pour l'information spatio-temporelle évolutive*. PhD thesis, Université de Grenoble.

PLUMEJEAUD C., MATHIAN H., GENSEL J. & GRASLAND C. (2011). Spatio-temporal analysis of territorial changes from a multi-scale perspective. *International Journal of Geographical Information Science*, **25**(10), 1597–1612.

REDMOND T., SMITH M., DRUMMOND N. & TUDORACHE T. (2008). Managing Change : An Ontology Version Control System. p.1̃0.

SHAW R., TRONCY R. & HARDMAN L. (2009). LODE : Linking Open Descriptions of Events. *ASWC*, **9**, 153–167.

VILLAZÓN-TERRAZAS B., VILCHES-BLÁZQUEZ L. M., CORCHO O. & GÓMEZ-PÉREZ A. (2011). Methodological Guidelines for Publishing Government Linked Data. In D. WOOD, Ed., *Linking Government Data*, p. 27–49. New York, NY : Springer New York.

VÖLKEL M. & GROZA T. (2012). SemVersion : an RDF-based ontology versioning system. p.9̃.

WELTY C., FIKES R. & MAKARIOS S. (2006). A reusable ontology for fluents in OWL. In *FOIS*, volume 150, p. 226–236.

Événements abstraits et états d'affaires « occurrent-facteurs »

Gilles Kassel

Laboratoire MIS, Université de Picardie Jules Verne
33 rue Saint-Leu, 80039 Amiens Cedex 1
Gilles.kassel@u-picardie.fr

Résumé : Dans cet article, nous défendons une conception abstraite, psychologique, de l'*événement*. Nous adoptons comme cadre formel d'ontologie un double réalisme ordinaire et conceptuel s'engageant vis-à-vis de l'existence de propriétés-relations et objets ordinaires, mais aussi de propriétés-relations et objets conceptuels. Parmi les objets conceptuels, nous admettons notamment les *propositions* et les *événements* et identifions une classe de propositions assertant l'*occurrence* (ou réalisation) d'événements. Ceci nous conduit à nous interroger sur la nature des entités concrètes réalisant les événements, des *faits* que nous nommons « occurrent-facteurs » (leur existence entraîne l'occurrence d'événements), par analogie avec le terme « véri-facteur » (ce qui rend vraies des propositions). En considérant des *états* et des *changements* comme événements, nous montrons que les faits en question peuvent se limiter à des faits ayant pour seuls constituants des particuliers.

Mots-clés : Événement, processus, réalisme ordinaire et conceptuel, entité concrète, entité abstraite, état d'affaires, véri-facteur, occurrent-facteur

1 Introduction

Récemment, pour rendre compte de la nature des entités communément qualifiées de « survenantes » (*processus*, *événements*, *états*, *changements d'état*), nous avons proposé un cadre ontologique dont les principes de structuration remettent en cause la distinction « continuant » *vs* « occurrent » sur laquelle se fondent, dans la communauté ontologie appliquée, la plupart des ontologies dites fondatrices (BFO, DOLCE, GFO). En bref, ce cadre repose sur 3 principes (Kassel, 2019, 2020) :

- Le monde physique est peuplé de particuliers – *objets* et *processus* – qui, en endurant, assurent sa stabilité tout autant que sa dynamicité ;
- Ces particuliers ont une vie en portant des *propriétés* et en entretenant des *relations* temporairement avec d'autres particuliers, de tels liens constituant des *faits* ;
- Des sujets cognitifs, plongés dans le monde physique, se représentent au moyen notamment d'*événements* l'histoire passée, présente et future du monde pour interagir.

Jusqu'à présent, en guise de défense de ce cadre, nous avons pris soin de montrer une filiation de nos conceptions avec des théories existantes (en ontologie formelle) et une cohérence d'ensemble. Ainsi, pour notre primitive *processus physique*, nous avons adopté la conception de Cleland (1990) du « processus causal de transformation de propriété », en phase avec les travaux de Stout (1997, 2003) et de Galton (2006, 2008). Concernant la catégorie d'*événement*, nous nous sommes placés dans les traces de Chisholm (1970) et de Wilson (1974) en proposant une conception *abstraite* de l'événement. Enfin, côté *faits*, nous nous sommes principalement référé à la *circonstance* de Fine (1982) et à l'*état d'affaires* d'Armstrong (1997).

Dans cet article, nous visons plus particulièrement à défendre notre conception de l'événement *abstrait* en la positionnant vis-à-vis de théories contemporaines en philosophie du

langage et de la logique. Cette philosophie reste traversée par une ligne de fracture, apparue au tournant du XX^e^ siècle, concernant la possibilité d'une sémantique cognitive. D'un côté, des philosophes comme Russell (1872-1970), Quine (1908-2000) et Davidson (1917-2003) ont affiché leur scepticisme (et c'est un euphémisme) vis-à-vis d'une quelconque ontologisation du *sens* (qu'elle soit psychologique ou idéale) jouant un rôle intermédiaire entre les phrases et leur référent[1]. D'un autre côté, se situant dans un courant de pensée prenant sa source avec le *complex significabile* stoïcien, des philosophes comme Bolzano (1781-1848), Brentano (1838-1917), Frege (1848-1925) et Meinong (1853-1920) ont au contraire donné du crédit à un « objectivisme » sémantique en reconnaissant au sens une certaine forme d'existence[2]. L'auteur se situe dans ce second courant, en positionnant l'événement du côté du sens.

Dans la suite de l'article, nous commençons par rappeler notre cadre ontologique de référence pour ce qui concerne le domaine des entités dites « survenantes », ce qui va nous conduire à identifier la figure de l'*événement abstrait* (Section 2). Nous nous dotons ensuite d'une théorie sémantique conférant à l'événement abstrait un rôle à la fois de constituant de sens et d'objet direct de référence (Section 3). Finalement, nous mettons à l'épreuve notre ontologie du monde et notre théorie sémantique en identifiant, pour différentes phrases assertant l'occurrence d'événements, les faits véri-facteurs (Section 4).

2 Processus, faits et événements (en résumé)

Dans cette section, nous donnons une caractérisation succincte de nos primitives – *processus*, *faits* et *événements* – rendant compte du domaine des entités « survenantes », telles que définies dans (Kassel, 2019, 2020). Nous en profitons pour préciser quelques engagements ontologiques de base, notamment le fait d'adopter une théorie *présentiste* du temps.

Pour introduire ces primitives, nous revenons sur l'histoire récente des théories philosophiques du changement. En des termes actuels, le changement temporel se définit comme le fait qu'une substance[3] porte des propriétés contradictoires (*F* et *non F*) à des temps différents. Par exemple, un objet *O* est froid à un temps T_1 et chaud à un temps T_2. Au début du XX^e^ siècle, Bertrand Russell (1903) propose la conception suivante du mouvement continu, en tant qu'un changement temporel de localisation spatiale : « Motion consists merely in the occupation of different places at different times ». Ainsi, un mouvement continu d'un objet *O* n'est *rien de plus* qu'une série de faits correspondant à l'occupation *Loc* par *O* de différentes positions Pos_i à des instants successifs I_j : $\langle O,Loc,Pos_1,I_1\rangle$, $\langle O,Loc,Pos_2,I_2\rangle$, *etc*[4]. Cette conception a été critiquée par de nombreux philosophes, au motif qu'elle ne rend pas compte du caractère dynamique du déplacement. Comme Henri Bergson (1946) a pu notamment l'exprimer, le mouvement est traité « comme s'il était fait d'immobilités ».

[1] Pour mémoire, dans son (1967a) *Truth and Meaning*, Davidson déclarait (p. 307) : « Paradoxically, the one thing meanings does not seem to do is oil the wheels of a theory of meaning – at least as long as we require of such a theory that it non-trivially gives the meaning of every sentence in the language. My objection to meanings in the theory of meaning is not that they are abstract or that their identity conditions are obscure, but that they have no demonstrated use ».

[2] L'ouvrage édité par Jocelyn Benoist (2006) présente de façon historique et critique ce courant.

[3] Dans cet article, nous utilisons le terme « substance » au sens d'un « substrat », autrement dit d'une entité susceptible de porter des propriétés.

[4] Le terme « changement de Cambridge » a été proposé par Peter Geach (1968, p. 13) pour dénoter cette conception du changement continu promue par des philosophes de Cambridge dont John McTaggart et Bertrand Russell. Cette conception porte également le nom de théorie « *at-at* ». Pour rendre compte du caractère continu du changement, la plupart des versions privilégient comme temps l'instant indivisible.

Avant d'aller plus loin sur la dynamicité du mouvement, on peut noter que la métaphysique contemporaine donne crédit à l'existence de *Faits*, ceux-là mêmes auxquels il est fait référence dans les formulations de la théorie 'at-at'. La thèse de l'existence des *Faits* a été défendue notamment par Kit Fine (1982) et David Armstrong (1997). Un *Fait* (ou « circonstance », selon la terminologie de Fine, ou encore « état d'affaires », pour reprendre le terme d'Armstrong), est une entité complexe constituée d'une substance (ex : 'Paul'), d'une propriété (ex : 'Être à côté de Marie') et d'un temps, l'instant 'I' : <Paul, Être à côté de Marie, I>)[5]. L'existence simultanée à un instant donné d'une substance et d'une propriété ne signifie pas pour autant que la substance exemplifie la propriété à cet instant. Le *Fait* correspond à un lien interne unissant, à un instant donné, substance et propriété/relation en une entité à part entière. L'argument principal de l'existence des *Faits* est qu'ils constituent un « véri-facteur », autrement dit ce qui rend vrai dans le monde des propositions comme 'Paul est à côté de Marie'. Les *Faits* auxquels il est fait référence dans la théorie 'at-at' du mouvement ont pour constituant une propriété de localisation spatiale d'une substance.

Revenons à la dynamique du mouvement. Il nous reste à expliquer comment il est possible à un objet d'entrer et de sortir d'une position plutôt que d'être statiquement dans une position. Pour ce faire, nous adoptons l'analyse de la philosophe Carol Cleland (1990), faisant appel à la notion d'*effort*, au cœur de la physique newtonienne. Considérons l'expérience consistant à faire tourner autour de soi un globe (ou tout autre objet) maintenu par une ficelle. Selon Cleland, le fait que l'on puisse sentir la tension dans la ficelle, autrement dit le fait que la *tendance* de l'objet à être éjecté soit un observable mesurable, constitue un argument décisif pour son existence physique (*Ibid.*, 273) :

> "Given their crucial role in physical explanation and theory, I propose that we admit operative tendencies to be elsewhere into our ontology as primitive properties of physical objects. We can think of them as physicists think of instantaneous vector quantities, viz., as uneliminable proclivities of varying degrees of strength."

Ainsi, pour rendre compte de la notion d'effort, Cleland pose l'existence d'une entité correspondant à un « état actif de mouvement », qu'elle nomme tour à tour « tendance opérante » ou « processus causal de transformation de propriété ». L'existence d'un tel processus est justement ce qui distingue le fait, pour un objet, de passer dynamiquement à travers des états plutôt que d'être statiquement dans des états différents.

Nous venons de caractériser le processus comme *capacité/disposition dynamique*, ou comme « moteur » d'un mouvement continu. Toujours suivant Cleland, nous généralisons la notion de processus comme moteur de *tout* type de changement continu, qu'il s'agisse d'un changement de température, de couleur, *etc.* Plus précisément, nous considérons que lors d'un changement continu, un même et unique processus est continuellement actif. Finalement, nous aboutissons à une conception du processus en tant qu'une entité endurante portant des propriétés et pouvant changer dans le temps (une marche peut s'accélérer, changer de direction)[6]. Une propriété importante que nous attribuons aux processus est d'être ancrés dans un objet : il s'agit du processus du mouvement d'*une balle*, du mûrissement d'*un fruit*, de la fonte d'*un glacier*, de l'oxydation *d'une pièce de métal*, *etc.* Pour rendre compte de ce lien fort entre objets et processus, nous reprenons à notre compte la relation d'*énaction* introduite par Galton et Mizoguchi (2009) : les objets *énactent* contingentement des processus.

[5] Nous optons pour une théorie *présentiste* du temps stipulant que seul le présent existe. De ce fait, nous n'accordons d'existence physique qu'à des instants indivisibles, mais de durée non nulle (Kassel, 2020).

[6] Notre conception du processus est proche de celle promue par Rowland Stout (1997, 2003) et Antony Galton (2006, 2008) en rapprochant les processus physiques des objets dans leur façon d'endurer dans le temps.

Pour définir maintenant nos *événements*, revenons sur l'analyse du changement et reprenons la caractérisation du mouvement continu. Prenons l'exemple d'une personne marchant. Comme nous venons de le définir, cette personne *énacte* un processus de marche – nommons le 'Marche$_{Proc}$' – ce qui la conduit à se déplacer en occupant successivement des positions distinctes. Envisageons maintenant l'histoire de la vie de la personne, en lien avec son déplacement sur une certaine période de temps. Nous pouvons par exemple considérer 'la marche de Paul jusqu'à la gare ce matin'. La thèse que nous défendons est que, ce faisant, nous conférons une existence à une nouvelle entité – nommons la 'Marche$_{Evén}$'. Cette entité, par contre, ne peut être physique. Rappelons en effet que nous adoptons une théorie présentiste du temps, ce qui nous empêche d'admettre des entités physiques étendues dans le temps. Qui plus est, sur un plan strictement physique, on notera que 'Marche$_{Evén}$' a pu donner lieu à plusieurs processus comme 'Marche$_{Proc}$', si Paul a flâné sur le trajet en s'interrompant et en reprenant sa marche. Nous posons donc que 'Marche$_{Evén}$' est un construit psychologique, un événement abstrait.

Dans Kassel (2020), nous envisageons différentes catégories de chroniques du monde et donc de catégories d'événements. C'est ainsi qu'à côté des *changements*, nous considérons les *états* rendant compte de la stabilité du monde. Une propriété importante des événements en général est qu'ils sont susceptibles d'*occurrer*, ou d'être *réalisés*. Cette propriété est distincte de celle de l'*existence* : un événement comme 'Marche$_{Evén}$' *existe* en étant pensé par un sujet ; l'événement *occurre* lorsque des processus comme 'Marche$_{Proc}$' sont actifs et que l'activité de ces processus fait que l'histoire du monde correspondant à 'Marche$_{Evén}$' est satisfaite.

En Section 4, nous identifions ces processus à des constituants de faits que nous qualifions d'« occurrent-facteurs », traduisant le fait que leur existence entraîne l'*occurrence* ou la réalisation d'événements. Avant cela, nous précisons la théorie sémantique que nous adoptons.

3 Fondements d'une théorie sémantique

L'enjeu de cette section est de se doter d'une théorie sémantique générale, qui accorde notamment une place à l'événement abstrait. Ceci va nous conduire à préciser nos engagements ontologiques vis-à-vis des entités abstraites et de leurs relations avec les entités concrètes. Nous adoptons tout d'abord un double réalisme ordinaire (pour le monde concret) et psychologique ou conceptuel (pour le monde abstrait) (§3.1). Notre réalisme conceptuel nous ouvre un espace pour accueillir le *sens* et, suivant une tradition contemporaine forte, nous faisons jouer à la *proposition*, que nous identifions à un contenu, le rôle de sens. Nous en profitons pour réhabiliter la notion d'*état d'affaires abstrait* et identifier nos événements à cette espèce d'entités (§3.2). Finalement, nous proposons, pour les faits concrets, de nous en tenir à la figure restreinte (par rapport à la proposition objective russellienne) de l'état d'affaires wittgensteinien (1922/1921) (§3.3).

3.1 Un double réalisme, ordinaire et psychologique

Précisons en premier lieu les fondements de l'ontologie formelle que nous adoptons. Cette ontologie emprunte à la *théorie de l'objet abstrait* d'Edward Zalta (1983)[7] et au *réalisme*

[7] Il s'agit plus précisément de la *théorie typée des objets abstraits* que Zalta élabore dans les chapitres 5 et 6 de son (1983) et qui intègre des *propriétés* et *relations abstraites*. Sa *théorie de l'objet abstrait* (de base), définie dans les premiers chapitres, est une élaboration de théories proposées par différents néoMeinongiens, dont Ernst Mally, William Rapaport et Terence Parsons.

naturel et conceptuel de Nino Cocchiarella (1996)[8]. Les fondements concernent tout à la fois la nature psychologique et la variété des entités abstraites considérées et leur articulation avec les entités concrètes. Ils reposent sur deux premières thèses :

(i) Le monde est peuplé d'entités *concrètes* et d'entités *abstraites* organisant une partition du monde ;

(ii) Ces deux parties distinctes du monde sont peuplées d'entités simples *saturées* et *insaturées* se combinant en des *complexes*.

La thèse (i) de la partition du monde en entités concrètes et abstraites est un principe métaphysique bien établi, même si le positionnement de la frontière demeure toujours une question ouverte, en témoigne le statut d'entité abstraite que nous revendiquons pour les événements. La distinction correspond à des modes distincts d'*être* : les entités *concrètes* existent indépendamment d'être pensées, au contraire des entités *abstraites* qui sont des construits psychologiques et sociaux. En adoptant cette caractérisation des entités abstraites, nous suivons Zalta et Cocchiarella dans le fait de les déloger d'un royaume platonicien d'entités existant en dehors de l'espace et du temps[9]. Ainsi, qu'elles soient concrètes ou abstraites, nous considérons que toutes les entités sont *dans le temps* : elles peuvent être *actuelles, passées* ou *futures*.

La thèse (ii), fermement revendiquée par Zalta et Cocchiarella, consiste à positionner la *prédication* comme principe ontologique de base. La prédication s'explique par l'existence d'entités *saturées* – des objets *complets*, dans un sens analogue aux substances aristotéliciennes – et d'entités *insaturées* se combinant pour constituer des *complexes*. Du côté concret, comme du côté abstrait, on distingue ainsi des objets et des propriétés-relations qui, lorsqu'ils se combinent, donnent des « complexes-propositions ». Les principes de combinaison relèvent respectivement des lois physico-biologiques et des lois régissant nos pensées et actes de langage : une proposition concrète consiste en un lien physico-biologique entre un objet concret et une propriété-relation concrète (l'objet est dit « exemplifier » la propriété-relation) ; une proposition abstraite tient en un acte de pensée (pouvant être extériorisé en un acte de discours) combinant un objet abstrait et une propriété-relation abstraite (on dit de l'objet qu'il « encode » la propriété-relation).

Dans la suite de la section 3, nous précisons nos engagements vis-à-vis de ces complexes-propositions abstraits et concrets. Avant cela, nous précisons le lien existant entre entités abstraites et concrètes, en posant une troisième thèse :

(iii) Les propriétés-relations abstraites *représentent* des propriétés-relations concrètes, tandis que les objets abstraits *représentent* des objets concrets.

Le verbe « représenter » est à entendre dans un sens large de *comporter un contenu se référant à*. Chez Cocchiarella (1996), cette thèse (iii) s'accompagne d'une théorie sémantique, que nous reprenons à notre compte. Nous la formulons en adoptant la terminologie de Coccharielle : le terme « concept » est utilisé en lieu et place de « propriété-relation abstraite ».

Une assertion telle « Paul est prudent » ou « Paul est à côté de Marie » s'analyse sémantiquement à deux niveaux. La structure prédicative de ces jugements en est le point de

[8] Le terme « réalisme naturel » désigne un réalisme aristotélicien consistant à conférer une existence aux objets ordinaires nous entourant (tables, maisons, personnes), mais également aux objets de taille micro et macroscopique. L'inconvénient de ce terme, lorsqu'il est opposé aux termes « réalisme conceptuel » ou « réalisme psychologique », est qu'il semble exclure de la nature les entités psychologiques, cognitives. Pour éviter cela, nous lui préférons dans cet article le terme « réalisme ordinaire ».

[9] Nous revenons sur ce point en §3.2, lorsque nous évoquons les conceptions idéales de la proposition chez Bolzano et Frege.

départ. À un premier niveau, ces assertions s'analysent comme la combinaison d'un *concept-référentiel* exprimé par le sujet « Paul » et d'un *concept-prédicat* exprimé respectivement par le prédicat « est prudent » et « est à côté de Marie ». À un second niveau, de façon dérivative, ces concepts représentent des objets concrets et des propriétés-relations concrètes. Ainsi, le sujet « Paul » tient pour deux entités distinctes, à savoir un concept-référentiel et un objet ordinaire. De même, le prédicat « est prudent » tient pour deux entités distinctes, à savoir un concept-prédicat et une propriété ordinaire. De façon importante, la théorie sémantique de Cocchiarella n'exige pas que chaque concept abstrait représente une entité concrète. En Section 4, nous allons justement nous émanciper de cette contrainte pour proposer (notamment) que la relation de proximité spatiale, exprimée par « Être à côté de », soit uniquement conceptuelle.

Parmi les entités abstraites saturées figurent la *proposition*. Nous précisons maintenant nos engagements vis-à-vis de cette entité. A côté de la proposition, nous comptons réhabiliter une certaine conception de l'*état d'affaires abstrait* pour positionner nos événements.

3.2 La proposition et l'événement abstraits

Historiquement, la proposition abstraite puise ses racines au milieu du XIX^{e} siècle dans la 'proposition en soi' (*Satz an sich*) de Bernard Bolzano. Elle s'est ensuite forgée au tournant du XX^{e} siècle avec, d'un côté, la 'pensée' (*Gedanke*) de Gotlob Frege, et, de l'autre, les théories psychologiques des actes de pensée et de discours établies par Franz Brentano et ses élèves[10].

Ces théories ont en commun d'avoir mis en scène différentes entités : des actes « occurrents » et des entités « continuants » leur étant associées, à savoir des contenus, des états d'affaires, des objets de référence. L'existence et la nature de ces entités continuent aujourd'hui de faire débat. Notamment, côté sens, se pose la question de savoir si nous devons lui accorder un véritable statut ontologique et, le cas échéant, si sa nature est d'être un occurrent (un acte) ou un continuant (un contenu)[11]. Le lecteur aura compris qu'en ayant fait le choix d'une sémantique cognitive, nous avons de fait opté pour la reconnaissance d'un contenu-propositionnel jouant le rôle de sens pour des actes de pensée et de discours. Également, la question est de savoir si des états d'affaires abstraits jouant le rôle d'objets de référence existent et, le cas échéant, quelle est leur nature. À cette dernière question, nous allons répondre positivement en identifiant nos événements abstraits à de tels états d'affaires. Nous allons donc prendre quelques engagements ontologiques supplémentaires pour rendre compte de « ce que l'on pense ou dit » et « ce à quoi on se réfère en pensant ou disant ».

Un premier engagement que nous souhaitons prendre, pour rendre compte de « ce que l'on pense ou dit », est le suivant : il existe des propositions *mentales* ayant un caractère *social*. Bolzano, nous le savons, a conçu sa proposition en soi comme idéale. Toutefois, contrairement à Frege (et à sa pensée), Bolzano n'a jamais dit que les propositions en soi étaient indépendantes

[10] Pour une présentation historique et critique de cette période, nous renvoyons le lecteur à Smith (1995), tout particulièrement au chapitre 6 (pp. 155-195) *Kasimir Twardowski : On Content and Object*, et à Benoist (2006), tout particulièrement son chapitre introductif (pp. 13-49) *Variétés d'objectivisme sémantique*.

[11] L'ouvrage édité par Friedericke Moltmann et Mark Textor (2017) se fait l'écho de ces débats contemporains entre des approches qualifiées respectivement de 'act first' et 'content first'. Parmi les partisans de l'approche 'act first' figurent de façon prééminente Scott Soames (2014) et Peter Hanks (2017). Leur position est d'identifier les propositions à des actes (ou plutôt à des types d'actes) consistant à prédiquer des propriétés d'objets. Ce faisant, leur conception s'oppose à celle d'un contenu endurant, que nous promouvons au contraire. Dans cet article, nous ne souhaitons pas développer un argumentaire détaillé en faveur de telle ou telle conception. Nous comptons plutôt nous focaliser sur les questions suivantes : quels objets et quelles propriétés reconnaissons-nous ? En sachant que notre intention est d'ouvrir le domaine des objets communément considérés à des événements abstraits dont l'*occurrence* est prédiquée.

du langage et de la pensée, seulement qu'elles étaient indépendantes du sujet qui les pense, les énonce ou les juge. Ceci explique que les élèves de Brentano, au corps défendant de leur maître, aient pu s'approprier la proposition en soi de Bolzano, selon un double geste : d'une part, en étoffant le domaine des propositions brentaniennes, jusque-là réduites à des propositions existentielles ; d'autre part en faisant en quelque sorte « gagner en immanence » les propositions bolzaniennes vis-à-vis de l'acte[12]. La théorie bolzanienne distingue déjà les propositions subjectives correspondant aux contenus d'actes datables de pensée et de discours et les propositions objectives, ou propositions en soi, dont la raison d'être est de porter une vérité objective. Le second geste que nous évoquons est parti du constat que ce que pensent et disent différents individus à différentes occasions (et indépendamment des langues utilisées) se ressemble, au point de paraître être la même chose. Sur la base de ce constat, Husserl a proposé que cette « même chose » relève d'un universel ou d'une espèce, dont les contenus datables soient les instances ou membres (de la même façon que l'universel de rougeur est présent dans les rougeurs contingentes d'objets matériels variés). La proposition en soi bolzanienne devient cet universel présent dans chaque proposition subjective (notre proposition mentale) en lui conférant un caractère social.

Intéressons-nous maintenant à « ce à quoi on se réfère en pensant ou disant ». Chez Bolzano, la proposition se réfère tout au plus à l'objet de sa représentation sujet (par exemple, à l'objet 'Paul' dans la proposition [Paul salue Marie]). L'objet peut être concret, comme dans le cas de la personne 'Paul', ou abstrait-idéal, comme dans le cas des entités mathématiques tel un nombre ou un triangle. Par ailleurs certaines représentations sont dites 'an-objectuelles' (*gegenstandlos*) lorsque l'objet est contradictoire, comme dans le cas du 'carré rond'. Brentano n'ira pas plus loin, tenté un temps d'élargir le domaine des objets visés à celui de 'choses' (*Dinge*) comme des collectifs, des parties d'objets matériels ou des êtres appartenant à un passé révolu, pour finalement y renoncer et restreindre aux entités du monde concret la possibilité d'être visées par des actes mentaux. Ce pas sera franchi par Twardowski et Meinong, s'assurant par là-même que chaque représentation ait à la foi un contenu et un objet (existant d'une certaine manière par le fait même d'être pensé et représenté). Pour notre propos dans cet article, notre intérêt concerne l'extension au jugement de cette doctrine d'une existence conjointe et distincte d'un contenu et d'un objet. Twardowski, très tôt, élabore une théorie du jugement en lui appliquant la distinction du contenu et de l'objet, ce dernier étant conçu comme un 'état d'affaires' (*Sachverhalt*). Le geste fondamental accompli par Twardowski est de considérer qu'une proposition telle [Paul salue Marie], qu'il identifie au contenu d'un jugement relationnel, se réfère ou donne accès à l'objet-état d'affaires 'Salutation de Paul à Marie'[13]. Nous sommes bien en présence de deux entités abstraites distinctes, la proposition et l'état d'affaires. Nous identifions ces états d'affaires abstraits à nos événements.

3.3 L'état d'affaires concret

Venons-en maintenant à élucider la nature du complexe concret auquel nous avons fait référence en §3.1 en le caractérisant comme une forme prédicative d'exemplification par un objet concret d'une propriété-relation également concrète.

À notre époque contemporaine, la figure de ce complexe concret est incarnée de façon prééminente par le « fait » armstrongien (1997), lui-même héritier de la proposition objective

[12] Cette filiation et la contribution de Husserl évoquée juste après sont retracées avec soin par Wolfgang Künne (2009).
[13] Nous nous fondons ici sur l'analyse de Arianna Betti (2005).

russélienne. Nous y avons fait référence en Section 2, à l'occasion de l'analyse du mouvement continu, en évoquant des faits de localisation spatiale qui s'obtiennent successivement lorsqu'un objet se meut. Selon le cadre ontologique posé en Section 2, de tels faits participent de l'ameublement du monde physique à côté des objets et des processus. D'une façon générale, ils se définissent comme un complexe constitué d'un objet physique et, soit d'une propriété (comme dans : <Livre, Être rouge>, <Paul, Marcher>), soit d'une relation (comme dans : <Livre, A pour prix, 50€>, <Paul, Être à côté, Marie>, <Paul, Dresser des plans sur la comète avec, Marie>).

Avant de poursuivre, il est fondamental de noter que dans la présentation que nous venons d'effectuer des complexes concrets, sous couvert de « généralité », en réalité nous avons évoqué deux espèces différentes d'entités dont la nature métaphysique et la justification de l'existence diffèrent. D'une part, nous avons évoqué des faits de la théorie 'at-at' n'étant constitués que d'individus (nous justifierons dans un instant cette affirmation) et dont l'existence est justifiée sur un plan exclusivement ontologique. D'autre part, nous avons évoqué les faits armstrongiens constitués notamment de propriétés-relations universelles et dont la justification de l'existence est essentiellement sémantique. Rappelons que, selon Armstrong, l'existence de ces faits représente la meilleure explication à fournir en termes d'entités véri-factrices qui rendent vraies des propositions comme « Le livre est rouge », « Paul marche », *etc.* Cette différence d'espèces est d'autant plus importante à souligner que nous nous apprêtons à accorder une existence aux « faits at-at » et à dénier toute forme d'existence aux « faits armstrongiens ».

Le coup de grâce contre l'existence des faits Armstrongiens nous paraît avoir été porté notamment par William F. Vallicella (2000). Vallicella considère qu'à partir du moment où un universel est considéré comme constituant d'un fait concret, à côté de particuliers, tout espoir est vain de concevoir une entité concrète liant les constituants en une unité : le lien représenté par notre notation <,,> – devient un constituant lui-même, mais alors on s'expose à une régression infinie de devoir supposer l'existence d'un nouveau lien liant le premier lien et ses constituants, et de même pour ce nouveau lien, etc.

Cette analyse nous paraît tout à fait pertinente et nous pousse, si nous voulons conserver un complexe concret dans notre ameublement du monde physique, à trouver une parade. A notre connaissance, on trouve une telle parade dans la littérature. Mulligan, Simons et Smith (1984) ont en effet proposé de revenir à ce qu'ils estiment être l'esprit de l'état d'affaires du *Tractatus* de Wittgenstein (1922/1921). Selon ces auteurs, l'état d'affaires tractarien repose uniquement sur des particuliers liés par des « relations de fondation mutuelle » :

> It is, we suggest, because analytic-philosophical interpreters of the Tractacus have standardly lacked a theory of lateral foundation relations, relations which may bind together individual objects, that they have been constrained to resort to views of the kind which see Sachverhalte as involving both individuals and universal properties. It is open to us here, however, to develop a view of Sachverhalte as involving individuals alone, linked together by relations of foundation. 'This speck is red' might be made true, on such a view, by a two-object Sachverhalt comprising the speck and an individual moment of redness linked by a relation of mutual foundation.

En se focalisant sur des entités appelées « moments » (et définies comme des entités existentiellement dépendantes d'autres entités, à l'instar des qualités de substances), Mulligan *et coll.* ont ouvert la voie à une approche cherchant à identifier les états d'affaires à des complexes constitués de seuls particuliers. On évite ainsi le problème évoqué supra d'entités hybrides constituées de particuliers et d'un universel. Par la suite, nous choisissons de suivre leur proposition. Ce choix relève d'un projet et, en ce sens, nécessite d'être évalué, ce que nous nous proposons de faire en Section 4.

4 Événements et états d'affaires occurrent-facteurs

Dans cette section, nous mettons à l'épreuve les théories ontologique et sémantique que nous venons d'établir. Pour des phrases que nous interprétons comme des assertions d'occurrence d'événement, nous commençons par proposer une forme logique reconnaissant à l'événement le statut d'entité abstraite que nous lui avons octroyé (§4.1). Nous envisageons ensuite différents types d'événement et, pour chaque type, nous identifions les faits « occurrent-facteur » permettant de décider de la véracité de l'occurrence de l'événement. Nous montrons alors que, sous certaines conditions portant sur la nature en général des propriétés-relations, il nous semble possible de nous en tenir à des états d'affaires wittgensteiniens (§4.2).

4.1 L'événement abstrait logique

Historiquement, nous devons à deux philosophes – Hans Reichenbach, dans son (1947) *Elements of Symbolic Logic*, et Donald Davidson, dans son (1967b) *The Logical Form of Action Sentences* – d'avoir proposé des formes logiques de phrases d'action, conférant à l'événement un statut ontologique. Les deux philosophes partagent la même doctrine sémantique de la référence directe : les phrases douées de vérité, qu'ils nomment « propositions », dénotent directement des entités, sans intermédiation d'un quelconque sens[14]. Leurs propositions sont réputées distinctes (Davidson a, du reste, fermement rejeté dans son (1967b) la forme logique proposée par Reichenbach). Nous les comparons sur l'exemple de la phrase ci-dessous :

(1) « Paul a déplacé la table »

La thèse défendue par Reichenbach est qu'une phrase comme (1) est équivalente sémantiquement à la phrase (2) « Un déplacement de la table par Paul a occurré », ces deux phrases donnant lieu à des structures logiques différentes, que nous notons respectivement (1-R) et (2-R) : selon (1-R), (1) prédique du sujet-objet « Paul » d'« avoir déplacé la table »[15] ; selon (2-R), (2) prédique du sujet-événement « Déplacement de la table par Paul » d'« avoir occurré ».

(1-R) DéplacerLaTable (Paul)
(2-R) Occurer (DéplacementDeLaTableParPaul)

Ces deux phrases dénotent, selon Reichenbach, une et une seule *situation* concrète (ce qui justifie qu'elles soient sémantiquement équivalentes), mais cette situation admet, comme décalque des formes logiques, deux décompositions distinctes : suivant (1-R), la situation a pour constituants l'objet physique 'Paul' et la propriété physique 'DéplacerLaTable'[16] ; suivant (2-R), la même situation a pour constituants l'événement physique 'DéplacementDeLaPorteParPaul' et la propriété physique 'Occurrer'. Comme on peut le

[14] Nous avons rappelé en Introduction la position de Davidson. Reichenbach, pour sa part, expose sa position en introduction de son (1947). Tout d'abord, selon Reichenbach, une phrase dans sa totalité dénote une *situation* concrète (*Ibid.*, pp. 14-15) : « Physical objects divide into *things*, such as individual human beings, tables, atoms, and *situations*, also called *states of affairs*, which constitute the denotata of sentences. Thus the sentence 'the battle-ship *Bismarck* was sunk' denotes a situation; the ship itself is a thing ». Par ailleurs, aucun intermédiaire n'existe entre la phrase et la situation (*Ibid.*, p. 15) : « When some logicians thought it necessary to distinguish between 'proposition' and 'sentence' they did so because they believed that there was a third thing between the sentence, *i.e.*, the linguistic expression, and the situation. Such a third thing is certainly unnecessary, and we shall therefore identify sentence and proposition ».

[15] Pour des raisons de simplification, nous omettons le temps dans la forme logique.

[16] La table étant un autre objet physique auquel fait référence la phrase (1), on peut préférer la forme logique (1-Rbis) Déplacer (Paul, Table), le prédicat « déplacer » se référant à une propriété binaire, ou relation.

constater, l'analyse de Reichenbach conduit à une promiscuité ontologique importante avec des *situations* côtoyant des *événements*, des *propriétés* et des *objets* physiques.

Envisageons, de façon complémentaire, la proposition de Davidson. La thèse défendue par Davidson est que tout verbe d'action, exprimant « ce que quelqu'un a fait », doit être construit comme comportant un argument implicite se référant, au moyen de variables ou de termes singuliers, à un événement. Davidson propose ainsi l'expression logique (1-D) quantifiée existentiellement, la variable 'e' prenant ses valeurs dans un domaine d'événements. Par la suite, la formulation (1-D) sera révisée en (1-DP) par Terence Parsons (1990) pour éviter l'usage de prédicats à arité variable. En vue de comparer avec la proposition de Reichenbach, on notera que ce dernier, dans (2-R), a considéré un terme singulier 'le déplacement de la table par Paul' se référant à un événement singulier. Toutefois, comme la phrase (1) laisse ouverte la référence à plusieurs actions de Paul, ce qui est le parti pris de Davidson avec (1-D), nous considérons plutôt l'expression (2-Rbis) comportant le prédicat 'DéplacementDeLaTableParPaul' se référant à un type d'événements.

(1-D) $(\exists e)$ Déplacer (Paul, Table, e)
(1-DP) $(\exists e)$ (Déplacer (e) $\wedge$ Agent (e, Paul) $\wedge$ Objet (e, Table))
(2-Rbis) $(\exists e)$ (DéplacementDeLaTableParPaul (e) $\wedge$ Occurrer (e))

Plutôt que de dégager les mérites de telle ou telle forme, nous nous apprêtons à pointer du doigt un problème d'ordre inférentiel qui, à notre sens, leur est commun. Davidson (1967b) attire justement notre attention sur ce point. Selon Davidson, en effet, à partir d'une phrase comme « Paul vole avec son vaisseau spatial vers l'Étoile du Soir », nous devrions en déduire que « Paul vole avec son vaisseau spatial vers l'Étoile du Matin », compte tenu de l'identité extensionnelle : Étoile du Soir = Étoile du Matin. Toujours selon Davidson, cette inférence est permise par une formule comme (1-DP), là où (2-Rbis) présenterait un comportement aberrant[17].

De fait, nous nous inscrivons en faux vis-à-vis de la validité de l'inférence et nous y voyons là un signe évident de l'inadéquation de l'approche extensionnelle proposée par les deux philosophes (en considérant des événements concrets). Nous considérons ainsi qu'à partir de phrases comme « Paul a déplacé la table » ou « Paul a déchiré le morceau de papier trainant sur la table » (pour garder les pieds sur terre…), il n'est pas possible d'inférer que « Paul a déplacé le seul objet dont Marie a hérité de ses parents » ou que « Paul a déchiré la dernière facture téléphonique ». Si ces phrases sont équivalentes du point de vue de la référence, elles ne sont – *a priori* – pas équivalentes sémantiquement. Les connaissances de sujets sont clairement en jeu : ainsi, Paul pousse ce qu'il considère être une table ; par contre, s'il ne dispose pas de la connaissance du fait que cette table est le seul objet dont Marie a hérité de ses parents, il ne peut penser qu'il pousse cet objet unique (le même raisonnement vaut pour la facture téléphonique). En privilégiant une interprétation *de dicto*, et non *de re*, pour le sens de ces phrases, nous tenons compte de la nature conceptuelle des événements et nous considérons - a priori – comme distincts des événements ayant des constituants conceptuels distincts, par exemple : 'Le déplacement par Paul de la table de salon' et 'Le déplacement par Paul de la table dont Marie a hérité'. La précaution que nous prenons, en disant « a priori », se justifie par le fait que nous nous situons dans un cadre général en considérant des concepts sociaux et non singuliers. S'il s'avère par contre que Paul dispose des deux concepts co-référentiels de la table

[17] Dans son (1967b), Davidson consacre deux pages à montrer l'inadéquation de (2-Rbis) sur ce point. Son argumentaire s'est toutefois révélé erroné, comme l'a montré Karl Pfeifer (1988).

Être la table de salon et *Être la table dont Marie a hérité*, nous pourrions convenir que les deux événements sont identiques pour Paul[18].

Pour rendre justice au caractère abstrait de l'événement, nous proposons d'adopter la théorie conceptualiste de la forme logique proposée par Cocchiarella (1996, 2001). Comme nous l'avons vu en §3.1, selon l'ontologie formelle de Cocchiarella, une phrase prédicative comme « S est P » s'analyse comme la combinaison d'un concept-référentiel (exprimé par « S ») et d'un concept-prédicable (exprimé par « est P »). Par exemple, une affirmation comme « tous les corbeaux sont noirs » s'analyse comme la combinaison du concept-référentiel exprimé par « tous les corbeaux » et du concept-prédicable exprimé par « sont noirs ». En symbolisant le concept-référentiel par '(∀xCorbeau)' et le concept-prédicable par 'Noir(x)', nous obtenons la forme complète : '(∀xCorbeau)Noir(x)'. De même, nous représenterions la phrase « des cygnes sont noirs » par l'expression : '(∃xCygne)Noir(x)'. Rappelons que, selon le réalisme conceptuel naturel de Cocchiarella, un même prédicat linguistique tient pour (ou signifie) en premier lieu un concept et, de façon dérivative, une propriété ou relation concrète. Il en est de même pour ces expressions avec les prédicats logiques.

Cette théorie est a priori compatible avec l'analyse de Reichenbach, en considérant qu'une phrase comme « E Occurre » exprime la combinaison d'un concept-référentiel-événement '(∃xE)' et du concept-prédicable 'Occurrer(x)'. En guise d'analyse de la phrase (1), nous obtenons :

(1-R&C) (∃xDéplacement/Agent(x,Paul)/Objet(x,Table))Occurrer(x)

Littéralement, (1-R&C) exprime qu'un événement de type 'Déplacement' – ayant pour 'Agent' 'Paul' et pour 'Objet' 'Table' – 'Occurre'. La description de l'événement est représentée à la Davidson, chaque clause introduite par '/' revenant à l'ajout d'un terme conjonctif : $(\exists xE /\varphi_1/\varphi_2/\ldots/\varphi_n)F(x) \leftrightarrow (\exists xE/\varphi_1 \wedge \varphi_2 \wedge \ldots \varphi_n)F(x)$. Par ailleurs, suivant les règles logiques – $(\exists xE/\varphi_1/\varphi_2/\ldots/\varphi_n)F(x) \vdash (\exists xE)F(x)$; $\vdash (\exists xE/\varphi_i/\varphi_j/\ldots/\varphi_k)F(x)$ $i,j,k \in \{1,\ldots,n\}$ –, nous pouvons déduire les faits suivants : « un déplacement occurre » (1-R&C') ; « un événement ayant Paul pour agent occurre » (1-R&C'') ; et « un événement ayant Table pour Objet occurre » (1-R&C''').

(1-R&C') (∃xDéplacement)Occurrer(x)
(1-R&C'') (∃xÉvénement/Agent(x,Paul))Occurrer(x)
(1-R&C''') (∃xÉvénement/Objet(x,Table))Occurrer(x)

On notera que la forme logique conceptuelle que l'on obtient préserve les avantages recherchés par Davidson (1967b), à savoir : (i) rendre compte de la polyadicité des verbes d'action, grâce à l'ajout d'un nombre arbitraire de termes conjonctifs dans la description de l'événement ; et (ii) valider, comme nous venons de le montrer, des inférences autorisées par notre description de l'événement. On retiendra par contre que le quantificateur existentiel porte sur un domaine d'événements *abstraits*. La différence majeure avec le traitement de Davidson est que les événements abstraits existent par le fait d'être pensés. L'*existence* des événements, au sens de Davidson, se transforme pour nos événements abstraits en une propriété d'*occurrence*. Cette dernière, comme nous allons le voir en §4.2, dépend de l'existence de faits concrets.

[18] On trouve une analyse comparable chez Zalta (2001).

4.2 États d'affaires occurrent-facteurs de l'événement

Nous venons d'identifier une classe de propositions consistant à prédiquer l'occurrence d'événements – dans notre nouvelle notation : '(∃xE)Occurrer(x)' – l'événement de type E existant pour un sujet *s* à un temps *t*. La proximité de la proposition et de l'événement fait que nous pouvons identifier les conditions de vérité de la proposition aux conditions d'occurrence de l'événement. En adoptant une théorie de la correspondance entre propositions/événements et états d'affaires, nous considérons ainsi que les états d'affaires « véri-facteurs », donnant à la proposition la valeur 'Vrai' ou 'Faux', sont ces mêmes états d'affaires « occurrent-facteurs » déterminant l'occurrence de l'événement. Rappelons que, concernant ces états d'affaires, nous comptons évaluer l'hypothèse de nous en tenir à des états d'affaires wittgensteiniens, plus précisément à des faits de la théorie 'at-at'. Dans la suite de cette section, nous utilisons le terme « fait » en ce sens.

Pour préciser notre notion d'*occurrence*, nous formulons une remarque complémentaire. En général c'est bien un ensemble de faits, et non un fait isolé, qui caractérisent l'occurrence d'un événement. Cet ensemble de faits détermine une région spatio-temporelle correspondant à la région spatiale occupée par les substances constituant les états d'affaires, couplée à la région temporelle couvrant les instants d'obtention des états d'affaires[19].

Dans la suite de cette section, nous envisageons des phrases exprimant l'occurrence de divers événements, des états et des changements. Pour chaque phrase, nous identifions les faits occurrent-facteurs.

Commençons par l'expression d'un état de qualité perceptive, par exemple (1a). Nous en rendons compte logiquement au moyen de la forme (1b) : dans cette expression, le prédicat 'ÊtreBlanc' représente un type d'événement qui se trouve être un concept-référentiel générique ; le prédicat 'Expr' représente la relation conceptuelle de participation d'une substance à un état[20]. Pour déterminer les faits occurrents-facteurs, nous faisons appel à la théorie classique des tropes stipulant l'existence d'une qualité particulière *inhérente* à l'objet 'Table' (dans une relation de dépendance spécifique mutuelle)[21]. Comme nous l'annoncions, nous considérons que plusieurs faits sont occurrent-facteurs, ce qui paraît nécessaire pour rendre compte d'une stabilité d'inhérence avec la même qualité '$\text{Blanc}_{\text{Table}}$'. L'identité et le nombre de ces faits dépendent du contexte d'interprétation de la véracité de la phrase (1a).

(1) a « La table est blanche »
b (∃xÊtreBlanc/Expr(x,Table))Occurrer(x)
c <Table, inhère, $\text{Blanc}_{\text{Table}}$,$I_1$>, <Table, inhère, $\text{Blanc}_{\text{Table}}$,$I_2$>, …

Profitons, puisque nous évoquons les tropes, pour signaler que nous identifions des sensations éprouvées par des sujets à des tropes. Nous proposons donc pour une phrase comme (2a) une interprétation analogue à celle de (1a). Les faits occurrent-facteurs sont des faits d'inhérence avec le trope '$\text{Inquiétude}_{\text{Paul}}$' (2c).

[19] Le lecteur notera qu'en faisant intervenir des régions spatio-temporelles et des états d'affaires, nous considérons des entités proches des *situations* dans la théorie de Jon Barwise et John Perry (1983). Les développements autour de cette théorie étant nombreux, et faute de place dans cet article, nous ne tenterons pas de comparaisons.

[20] Le lecteur notera que, en toute rigueur dans (1b), la constante 'Table' devrait être développée pour représenter un acte référentiel visant un objet physique *simpliciter*, celui-ci « comptant pour » (au sens de la construction sociale de Searle) une table. Nous devrions ainsi prendre en compte l'existence d'un fait social d'attribution d'une fonction. Par la suite, nous continuerons à user de telles simplifications.

[21] Pour une présentation récente de la théorie des tropes, le lecteur est invité à se référer à la synthèse d'Anthony Fisher (2019).

(2) a « Paul est inquiet »
b (∃xÊtreInquiet/Expr(x,Paul))Occurrer(x)
c <Paul, inhère, Inquiétude$_{Paul}$,I$_1$>, <Paul, inhère, Inquiétude$_{Paul}$,I$_2$>, …

Poursuivons avec l'expression d'un état de proximité spatiale entre deux objets, par exemple (3a), dont nous rendons compte logiquement au moyen de la forme logique (3b). Pour identifier les faits occurrent-facteurs (3c), nous adoptons l'analyse suivante. Tout d'abord, dans la suite logique des exemples précédents, nous posons l'existence de deux séries de faits de localisation, se rapportant respectivement à Paul et Marie (les crochets dénotent des séries homogènes de faits). Les faits en question sont des faits d'occupation (prédicat 'Loc') de positions spatiales (constantes 'Pos$_i$') à des instants (resp. 'I$_i$' et 'J$_i$'). Chacune de ces séries correspond, en quelque sorte, à un état de localisation d'une personne. Pour tenir compte du fait que ces états sont concomitants, nous avons ajouté comme condition que des instants I$_i$ correspondent à des instants J$_i$. Le lecteur aura noté que, dans notre inventaire des faits occurrent-facteurs (3c), nous n'avons pas fait apparaître la distance entre les régions spatiales occupées par 'Paul' et 'Marie', laquelle intervient pourtant pour évaluer leur « proximité ». Notre inventaire est donc incomplet. Mais on notera à ce propos que cette distance préexiste au fait que 'Paul' et 'Marie' occupent les régions spatiales en question et ceci nous amène à conclure que les deux séries de faits recensées dans (3c) n'ajoutent rien de plus dans le monde spatio-temporel. En d'autres termes, nous en concluons que le prédicat 'ÊtreÀCôté' ne représente qu'une relation conceptuelle (il n'existe pas de relation ordinaire correspondante).

(3) a « Paul est à côté de Marie »
b (∃xÊtreÀCôté/Expr(x,Paul)/Objet(x,Marie))Occurrer(x)
c [<Paul, loc, Pos$_{I1}$, I$_1$>, <Paul, loc, Pos$_{I2}$, I$_2$>, …] ; [<Marie, loc, Pos$_{J1}$, J$_1$>, <Marie, loc, Pos$_{J2}$, J$_2$>, …] (*certains I$_i$ coïncident avec des J$_i$; l'immobilité de Paul et de Marie peut entraîner que des I$_i$ (resp. J$_i$) soient identiques entre eux*)

Pour rester sur des états, mais des états impliquant des processus, considérons les expressions d'événements-processuels (4a) et (5a). De tels événements sont habituellement qualifiés de « faires » dans le sens où ils rendent compte de changements dans le monde attribuables à des substances. La signification que nous accordons à (4a) (resp. (5a)) est qu'un événement de type 'Marche' (resp. 'Déplacement') est en train d'occurrer (4b) (resp. (5b)). Le concept-référentiel 'Marche' (resp. 'Déplacement') tient pour un type d'événement. Les faits occurrent-facteurs sont des faits d'énaction de processus (4c) (resp. (5c)). Dans (4c), 'Marcher$_{\#i}$' est à considérer comme une instance du type de processus 'Marcher', à distinguer donc de l'instance du type d'événement 'Marche' (nous retrouvons la distinction entre 'Marche$_{Proc}$' et 'Marche$_{Event}$' discutée en Section 2). Dans (5c), nous avons représenté le fait que Paul énacte un processus 'Proc$_{\#i}$', dont le type n'est pas précisé, lequel *perpétue* le processus 'Déplacer$_{\#j}$' énacté par la table[22].

(4) a « Paul marche »
b (∃xMarche/Agent(x,Paul))Occurrer(x)
c [<Paul, énacte, Marcher$_{\#i}$, I$_1$>, <Paul, énacte, Marcher$_{\#i}$, I$_2$>, …]

(5) a « Paul déplace la table »

[22] Nous empruntons ici à Galton (2012) la relation de *perpétuation* entre processus. Cette relation est à entendre comme une propagation causale entre deux processus déjà existants : un premier processus actif entretient l'activité d'un second processus. Par exemple, tant que Paul pousse une table, celle-ci se déplace.

b (∃xDéplacement/Agent(x,Paul)/Patient(x,Table))Occurrer(x)

c [<Paul, énacte, $Proc_{\#i}$,I_1>, <Paul, énacte, $Proc_{\#i}$,I_1>,…] ; [<Table, énacte, $Déplacer_{\#j}$, I_1>, <Table, énacte, $Déplacer_{\#j}$, I_1>,…] ; [<$Proc_{\#i}$, perpétue, $Déplacer_{\#j}$,I_1>, <$Proc_{\#i}$, perpétue, $Déplacer_{\#j}$,I_2>, …]

Arrêtons-nous sur le traitement de ces événements-processuels. Ces exemples nous montrent qu'il est possible de distinguer, côté sens, des événements abstraits conceptuels et, côté référence, des énactions de processus concrets. Nous pourrions aller plus loin en montrant comment des adverbes modifient le sens de tels énoncés. Dans un cas comme « Paul marche rapidement », l'adverbe qualifie la vitesse du processus. Au contraire, dans un cas comme « Paul s'est soudainement mis à marcher », l'adverbe « soudainement » qualifie l'événement en lui-même : il lui correspond un fait événementiel abstrait[23].

A ce stade, établissons un bilan. Comme nous l'avions envisagé, les seuls faits occurrent-facteurs auxquels nous avons eu recours au sein des séries de faits sont des faits wittgensteiniens. Les relations identifiées - « inhère », « localisation », « énacter », « perpétue » - peuvent être considérées comme des relations de fondation physiques. La preuve en est que ces relations ne sont pas exprimées par des termes de la langue naturelle. A contrario, ceci signifie que la catégorie des relations « ordinaires » devient sans objet et que nous devons considérer la grande majorité des relations comme des relations conceptuelles, comme nous l'avons fait avec la relation 'Être à côté'. Ce constat tient sur un échantillon de phrases et nécessitera d'être consolidé. Rappelons que le fait de considérer des processus physiques à côté des objets physiques, d'attribuer aux objets physiques des qualités particulières pouvant évoluer dans le temps, est un aspect déterminant pour le bilan que nous venons de dresser.

5 Conclusion

Dans cet article, nous avons présenté une défense du cadre ontologique que nous promouvons couramment, concernant notamment la nature abstraite des événements et la nature concrète des processus, en le positionnant vis-à-vis de théories contemporaines du *sens* et de la *référence*. Nous avons adopté à cette occasion un double réalisme ordinaire et conceptuel, qui s'est révélé déterminant pour analyser la nature des propriétés-relations et distinguer celles étant uniquement conceptuelles de celles qui représentent des propriétés-relations ordinaires.

En guise de propriétés-relations ordinaires, nous avons émis l'hypothèse de nous en tenir à des propriétés-relations dénommées par Mulligan *et al.* (1984) « relations de fondation mutuelle », ce qui revient à ne considérer dans le monde physique que des faits wittgensteiniens constitués de particuliers. En considérant un petit échantillon de phrases assertant l'occurrence d'événements, correspondant à des états mais également des changements, nous avons pu montrer le bien-fondé de cette hypothèse.

Récemment, la philosophe Arianna Betti, dans son (2015) *Against facts*, très critique quant à l'existence de faits armstrongiens comportant un universel comme constituant, est arrivée à un constat proche. En guise de « relation de fondation mutuelle », Betti a repris à son compte la notion russelienne de « relation reliante ». La clef de voûte de ces théories ontologique et sémantique réside très clairement dans la nature des relations que l'on considère. Nous considérons cette question de la nature des relations comme le principal chantier qui s'ouvre devant nous.

[23] Faute de place, nous ne pouvons détailler ici les traitements. Le lecteur intéressé peut se référer à (Kassel, 2018).

Références

ARMSTRONG, D.M. (1997). *A world of states of Affairs*. Cambridge University Press.

BARWISE, J. & PERRY, J. (1983). *Situations and Attitudes*. Cambridge, MA: The MIT Press.

BENOIST, J. (ed.)(2006). *Propositions et états de choses. Entre être et sens*. Paris, Librairie Philosophique J. Vrin.

BERGSON, H. (1946). *The creative Mind*. New York, Philosophical Library.

BETTI, A. (2005). Propositions et états de choses chez Twardowski. *Dialogue*, 14, 469-92.

BETTI, A. (2015). *Against facts*. The MIT Press.

CHISHOLM, R.M. (1970). Events and Propositions. *Noûs*, 4, 15-24.

CLELAND, C.E. (1990). The Difference Between Real Change and Mere Cambridge Change. *Philosophical Studies*, 60, 257-280.

COCCHIARELLA, N.B. (1996). Conceptual Realism as a Formal Ontology. In R. POLI & P. SIRNOR (eds.), *Formal Ontology*, Dordrecht: Kluwer Academic Press.

COCCHIARELLA, N.B. (2001). Logic and Ontology. *Axiomathes*, 12, 117-150.

DAVIDSON, D. (1967a). Truth and Meaning. *Synthese*, 17(3), 304-323.

DAVIDSON, D. (1967b). The Logical Form of Action Sentences. In N. RESCHER (ed.), *The Logic of Decision and Action* (pp. 81-95), Pittsburgh: University of Pittsburgh Press.

FINE, K. (1982). First-Order Modal Theories III – Facts. *Synthèse*, 53, 43-122.

FISHER, A.R.J. (2019). *Abstracta* and Abstraction in Trope Theory. *Philosophical Papers*, https://www.tandfonline.com/doi/full/10.1080/05568641.2019.1571938

GALTON, A. (2006) On What Goes On: The ontology of processes and events. In R. FERRARIO & W. KUHN (eds.), proc. of the *Fourth International Conference on Formal Ontology in Information Systems (FOIS2006)*, pp. 4-11.

GALTON, A. (2008). Experience and History: Processes and their Relation to Events. *Journal of Logic and Computation*, 18(3), 323-40.

GALTON, A. (2012). States, Processes and Events, and the Ontology of Causal Relations. In M. DONNELLY & G. GUIZZARDI (eds.), Proc. of the *7th Int. Conf. on Formal Ontology in Information Systems* (pp. 279-92), IOS Press.

GALTON, A. & MIZOGUCHI, R. (2009). The water falls but the waterfall does not fall: New perspectives on objects, proceses and events. *Applied Ontology*, 4, 71-107.

GEACH, P. (1968). What actually Exists? In *Proc. of the Aristotelian Society*, Supplementary Volumes, 42, pp. 7-16.

HANKS, P. (2017). Propositions, Synonymy, and Compositional Semantics. In F. MOLTMANN & M. TEXTOR (eds.), *Act-Based Conceptions of Propositional Content: Contemporary and Historical Perspectives* (pp.235-53), Oxford University Press.

KASSEL, G. (2018). Ontologie de l'action et formes logiques des phrases d'action : de nouvelles perspectives. In T. DE LIMA et S. DOUTRE (eds.), Actes des *12èmes Journées d'Intelligence Artificielle Fondamentale*, Amiens, 13-15 juin, 2018.

KASSEL, G. (2019). Processes Endure, Whereas Events Occur. In S. BORGO, R. FERRARIO, C. MASOLO & L. VIEU (eds.), *Ontology Makes Sense* (pp. 177-193), IOS Press.

KASSEL, G. (2020). Physical processes, their life and their history. *Applied Ontology*. *Sous presse*.

KÜNNE, W. (2009). Bolzano and (Early) Husserl on Intentionality. In G. PRIMIERO & Sh. RAHMAN (eds.), *Acts of Knowledge: History, Philosophy and Logic, Essays Dedicated to Göran Sundholm* (pp. 95-140), London: College Publications.

MOLTMANN, F. (2013). *Abstract Objects and the Semantics of Natural Language*. Oxford University Press.

MOLTMANN, F. & TEXTOR, M. (eds.)(2017). *Act-Based Conceptions of Propositional Content: Contemporary and Historical Perspectives*, Oxford University Press.

MULLIGAN, K., SIMONS, P. & SMITH B. (1984). Truth-Makers. *Philosophy and Phenomenological Research*, 44, 287-321.

PARSONS, T. (1990). *Events in the Semantics of English*. Cambridge (Mass.): MIT Press.

PFEIFER, K. (1988). A short vindication of Reichenbach's "event-splitting". *Logique et Analyse*. 31(121-122), 143-152.

REICHENBACH, H. (1947). *Elements of Symbolic Logic*. New York: Macmillan.

RUSSELL, B. (1903). *Principles of Mathematics*. Cambridge, UK: Cambridge University Press.

SMITH, B. (1995). *Austrian Philosophy, Brentano's Legacy*. Chicago, Open Court.

SOAMES, S. (2014). Cognitive propositions. In J.C. KING, S. SOAMES & J. SPEAKS (eds.), *New thinking about propositions* (pp. 91-124), Oxford: Oxford University Press.

STOUT, R. (1997). Processes. *Philosophy*, 72(279), 19-27.

STOUT, R. (2003). The life of a process. In G. DEBROCK (ed.), *Process Pragmatism: Essays on a Quiet Philosophical Revolution* (pp. 145-57), Rodopi.

VALLICELLA, W.F. (2000). Three Conceptions of States of Affairs. *Noûs*, 34(2), 237-259.

WILSON, N. (1974). Facts, Events, and Their Conditions. *Philosophical Studies*, XXV, 303-321.

ZALTA, E.N. (1983). *Abstract Objects. An Introduction to Axiomatic Metaphysics*. D. REIDEL Publishing Company.

ZALTA, E.N. (2001). Fregean Senses, Modes of Presentation, and Concepts. *Philosophical Perspectives*, *Noûs* Supplement, 15(2001), 335-359.

WITTGENSTEIN, L. (1922/1921). *Tractatus Logico-Philosophicus*. London, Routledge and Kegan Paul, 1922; trad. anglaise de *Logisch-Philosophische Abhandlung*, Wilhelm Ostwald (ed.), *Annalen der Naturphilosophie*, 14, 1921.

MEMORAe-CWE : un système collaboratif de systèmes d'information à base d'ontologies

Siying Li[1], Marie-Hélène Abel[1], Elsa Negre[2]

[1] Sorbonne universités, Université de Technologie de Compiègne, CNRS UMR 7253, HEUDIASYC, 60203 Compiègne
{siying.li, marie-helene.abel}@utc.fr

[2] Université Paris-Dauphine, PSL Research University, CNRS UMR 7243, LAMSADE, 75016 Paris, France
elsa.negre@dauphine.fr

Résumé : Intégrant divers outils de collaboration, un environnement de travail collaboratif basé sur le web, support à la collaboration, vise à faciliter la production et le partage de ressources. Durant les processus collaboratifs les ressources produites sont de fait stockées de manière distribuée au sein des systèmes/outils rendus accessibles par l'environnement. Le problème de leur accès se pose, il ne s'agit pas d'autoriser un accès individualisé dans chaque système mais bien de proposer un accès contextualisé à partir de l'environnement de travail collaboratif. Dans le cadre de notre recherche, nous considérons un environnement de travail collaboratif comme un système collaboratif de systèmes d'information développé à partir d'une ontologie définissant le concept de collaboration et son contexte. C'est à partir de cette ontologie que les fonctionnalités de l'environnement sont développées et permettent notamment d'organiser le partage et l'accès aux ressources produites au sein des différents systèmes. Les traces sémantiques de collaboration enregistrées peuvent alors être utilisées pour effectuer des recommandations de ressources aux collaborateurs/utilisateurs.
Mots-clés : Environnement de Travail Collaboratif, Système de Systèmes d'Information, Ontologie, Contexte de collaboration.

1 Introduction

Grâce au développement des technologies de l'information, de nombreux outils de collaboration sont mis à la disposition des utilisateurs, tels que les outils de mail et de chat en temps réel (Xu *et al.*, 2008; Wang, 2016). Ces outils numériques peuvent être intégrés dans un Environnement de Travail Collaboratif (ETC) basé sur le web (Martínez-Carreras *et al.*, 2007). De tels environnements permettent aux utilisateurs de collaborer au-delà de la limite des distances géographiques et rendent le travail collaboratif dans les entreprises plus agile et flexible. Pendant leur collaboration, les utilisateurs doivent généralement utiliser différents outils au sein desquels ils sont amenés à générer diverses ressources (par exemple, des documents, des vidéos, etc.). Le problème de leur accès se pose, il ne s'agit pas d'autoriser un accès individualisé dans chaque système mais bien de proposer un accès contextualisé à partir de l'ETC. Dans le cadre de notre travail nous visons l'organisation de telles ressources distribuées au moyen d'un référentiel partagé permettant un accès centralisé au sein d'un ETC.

Dans les ETCs, les ressources sont stockées dans différents outils de collaboration qui sont des systèmes d'information autonomes (Neto *et al.*, 2017). Ces systèmes, ainsi que l'environnement lui-même, peuvent être vus comme un Système de Systèmes d'Information (SdSI) (Saleh & Abel, 2016). Compte tenu des différentes gestions de ressources dans des SdSIs (Maier, 1998; Dahmann & Baldwin, 2008), nous proposons de construire un ETC comme un SdSI collaboratif basé sur une ontologie de collaboration tenant compte de son contexte afin que chaque ressource puisse être gérée au moyen d'une base de connaissances (Saleh & Abel, 2016) et stockée dans le système d'information où elle a été produite.

Les ressources produites et/ou utilisées lors d'une collaboration dépendent de l'objectif de la collaboration et plus généralement de son contexte (Li *et al.*, 2019). Il semble donc nécessaire de tenir compte de ce dernier pour les décrire afin de faciliter leur utilisation. A cette fin, nous avons fait le choix de construire une ontologie du contexte de collaboration. Cette dernière sera utilisée pour l'accès aux ressources mais également effectuer leur recommandation contextuelle.

Cet article est structuré de la manière suivante. La section 2 présente les principaux concepts et approches liés aux définitions d'un ETC, à l'architecture des SdSIs et aux recommandations contextuelles. La section 3 décrit nos contributions pour (i) adapter une architecture de SdSI basé sur une ontologie aux ETCs, (ii) mettre en oeuvre une ontologie du contexte de collaboration, (iii) développer un prototype de ETC basé sur l'architecture et l'ontologie du contexte de collaboration définies. Nous discutons ensuite des forces et faiblesses du prototype en section 4. Enfin, nous concluons et proposons quelques perspectives de recherche dans la section 5.

2 Etat de l'art

Dans cette section, nous présentons ce qu'est un Environnement de Travail Collaboratif (ETC) et discutons de ses fonctions. Nous précisons également la notion de Système de Systèmes d'Information (SdSI). Enfin, nous expliquons en quoi consistent les recommandations contextuelles et leurs avantages.

2.1 Environnement de Travail Collaboratif - ETC

Pour atteindre un objectif commun (Oliveira *et al.*, 2011), une collaboration implique deux personnes ou plus et comprend un ensemble d'actions réalisées par des acteurs humains agissant pour le compte du collaborateur correspondant (Li *et al.*, 2018b,a). De nos jours, de plus en plus de personnes collaborent à distance à l'aide de diverses technologies, telles que le Web/Internet, les technologies de l'information et de la communication (TIC) et les technologies dans le domaine du travail coopératif assisté par ordinateur (TCAO) (Martínez-Carreras *et al.*, 2007; Su & Casamayor, 2009). Cela conduit à l'émergence d'un nouvel espace collaboratif, un ETC où les utilisateurs peuvent travailler ensemble en groupes spontanés et dynamiques assemblés de manière collaborative (Prinz *et al.*, 2006).

Les ETCs, en particulier ceux basés sur le web, permettent des collaborations numériques entre les utilisateurs en groupes (Bafoutsou & Mentzas, 2002). Chaque groupe dispose d'un espace accessible à ses membres (utilisateurs) (Bentley *et al.*, 1997). Cela permet à tous les membres d'un groupe de travailler dans un espace partagé au sein du groupe (Bafoutsou & Mentzas, 2002). Par ailleurs, différentes applications de groupe peuvent être intégrées dans les ETCs en tant qu'outils de collaboration (Martínez-Carreras *et al.*, 2007). En y ajoutant ces outils, les ETCs peuvent donner aux utilisateurs la possibilité de collaborer dans plusieurs groupes en même temps. D'autres outils basés sur les TICs sont également disponibles dans les ETCs, tels que des outils de mail, de partage de documents et de gestion de projets (Wang, 2016; Bafoutsou & Mentzas, 2002; Truong *et al.*, 2008).

Selon Martínez-Carreras *et al.* (2007); Su & Casamayor (2009); Prinz *et al.* (2006); Bafoutsou & Mentzas (2002), les ETCs offrent des services tels que :

1) Permettre aux utilisateurs de collaborer dans le temps et l'espace ;
2) Soutenir les différentes activités des utilisateurs pendant leurs collaborations, telles que la communication, la coordination et l'interaction ;
3) Intégrer et fournir différents outils de collaboration : les outils asynchrones [1] (par exemple, mail et Wiki) et les outils synchrones [2] (par exemple, les systèmes de chat en temps réel et de communication par vidéo) ;
4) Fournir des services flexibles aux utilisateurs afin de les aider dans leurs collaborations ;
5) Autoriser l'interopérabilité avec différents outils de collaboration ;
6) Augmenter la productivité et la créativité dans les processus collaboratifs ;
7) Améliorer la pensée critique et la capacité d'analyse et de résolution des problèmes des utilisateurs.

1. Les outils asynchrones permettent aux utilisateurs de collaborer à différents moments (Xu *et al.*, 2008).
2. Les outils synchrones permettent aux utilisateurs de collaborer en même temps (Wang, 2016).

Plusieurs travaux de recherche ont déjà développé des ETCs dans différents domaines. Par exemple, Su & Casamayor (2009) ont réalisé un ETC pour promouvoir la conception durable du mobilier. Truong *et al.* (2008) ont fusionné plusieurs outils de collaboration dans un ETC dédié aux collaborations en équipe. Parmi ces ETCs existants, il existe une difficulté partagée et non résolue : le passage d'un outil de collaboration à un autre impose une charge liée aux ressources (par exemple, documents, ...) aux utilisateurs (ter Hofte, 1998). Lorsque les utilisateurs alternent entre des collaborations ou effectuent plusieurs activités en utilisant différents outils de collaboration, ils ont besoin de copier et/ou déplacer des ressources entre ces outils. Particulièrement, ces ressources sont stockées dans différentes bases de données, soit dans les outils où elles ont été produites, soit dans l'ETC lui-même. Cela rend plus difficile l'accès et la gestion des ressources dans un ETC. De ce fait, cela pose aussi des problèmes sur la façon d'intégrer ces outils au sein d'un ETC (Prinz *et al.*, 2006).

Dans le cadre de notre travail, nous visons un accès centralisé aux ressources via l'ETC. A cette fin il est nécessaire de penser dès la conception de l'ETC la façon de lui lier les outils de collaboration qu'il rendra accessible à l'utilsateur/collaborateur. Chaque outil est un système autonome d'information avec sa propre base de données. Les outils et le noyau de l'ETC forment un système favorisant les collaborations entre utilisateurs. Un tel système peut être considéré comme un Système de Systèmes d'Information (SdSI) (Saleh & Abel, 2016).

2.2 Système de Systèmes d'Information - SdSI

La notion de Système de Systèmes d'Information (SdSI) est conceptuellement dérivée de celle de **Système de Systèmes (SdS)**. Un SdS est un nouveau type de système formé par la relation qu'il entretient avec ses composants, qui sont eux-mêmes des systèmes indépendants (Assaad *et al.*, 2016). Dans un tel système, on peut distinguer le *système global* et des *systèmes composants*. Les systèmes composants sont des systèmes indépendants et hétérogènes. Le système global consiste en un système faisant le lien entre les différents systèmes composants (Assaad *et al.*, 2016). En ce qui concerne l'ETC, ses outils intégrés de collaboration sont les systèmes composants, tandis que l'ETC lui-même correspond au système global.

Un SdS peut être classifié selon le lien qu'il met en place avec les différents systèmes composants (Maier, 1998; Dahmann & Baldwin, 2008). On distingue quatre catégories :

- Un **SdS dirigé - *directed*** est construit pour répondre à des objectifs spécifiques et il est géré de manière centralisée (par exemple, les systèmes responsables du développement des systèmes futurs de combat au sein du Ministère de la Défense des États-Unis (Dahmann & Baldwin, 2008)).
- Un **SdS reconnu - *aknowledged*** a une gestion centrale et des ressources communes. Néanmoins, les systèmes composants maintiennent leurs propriétés indépendantes et leurs objectifs. Les changements dans le SdS sont basés sur la collaboration entre le SdS et les systèmes composants (par exemple, un centre des opérations aériennes (Dahmann & Baldwin, 2008)).
- Un **SdS collaboratif** n'a pas de gestion centrale. Les systèmes composants collaborent entre eux pour atteindre les objectifs centraux (par exemple, Internet [3] (Maier, 1998; Dahmann & Baldwin, 2008)).
- Un **SdS virtuel** n'a ni gestion centrale ni d'objectifs communs reconnus au niveau central. Il est le résultat de l'interaction entre ses composants, alors que les objectifs sont inconnus. Ce SdS est maintenu par des mécanismes invisibles (par exemple, World Wide Web [4] (Maier, 1998)).

De plus, si chaque système composant d'un SdS est un système d'information, ce SdS est un SdSI (Saleh & Abel, 2016). Le système d'information contient un ensemble de composants interdépendants qui effectuent des activités visant à collecter, traiter, stocker et distribuer des informations, tandis qu'un système est un ensemble d'éléments dynamiquement

3. L'internet est un réseau mondial d'ordinateur à ordinateur, dont éléments sont des réseaux informatiques et des sites informatiques importants (Maier, 1998).

4. Le World Wide Web est un système de systèmes qui n'existe qu'au niveau des couches supérieures du protocole (Maier, 1998).

interdépendants pour effectuer des activités visant à atteindre un objectif spécifique (Neto *et al.*, 2017). Un SdSI peut être vu comme un SdS spécial concernant l'information. De ce fait, le SdSI peut aussi être classifié selon les quatre catégories (dirigé, reconnu, collaboratif, virtuel). Par exemple, Internet est un SdSI collaboratif qui rassemble différents systèmes d'information pour fournir divers services informatiques aux utilisateurs (voir la figure 1).

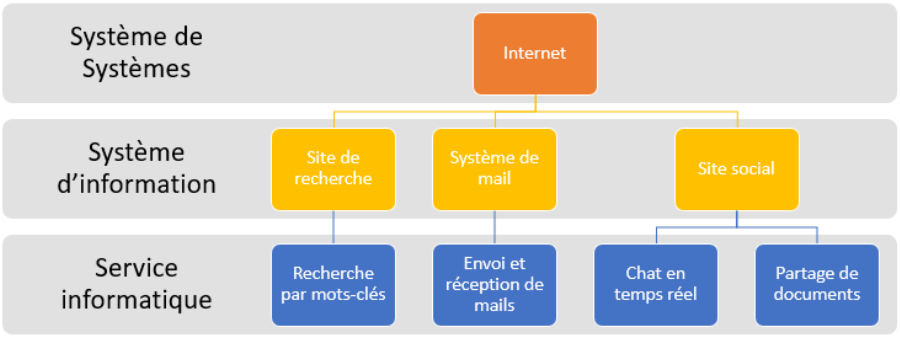


FIGURE *1 – Internet vu comme un SdSI collaboratif.*

Concrètement, chaque outil de collaboration intégré dans un ETC est un système d'information indépendant et autonome avec sa propre base de données et sa propre gestion de ressources. Un ETC peut être considéré comme un SdSI, qui peut également être classifié selon les quatre catégories des SdSIs. Par exemple, l'ETC créé par Su & Casamayor (2009) est reconnu, parce qu'il dispose d'une gestion centrale et de ressources communes. Cependant l'accès et la gestion de ressources communes dans de tels ETCs peut créer une barrière pour les personnes (cf. Section 2.1).

Pour organiser les ressources réparties entre les systèmes d'information dans un ETC, on peut considérer qu'un ETC est un SdSI collaboratif avec une gestion centrale des ressources. En effet, il n'existe pas de ressources communes dans un tel ETC. Au contraire, chaque ressource appartient exclusivement au système d'information où elle a été générée. Cependant, toutes les ressources peuvent être visibles, accessibles et gérées de manière centralisée au sein de l'ETC. Notamment, un SdSI collaboratif basé sur une ontologie, qui peut gérer des informations à travers des systèmes séparés, a été développé (Saleh & Abel, 2016). Compte tenu de la relation entre les ETCs et les SdSIs collaboratifs, nous décidons donc de mettre en oeuvre un ETC sous la forme d'un SdSI collaboratif développé à partir d'une ontologie définissant le concept de collaboration et son contexte. C'est à partir de cette onologie que les fonctionnalités de l'environnement sont développées et permettent notamment d'organiser le partage et l'accès aux ressources produites au sein des différents systèmes. Les traces sémantiques de collaboration enregistrées peuvent alors être utilisées pour effectuer des recommandations de ressources aux collaborateurs/utilisateurs.

2.3 Recommandations contextuelles

Les recommandations sont générées par des Systèmes de Recommandation (SRs) qui sont conçus et appliqués pour trouver l'information la plus pertinente aux besoins des utilisateurs et la transférer aux utilisateurs (Nunes & Jannach, 2017). Généralement, les SRs traitent de deux dimensions des données : les utilisateurs et les items (Adomavicius *et al.*, 2005). Ici, *items* est le terme générique pour indiquer les objets que des SRs recommandent aux utilisateurs, tandis que *utilisateurs* indiquent les personnes qui recevront ces objets recommandés (Ricci *et al.*, 2011). Par exemple, dans un ETC, des ressources peuvent être recommandées aux utilisateurs pour les aider lors de collaborations. Un item se réfère à une ressource.

Pour fournir des recommandations, les SRs doivent identifier et trier les items en fonction de leur utilité (scores) (Ricci *et al.*, 2011) qui indique comment un utilisateur particulier

aime un item spécifique (Adomavicius & Tuzhilin, 2005). Ainsi, la tâche principale des SRs est la suivante : étant donné un ensemble initial de scores que les utilisateurs donnent explicitement ou implicitement pour les items, les SRs essaient de calculer/comparer les scores inconnus/manquants des items et décident quels items recommander, sur la base des données des utilisateurs et des items (Ricci *et al.*, 2011) (Adomavicius & Tuzhilin, 2011).

Adomavicius & Tuzhilin (2005) ont modélisé la fonction d'utilité en utilisant deux dimensions de données comme suit :

$$R_{SR\,2D} : Utilisateur \times Item \rightarrow Score \quad (1)$$

où $Score$ est un ensemble ordonné (par exemple, des entiers non négatifs ou des nombres réels dans un certain intervalle).

Récemment, une nouvelle branche des SRs, les Systèmes de Recommandation Contextuels (SRCs), a été proposée par Adomavicius *et al.* (2005). Un SRC est construit en incorporant le contexte dans les processus de génération de recommandations, ce qui peut déboucher sur des recommandations plus précises (Palmisano *et al.*, 2008), notamment des recommandations contextualisées.

Les SRCs visent à recommander des items aux utilisateurs (Ricci *et al.*, 2011) en fonction du contexte. Ici, le contexte est toute information qui peut être utilisée pour caractériser la situation des utilisateurs, des items, ou l'interaction entre les utilisateurs et les items (Dey, 2001). Dans l'exemple ci-dessus, le contexte indique le fait que les utilisateurs collaborent entre eux dans certains groupes au sein d'un ETC. Basées sur les trois dimensions des données : utilisateurs, items et contextes, l'utilité (scores) des items dans les SRCs indiquent comment un utilisateur particulier aime un item spécifique (Adomavicius & Tuzhilin, 2005) dans un certain contexte. Par conséquent, la fonction d'utilité (scores) d'un SRC est (Adomavicius *et al.*, 2005) :

$$R_{SRC} : Utilisateur \times Item \times Contexte \rightarrow Score \quad (2)$$

Dans le cadre de nos recherches, le contexte dans l'équation (2) indique les informations contextuelles sur les interactions entre des utilisateurs et des ressources au sein des ETCs. Ces interactions visent notamment à atteindre les objectifs communs des collaborations entre utilisateurs. Cela implique que les informations contextuelles sont pertinentes par rapport aux collaborations. Ainsi, ces informations contextuelles appartiennent également aux contextes de collaboration. En considérant les ETCs comme des SdSIs collaboratifs basés sur une ontologie, une ontologie de contexte de collaboration peut être mise en œuvre et appliquée dans les ETCs pour recommander des ressources aux utilisateurs. Ces recommandations sont destinées à améliorer les collaborations des utilisateurs dans certains groupes, contribuant ainsi à rendre plus efficace le travail collaboratif dans les ETCs.

3 Contributions

Cette section présente le problème de l'accès centralisé et l'organisation des ressources dans des Environnements de Travail Collaboratif (ETCs) à l'aide d'un scénario de travail collaboratif (Section 3.1). Pour résoudre de tels problèmes, les ETCs peuvent être construits en adaptant l'architecture d'un Système de Systèmes d'Information (SdSI) collaboratif basé sur une ontologie (Section 3.2) et en appliquant une ontologie de contexte de collaboration (Section 3.3).

Dans le cadre de nos recherches, basé sur cette architecture et sur l'ontologie du contexte de collaboration, un prototype correspondant d'ETC a été développé, qui gère les ressources distribuées dans des outils différents. Enfin, le prototype est présenté (Section 3.4). Il permet de faciliter les collaborations et de générer des recommandations de ressources aux utilisateurs.

IC 2020

3.1 Un scénario de travail collaboratif

L'entreprise FileX vise à développer diverses applications d'éditeurs et de traducteurs de fichiers. Actuellement, l'entreprise se concentre sur deux applications : une application web et une application Android d'un éditeur de fichiers. Les deux applications sont respectivement réalisées par deux groupes. En particulier, Lucie, Mary et Steve collaborent pour l'application web. Lucie, Emma et Leo travaillent ensemble pour l'application Android. Par ailleurs, l'application desktop de l'éditeur de fichiers a été réalisée par Lucie, Majd et Nathalie.

Au cours du développement des deux applications, Lucie trouve un document intéressant ("File editor document 1") et une vidéo utile ("File editor video"). Elle souhaite donc les partager séparément dans les deux groupes de travail collaboratif et en discuter avec d'autres personnes. Cependant, les deux ressources sont stockées différemment. La vidéo est accessible sur YouTube, tandis que le document est enregistré dans l'ordinateur de Lucie. De plus, lors de ses discussions dans les deux groupes, Lucie aimerait comparer les différentes idées des personnes de ses deux groupes sur l'éditeur de fichiers.

Pour permettre à Lucie de travailler plus efficacement dans son travail, un ETC peut être utilisé dans son entreprise, FileX. Un tel ETC permet non seulement de gérer les ressources stockées dans les différents outils de collaboration, mais aussi de relier ces ressources à un vocabulaire commun qui décrit les objectifs des collaborations de Lucie. Tout en développant un tel ETC, une architecture de SdSI collaboratif basé sur une ontologie est adaptée pour construire des ETCs permettant d'avoir accès aux ressources. Parallèlement, une ontologie de contexte de collaboration est également utilisée pour gérer ces ressources avec des vocabulaires partagés d'objectifs de collaboration.

3.2 Une architecture d'Environnement de Travail Collaboratif

Saleh & Abel (2016) ont proposé une architecture leader/suiveur de SdSI collaboratif (voir partie (a) de la figure 2). Dans cette architecture, un SdSI collaboratif est composé d'un système leader qui est le système global et de multiples systèmes suiveurs qui représentent les systèmes composants. De plus, le système global contient une base ontologique de connaissances pour gérer collectivement les informations entre les systèmes d'information.

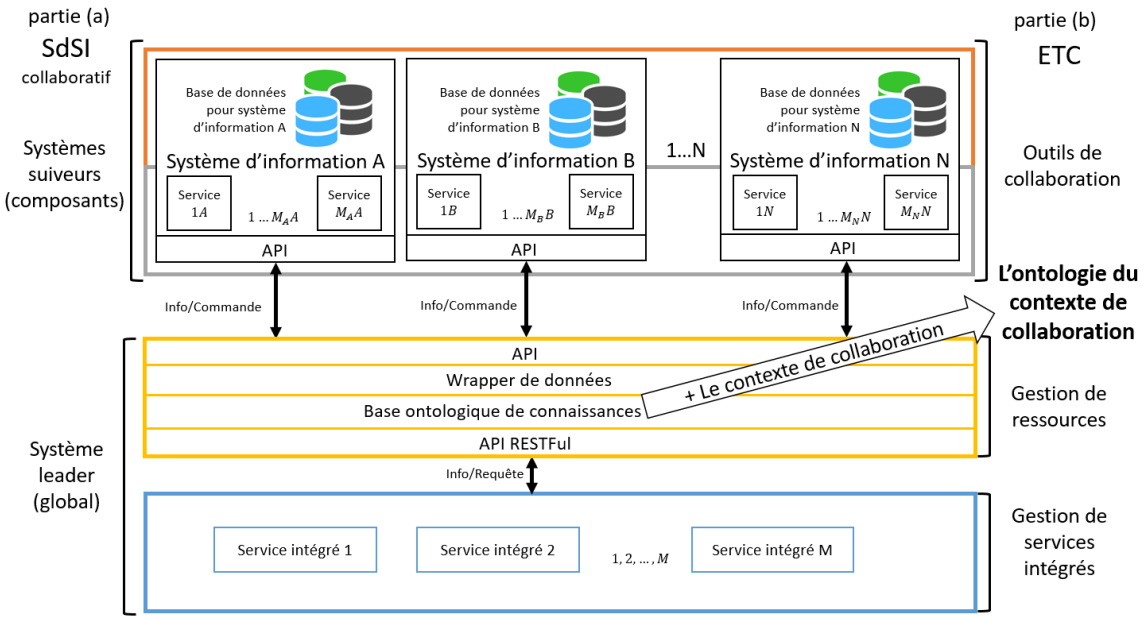


FIGURE 2 – *Considérer un ETC comme un SdSI collaboratif basé sur une ontologie.*

En considérant un ETC comme un SdSI collaboratif basé sur une ontologie, cette architecture leader/suiveur peut aussi servir d'architecture pour un ETC (voir partie (b) de la figure 2). Les systèmes composants sont des outils de collaboration qui fournissent leurs propres services (par exemple, Service 1A, ..., $M_A A$ dans la figure 2). Ils peuvent volontairement choisir de quitter ou de rejoindre un ETC pour y intégrer des services destinés à supporter les collaborations (par exemple, Service intégré 1, ..., M dans la figure 2). En outre, les quatre parties de l'architecture peuvent gérer les ressources dans l'ETC de la manière suivante : *API*, *Wrapper de données*, *Base ontologique de connaissances* et *API RESTFul*. *API* échange des informations de ressources avec les outils de collaboration. Ensuite, *Wrapper de données* vise à gérer les informations collectées via *API* et à les stocker dans une *base ontologique de connaissances*. Enfin, *API RESTFul* offre des interfaces uniformes pour l'accès et la manipulation de ressources (Lucchi *et al.*, 2008). Une telle architecture (voir partie (b) de la figure 2) permet aux ressources situées dans des outils indépendants d'être visibles et accessibles dans des ETCs sans modifier leur lieu de stockage (source).

Considérant que les collaborations sont orientées par l'objectif (Oliveira *et al.*, 2011; Li *et al.*, 2018a), les ressources utilisées dans les collaborations sont pertinentes par rapport à leurs objectifs. De ce fait, les ressources doivent être gérées en fonction des objectifs de collaboration dans les ETCs. Toutefois, les ressources et les objectifs sont influencés par le contexte de collaboration, par exemple les activités des collaborateurs (Li *et al.*, 2019). De ce fait, nous devons prendre en compte le contexte de collaboration dans la base ontologique de connaissances pour gérer des ressources. Ainsi, une ontologie du contexte de collaboration (Li *et al.*, 2019) est mise en place comme base de connaissances dans l'architecture de l'ETC (voir partie (b) de la figure 2). Cette architecture contribue à la construction d'un prototype d'ETC où les ressources sont gérées au sein du contexte de collaboration.

3.3 Une ontologie du contexte de collaboration

L'ontologie du contexte de collaboration **MEMORAe-Collaboration-Context (MCC)** (Li *et al.*, 2019) applique un concept de groupe d'utilisateurs, mcc :UserGroup (voir la figure 3), pour représenter des collaborations. Autour de ce concept, huit dimensions du contexte de collaboration sont représentées par différents concepts et/ou leurs inter-relations : Objectif, Collaborateur, Activité, Ressource, Temps, Lieu, Relation et Satisfaction (Li *et al.*, 2019). Un mcc :UserGroup est établi pour une période de temps, a un objectif et fournit un espace de groupe accessible à ses membres (au moins deux) avec leurs comptes utilisateurs.

Dans les espaces de groupe, les utilisateurs peuvent accéder et consulter les ressources distribuées dans les différents outils de collaboration via leurs comptes d'utilisateurs. Comme montré dans la figure 3, les ressources sont contenues dans la base de données où elles ont été stockées à l'origine, soit dans un ETC (représenté par ms :LeaderSystem), soit dans un outil de collaboration (représenté par ms :WebBasedApplication et ms :SandAloneSystem) intégré dans l'ETC. Pour associer les ressources dans les outils de collaboration avec l'ETC, ms :ReferenceKey est appliqué. Chaque ms :ReferenceKey est inclus dans l'ETC et possède un mc2 :IndexKey qui est visible dans certains espaces de groupe. Cela permet de lier des ressources avec des groupes d'utilisateurs. En utilisant mc2 :IndexKey et ms :ReferenceKey, toutes les ressources sont accessibles et visibles dans les groupes d'utilisateurs de l'ETC. Lorsque des personnes effectuent différentes activités sur des ressources (par exemple, partage, vote et/ou suppression de ressources), ce sont leurs mc2 :IndexKey et ms :ReferenceKey qui sont utilisés et modifiés, plutôt qu'elles-mêmes dans les bases de données. Les activités possibles sur des ressources dans l'ETC sont présentées dans la figure 4. Ces activités peuvent être enregistrées en tant que traces d'activité des utilisateurs dans des groupes d'utilisateurs et classées en 6 catégories : création, ajout, partage, modification, accès et suppression de ressources (marqués par des rectangles dans la figure 4).

Par ailleurs, mcc :Goal (voir les figures 3 et 4) représente l'objectif de la collaboration. Comme les collaborations sont pertinentes les unes pour les autres au sein d'une entreprise ou d'une organisation, nous appliquons un graphique organisationnel des connaissances pour définir un vocabulaire commun décrivant les objectifs de collaboration. Chaque noeud du graphe est un concept utile pour les objectifs de collaboration au sein d'une organisation,

IC 2020

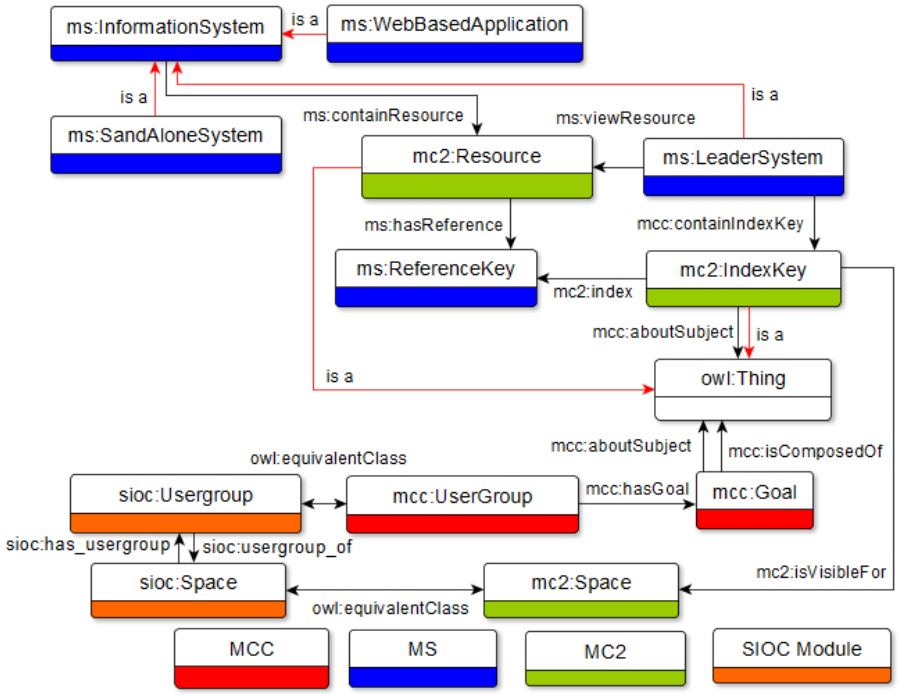


FIGURE *3 – Module de ressources dans MCC (Li* et al.*, 2019).*

représenté par owl :Thing (voir les figures 3 et 4). Parallèlement, chaque mc2 :IndexKey est aussi lié à owl :Thing. Cela signifie que les ressources et les activités sur des ressources dans l'ETC peuvent être attachées aux objectifs de collaboration par MCC.

L'application de MCC comme base ontologique de connaissances permet d'indexer et de gérer les ressources avec un vocabulaire commun des objectifs de collaboration. Cela permet de déterminer l'utilité des ressources et de générer des recommandations de ressources aux utilisateurs pour faciliter leurs collaborations dans des ETCs.

3.4 Prototype : MEMORAe-CWE

Basé sur l'architecture d'ETC (cf. Section 3.2) et sur l'ontologie du contexte de collaboration (cf. Section 3.3), un prototype d'ETC a été développé : **MEMORAe-CWE**. Cet prototype est développé à partir d'une plateforme MEMORAe, qui a déjà été testée et utilisée par des utilisateurs réels à l'Université de Technologie de Compiègne.

Dans MEMORAe-CWE, les utilisateurs ont accès à diverses organisations. Une fois qu'une organisation est sélectionnée, les utilisateurs peuvent parcourir le graphe de connaissances de l'organisation. Les concepts concernant les objectifs de collaboration dans l'organisation sont affichés via les noeuds dans le graphe, et leurs relations via les liens. Chaque noeud et lien a sa propre description. Dans le scénario de travail collaboratif (cf. Section 3.1), l'entreprise FileX définit les concepts avec des descriptions dans un graphe de connaissances. Tous les employés de FileX peuvent consulter le graphe au sein de MEMORAe-CWE (voir la figure 5).

Chaque organisation est composée d'un ou plusieurs groupes d'utilisateurs dont les membres collaborent les uns avec les autres pour atteindre un objectif commun. Dans le scénario, les deux groupes pour le développement d'applications différentes sont deux groupes d'utilisateurs dans MEMORAe-CWE. Leurs objectifs sont de développer l'application correspondante de l'éditeur de fichiers. Particulièrement, chaque groupe d'utilisateurs fournit un espace

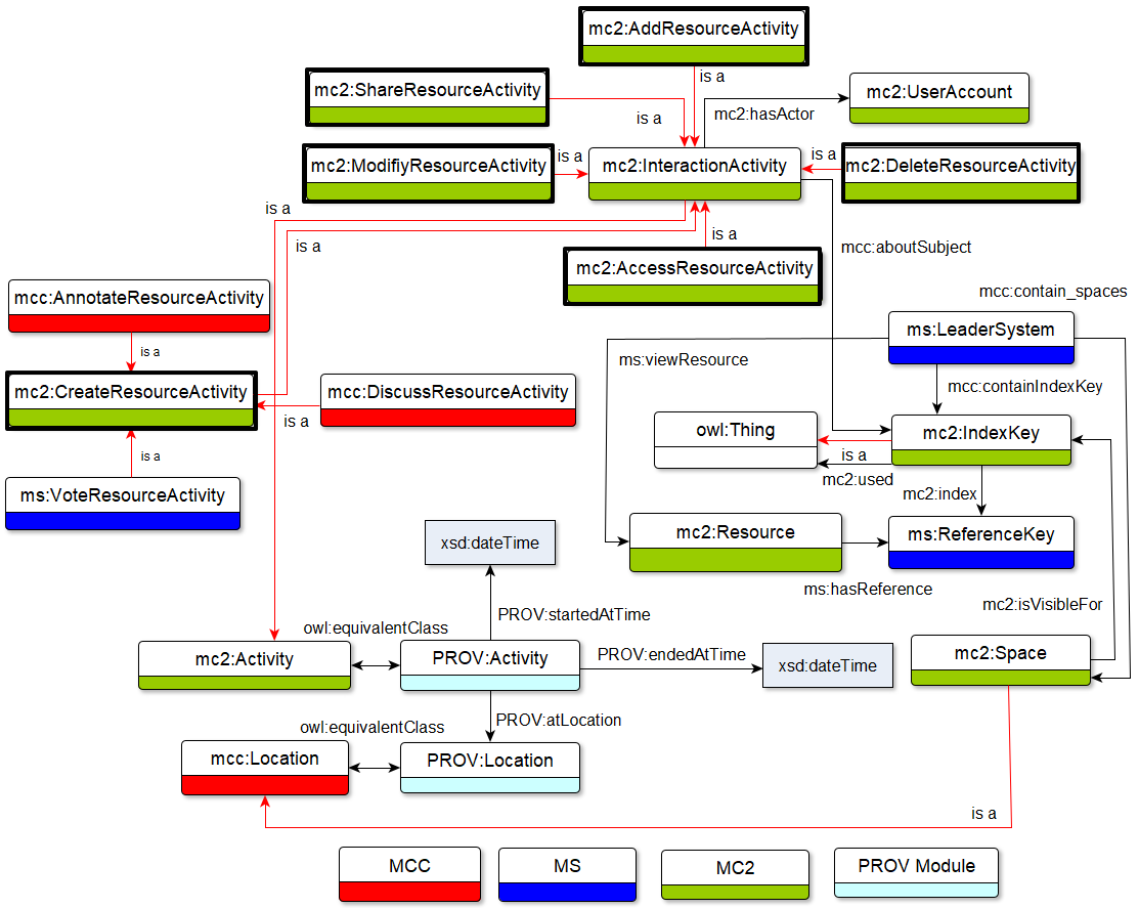


FIGURE *4 – Module de activité dans MCC (Li* et al.*, 2019).*

IC 2020

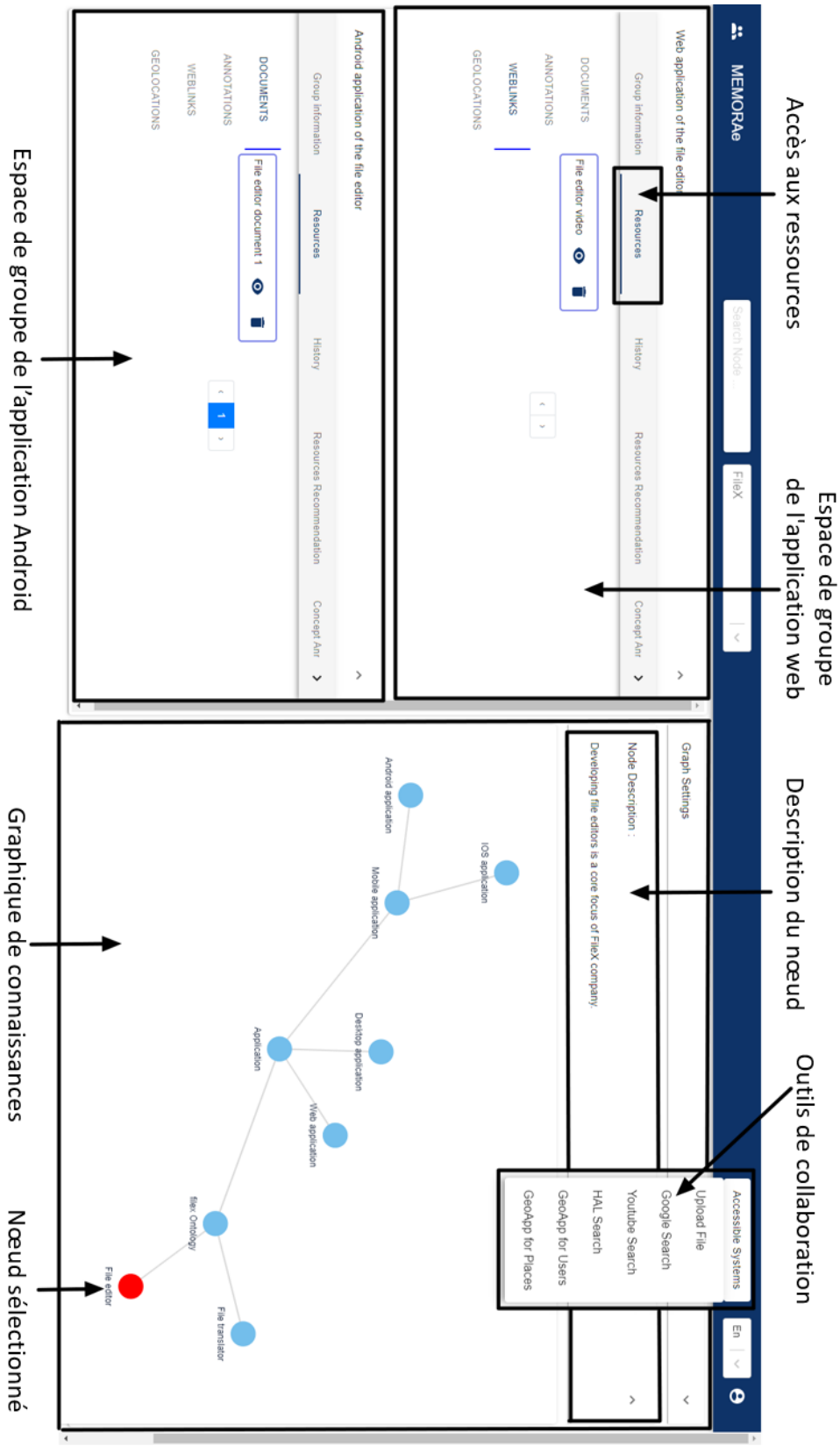


FIGURE 5 – *L'interface de Lucie dans MEMORAe-CWE.*

de groupe qui permet à ses membres d'interagir avec des ressources distribuées dans différents outils de collaboration. Dans le scénario, Lucie peut partager avec Mary et Steve le lien web de la vidéo YouTube sur les éditeurs de fichiers dans l'espace du groupe. Ici, le lien web est une ressource qui est stockée dans YouTube mais accessible et visible dans un groupe d'utilisateurs de MEMORAe-CWE (voir la figure 5). En plus d'un lien web, une ressource peut également être un document, une annotation, un vote, un commentaire ou un lieu géographique.

Par ailleurs, toutes les ressources de MEMORAe-CWE sont indexées avec des noeuds dans les graphes de connaissance. Dans le scénario, le lien web partagé par Lucie est indexé et visible avec le noeud *File editor*. Les noeuds d'un graphe de connaissances définissent un vocabulaire partagé utilisé par les collaborateurs pour décrire les objectifs de collaboration. Notamment, les ressources sont visibles et accessibles uniquement avec les noeuds indexés. Cela aide les personnes à organiser les ressources autour des objectifs de collaboration. Lorsqu'un graphe de connaissances est partagé au sein d'une organisation, tous les membres des groupes dans l'organisation peuvent indexer les ressources dans le graphe. Cela permet aux utilisateurs de comprendre et de visualiser l'indexation des ressources dans une organisation à partir d'un seul graphe. Lorsque Lucie sélectionne le noeud *File editor* dans MEMORAe-CWE, elle peut consulter toutes les ressources indexées au sein des deux groupes d'utilisateurs (voir la figure 5), et différencier les ressources dans des groupes distincts.

MEMORAe-CWE peut intégrer des outils de collaboration pour aider les utilisateurs à mener diverses activités sur les ressources. Ces outils se trouvent dans le menu *Accessible Systems* de la figure 5. Dans le scénario, les activités de Lucie dans le groupe d'utilisateurs de l'application Android sont indiquées dans la figure 6. Ces activités peuvent être enregistrées sous forme de traces d'activité et visualisées par type d'activité (voir la figure 6), type de ressource, nom de la ressource, nom du concept (index), date et acteur (utilisateur). Grâce à ces traces d'activité sur des ressources, les ressources peuvent être recommandées aux utilisateurs dans les groupes d'utilisateurs correspondants. Ces recommandations visent à faciliter les collaborations des utilisateurs dans MEMORAe-CWE.

Considérant que les ressources indexées et les concepts (noeuds) pertinents sont deux ensembles dans un groupe d'utilisateurs, alors un groupe d'utilisateurs c peut être représenté comme l'union de ces deux ensembles $R^c \cup S^c$. $R^c = \left\{ r_j^c | j \leq N_r^c, j \in N^+ \right\}$ est l'ensemble de toutes les ressources indexées dans le groupe d'utilisateurs c; N_r^c est le nombre de ressources dans le groupe d'utilisateurs c; $S^c = \{ s_k^c | k \leq N_s^c, k \in N^+ \}$ est l'ensemble de tous les concepts pertinents qui indexent les ressources dans le groupe d'utilisateurs c; N_s^c est le nombre de concepts dans le groupe d'utilisateurs c.

Ensuite, le problème de recommandation contextuelle de ressources dans MEMORAe-CWE est formulé comme suit :

Étant donné un concept s et un utilisateur u dans un groupe d'utilisateurs c avec deux ensembles de ressources R^c et de concepts S^c, les K meilleures ressources $i(i \notin R^c)$ qui peuvent être indexées avec le concept s dans le groupe d'utilisateurs c avec les plus grandes probabilités seront recommandées à l'utilisateur u pour faciliter sa collaboration dans le groupe c. Ici, le contexte comprend le concept s et le groupe d'utilisateurs c.

Les étapes pour générer des recommandations contextuelles [5] sont :

1. Calculer la similarité $S(c_i, c)(i = 1, 2, ..., N-1)$ entre les ensembles de ressources dans le groupe d'utilisateurs c et les autres groupes d'utilisateurs $c_i(c_i \neq c)$ sur un graphe partagé des connaissances. Ici, N est le nombre de groupes d'utilisateurs dont l'utilisateur u est membre.
2. Ordonner les groupes d'utilisateurs c_i en fonction de $S(c_i, c)$ par ordre décroissant.
3. Filtrer les ressources non pertinentes $i(i \notin R^c)$ qui ne sont pas incluses dans c_i en prenant les K plus élevés $S(c_i, c)$ ou non indexées avec le concept s, et obtenir les ressources pertinentes i'.

5. Notamment, $score(u, i')$ est donnée en fonction des activités de vote des utilisateurs sur la ressource i', ce score appartient à l'intervalle $[0, 5]$. Dans cet article, nous ne discutons pas de la manière dont cela est calculé.

IC 2020

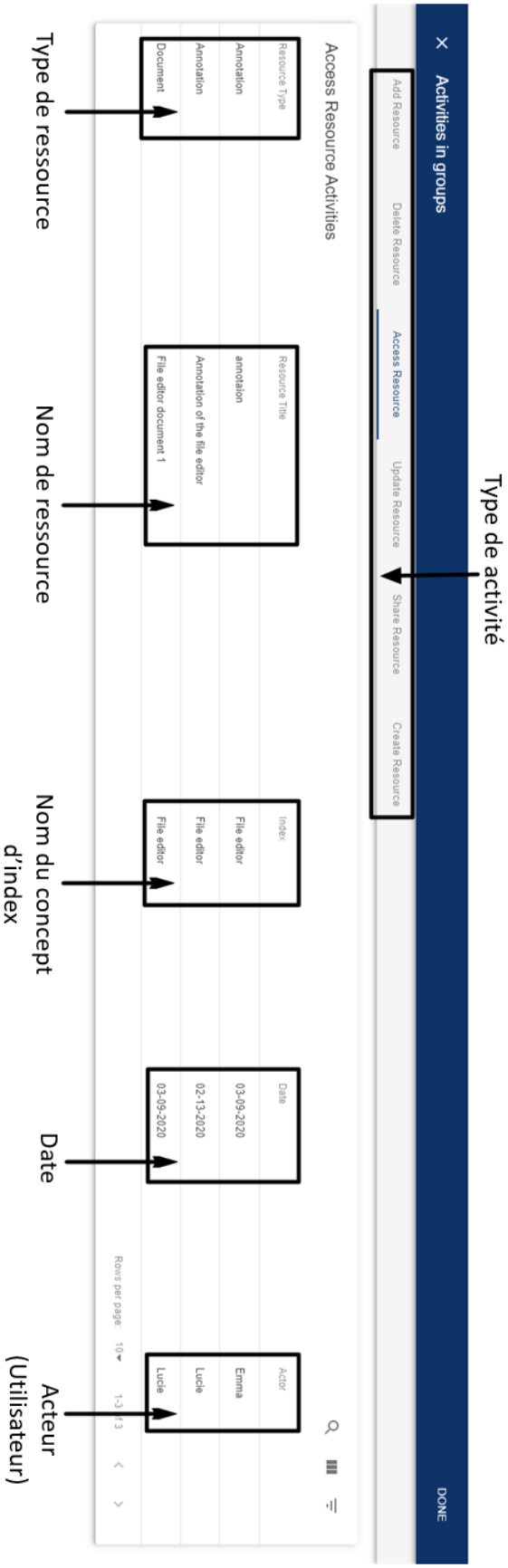


FIGURE 6 – *Les activités de Lucie dans le groupe d'utilisateurs de l'application Android.*

4. Recommander/Retourner les ressources ayant les K plus élevés $score(u, i')$ à l'utilisateur u dans le groupe d'utilisateurs c.

Pour calculer la similarité entre deux ensembles de ressources dans les groupes d'utilisateurs, nous devons repérer les ressources communes. Inspirée de l'indice de Tversky (Tversky, 1977) et d'une similarité sémantique (Carrer-Neto *et al.*, 2012), la similarité sémantique entre un groupe d'utilisateurs c et un autre groupe d'utilisateurs x est :

$$S(x,c) = \sum_{a=1}^{M} \frac{|R^x_{s_a} \cap R^c_{s_a}| \times Weight(s_a)}{|R^x_{s_a} \cap R^c_{s_a}| + \alpha|R^x_{s_a} - R^c_{s_a}| + \beta|R^c_{s_a} - R^x_{s_a}|} \quad (3)$$

où $M = |S^x \cap S^c|$ est le nombre de concepts indexant les ressources dans les deux groupes d'utilisateurs x et c ; $R^c_{s_a}$ est l'ensemble de toutes les ressources indexées avec le concept s_a dans le groupe d'utilisateurs c ; $R^x_{s_a} - R^c_{s_a}$ est le complément de $R^c_{s_a}$ dans $R^x_{s_a}$; $|R^x_{s_a} \cap R^c_{s_a}|$ est le nombre de ressources communes indexées avec le concept s_a dans les groupes d'utilisateurs x et c ; $Weight(s_a)$ est le poids du concept $s_a (a = 1, 2, ..., M)$; $\alpha, \beta \geq 0$.

De plus, la valeur de K dans les étapes est spécifiée par l'utilisateur u. Dans le scénario, nous considérons $K = 1$ pour illustrer comment une ressource peut être recommandée à l'utilisateur u dans le groupe d'utilisateurs c pour le concept s. Dans ce cas, Lucie peut recevoir une recommandation de ressources sur le concept *File editor* dans le groupe d'utilisateurs de l'application web. L'utilisateur u est Lucie ; le concept s est *File editor* ; le groupe d'utilisateurs de l'application web est c ; c_1 et c_2 sont des groupes d'utilisateurs de l'application desktop et de l'application Android. Spécifiquement, les ressources dans c et c_2 sont uniquement indexées avec le concept s. Nous avons $R^c = \{"File\ editor\ video"\}$ et R^{c_2} contenant *"File editor document 1"* et *"Annotation of the file editor"*. Quant à c_1, $R^{c_1}_s$ contient *"File editor video"*, *"File editor document 1"*, et *"Comments for the file editor"*.

A la suite de ces étapes, nous devons calculer $S(c_1, c)$ et $S(c_2, c)$. Supposons que $Weight(s) = 1$ et $\alpha = \beta = 1$, l'intervalle de $S(x, c)$ est donc $[0, 1]$. Avec des scores appartenant à $[0, 5]$, nous supposons : $score(u, "File\ editor\ document\ 1") = 4.7$ et $score(u, "Comments\ for\ the\ file\ editor") = 3.8$. Puis, nous pouvons obtenir $S(c_1, c) = \frac{1\times1}{1+2+0} + 0 = \frac{1}{3} \approx 0.333$ et $S(c_2, c) = \frac{0\times1}{0+2+1} + 0 = 0$. $S(c_1, c) > S(c_2, c)$ indique que c_1 a plus de ressources communes indexées avec le concept *File editor*. En d'autres termes, c_1 est plus similaire à c qu'à c_2.

Ensuite, les ressources que nous pouvons recommander à Lucie sont *"File editor document 1"* et *"Comments for the file editor"*, car *"File editor video"* est déjà indexé dans c. De plus, *"Annotation of the file editor"* est filtré/écarté car il n'existe pas dans c_1 avec $S(c_1, c)$ qui est la valeur la plus élevée dans les similarités. Enfin, la valeur la plus élevée, $score(u, "File\ editor\ document\ 1")$, implique que *"File editor document 1"* doit être recommandé à Lucie.

Grâce à MEMORAe-CWE, les utilisateurs sont capables de collaborer avec d'autres via les groupes d'utilisateurs. Dans chaque groupe, les ressources sont accessibles et organisées de manière centralisée par un graphe de connaissances qui définit un vocabulaire commun d'objectifs de collaboration. Cela permet d'indexer les ressources avec des objectifs de collaboration et de générer des recommandations contextuelles de ressources aux utilisateurs pour faciliter leurs collaborations dans un tel ETC.

4 Discussion

Considérer un Environnement de Travail Collaboratif (ETC) comme un Système de Systèmes d'Information (SdSI) collaboratif basé sur une ontologie peut améliorer l'intégration de différents outils collaboratifs. Cela permet également d'accéder aux ressources qui sont distribuées dans des outils distincts. Par conséquent, les ressources sont des composants directement accessibles (Lucchi *et al.*, 2008) dans un tel ETC. Cela apporte une approche plus légère d'accès et de gestion des ressources dans les ETCs.

Grâce à l'ontologie du contexte de collaboration (Li *et al.*, 2019), les ressources dans l'ETC peuvent être organisées et indexées avec un vocabulaire commun utilisé pour décrire les objectifs de collaboration dans l'ETC. Avec ces informations, les utilisateurs peuvent prendre les ajustements appropriés pendant les collaborations. Par exemple, pour préparer une collaboration, MCC peut être utilisé pour trouver des ressources indexées avec des concepts pertinents dans le vocabulaire commun.

De plus, les utilisateurs de l'ETC peuvent également annoter et voter pour des ressources lorsqu'ils collaborent au sein d'un groupe d'utilisateurs. Ces annotations et votes sont stockés dans l'ETC et sont accessibles aux membres du groupe d'utilisateurs. Cela permet de fournir des informations contextuelles de collaboration aux utilisateurs afin qu'ils puissent s'adapter pour mieux collaborer avec les autres membres du groupe. L'indexation conjointe des ressources et des objectifs de collaboration aide à générer des recommandations de ressources aux utilisateurs.

Enfin, le contexte de collaboration défini par MCC permet de formuler des recommandations contextuelles pour améliorer les collaborations. De telles recommandations ont déjà été produites par d'autres études. Par exemple, Liu *et al.* (2018) ont concentré leurs efforts sur les recommandations contextuelles des collaborateurs académiques pour soutenir les collaborations scientifiques. Cependant, ces études antérieures se concentrent généralement sur le contexte de l'utilisateur ou sur le contexte de l'item [6]. Aucune d'entre elles ne tient compte à la fois des utilisateurs et des items. Avec le contexte de collaboration défini dans MCC, les utilisateurs et les items (ressources) sont pris en compte conjointement dans les collaborations au sein d'un ETC.

5 Conclusion

Dans cet article, nous nous concentrons sur le problème de l'accès centralisé et l'organisation des ressources dans un Environnement de Travail Collaboratif (ETC). Nous avons développé un prototype d'ETC en adaptant une architecture de Système collaboratif de Systèmes d'Information (SdSI collaboratif) basé sur une ontologie et en appliquant une ontologie de contexte de collaboration.

A partir des travaux de la littérature, nous avons justifié la relation entre ETC et SdSI, et expliqué pourquoi un ETC peut être considéré comme un SdSI collaboratif basé sur une ontologie. Nous avons étudié comment une ontologie du contexte de collaboration peut être implémentée dans un ETC via l'architecture d'un SdSI collaboratif basé sur une ontologie. Le prototype correspondant a ensuite été présenté. Les caractéristiques de ce prototype ont finalement été discutées.

Nos perspectives de recherche comprennent la rélisation et l'amélioration de l'algorithme de recommandation et le développement du système de recommandation contextuel correspondant en tant que nouvel outil intégré à notre prototype. De plus, pour évaluer ce prototype, nous prévoyons de le tester auprès des étudiants de l'Université de Technologie de Compiègne (UTC) pour la gestion de leur travail collaboratif dans le cadre d'une veille technologique.

Références

ADOMAVICIUS G., SANKARANARAYANAN R., SEN S. & TUZHILIN A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, **23**(1), 103–145.

ADOMAVICIUS G. & TUZHILIN A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6), 734–749.

ADOMAVICIUS G. & TUZHILIN A. (2011). Context-aware recommender systems. In *Recommender systems handbook*, p. 217–253. Springer.

6. Dans leur recherche (Liu *et al.*, 2018), un item est un collaborateur, tandis que nous nous sommes intéressés à générer des recommandations contextuelles de ressources.

ASSAAD M. A., TALJ R. & CHARARA A. (2016). A view on systems of systems (sos). In *20th World Congress of the International Federation of Automatic Control (IFAC WC 2017) - special session*, Toulouse, France.

BAFOUTSOU G. & MENTZAS G. (2002). Review and functional classification of collaborative systems. *International journal of information management*, **22**(4), 281–305.

BENTLEY R., APPELT W., BUSBACH U., HINRICHS E., KERR D., SIKKEL K., TREVOR J. & WOETZEL G. (1997). Basic support for cooperative work on the world wide web. *International journal of human-computer studies*, **46**(6), 827–846.

CARRER-NETO W., HERNÁNDEZ-ALCARAZ M. L., VALENCIA-GARCÍA R. & GARCÍA-SÁNCHEZ F. (2012). Social knowledge-based recommender system. application to the movies domain. *Expert Systems with applications*, **39**(12), 10990–11000.

DAHMANN J. S. & BALDWIN K. J. (2008). Understanding the current state of us defense systems of systems and the implications for systems engineering. In *2008 2nd Annual IEEE Systems Conference*, p. 1–7 : IEEE.

DEY A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, **5**(1), 4–7.

LI S., ABEL M.-H. & NEGRE E. (2018a). Contact and collaboration context model. In *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, p. 1–6 : IEEE.

LI S., ABEL M.-H. & NEGRE E. (2018b). Modèle de contexte de collaboration : pour qui, pourquoi, comment ? In *29es Journées Francophones d'Ingénierie des Connaissances (IC 2018)*, p. 229–243, Nancy, France.

LI S., ABEL M.-H. & NEGRE E. (2019). Towards a collaboration context ontology. In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, p. 93–98 : IEEE.

LIU Z., XIE X. & CHEN L. (2018). Context-aware academic collaborator recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 1870–1879 : ACM.

LUCCHI R., MILLOT M. & ELFERS C. (2008). Resource oriented architecture and rest. *Assessment of impact and advantages on INSPIRE, Ispra : European Communities*.

MAIER M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering : The Journal of the International Council on Systems Engineering*, **1**(4), 267–284.

MARTÍNEZ-CARRERAS M. A., RUIZ-MARTÍNEZ A., GOMEZ-SKARMETA F. & PRINZ W. (2007). Designing a generic collaborative working environment. In *IEEE International Conference on Web Services (ICWS 2007)*, p. 1080–1087 : IEEE.

NETO V. V. G., ARAUJO R. & DOS SANTOS R. P. (2017). New challenges in the social web : Towards systems-of-information systems ecosystems. In *Anais do VIII Workshop sobre Aspectos da Interação Humano-Computador para a Web Social*, p. 1–12 : SBC.

NUNES I. & JANNACH D. (2017). A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, **27**(3-5), 393–444.

OLIVEIRA I., TINOCA L. & PEREIRA A. (2011). Online group work patterns : How to promote a successful collaboration. *Computers & Education*, **57**(1), 1348–1357.

PALMISANO C., TUZHILIN A. & GORGOGLIONE M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, **20**(11), 1535–1549.

PRINZ W., LOH H., PALLOT M., SCHAFFERS H., SKARMETA A. & DECKER S. (2006). Ecospace–towards an integrated collaboration space for eprofessionals. In *2006 International Conference on Collaborative Computing : Networking, Applications and Worksharing*, p. 1–7 : IEEE.

RICCI F., ROKACH L. & SHAPIRA B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, p. 1–35. Springer.

SALEH M. & ABEL M.-H. (2016). Moving from digital ecosystem to system of information systems. In *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, p. 91–96 : IEEE.

SU D. & CASAMAYOR J. (2009). Web-based collaborative working environment and sustainable furniture design. In *Proceedings of the 19th CIRP Design Conference–Competitive Design* : Cranfield University Press.

TER HOFTE G. H. (1998). Working apart together : Foundations for component groupware. p. 34–52.

TRUONG H.-L., DUSTDAR S., BAGGIO D., CORLOSQUET S., DORN C., GIULIANI G., GOMBOTZ R., HONG Y., KENDAL P., MELCHIORRE C. *et al.* (2008). Incontext : A pervasive and collaborative working environment for emerging team forms. In *2008 International Symposium on*

Applications and the Internet, p. 118–125 : IEEE.
TVERSKY A. (1977). Features of similarity. *Psychological review*, **84**(4), 327.
WANG N. (2016). *Towards a competency recommender system from collaborative traces*. PhD thesis, Université de Technologie de Compiègne.
XU J., ZHANG J., HARVEY T. & YOUNG J. (2008). A survey of asynchronous collaboration tools. *Information Technology Journal*, **7**(8), 1182–1187.

Vers une ontologie des interactions HTTP

Mathieu Lirzin[1,2], Béatrice Markhoff[2]

[1] Néréide, 8 rue des déportés, 37000 Tours, France
mathieu.lirzin@nereide.fr

[2] LIFAT EA 6300, Université de Tours, Tours, France
beatrice.markhoff@univ-tours.fr

Résumé : Les systèmes d'information d'entreprise ont adopté les bases du Web pour les échanges entre programmes hétérogènes. Ces programmes fournissent et consomment via des APIs Web des ressources identifiées par des URIs, dont les représentations sont transmises via HTTP. Par ailleurs HTTP reste au coeur de tous les développements du Web (Web sémantique, données liées, IoT...). Ainsi les situations où un programme doit pouvoir raisonner sur des interactions (requête-réponse) HTTP se multiplient. Cela suppose de disposer d'une spécification formelle explicite d'une conceptualisation partagée de ce qu'est une interaction HTTP. Une proposition de vocabulaire RDF existe, développée dans l'optique de réaliser des tests de conformité d'applications Web et enregistrer les résultats de ces tests. Ce vocabulaire a déjà été réutilisé pour d'autres applications. Dans cet article nous décrivons comment nous l'adaptons pour les besoins de notre application, dans l'optique que cette adaptation serve plus largement.

Mots-clés : HTTP, interaction HTTP, logique de description, ontologie, RDF, OWL

1 Introduction

Le Web repose sur trois standards : l'adressage par liens hypermédia (URI), le langage de balisage HTML et le protocole HTTP [1]. Le protocole HTTP est également utilisé dans les systèmes d'information d'entreprises, en particulier pour les échanges entre programmes hétérogènes : il suffit de mettre en oeuvre un serveur HTTP d'un côté et un client HTTP de l'autre. Malgré sa simplicité apparente ce protocole permet de gérer tous les aspects des communications client-serveur tout en maintenant sa capacité d'évolution pour prendre en compte de nouveaux aspects. Cette large couverture se reflète dans le volume de la spécification de HTTP 1.1, qui comprend 8 RFC de l'IETF, eux-mêmes précisés et étendus par d'autres RFC. Cette spécification doit être respectée non seulement par les concepteurs de serveurs et de navigateurs Web mais aussi par tout développeur de service Web et plus généralement encore tout développeur d'application qui doit utiliser un service accessible par HTTP. L'objectif de la proposition présentée dans cet article est de formaliser la spécification du protocole HTTP pour pouvoir décrire des *interactions* qui le respectent et, plus précisément, pouvoir vérifier automatiquement le respect du protocole. Depuis que le Web existe plusieurs propositions ont été élaborées pour décrire des interactions entre clients et serveurs Web, avec divers moyens et objectifs. La plupart sont exploitables par des programmes, mais *seulement à un niveau syntaxique*. Or *nous avons besoin d'une ontologie des interactions HTTP pour développer une validation d'API Web par rapport à une spécification formelle de ses fonctionnalités*. La description ontologique d'interactions HTTP se heurte à une difficulté importante, celle de représenter les liens hypermédia et leurs usages, à la fois comme représentants (identifiants) de ressources et comme moyens d'accès à des représentations (ou descriptions) de ces ressources, voire à la ressource elle-même (si document Web). Ainsi les usages des URIs dans HTTP relèvent à la fois des données et du contrôle sur ces données. Cet aspect est puissant en matière d'expressivité pour les développeurs qui écrivent des programmes qui interagissent à travers ces données et HTTP. Mais il est ardu de le décrire formellement et il demeure également difficile d'écrire des programmes qui reposent sur HTTP qui soient fiables et robustes face aux évolutions.

1. Standards maintenus par l'IETF (https ://tools.ietf.org) : respectivement, rfc7320, rfc2854 et rfc7230.

Notre proposition dans cet article est d'une part de commencer à formaliser la description d'interactions en HTTP en utilisant une logique de description pour pouvoir valider cette formalisation via un raisonneur, et d'autre part d'apporter des éléments de réponse quant à la difficulté qui vient d'être évoquée. Nous synthétisons ainsi nos contributions :

— Nous utilisons la logique de description $\mathcal{SROIQ}^{(\mathcal{D})}$ pour décrire nos propositions, lesquelles sont construites sur les bases d'un vocabulaire initié dans le cadre du W3C et destiné à représenter HTTP en RDF [2] (Koch *et al.*, 2017a), vocabulaire que par la suite nous nommons HTTPinRDF. La logique de description nous permet de caractériser précisément les classes et les propriétés définies dans ce vocabulaire et d'introduire de façon cohérente celles que nous proposons.

— Les en-têtes d'un message HTTP ont la forme d'une table d'associations (nom, valeur), avec des valeurs dont les types sont hétérogènes. Nous en proposons (section 4.5) une représentation générique, tout en montrant comment en avoir aussi une représentation spécifique plus précise, par exemple pour l'élément d'en-tête standard `Location` dont la valeur est un URI. Cela précise et simplifie la représentation adoptée par le vocabulaire W3C RDF HTTP, où toutes les valeurs sont des littéraux.

— Le contenu du corps d'un message HTTP peut avoir différent formats, ce qui est précisé à l'aide de déclarations dites *types de media*. Ce mécanisme générique offre une grande souplesse en ce qui concerne les types de données échangées, mais il rend délicat la représentation formelle de ces données. Nous proposons (section 4.6) de représenter en RDF à la fois le message (requête ou réponse), ses métadonnées et les données contenues dans son corps, pour un sous-ensemble des types de contenus autorisés par HTTP. Cela rend possible d'exploiter chacun de ces constituants avec les outils du Web sémantique (que ce soit SPARQL, SHACL, STTL, ...).

— Un URI est un identifiant qui respecte une syntaxe bien définie, il identifie une ressource, il permet d'accéder à des représentations de cette ressource, si cette ressource est un service Web il permet de lui communiquer des valeurs de paramètres, etc. Précisément dans le contexte d'une requête HTTP, une partie optionnelle de l'URI est très fréquemment utilisée pour paramétrer le comportement de la procédure responsable du traitement des requêtes côté serveur. Nous proposons (section 4.3) une représentation des différentes parties syntaxiques d'un URI et en particulier de la série de (clé, valeur) représentant ses paramètres, lorsqu'ils existent. Cela permet des références explicites à ces différents composants, ce qui est utile par exemple pour vérifier leur présence.

— Nos propositions sont implémentées [3] avec Protégé en OWL 2 DL, qui correspond à la logique de description $\mathcal{SROIQ}^{(\mathcal{D})}$ où à la fois classes et individus sont identifiés par des URI. Nous utilisons le raisonneur HermiT pour valider la satisfaisabilité de l'ontologie résultante et sa consistence. Nous fournissons également un ensemble d'instances qui représentent des cas concrets d'interaction HTTP, afin de vérifier les requêtes SPARQL qui permettent de répondre aux *Questions de Compétence* (notées **QC**) qui définissent nos besoins en matière de représentation d'interaction HTTP.

Dans la section 2 nous introduisons le protocole HTTP en même temps que sa représentation dans HTTPinRDF. Nous précisons les contours de nos propositions en décrivant les limites de ce vocabulaire par rapport à nos objectifs en section 3. Nous présentons nos propositions en section 4 et leur évaluation en section 5, puis nous les situons dans l'état de l'art concernant la description d'interactions HTTP en section 6, avant de conclure en section 7.

2 Notions préliminaires

HTTP est un standard géré par l'*Internet Engineering Task Force* (IETF). Partant des premières proposition de Tim Berners-Lee, un moyen d'indiquer les formats des données transmises y a rapidement été introduit (en-têtes MIME), puis ont été ajoutées des possibilités de préciser de nombreuses caractéristiques liées à l'efficacité des communications (connexions

2. `https://www.w3.org/TR/HTTP-in-RDF10/`

3. `https://labs.nereide.fr/mthl/http`

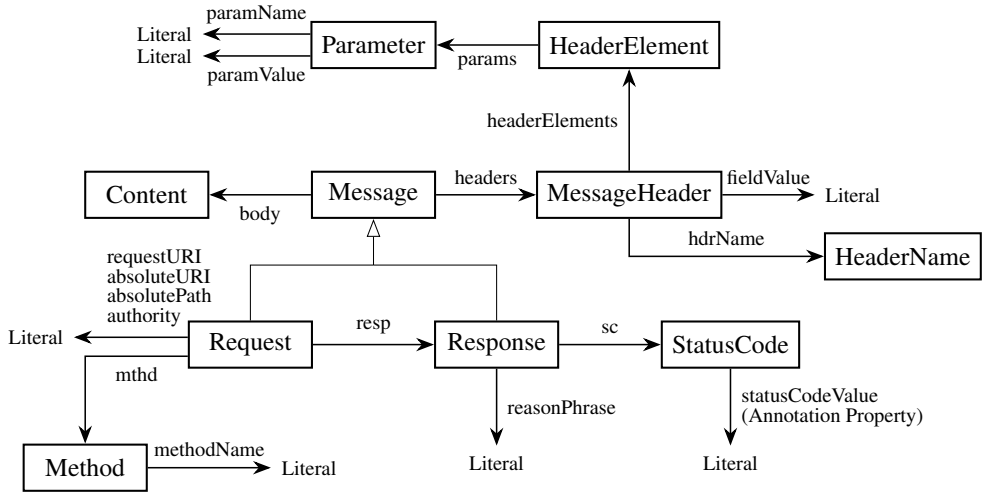


FIGURE *1 – Principaux éléments de HTTPinRDF.*

persistantes, mécanismes de cache) et à leur sécurité (HTTPS, chiffrement). Tout ceci résulte en un pouvoir d'exprimer des informations riches et variées sur les interactions client-serveur. La dernière spécification de HTTP 1.1, produite en 2014, est déclinée en 8 documents appelés RFC, dont les principaux sont la RFC 7230 où sont définis la syntaxe des messages et les aspects du protocole qui concernent le routage, et la RFC 7231 où sont décrits la sémantique des message et leurs contenus. Malgré toutes les précisions apportées dans ces divers documents et les dernières extensions décrites dans ceux associés à HTTP 2 [4], ce standard sur lequel reposent la grande majorité des applications actuelles est volontairement maintenu ouvert pour laisser de la place à d'autres innovations.

La connaissance des RFC de l'IETF est nécessaire pour tout développeur, aussi nous pensons que des outils intelligents devraient exister pour guider les développements qui doivent les respecter. Pour que de tels programmes puissent exploiter non seulement les règles syntaxiques mais aussi la sémantique des interactions HTTP il faut qu'ils disposent d'une ontologie de ces interactions. Un vocabulaire RDF dédié au protocole HTTP (Koch *et al.*, 2017a) a été développé il y a quelques années, dans le cadre d'un outil d'évaluation de l'accessibilité des applications Web. Les interactions de clients Web avec l'application Web testée doivent être enregistrées pour permettre l'évaluation de son accessibilité. Ce vocabulaire RDF fournit la structure de tels enregistrements et pour cela il décrit essentiellement les en-têtes des messages HTTP échangés. Il est utilisé dans EARL, un autre format élaboré dans le même cadre et dédié à l'expression des résultats des évaluations. HTTPinRDF répond aux besoins de ce projet sans pour autant prétendre représenter toute interaction possible via HTTP : il a donc été laissé dans l'état ouvert de *W3C Working Group Note* en 2017.

Cette proposition peut être considérée comme une ontologie d'application (Guarino, 1998), dont le but est de représenter des spécificités propres aux interactions HTTP sur lesquelles reposent les API Web, tandis qu'une ontologie de domaine à laquelle la relier va permettre par exemple de décrire plus généralement un problème, un algorithme ou une fonction, qu'ils soient implémentés par une API Web ou autrement. La figure 1 donne une représentation visuelle des principales classes et propriétés de HTTPinRDF (Koch *et al.*, 2017a). Ce vocabulaire comprend 14 classes et 25 propriétés dans l'espace de noms `http`, plus 11 classes et de nombreuses propriétés (dont celles du vocabulaire Dublin Core DCterm) définies dans quatre autres espaces de noms dédiés respectivement à la représentation des contenus, des en-têtes, des méthodes des message qui sont des requêtes et des codes de statut des messages qui sont des réponses. Ces différents espaces de noms sont présentés dans la table 1.

Le protocole HTTP spécifie une interaction sous la forme d'un échange de messages re-

4. `https://tools.ietf.org/html/rfc7540`

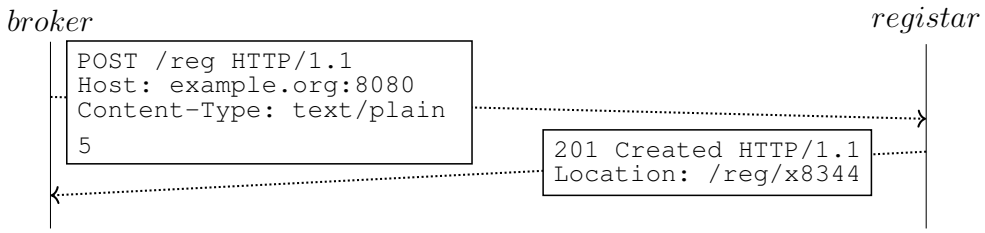


FIGURE *2 – Interaction entre un client "broker" et un serveur "registar".*

TABLE *1 – Préfixes utilisés en Figure 3.*

Prefix	**Namespace**
http :	`http://www.w3.org/2011/http#`
mthd :	`https://www.w3.org/2011/http-methods#`
hdr :	`http://www.w3.org/2011/http-headers#`
sc :	`http://www.w3.org/2011/http-statutCodes#`
cnt :	`http://www.w3.org/2011/content#`
dct :	`http://purl.org/dc/terms/`

quête/réponse : un client envoie un message qui est une requête à un serveur et ce dernier retourne au client un message qui est une réponse. Du point de vue architectural les choses sont un peu plus complexes, il y a des intermédiaires (proxy/passerelles) par lesquels passent les messages, qui peuvent les rediriger, les garder en cache, etc. Mais du point de vue de l'agent client devant utiliser le service offert par le serveur, ces détails n'ont pas d'impact sur le modèle qui lui est utile, celui de l'échange de messages requête/réponse. Comme l'illustre la figure 1, HTTPinRDF représente une interaction HTTP par une relation `resp` depuis un message requête vers un message réponse. Que ce soit une requête ou une réponse, un message a un corps et des en-têtes, lequels forment un ensemble d'associations (nom, valeur). Un message requête est caractérisé en plus par un URI et une méthode, tandis qu'à un message réponse est associé un code de statut. Dans HTTPinRDF, l'URI de la requête est représenté avec un ensemble de propriétés qui toutes ont pour objet un littéral. La méthode de la requête est représentée par une classe. Le code de statut de la réponse est représenté par un littéral et une classe. Pour montrer comment s'utilise ce vocabulaire, nous donnons en figure 2 un exemple d'interaction HTTP. Un agent client nommé *broker* envoie à un serveur nommé *registar* une requête pour que lui soient attribués cinq identifiants. La requête contient le nombre d'identifiants voulus et la réponse comprend le code de statut 201 qui ici dénote la création d'une nouvelle ressource correspondant à la collection d'identifiants ainsi enregistrés. Nous montrons en figure 3 la représentation en RDF de cette interaction (en Turtle), respectant HTTPinRDF. Les définitions des préfixes sont dans la table 1. Le graphe RDF consiste en une paire d'instances des classes `http:Request` et `http:Response`, reliées par la propriété `http:resp` property. Dans cet exemple la propriété utilisée pour représenter l'URI de la requête est `http:absolutePath`, sous-propriété de `http:requestURI`. Pour l'en-tête nous utilisons les propriétés `hdrName` / `fieldValue` parce que ses éléments sont des champs d'en-tête prédéfinis par l'IETF, mais le vocabulaire permet aussi de décrire n'importe quelle paire (nom, valeur) en utilisant la classe `Parameter`.

Un point important de HTTPinRDF est qu'il réifie les méthodes, en-têtes et codes de statut et créer des instances pour tous ceux qui sont définis précisément dans les RFC, tout en les représentant aussi par des littéraux (chaînes de caractères), solution utilisable pour les en-têtes et codes de statut non standards (propres à une application). C'est un moyen de respecter le caractère d'extensibilité du protocole HTTP. Dans l'exemple en Figure 2, nous n'utilisons que des méthodes, en-têtes, et code de statut standards, c'est pourquoi ces éléments sont identifiés par un URI et non un littéral.

```
:req a http:Request ;
   http:mthd mthd:POST ;
   http:absolutePath "/reg" ;
   http:headers [
       http:hdrName hdr:Host ;
       http:fieldValue "example.org:8080"
       ] , [
       http:hdrName hdr:ContentType ;
       http:fieldValue "text/plain"
       ];
   http:resp :resp ;
   http:body [
       a cnt:ContentAsBase64 ;
       dct:isFormatOf [
           a cnt:ContentAsText ;
           cnt:chars "5"
           ]
       ] .

:resp a http:Response ;
   http:sc sc:Created ;
   http:headers [
       http:hdrName hdr:Location ;
       http:fieldValue "/reg/x8344"
       ] .
```

FIGURE *3 – Représentation en Turtle selon HTTPinRDF de l'exemple en Figure 2.*

Une interaction, instance de la propriété `http:resp`, est caractérisée par le code de statut de la réponse. Avec HTTPinRDF il faut identifier la classe associée au numéro de ce code de statut. Un numéro de code de statut doit avoir trois chiffres et sa classe est déterminée par son premier chiffre de la façon suivante :

Classe	Informational	Successful	Redirection	Client Error	Server Error
Codes de statut	$[\![100, 199]\!]$	$[\![200, 299]\!]$	$[\![300, 399]\!]$	$[\![400, 499]\!]$	$[\![500, 599]\!]$

Ces classes sont définies dans l'espace de noms associé au préfixe `sc`. Chaque code de statut est une instance de la classe correspondante. Dans notre exemple, le code de statut est 201, il est représenté par l'individu `sc:Created`, de la classe `sc:Successful`.

Il est important d'observer enfin comment est représenté le corps du message, qui contient le contenu communiqué. Le protocole HTTP permet l'usage de multiple formats pour une même ressource. Un format est identifié par un type de media (Media-Type) dans l'élément d'en-tête `Content-Type`. Dans notre exemple la requête contient le littéral « 5 » avec le Media-Type `text/plain`. La classe `cnt:Content` est un moyen d'associer plusieurs représentations à un même contenu. La propriété `http:body` est toujours présente avec pour valeur `cnt:ContentAsBase64` mais peut aussi être utilisée avec la propriété `dct:hasFormat` pour un contenu `cnt:ContentAsText`.

3 Présentation du problème

D'autres travaux se sont appuyés sur HTTPinRDF, par exemple dans (Verborgh *et al.*, 2017) les auteurs l'utilisent pour définir RESTdesc, un cadre de spécification d'API Web hypermédia (et de composition automatique). « API Web hypermédia » fait ici référence aux API Web qui suivent les principes du style architectural *Representational State Transfer*

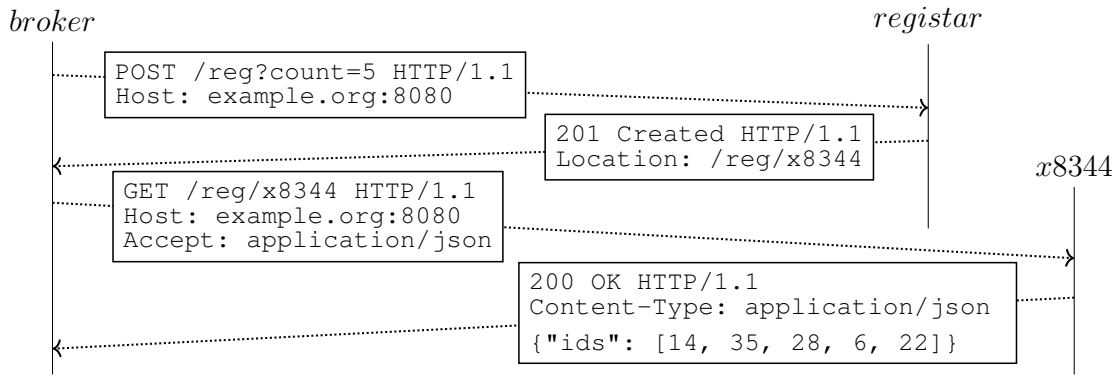


FIGURE *4 – Conversation hypermédia entre un client "broker" et un serveur "registar"*

(REST) [5] (Fielding & Taylor, 2002), et plus particulièrement le principe appelé HATEOAS, qui consiste à utiliser les liens hypermédia pour le contrôle de l'état de l'application client, en d'autres termes pour lui indiquer les continuations possibles des traitements. Nous souhaitons concevoir une vérification automatique de la conformité d'une telle API aux besoins d'une application cliente ou, inversement, de la conformité de l'utilisation de l'API Web par une application cliente. Pour cela nous avons besoin de représenter sous la forme d'un graphe RDF une interaction HTTP dans laquelle le client est amené à utiliser dynamiquement les liens hypermédia fournis par le serveur pour découvrir les actions et ressources possibles pour poursuivre. Or HTTPinRDF ne permet pas de représenter cela.

Pour s'en convaincre, prenons l'exemple de la figure 4, plus proche d'une application réelle que celui de l'exemple de la figure 2. Ici nous avons une conversation composée de deux interactions HTTP, c'est-à-dire deux paires requête/réponse. La première est semblable à l'exemple en Figure 2 à ceci près que le nombre voulu d'identifiants est passé dans une partie de l'URI de la première requête (POST), avec le paramètre de requête `count`. C'est une façon de faire très courante car compatible avec le traitement des formulaires par les navigateurs. La deuxième interaction consiste à interroger le lien fourni par le serveur via l'élément `Location` de l'en-tête de la réponse de la première interaction. Le déréférencement de ce lien fournit une réponse dont le contenu du corps est en format JSON, avec une structure plus complexe que du simple texte (ce à quoi se limitait l'exemple en Figure 2). Il n'est pas possible de représenter la sémantique de cet exemple avec un graphe RDF limité au vocabulaire HTTPinRDF car il ne permet pas de décrire la partie requête d'un URI, ni d'identifier un URI servant dans une interaction suivante, qu'il soit dans un en-tête ou dans un corps de message.

Un autre point qui nécessite de réviser HTTPinRDF est que ses auteurs font référence à la RFC 2616, qui a été supplantée par d'autres documents et en particulier par la RFC 7231, dédiée à la sémantique de HTTP 1.1, laquelle est préservée pour l'essentiel dans les nouvelles versions du protocole. Les changements que nous présentons dans la section suivante sont destinés à prendre en compte la RFC 7231 pour représenter des enchaînements d'interactions HTTP, où le client utilise les liens hypermédia pour fournir des valeurs de paramètres au serveur (en particulier dans l'URI de ses requêtes) et exploite aussi les liens hypermédia que le serveur lui fournit dans une partie ou une autre de ses réponses. Notre objectif est de pouvoir formuler des règles de validité sur les interactions ainsi représentées. Sans prétendre à l'exhaustivité nous exprimons des éléments de telles règles sous la forme de questions de compétence (QC) afin de pouvoir vérifier l'utilité de nos propositions par rapport à nos objectifs. Le principe des QC est de refléter les besoins fonctionnels qui fondent les engagements ontologiques d'une représentation de connaissances (Noy & Mcguinness, 2001; Blomqvist *et al.*, 2010; Hitzler *et al.*, 2016).

5. Où l'application se définit par les ressources (distantes) qu'elle utilise, tandis qu'avec SOAP elle se définit par les appels à des procédures distantes.

QC 1 (Type de média)
Quel est le type de média associé à un corps de message ?

QC 2 (Résultat d'interaction)
Quel est le numéro du code de statut d'une interaction ?

QC 3 (Valeur d'élément d'en-tête, comme Location)
Quel est l'URI fourni via l'élément `Location` *d'un en-tête de réponse ?*

QC 4 (Résultat de conversation)
Quel est le code de statut associé à la réponse de la dernière interaction d'une conversation donnée ? Par exemple celui de l'exemple en Figure 4 ce serait 200*.*

QC 5 (Négociation de contenu)
Est-ce que le type de média du corps d'une réponse correspond à l'un de ceux déclarés par un élément d'en-tête `Accept` *dans la requête correspondante ?*

QC 6 (Contenu du corps de message quand il est en RDF)
Quelles sont les valeurs d'une propriété RDF donnée p *dans le corps d'un message lui-même en RDF ?*

QC 7 (Paramètres de la partie requête dans un URI)
Quelle est la valeur d'un paramètre donné, par exemple `age`*, passé dans l'URI d'un message requête ?*

4 Contributions à une ontologie des interactions HTTP

Nous utilisons la logique de description $\mathcal{SROIQ}^{(\mathcal{D})}$, associée à OWL2 DL, pour décrire l'ontologie résultant de nos propositions, montrée en Figure 5, qui repose en grande partie sur HTTPinRDF. Nous conservons toutefois dans cet article la terminologie RDF en parlant de classes et propriétés plutôt que de rôles et de concepts. Nous notons $\top$ la classe de toutes les instances et $\mathcal{D}$ la classe des littéraux (qui ne peuvent pas être sujet d'une propriété et ont une interprétation unique). Les classes $\top$ and $\mathcal{D}$ sont disjointes.

4.1 Message

Un message peut avoir une collection d'éléments d'en-tête, ce que nous représentons comme dans HTTPinRDF par la propriété $headers$ mais dont les objets ici sont de la classe $Headers$, et un corps, représenté par la propriété $body$ qui n'apparaît qu'une seule fois et dont l'objet est de la classe $Content$. En-tête et corps sont optionnels.

$$Message \sqsubseteq \forall headers.Headers \sqcap \forall body.Content \qquad \top \sqsubseteq \leq_1 body.\top$$

Les détails des classes $Headers$ et $Content$ sont donnés en section 4.5 et section 4.6 respectivement. Un message peut être une requête ou une réponse mais pas les deux à la fois.

$$Message \equiv Request \sqcup Response \qquad Request \sqcap Response \sqsubseteq \bot$$

La propriété $resp$ relie une requête à une réponse.

$$\exists resp.\top \sqsubseteq Request \qquad \top \sqsubseteq \forall resp.Response$$

Il peut y avoir plusieurs réponses pour une même requête, avec les restrictions décrites en section 4.4.

4.2 Requête

Une requête a une méthode et un URI. Nous définissons la classe $Method$ comme superclasse de toutes les méthodes de requête standards en utilisant des *nominaux*. Cela n'est pas une équivalence car de nouvelles méthodes peuvent être définies.

$$\{\text{GET}, \text{HEAD}, \text{POST}, \text{PUT}, \text{DELETE}, \text{CONNECT}, \text{OPTIONS}, \text{TRACE}, \text{PATCH}\} \sqsubseteq Method$$
$$Request \sqsubseteq Message \sqcap \exists mthd.Method \sqcap \exists uri.URI$$
$$\top \sqsubseteq \leq_1 mthd.Method \qquad \top \sqsubseteq \leq_1 uri.URI$$

Les propriétés $mthd$ et uri sont fonctionnelles. La valeur associée à la propriété uri est une instance de la classe URI, que nous introduisons pour décrire les composants syntaxiques d'un URI tels que définis dans la RFC 3986 (Berners-Lee *et al.*, 2005) dont est tiré l'exemple d'URI complet ci-dessous :

$$\underbrace{\texttt{http}}_{scheme}\texttt{://}\underbrace{\texttt{example.com:8042}}_{authority}\underbrace{\texttt{/over/there}}_{path}\texttt{?}\underbrace{\texttt{name=ferret}}_{query}\texttt{\#}\underbrace{\texttt{nose}}_{fragment}$$

Les syntaxes que peut prendre concrètement l'URI d'une requête sont données dans (Fielding & Reschke, 2014a, Section 5.3). Certaines nécessitent de combiner la valeur de l'élément d'en-tête `Host` et du schéma de protocole (`http` ou `https`) pour obtenir l'URI complet. Les différentes parties de l'URI sont représentées avec les propriétés $scheme$, $authorithy$, $path$, $query$, $fragment$ (cf. exemple ci-dessus) de la classe URI, qui ont des valeurs littérales. Elles servent à décrire toute forme que prend concrètement l'URI. Cette représentation diffère de HTTPinRDF, où différentes propriétés sont définies pour la classe $Request$ (`http:requestURI`, `http:absolutePath` et `http:absoluteURI`) dont les valeurs sont des littéraux. Le littéral qui serait valeur de `http:absoluteURI` avec HTTPinRDF est ici représenté en utilisant ensemble $scheme$, $authorithy$, $path$, éventuellement $query$ et peut-être $fragment$. Nous ajoutons une propriété $idRes$ à la classe URI qui prend comme valeur l'URI représentée, sous forme de chaîne de caractères. Ce littéral est l'URI d'une ressource et l'instance de la classe URI sert à représenter ses différentes formes concrètes possibles. Pour pouvoir répondre à **QC 7** nous associons aussi aux instances de la classe URI une propriété `queryParams` afin de détailler la partie $query$, comme précisé dans la section suivante.

4.3 Paramètres de la partie *query* d'une URI

La description d'une requête HTTP nécessite de représenter des paramètres. Les paramètres peuvent être passés de différentes façons, la plus basique étant d'utiliser la partie $query$ de l'URI de la requête, partie qui se situe entre `?` et `#`. Pour une application Web cette partie se conforme au type de media `application/x-www-form-urlencoded` [6] qui permet de passer des paires (clé,valeur) comme arguments, que nous dénotons (k,v) dans l'exemple suivant.

$$\underbrace{\overbrace{\texttt{age}}^{k}\texttt{=}\overbrace{\texttt{54}}^{v}\texttt{\&}\overbrace{\texttt{id}}^{k}\texttt{=}\overbrace{\texttt{XPZIJ4}}^{v}}_{query}$$

Avec la propriété $query$ nous pouvons accéder à la valeur littérale, mais nous voulons aussi accéder aux paires (clé,valeur), pour cela nous définissons la propriété $queryParams$ et la classe $Parameter$ dont les instances ont un nom (et un seul) et une valeur (et une seule).

$$URI \sqsubseteq \forall queryParams.Parameter \qquad Parameter \equiv \exists name.\mathcal{D} \sqcap \exists value.\mathcal{D}$$

6. `https://url.spec.whatwg.org/#concept-urlencoded`

4.4 Réponse

Une réponse a un statut, instance de la classe $Status$ et accessible par la propriété sc, elle-même étant caractérisée par un nombre à trois chiffres via la propriété $code$.

$$Response \sqsubseteq Message \sqcap \exists sc.Status \qquad Status \sqsubseteq \exists code.[\![000, 999]\!]$$
$$\top \sqsubseteq \leq_1 sc.\top \qquad \mathcal{D} \sqsubseteq \leq_1 code.\mathcal{D}$$

Les propriétés sc et $code$ sont fonctionnelles. Nous utilisons la notation compacte $[\![000, 999]\!]$ pour dénoter le type des entiers positifs inférieurs ou égaux à 999 qui s'écrirait ainsi en OWL 2 turtle :

```
:threeDigit a rdfs:Datatype ;
    owl:equivalentClass [
        a rdfs:Datatype ;
        owl:onDatatype xsd:nonNegativeInteger ;
        owl:withRestrictions ([ xsd:maxInclusive 999 ])] .
```

Il existe une bijection entre une instance de $Status$ et son code. Par exemple nous pouvons définir $Created$ comme étant l'unique instance de $Status$ ayant le code 201 (qui indique *une nouvelle ressource a été créée avec succès*).

$$\{Created\} \equiv Status \sqcap \exists code.201$$

Les codes de statut sont des éléments syntaxiques qui dénotent le sens du statut de réponse. Même si chaque instance de $Statut$ a un sens spécifique, elles peuvent être regroupées en sous-classes de $Status$ de la façon suivante.

$$Successful \equiv Status \sqcap \exists code.[\![200, 299]\!] \quad ClientError \equiv Status \sqcap \exists code.[\![400, 499]\!]$$
$$Redirection \equiv Status \sqcap \exists code.[\![300, 399]\!] \quad ServerError \equiv Status \sqcap \exists code.[\![500, 599]\!]$$
$$Informational \equiv Status \sqcap \exists code.[\![100, 199]\!]$$

Les instances de ces classes qualifient des *réponses finales* à l'exception de celles de la classe $Informational$ qui définissent des *réponses intermédiaires* qui seront suivies d'une *réponse finale* (Fielding & Reschke, 2014b, Section 6.2). Ainsi plusieurs réponses peuvent être associées à une requête mais une seule peut être une *réponse finale*.

$$\exists sc.Informational \sqsubseteq Interim \qquad Final \equiv Response \sqcap \neg Interim$$

Definition 1 (Interaction)
Une interaction *est une instance* $resp(q, r)$ *de la propriété* $resp$ *telle que* q *est une instance de* $Request$ *et* r *est une instance de* $Final$.

Dans la figure 5 nous ne représentons pas les sous-classes de $Status$, comme nous ne représentons pas non plus celles de $StatusCode$ en Figure 1. Nous ne conservons pas la propriété $reasonPhrase$ car cette précision syntaxique ne porte pas de sens en elle-même.

4.5 Eléments d'en-tête

Un message peut avoir un ensemble d'éléments d'en-tête. Nous représentons ces éléments par une classe $Header$, ses instances étant caractérisées par un unique nom (propriété $hdrName$) et une unique valeur (propriété $fieldValue$), tous deux littéraux.

$$Header \sqsubseteq \exists hdrName.\mathcal{D} \sqcap \exists fieldValue.\mathcal{D}$$

Cette représentation générique utilisant un littéral pour toute valeur peut être précisée pour les éléments d'en-tête prédéfinis comme `Location` ou `Content-Type`, en définissant une

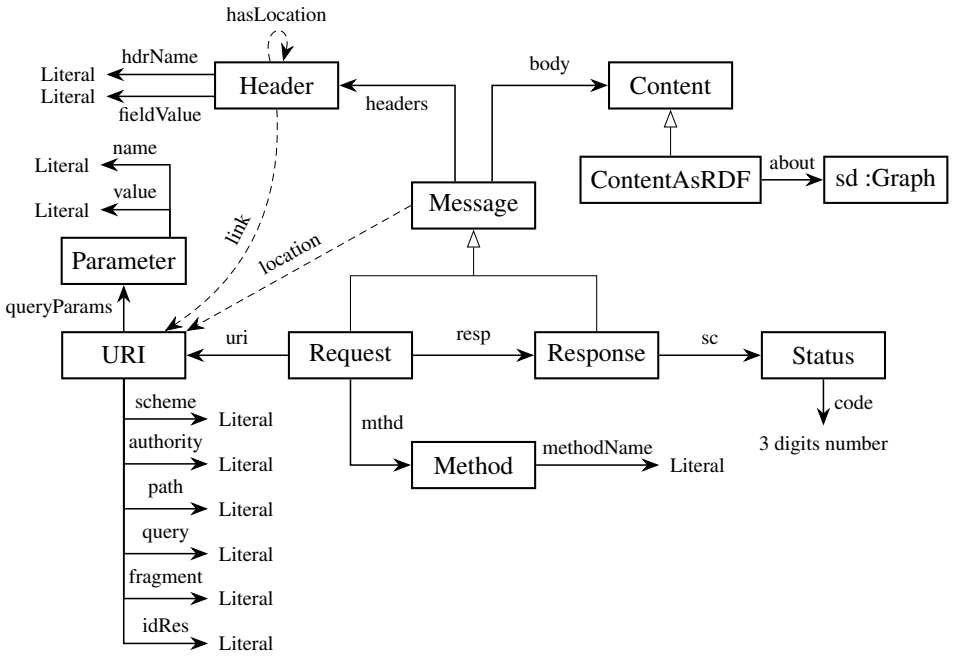


FIGURE *5 – Principaux éléments de notre ontologie des interactions HTTP.*

chaîne de propriétés qui permette de de faire référence à leur valeur directement depuis le message, avec une propriété ayant leur nom. Cela peut grandement simplifier les réponses pour **QC 1** et **QC 3**. Par exemple pour un élément d'en-tête `Location` (c'est-à-dire dont la propriété $hdrName$ a la valeur `"Location"`) nous définissons d'abord une propriété $link$ dont la valeur correspond à une instance de la classe URI représentant la valeur littérale de la propriété $fieldValue$, ce qui rend possible d'isoler des parties de cette valeur littérale si besoin. Puis nous disons qu'une instance de $Header$ ayant pour valeur de $hdrName$ `"Location"` a une propriété $link$.

$$Header \sqcap \exists hdrName.\{\texttt{"Location"}\} \sqsubseteq \exists link.\textbf{\textit{URI}}$$

Nous pouvons ensuite définir une propriété $location$. Pour cela nous introduisons une propriété réflexive $hasLocation$ uniquement pour les instances de la classe $Header$ qui ont `"Location"` pour valeur de $hdrName$.

$$Header \sqcap \exists hdrName.\{\texttt{"Location"}\} \equiv \exists hasLocation.Self$$
$$headers \circ hasLocation \circ link \sqsubseteq location$$

Ce mécanisme est illustré par les flèches en pointillés dans la figure 5 pour l'exemple de `Location`. Nous proposons de faire de même pour tous les éléments d'en-tête prédéfinis, donc aussi pour `Content-Type`. L'approche de HTTPinRDF est différente, la vue littérale des éléments d'en-têtes est raffinée avec la propriété `http:headerElements` dont les objets peuvent être décomposés avec `http:elementName`, `http:elementValue` ou avec `http:params` de façon générique. Nous conservons la généricité tout en proposant des raccourcis pour les en-têtes prédéfinis comme `Location`, ce qui permet une écriture plus succincte des requêtes mais augmente la taille de la TBox.

4.6 Contenu du corps

Comme définie dans la section 4.1, la propriété $body$ donne accès aux données du message. Son objet est instance de la classe $Content$. Dans HTTPinRDF la représentation du

contenu du corps du message est déléguée à un vocabulaire externe nommé Content (Koch *et al.*, 2017b) qui prend en compte le fait qu'une ressource peut être associée à plusieurs représentations dans différents formats. Cependant il restreint le co-domaine de $body$ au format $ContentAsBase64$ ce qui doit pouvoir être précisé puisqu'en HTTP le véritable format est indiqué par les éléments d'en-tête `Content-Type` et `Content-Encoding`. Le premier est obligatoire pour tout message ayant une propriété $body$ et sa valeur est un type de media $\mathcal{M}$. Ces éléments d'en-tête permettent au récepteur d'un message de savoir comment interpréter le contenu du corps. Par exemple lorsque `Content-Type` est `text/plain` et qu'il n'y a pas de `Content-Encoding` la valeur de $body$ devrait être directement une instance de la classe $ContentAsText$.

$$\exists body.Content \sqsubseteq \exists \texttt{content-type}.\mathcal{M}$$

Pour **QC 2** nous aimerions désigner les liens fournis dans un message. Certains sont dans les valeurs d'éléments d'en-tête comme `Location` mais d'autres peuvent être dans le corps lorsque le `Content-Type` correspond à un format hypermédia. RDF étant un moyen naturel de représenter ces liens, nous proposons que le contenu du corps soit disponible en RDF, ce qui, moyennant la précaution de correctement encapsuler ce graphe de contenu, permet de représenter à la fois le message et son contenu en RDF. Pour cela nous introduisons la classe $ContentAsRDF$ comme sous-classe de $Content$. La propriété $about$ sert de lien entre le graphe du message et celui du contenu, encapsulé dans un graphe nommé tel que défini dans la spécification de SPARQL 1.1.

$$ContentAsRDF \sqsubseteq Content \sqcap \exists about.\top \quad \top \sqsubseteq \forall about.Graph \quad \top \sqsubseteq \leq_1 about.\top$$

Cette propriété est fonctionnelle. Par exemple la représentation RDF d'un corps de message décrivant la ressource `:foo` peut se faire avec le langage TriG comme suit :

```
:B {
    :foo :ids (1 2 3) ;
         :date "2003-02-10"^^xsd:date .
}
```

Cette représentation peut ensuite être associée au contenu du corps de message présent dans le graphe par défaut de la manière suivante :

```
:m a http:Message ;
   http:body :b .
:b a cnt:ContentAsRDF ;
   cnt:about :B .
:B a sd:Graph .
```

Pour s'assurer d'avoir des contenus qui soient dans un format RDF (comme RDF/XML, JSON-LD ou Turtle) il est possible d'adopter la notion de *présentation RDF* (Lefrançois, 2018), qui fournit un moyen de transformer un format non-RDF en un graphe RDF et à l'inverse un moyen de ramener un graphe RDF dans le format non-RDF de départ.

5 Evaluation

La question de l'évaluation d'ontologie a reçu beaucoup d'attention et peut être considérée selon différentes catégories : logique, structurelle et fonctionnelle (Tartir *et al.*, 2010). La catégorie logique regroupe les dimensions de qualité qui peuvent être évaluées avec un raisonneur, par exemple la satisfaisabilité. Notre implémentation en OWL 2 DL nous permet de vérifier les inférences possibles avec le raisonneur Hermit, pour en particulier détecter les incohérences. Nous avons aussi vérifié l'absence d'incohérence par la *provocation d'erreur* au moment de l'instanciation de l'ontologie avec un ensemble d'individus représentatif. Une représentation visuelle du graphe RDF des individus correspondant à l'exemple en Figure 4 est fourni en Figure 6.

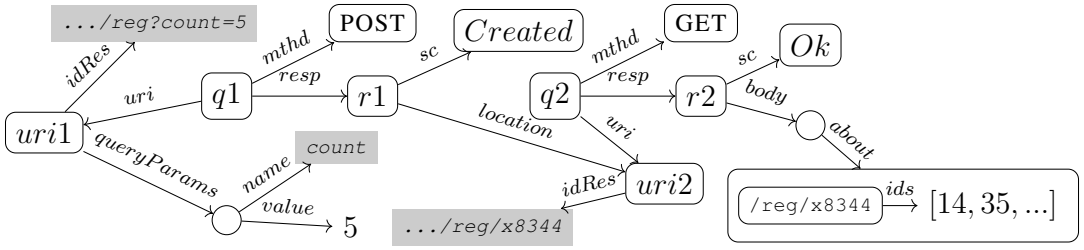


FIGURE *6 – Graphe RDF de l'exemple en Figure 4*

La catégorie *structurelle* est composée de dimensions non contextuelles qui peuvent être mesurées de manière quantitative. Par exemple en utilisant OntoMetrics [7] mais aussi la popularité ou le degré de couplage avec d'autres resources du Web des données. Dans notre cas, des dimensions importantes appartenant à cette catégorie sont les suivantes :

flexibilité Est-ce que l'ontologie est facilement adapable à une variété d'usages ? Nous répondons à cette question avec notre proposition pour les en-têtes, le corps du message et les paramètres de requêtes, qui améliorent la réutilisabilité potentielle de l'ontologie par rapport à HTTPinRDF.

transparence Est-ce que l'ontologie est facilement analysable ? Nous répondons à cette question avec notre formalisation en logique de description, implémentée en OWL2 DL, qui permet de l'explorer et la tester avec Protégé et les raisonneurs qu'il intègre.

Ergonomie cognitive Est-ce que l'ontologie est facilement compréhensible et exploitable par l'utilisateur ? Nous répondons à ce point en écrivant cet article pour expliquer nos choix, en documentant cette ontologie, et en la mettant à disposition en ligne.

Conformité à l'expertise Est-ce que l'ontologie est conforme avec les connaissances qu'elle représente ? Dans notre cas nous représentons mieux la sémantique des RFCs en introduisant des classes et propriétés en substitut de littéraux, par exemple pour les URIs. Cette dimension est aussi liée aux propriétés fonctionnelles, qui dans le cadre d'une ontologie d'application comme celle présentée ici sont d'une importance cruciale.

La catégorie *fonctionnelle* regroupe les dimensions de qualité liées aux usages et fonctions contextualisées. Nous addressons cela en écrivant des requêtes SPARQL à partir de notre ontologie dans le but de répondre aux *questions de compétence* exprimées en section 3.

QC 1 Quel est le type de média associé à un corps de message ?

```
SELECT ?m ?mt
WHERE {
  ?m a http:Message .
  ?m http:content-type ?mt .
}
```

QC 2 Quel est le numéro du code de statut d'une interaction ?

```
SELECT ?status
WHERE {
  ?q http:resp ?r .
  ?r http:sc/http:code ?status .
}
```

QC 3 Quel est l'URI fourni via l'élément `Location` d'un en-tête de réponse ?

```
SELECT ?next
WHERE {
  ?q0 http:resp/http:location ?next .
}
```

7. https://ontometrics.informatik.uni-rostock.de/ontologymetrics

QC 4 Quel est le code de statut associé à la réponse de la dernière interaction de la conversation en Figure 4 ?

```
SELECT ?status
WHERE {
  ?q0 http:resp/http:location/http:idRes ?next .
  ?q1 http:uri/http:idRes ?next .
  ?q1 http:resp/http:sc ?status .
}
```

QC 5 Est-ce que le type de média du corps d'une réponse correspond à l'un de ceux déclarés par un élément d'en-tête `Accept` dans la requête correspondante ?

```
ASK {
  ?q http:resp ?r .
  ?q http:accept/http:media-type ?mt1 .
  ?r http:content-type ?mt2 .
  FILTER (CONTAINS(STR(?mt1), STR(?mt2))
          || CONTAINS(STR(?mt2), STR(?mt1)))
}
```

QC 6 Quelles sont les valeurs de la propriété RDF `ex:ids` dans le corps de la deuxième réponse de la conversation en Figure 4 ?

```
SELECT ?ids
WHERE {
  ?m http:body/cnt:about ?G .
  GRAPH ?G { ?x ex:ids/rdf:rest*/rdf:first ?ids } .
}
```

Ici la propriété `ex:ids` associe la racine de la représentation du contenu avec une liste d'identifiants de type `rdf:List`.

QC 7 Quelle est la valeur d'un paramètre donné, par exemple `age`, passé dans l'URI d'un message requête ?

```
SELECT ?age
WHERE {
  ?q http:uri/http:queryParams ?p .
  ?p http:name "age" .
  ?p http:value ?age
}
```

6 Positionnement dans l'état de l'art

Notre objectif est de guider le travail de développeurs d'applications web : pour leur expliquer leurs erreurs un programme doit pouvoir traiter les différents éléments de HTTP et leur sens, il doit donc s'appuyer sur une ontologie les représentant. Cette ontologie peut aussi être un cadre pour la définition de spécification d'API Web, sous la forme de règles de validation. Dans cette section nous considérons donc le problème de spécification d'API Web, qui motive notre travail, sous différents points de vue, à travers les utilisations passées et actuelles des propositions existantes. D'un point de vue historique, aux débuts de la programmation d'applications Web on parlait de services Web, définis, décrits et publiés avec SOAP, WSDL et UDDI (Weerawarana *et al.*, 2005). De nombreux travaux visaient aussi la description de ces services Web à un niveau sémantique, avec OWL-S (Martin & et., 2005) ou encore SAWSDL (Kopecký *et al.*, 2007). Pour les interactions client-serveur, ce type de services Web repose sur une forme de communication dite Remote Procedure Call (RPC), popularisée par la programmation orientée objet, qui d'une part ignore le principe des liens hypermédia au coeur de l'architecture logicielle du Web et d'autre part consiste à définir et exposer des ensembles arbitraires d'opérations au lieu des méthodes définies dans HTTP. Ils ont été beaucoup étudiés et développés et de grandes entreprises les utilisent encore couramment. Pour les développeurs les spécifications en WSDL fournissent un excellent support

pour savoir comment utiliser un tel service, mais à notre connaissance aucune tentative visant à automatiser leur utilisation ne s'est avérée probante. Pourtant WSDL permet de fournir une description lisible par une machine, des moyens de faire appel au service, avec quels paramètres, et quels résultats sont retournés (Weerawarana *et al.*, 2005). Les services y sont décrits comme des collections de points d'accès réseau (ports), qui associent un URL avec une liaison dans laquelle sont décrits le protocole concret et les formats de message, à savoir les opérations supportées et les formes de données échangées (schémas). SAWSDL (Kopecký *et al.*, 2007) représente un ensemble d'attributs supplémentaires pour ajouter des annotations sémantiques aux différentes parties d'un document WSDL. Cela permet aux concepteurs de service Web de relier des déclarations WSDL et des schémas XML à des ontologies.

La suite de l'histoire des applications Web est dominée par les services REST ou les services RESTful (Richardson *et al.*, 2013) dits encore API Web hypermédia (Verborgh *et al.*, 2017), qui ont pris leur essor en même temps que l'émergence des Linked Data et la popularité croissante des formats JSON et JSON-LD liés au langage de développement web Javascript. Avant de considérer les propositions de spécification de tels services, il est important de remarquer comme est universel le besoin d'un moyen de fournir de l'information sur les fonctionnalités d'un service, ainsi que sur la forme et le sens des données échangées. C'est la vocation de SAWSDL pour les services qui suivent SOAP, ou de OWL-S (Martin & et., 2005) qui est proposé pour tout type de services. Ce dernier est d'ailleurs comparable à une autre proposition plus récente d'une ontologie pour la description de fonctions, Function Ontology (De Meester *et al.*, 2016), elle aussi destinée à compléter par un niveau plus abstrait les spécifications de services Web, lesquelles sont très couplées avec la technologie [8]. Il s'agit de permettre de déclarer et décrire n'importe quel problème, fonction ou algorithme, que cela soit implanté avec un service Web ou autrement. OWL-S et Function Ontology ont en commun de répondre aux besoins de descriptions pour la découverte, l'invocation et la composition automatique de services. OWL-S est une proposition ancienne (W3C Member Submission 2004) alors que Function Ontology est à un stade de première ébauche non officielle, conçue pour les applications du Web sémantique. Les deux propositions comprennent un mécanisme pour relier le niveau plus élevé des descriptions fonctionnelles au niveau plus concret des descriptions d'API Web, ces dernières pouvant être spécifiées en utilisant WSDL, ou l'ontologie HTTP, ou encore Hydra (Lanthaler & Gütl, 2013).

Hydra est un formalisme pour décrire et utiliser les API Web hypermédia, qui comme on l'a vu respectent le style architectural REST et sont plus simples à déployer et à utiliser que les services définis avec SOAP (Upadhyaya *et al.*, 2011; Richardson *et al.*, 2013). Ce sont ces API Web qui sont maintenant les plus développées et utilisées. OpenAPI (Initiative, 2018) est une initiative très suivie qui propose d'associer aux API Web une documentation lisible par un programme, qui peut être compilée en une page web, devenant ainsi découvrable avec un navigateur et un moteur de recherche standard. C'est un niveau de description seulement syntaxique qui permet pour ces services ce que WSDL offre pour les services SOAP, ou encore ce que permet WADL pour des services REST (JSON replaçant XML). De son côté Hydra offre un moyen d'exprimer une description sémantique avec l'objectif de simplifier encore plus le développement de services RESTful, tout en exploitant le potentiel du Linked Data (Lanthaler & Gütl, 2013), aspect qui n'est pas considéré par l'initiative OpenAPI. Hydra se présente comme un vocabulaire RDFS qui permet d'un côté de décrire les API Web et de l'autre d'augmenter les Linked Data avec des contrôles hypermédia (il permet de spécifier quels URI dans un graphe RDF sont prévus pour être déréférencés). Le W3C accueille par ailleurs le développement du standard Linked Data Platform (LDP) (Speicher *et al.*, 2015) qui a pour but de permettre aux fournisseurs de données liées d'exposer leurs jeux de données à la manière RESTful et comprend un modèle pour interagir (lire et écrire) avec ces données. Cela rejoint l'intention de Hydra de permettre de décrire des API Web RESTful qui consomment et produisent des données liées. Hydra est la proposition la plus proche de l'ontologie HTTP que nous proposons, elle en diffère toutefois par l'usage de classes et de propriétés pour représenter les composants de HTTP et des URI qui sont plus abstraites, tout en imposant de manipuler des notions propres aux données liées qui ne sont pas forcément familières pour

8. `https://w3id.org/function/spec`

les développeurs d'applications Web actuelles. Hydra est utilisé par plusieurs applications, par exemple l'une d'elles consiste à automatiser la découverte et l'utilisation de services Web grâce à des micro-services SPARQL (Michel *et al.*, 2019), qui permettent aux programmes d'interroger une API Web comme Flickr en utilisant SPARQL.

RESTdesc (Verborgh *et al.*, 2012, 2017) est aussi un format de description sémantique d'API Web hypermédia, qui utilise l'ontologie HTTP plutôt que Hydra, et Notation3 Logic (Berners-Lee *et al.*, 2007) plutôt que RDFS ou OWL. Il est conçu pour la découverte et la composition automatique de tels services, par des agents intelligents. Pour ses auteurs en effet, l'utilisation des liens hypermédia et des données liées doit permettre de développer des agents intelligents qui naviguent à travers les API et choisissent la poursuite de leurs actions au cours de cette navigation, comme le font les humains avec les sites web. Cette vision est évidemment enthousiasmante mais elle reste éloignée des pratiques actuelles dans les systèmes d'information d'entreprise : la composition des services est encore programmée à la main, à partir d'un ensemble précis de services, et même ainsi, avec tout le soin mis dans ces développements, de nombreux problèmes se posent lorsque les services évoluent. Notre objectif est ainsi plus modeste que celui de RESTdesc, au moins dans un premier temps. Pour assister le travail quotidien des producteurs et consommateurs d'API Web nous cherchons à valider une API Web concrète par rapport à une spécification en expliquant les erreurs éventuelles. C'est pourquoi nous nous attachons à la représentation formelle de la connaissance que partagent ces développeurs, concernant les interactions HTTP.

7 Conclusion

Nous avons présenté une descritpion ontologique des interactions HTTP en nous basant sur une étude en profondeur de la RFC 7231. Notre point de départ était HTTPinRDF proposé par le W3C qui est utilisé dans différent contextes. Comme il s'appuie sur une interprétation limitée des RFC (plus proche de la syntaxe HTTP que de sa sémantique), il ne nous permet pas d'exprimer les requêtes qui correspondent aux questions de compétences représentant nos besoins. Nous avons mené une analyse formelle en utilisant à la fois la logique de description et OWL 2 DL pour introduire notre proposition de son évolution vers une ontologie HTTP qui répond à nos besoins comme montré par notre effort d'évaluation. En ce qui concerne les travaux liés, HTTPinRDF est la seule proposition qui est directement liée à notre objectif de décrire sémantiquement le protocole HTTP, même si l'objectif plus général de décrire des services Web est celui de nombreux travaux. Les réponses actuelles à cet objectif plus général ne sont pas directement utilisables pour valider une API Web concrète par rapport à la spécification d'HTTP, en expliquant aux développeurs leur potentielle erreurs.

La spécification RFC 7231 étant conséquente notre effort de formalisation est incomplet et de nombreux aspects restent à étudier. De plus nous avons observé dans le cas des paramètres de requête que les pratiques usuelles, qui sont importantes à prendre en compte du fait de leur large adoption, reposent sur des extensions *ad-hoc* de la spécification d'un URI. Nous sommes conscients des nombreuses limitations de notre approche. Par exemple nous ne sommes pas en mesure de décrire l'évolution des représentations. Nous n'exprimons pas non plus les relations de dépendance temporelle qui existent entre les messages. Une autre limitation vient de *l'hypothèse du monde ouvert* qui signifie qu'il n'est pas possible de vérifier l'absence de certains type d'instance, ce qui est utile dans un contexte de validation. Par exemple nous pourrions vouloir vérifier que les réponses associées à la méthode `HEAD` n'ont pas de corps. Combiner OWL et SHACL pourrait être la solution et un travail à venir est d'exploiter l'ontologie pour instrumenter l'exécution des interactions HTTP soit au niveau du client, soit au niveau du serveur. Les inférences logiques réalisées par le raisonneur OWL 2 permettront de vérifier la conformité de ces interaction avec la spécification du protocole. Par exemple il pourrait vérifier pour chaque message que l'en-tête `Content-Type` est défini de manière appropriée en présence d'un corps. Pour réaliser cela, nous travaillons sur un programme qui convertit des messages HTTP en graphes RDF utilisant notre ontologie.

Références

BERNERS-LEE T., CONNOLLY D., KAGAL L., SCHARF Y. & HENDLER J. A. (2007). N3logic : A logical framework for the world wide web. *Theory Pract. Log. Program.*, **8**, 249–269.

BERNERS-LEE T., FIELDING R. & MASINTER L. (2005). Uniform resource identifier (uri) : Generic syntax.

BLOMQVIST E., PRESUTTI V., DAGA E. & GANGEMI A. (2010). Experimenting with extreme design. In P. CIMIANO & H. S. PINTO, Eds., *Knowledge Engineering and Management by the Masses*, p. 120–134, Berlin, Heidelberg : Springer Berlin Heidelberg.

DE MEESTER B., DIMOU A., VERBORGH R. & MANNENS E. (2016). An Ontology to Semantically Declare and Describe Functions. In *The Semantic Web*, p. 46–49, Cham : Springer International Publishing.

FIELDING R. & RESCHKE J. (2014a). Hypertext transfer protocol (http/1.1) : Message syntax and routing.

FIELDING R. & RESCHKE J. (2014b). Hypertext transfer protocol (http/1.1) : Semantics and content.

FIELDING R. T. & TAYLOR R. N. (2002). Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, **2**(2), 115–150.

GUARINO N. (1998). Formal ontology and information systems. In *Proceedings of Formal Ontology in Information Systems*, p. 3–15 : IOS Press.

P. HITZLER, A. GANGEMI, K. JANOWICZ, A. KRISNADHI & V. PRESUTTI, Eds. (2016). *Ontology Engineering with Ontology Design Patterns - Foundations and Applications*, volume 25. IOS Press.

INITIATIVE O. (2018). The OpenAPI Specification.

KOCH J., VELASCO C. & ACKERMANN P. (2017a). HTTP Vocabulary in RDF 1.0. W3C Working Group.

KOCH J., VELASCO C. & ACKERMANN P. (2017b). Representing content in rdf 1.0.

KOPECKÝ J., VITVAR T., BOURNEZ C. & FARRELL J. (2007). SAWSDL : semantic annotations for WSDL and XML schema. *Internet Computing, IEEE*, **11**, 60–67.

LANTHALER M. & GÜTL C. (2013). Hydra : A Vocabulary for Hypermedia-Driven Web APIs. *LDOW*, **996**.

LEFRANÇOIS M. (2018). RDF presentation and correct content conveyance for legacy services and the web of things. In *Proceedings of the 8th International Conference on the Internet of Things*, p.4̃3 : ACM.

MARTIN D. & ET. A. (2005). Bringing Semantics to Web Services : The OWL-S Approach. In *Semantic Web Services and Web Process Composition*, p. 26–42, Berlin, Heidelberg : Springer Berlin Heidelberg.

MICHEL F., FARON-ZUCKER C., CORBY O. & GANDON F. (2019). Enabling automatic discovery and querying of web APIs at web scale using linked data standards. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019.*, p. 883–892.

NOY N. & MCGUINNESS D. (2001). Ontology development 101 : A guide to creating your first ontology. *Knowledge Systems Laboratory*, **32**.

RICHARDSON L., AMUNDSEN M. & RUBY S. (2013). *RESTful Web APIs*. O'Reilly Media.

SPEICHER S., ARWE J. & MALHOTRA A. (2015). *Linked Data Platform 1.0, W3C Recommendation*. Rapport interne, W3C.

TARTIR S., ARPINAR I. & SHETH A. (2010). *Ontological Evaluation and Validation*, p. 115–130.

UPADHYAYA B., ZOU Y., XIAO H., NG J. & LAU A. (2011). Migration of SOAP-based services to RESTful services. In *2011 13th IEEE International Symposium on Web Systems Evolution (WSE)*, p. 105–114 : IEEE.

VERBORGH R., ARNDT D., HOECKE S. V., ROO J. D., MELS G., STEINER T. & GABARRÓ J. (2017). The pragmatic proof : Hypermedia API composition and execution. *Theory Pract. Log. Program.*, **17**(1), 1–48.

VERBORGH R., STEINER T., VAN DEURSEN D., COPPENS S., VALLÉS J. G. & VAN DE WALLE R. (2012). Functional descriptions as the bridge between hypermedia APIs and the semantic web. In *Proceedings of the third international workshop on RESTful design*, p. 33–40 : ACM.

WEERAWARANA S., CURBERA F., LEYMANN F., STOREY T. & FERGUSON D. F. (2005). *Web Services Platform Architecture : SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. USA : Prentice Hall PTR.

Propagation contextuelle des propriétés pour les graphes de connaissances : une approche fondée sur les plongements de phrases

Pierre-Henri Paris, Fayçal Hamdi, Samira Si-said Cherfi

Conservatoire National des Arts et Métiers, Paris, France
pierre-henri.paris@upmc.fr, faycal.hamdi@cnam.fr, samira.cherfi@cnam.fr

Résumé : Avec le nombre sans cesse croissant de graphes de connaissances RDF, le nombre d'interconnexions entre ces graphes à l'aide de la propriété *owl:sameAs* a explosé. De plus, comme plusieurs travaux précédents l'ont indiqué, l'identité définie par la sémantique de *owl:sameAs* est peut-être trop rigide dans de nombreux cas. En effet, l'identité devrait être considérée comme dépendante du contexte. Ces faits conduisent à des données de mauvaise qualité lors de l'utilisation des capacités d'inférence de *owl:sameAs*. L'identité contextuelle pourrait être une voie possible vers une connaissance de meilleure qualité. Contrairement à l'identité classique, avec l'identité contextuelle, seules certaines propriétés peuvent être propagées entre des entités contextuellement identiques. Dans la continuité de ces travaux sur l'identité contextuelle, nous proposons une approche, fondée sur les plongements de phrases, pour trouver semi-automatiquement un ensemble de propriétés, pour un contexte d'identité donné, qui peuvent être propagées entre des entités contextuellement identiques. Les cas d'utilisation fournis montrent que l'identification des propriétés qui peuvent être propagées aide les utilisateurs à obtenir les résultats souhaités qui répondent à leurs besoins lorsqu'ils interrogent un graphe de connaissances, c'est-à-dire des réponses plus complètes et plus précises.
Mots-clés : RDF, identité contextuelle, graphe de connaissances, données liées, plongement de phrases.

1 Introduction

Les graphes de connaissances RDF tels que Wikidata [1] ou DBpedia [2] sont en croissance constante en matière de nombre et d'usage. Cette croissance va de pair avec une augmentation du nombre d'entités décrites dans ces graphes et entraîne un problème pour les éditeurs de données et leurs utilisateurs : **comment savoir si deux entités sont les mêmes ou pas ?** Pour interconnecter les graphes de connaissances, la propriété *owl:sameAs* a été définie par le W3C [3] en 2004 pour relier des entités qui sont les mêmes. En effet, un objet (du monde réel) peut être décrit parmi plusieurs graphes de connaissances, et ces descriptions sont liées entre elles grâce à la propriété *owl:sameAs*. Cependant, la définition sémantique de *owl:sameAs* est très stricte. Elle est fondée sur la définition de l'identité de Leibniz, c'est-à-dire l'identité des indiscernables :

$$\forall x, \forall y(\forall p, \forall o, (\langle x, p, o\rangle \; et \; \langle y, p, o\rangle) \rightarrow x = y) \tag{1}$$

Et sa réciproque, l'indiscernabilité des identiques :

$$\forall x, \forall y(x = y \rightarrow \forall p, \forall o, (\langle x, p, o\rangle \rightarrow \langle y, p, o\rangle)) \tag{2}$$

C'est pourquoi deux entités sont considérées comme identiques si et seulement si elles partagent tous leurs couples ⟨*propriété, valeur*⟩ dans tous les contextes possibles et imaginables. En d'autres termes, deux identités sont identiques si **toutes leurs propriétés sont indiscernables** pour chaque valeur. Une fois qu'un lien d'identité est établi entre deux entités, il est possible d'utiliser les couples ⟨*propriété, valeur*⟩ de l'une sur l'autre et inversement. C'est ce

1. https://www.wikidata.org
2. https://wiki.dbpedia.org/
3. https://www.w3.org/TR/owl-ref/

que l'on nomme "propagation" dans cet article. Cependant, il faut noter que c'est une affirmation très forte que d'établir que deux objets sont identiques quel que soit le contexte. D'un point de vue philosophique, il y a de multiples contre-arguments à la définition de l'identité selon Leibniz. Par exemple, si nous considérons deux verres d'un même ensemble, ils sont indiscernables dans leurs fonctions et apparences, mais ils n'en sont pas moins deux objets physiques distincts. Ou encore, un bateau dont on changerait toutes les pièces au cours du temps sera-t-il toujours le même au cours du temps ?

Il s'agit aussi d'un problème technique à cause de l'hypothèse du monde ouvert (Drummond & Shearer (2006)) d'une part, et de l'intention de l'éditeur de données d'autre part. En effet, l'éditeur pourra avoir en tête une intention d'utilisation différente de celle de l'utilisateur. En outre, lorsque les données sont publiées, il est presque impossible de connaître le **consensus** qui sous-tend la décision de créer un tel lien. Plusieurs travaux (Halpin *et al.* (2010) ou Ding *et al.* (2010)) ont démontré que, dans certains cas, l'utilisation de *owl:sameAs* était inadéquate. En effet, les liens établis ne peuvent être considérés comme vrais que dans des contextes spécifiques. De nos jours, le problème de l'identité reste l'un des plus importants pour l'industrie travaillant avec des graphes de connaissances (Noy *et al.* (2019)).

En première approximation, une identité contextuelle entre deux entités pourrait être considérée comme un sous-ensemble Π des propriétés de ces entités pour lesquelles les entités partagent les mêmes valeurs pour chaque $p \in \Pi$. Π est l'ensemble d'indiscernabilité.

Exemple 1
Deux médicaments génériques différents Drug1 et Drug2 peuvent être identiques en ce qui concerne leur principe actif. Si un graphe de connaissances contient les triplets $\langle$ *Drug1 activeIngredient Molecule1* $\rangle$ *et* $\langle$ *Drug2 activeIngredient Molecule1* $\rangle$*, alors* $Drug1 \equiv_{activeIngredient} Drug2$ *lorsque l'ensemble d'indiscernabilité est* $\Pi = \{activeIngredient\}$.

L'une des principales caractéristiques de *owl:sameAs* est de pouvoir **propager toutes les propriétés** d'une entité vers d'autres entités identiques. Si $A = B$, alors toutes les caractéristiques de A sont aussi valables pour B, et inversement. Ainsi, *owl:sameAs* permet de découvrir plus de connaissances et d'accroître la complétude. De la même manière, l'identité contextuelle doit aider à découvrir **plus de connaissances et à accroître la complétude**, mais seulement dans des circonstances spécifiques. Pour être utile, une identité contextuelle doit préciser ce qui se passe avec les propriétés qui ne font pas partie de l'ensemble d'indiscernabilité. En d'autres termes, **une identité contextuelle** doit aussi pouvoir permettre de **propager certaines propriétés**.

Exemple 2
En reprenant l'exemple 1, établir seulement $Drug1 \equiv_{activeIngredient} Drug2$ *est d'un intérêt limité, puisqu'en dehors de* $activeIngredient$*, nous ne savons pas quoi faire des autres propriétés des médicaments. En considérant que* $activeIngredient$ *est l'ensemble d'indiscernabilité, nous savons, en tant qu'être humains, que la propriété* $targetDisease$ *est propageable, et nous pouvons conclure que si la déclaration* $\langle$ *Drug1 targetDisease Disease1* $\rangle$ *existe alors* $\langle Drug2\ targetDisease\ Disease1 \rangle$ *aussi. A l'inverse, nous savons que la propriété* $excipient$ *n'est pas nécessairement propageable.*

De plus, la capacité à propager une propriété entre entités dépend du contexte, c'est-à-dire que la même propriété peut se propager dans un contexte C_1 et ne pas se propager dans un autre contexte C_2. Par exemple, la propriété "maladie ciblée" se propage entre deux médicaments si le contexte est la propriété "principe actif". Mais si le contexte est "produit par", alors "maladie ciblée" ne sera très certainement pas propageable entre deux médicaments.

Plusieurs travaux ont tenté de proposer une solution à l'identité contextuelle. Beek *et al.* (2016), Idrissou *et al.* (2017) et Raad *et al.* (2017) ont défini trois façons différentes de traiter l'identité dans un contexte donné. Mais aucun de ces travaux ne propose de solutions pour découvrir des propriétés qui peuvent être propagées dans un contexte spécifique.

Questions de recherche : Avec un contexte d'identité donné entre deux entités, comment trouver les propriétés qui peuvent être propagées ? Est-il possible de trouver ces propriétés propageables (semi-)automatiquement ?

Dans cet article, en reprenant la définition de Idrissou *et al.* (2017), nous proposons une approche pour **trouver les propriétés propageables** afin de faciliter la découverte de connaissances pour les utilisateurs de graphes de connaissances. Nous utilisons une méthode de plongement de phrases existante fondée sur les réseaux de neurones pour découvrir des propriétés propageables pour un contexte donné. Nous avons validé notre approche avec des expériences qualitatives.

Le reste du document est organisé comme suit. Dans la section suivante, nous présentons les travaux connexes. Dans la section 4, nous présentons notre approche. Dans la section 5, nous présentons les expériences qualitatives que nous avons menées. Enfin, nous concluons et définissons les prochaines orientations de nos travaux futurs.

2 Travaux connexes

Dans la première partie de cette section, nous décrivons les articles qui ont souligné les problèmes soulevés par l'usage de *owl:sameAs*. La deuxième partie concerne les propositions qui s'attaquent à ces problèmes.

2.1 Crise de l'identité

Comme décrit dans Horrocks *et al.* (2006), le but de la propriété *owl:sameAs* est de relier deux entités qui sont strictement identiques, c'est-à-dire que les deux entités sont identiques dans tous les contextes possibles. *owl:sameAs* a une sémantique stricte permettant de déduire de nouvelles informations. De nombreux outils existants produisent de tels liens *owl:sameAs* (Ferrara *et al.* (2011)), et plusieurs enquêtes sont disponibles (voir Ferrara *et al.* (2011), Achichi *et al.* (2015) et Nentwig *et al.* (2017)).

Toutefois, aucune de ces approches ne tient compte des liens d'identité contextuels. Leur but est de découvrir des liens d'identité qui seraient toujours valables. Ceci est, d'un point de vue philosophique, difficile à obtenir comme le souligne la définition de l'identité de Leibnitz.

Dès 2002, Guarino & Welty (2002) a soulevé la question de l'identité pour les ontologies. Surtout lorsque le temps est impliqué, affirmer que deux choses sont identiques devient un problème philosophique. Les auteurs ont proposé de n'impliquer dans l'identité que les propriétés essentielles, c'est-à-dire les propriétés qui ne peuvent pas changer. Comme indiqué par exemple dans Halpin *et al.* (2010) ou Ding *et al.* (2010), en raison de la sémantique stricte de *owl:sameAs*, la charge des éditeurs de données pourrait être trop lourde. En fait, ces liens ne sont pas souvent utilisés de manière adéquate. Certains peuvent être simplement erronés et, plus insidieusement, certains peuvent dépendre du contexte, c'est-à-dire que le lien ne tient pas dans tous les contextes possibles parce qu'il est difficile d'obtenir un consensus sur la validité d'une déclaration. Le sens donné par un modélisateur de données peut ne pas correspondre à ce qu'attend un utilisateur de données. Cette utilisation abusive de *owl:sameAs* est souvent appelée "crise de l'identité" (Halpin *et al.* (2010)).

2.2 Identité contextuelle

Beek *et al.* (2016) ont abordé cette question en construisant un treillis de contextes d'identité où les contextes sont définis comme des ensembles de propriétés. Cela correspond à la première approximation proposée dans la Section 1. Toutes les entités identiques dans un contexte partagent les mêmes valeurs pour chaque propriété de ce contexte. Ainsi, un contexte est un ensemble de propriétés indiscernables pour une entité. Cependant, les auteurs ne donnent pas d'indications sur l'utilisation de propriétés n'appartenant pas à de tels contextes. Raad *et al.* (2017) ont proposé un algorithme appelé DECIDE pour calculer les contextes, où les contextes d'identité sont définis comme des sous-ontologies. Mais comme dans le précédent article, les propriétés des entités qui ne sont pas dans la sous-ontologie

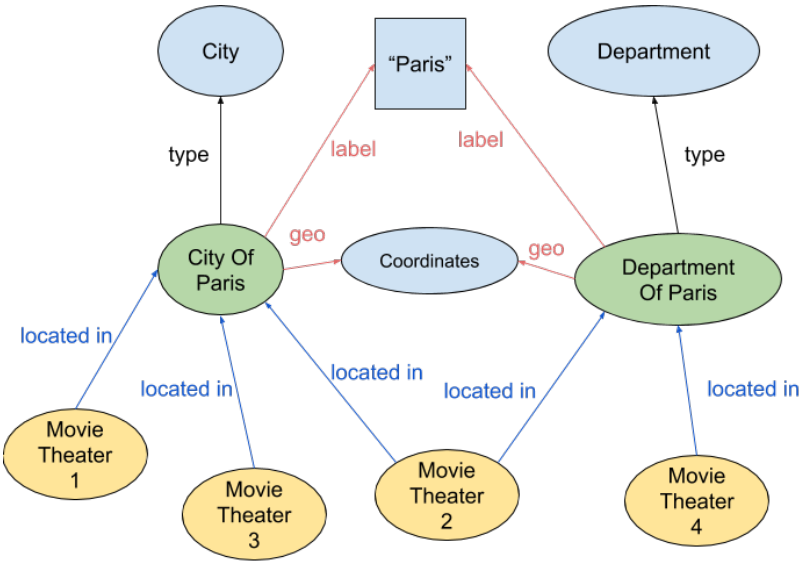


FIGURE *1 – Extrait d'un graphe de connaissances sur Paris. Les propriétés en rouge sont indiscernables pour la ville et le département. Les propriétés en bleu se propagent tant que les propriétés en rouge sont indiscernables.*

sont ignorées. Ainsi, dans les deux travaux précédents, il y a une limitation concernant les propriétés qui n'appartiennent pas à un contexte. Cette limitation réduit l'intérêt d'utiliser de telles approches. En effet, l'un des objectifs d'un contexte d'identité est de définir une relation d'identité entre deux entités afin d'utiliser les informations de l'une à l'autre. La solution de Idrissou *et al.* (2017) implique une telle propagation des propriétés, et donc, augmente la complétude d'une entité selon un contexte. Toutefois, cette proposition exige que l'utilisateur donne en entrée les propriétés de propagation et les propriétés indiscernables. Elle laisse donc à l'utilisateur la charge d'identifier et de fournir le contexte et les propriétés. L'utilisateur doit fournir les deux ensembles de propriétés indiscernables et propageables.

Dans ce travail, nous proposons d'enlever partiellement cette charge à l'utilisateur, c'est-à-dire de **calculer semi-automatiquement l'ensemble de propagation des propriétés étant donné un ensemble de propriétés indiscernables**. Pour cela, nous utiliserons une approche de plongement de phrases (présentée dans la section 4.3) pour calculer les plongements des (descriptions des) propriétés afin de découvrir les **propriétés propageables** pour un contexte d'identité donné (tel que défini dans Idrissou *et al.* (2017)).

3 Motivation

Parfois, les entités du monde réel peuvent être proches quant à leurs propriétés, mais pas exactement les mêmes. Par exemple, la capitale française, Paris, est à la fois une ville et un département. Tout en considérant que la ville et le département sont les mêmes en ce qui concerne leur géographie, ils sont deux entités distinctes sur le plan administratif (ou juridique). Maintenant, supposons que les deux Paris soient représentés dans un graphe de connaissances en tant qu'entités distinctes, et que les deux soient liés à des cinémas (éventuellement distincts). Si l'on veut récupérer les salles de cinéma situées dans la ville de Paris, les résultats ne seront pas complets si certains d'entre eux sont liés au département (voir Figure 1).

Un citoyen français peut connaître cette vérité, mais comment permettre à un agent automatisé de découvrir ce fait ? L'identité contextuelle est une réponse possible à cette question, c'est-à-dire un ensemble de propriétés pour lesquelles les valeurs sont les mêmes pour les deux entités. Dans le présent exemple, les deux Paris (ville et département) sont géographi-

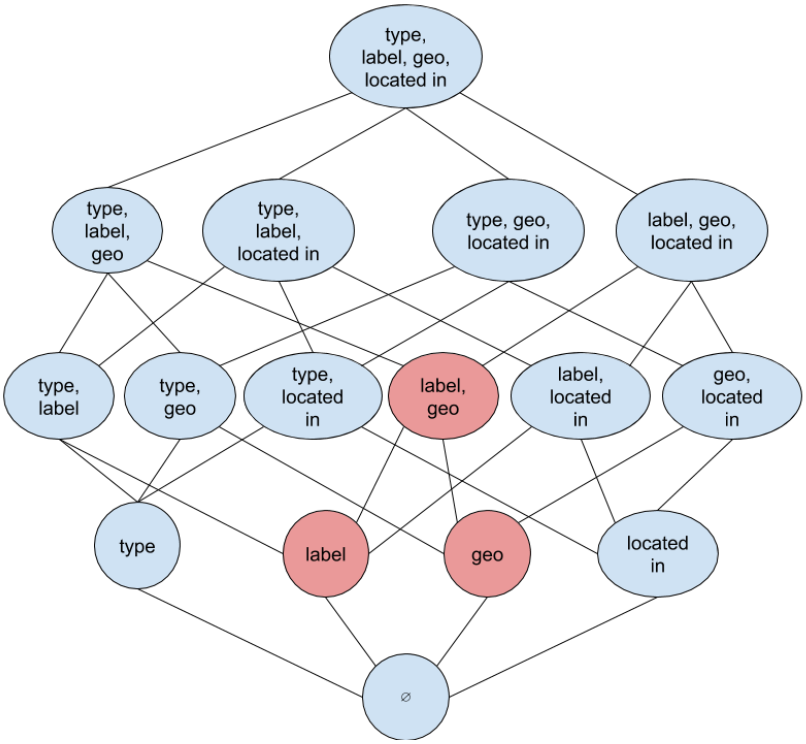


FIGURE 2 – *Treillis d'identité simplifié de la figure 1 : chaque nœud est un ensemble de propriétés. Seuls les nœuds rouges ont des entités similaires.*

quement identiques et certaines propriétés liées à la géographie pourraient être **propagées**. Dans la figure 1, les propriétés rouges (*geo* et *label*) sont indiscernables (ont les mêmes valeurs) et les propriétés bleues (*located in*) se propagent. Dans le monde réel, les cinémas situés dans la ville ou le département selon le graphe de connaissances sont en fait situés au même endroit. Malgré le fait que les deux entités ne partagent pas les mêmes valeurs pour la propriété *located in*, celle-ci est liée au contexte géographique. En effet, pour un agent humain, la propriété *located in* pourrait être évidemment propagée entre les deux entités.

Alors que nous savons que les quatre cinémas sont situés à Paris, la requête dans Listing 1 ne donnera que les cinémas 1, 2 et 3 (voir Figure 1).

```
SELECT DISTINCT ?movieTheater WHERE {
    ?movieTheater :locatedIn :CityOfParis .
}
```

Listing 1 – Requête SPARQL récupérant tous les cinémas de Paris.

Ainsi, la découverte de tels contextes d'identité entre entités pourrait améliorer les résultats des requêtes. Notre intuition est inspirée par la première loi de Tobler (Tobler (1970)), c'est-à-dire :

> "*Tout interagit avec tout, mais deux objets proches ont plus de chances de le faire que deux objets éloignés.*"

Par conséquent, **nous émettons l'hypothèse que, d'un point de vue sémantique, plus une propriété est proche du contexte d'identité, plus elle est susceptible d'être une bonne candidate à la propagation**. Dans l'exemple précédent, *located in* fait clairement référence à la géographie, et le contexte d'identité concerne la géographie puisqu'il est composé de coordonnées géographiques. L'idée est donc de **calculer une distance entre les propriétés indiscernables** et **les propriétés candidates à la propagation**. Par conséquent, les nombres,

et dans notre cas les vecteurs numériques sont les mieux adaptés pour calculer cette distance. Une représentation numérique de la description textuelle de chaque propriété grâce à la valeur donnée par $rdfs{:}comment$ ou $schema{:}description$ peut fournir une base pour obtenir ce vecteur. En effet, le plongement des descriptions des propriétés nous donne des vecteurs numériques dont la distribution dans l'espace vectoriel respecte la similarité sémantique des phrases.

Dans ce qui suit, nous décrirons notre approche avec plus de détails.

4 Approche

Dans cette section, avant de plonger plus avant dans l'approche de base, nous donnons quelques définitions nécessaires par la suite pour décrire l'approche.

4.1 Préliminaires

Comme nous l'avons vu dans la section 2, plusieurs propositions ont été faites pour définir un contexte d'identité. Nous avons choisi celle de Idrissou *et al.* (2017) (Définition 1) car elle est la seule à prendre en compte la propagation des propriétés. Ils donnent la définition suivante du contexte d'identité :

Définition 1
*(**Contexte d'identité**) Un contexte d'identité $\mathcal{C} = (\Pi, \Psi, \approx)$ est défini par deux ensembles de propriétés (Π et Ψ) et une **procédure d'alignement** ($\approx$). Π est l'**ensemble d'indiscernabilité** des propriétés (équation 3) et Ψ est l'**ensemble de propagation** des propriétés (equation 4). Dans la suite, x et y sont des entités.*

$$\begin{aligned} x =_{(\Pi,\Psi,\approx)} y \leftrightarrow \forall (p_1, p_2) \in \Pi^2 \text{ avec } p_1 \approx p_2 \\ \text{et } \forall v_1, v_2 \text{ avec } v_1 \approx v_2 : \langle x, p_1, v_1 \rangle \leftrightarrow \langle y, p_2, v_2 \rangle \end{aligned} \tag{3}$$

$$\begin{aligned} x =_{(\Pi,\Psi,\approx)} y \rightarrow \forall (p_1, p_2) \in \Psi^2 \text{ avec } p_1 \approx p_2 \\ \text{et } \forall v_1, v_2 \text{ avec } v_1 \approx v_2 : \langle x, p_1, v_1 \rangle \leftrightarrow \langle y, p_2, v_2 \rangle \end{aligned} \tag{4}$$

*De plus, nous définissons le **niveau d'un contexte** $|\Pi_{\mathcal{C}}|$ comme étant le nombre de propriétés indiscernables (dans Π).*

Dans le cas où des entités similaires selon un contexte d'identité appartiennent au même graphe de connaissances, il n'est pas nécessaire d'avoir une procédure d'alignement.

Une entité peut avoir plusieurs contextes d'identité, en fonction des propriétés de l'ensemble d'indiscernabilité Π. En effet, deux combinaisons différentes de propriétés peuvent donner des ensembles différents d'entités similaires. Par exemple, si l'on choisit comme ensemble d'indiscernabilité la propriété "principe actif", pour un médicament donné med, on aura tous les médicaments qui ont le même principe actif que med. A contrario, si l'on choisit comme ensemble d'indiscernabilité la propriété "produit par" pour la graine med, on aura tous les médicaments produits par la société qui produit med. Dans les deux cas, la graine est la même (med), mais les entitiés similaires des deux ensembles d'indiscernabilités ont de grandes chances d'être différentes. Pour finir, l'ensemble d'indiscernabilité composé des deux propriétés "principe actif" et "produit par" donnera comme entités similaires celles qui sont produites par l'entreprise fabriquant med et qui ont le même principe actif. Il se peut bien entendu qu'il n'existe pas de médicament correspondant à ce cas.

Le treillis d'identité de tous les contextes d'identité d'une entité e est défini comme suit (voir Figure 2) :

Définition 2
*(**Treillis d'identité**) Un treillis d'identité $\mathcal{L}$ est un treillis dont chaque élément est un contexte d'identité. L'inclusion ensembliste entre les ensembles d'indiscernabilités de chaque contexte est la relation binaire responsable de l'ordre partiel.*

La dernière notion est la graine d'un treillis ou d'un contexte que nous définissons comme suit :

Définition 3
*(**Graine d'un treillis ou d'un contexte**) Chaque contexte d'identité d'un treillis d'identité est construit à partir de la **même entité** e. Cette entité e est appelée la graine du treillis d'identité.*

En effet, pour construire un treillis d'identité, nous devons partir d'une graine, même si le treillis pourrait potentiellement être valide avec une autre graine, c'est-à-dire que deux graines peuvent donner le même treillis d'identité.

Maintenant que nous avons défini les concepts nécessaires, nous allons expliquer le cœur de notre approche.

4.2 Calcul des contextes

Dans cette section, nous expliquons comment calculer un treillis et ses contextes.

Nous présentons l'Algorithme 1 qui permet le calcul du treillis d'identité. Il prend en entrée l'entité graine, le graphe de connaissances source auquel la graine appartient, le graphe de connaissances cible (éventuellement le même que le graphe de connaissances source) et une procédure d'alignement si les deux graphes de connaissances sont distincts. L'idée principale est de commencer par calculer les ensembles d'indiscernabilité de bas niveau avec chaque propriété de la graine et enfin de combiner ces ensembles pour obtenir des ensembles d'indiscernabilité de plus haut niveau. **Lorsqu'on construit un contexte, le premier élément est son ensemble d'indiscernabilité, à partir duquel on obtient ensuite des entités similaires, pour finalement obtenir des propriétés candidates à la propagation et, en conclusion, des propriétés de propagation.**

La première étape, ligne 3, consiste à calculer tous les contextes d'identité de niveau 1 (voir Définition 1). En effet, pour chaque propriété p de la graine, il existe exactement un contexte d'identité (son ensemble d'indiscernabilité est $\Pi = \{p\}$). Par la suite, les contextes d'identité n'ayant qu'une seule propriété d'indiscernabilité seront fusionnés pour donner des contextes d'identité de niveau 2, puis de niveau 3, etc. Ensuite, nous récupérons l'ensemble $entities_p$ des entités similaires à la graine qui ont la (les) même(s) valeur(s) pour la propriété donnée p. Si p est multi-valué (plusieurs o pour p), alors les entités dans $entities_p$ sont similaires à la graine pour toutes les valeurs o de sorte que $\langle seed\ p\ o \rangle$, c'est-à-dire $\forall p, (\forall o, \langle seed, p, o \rangle \rightarrow \forall e \in entities_p, \langle e, p, o \rangle)$. Il est à noter que, lors du remplissage de $entities_p$, nous ne recherchons que les entités qui ont le(s) même(s) type(s) que la graine. C'est parce que nous voulons éviter les résultats absurdes, par exemple comparer une personne avec un avion. Elle présente également l'avantage de réduire considérablement le nombre de contextes d'identité possibles à calculer. Enfin, sur la base de $entities_p$, nous calculons l'ensemble de propagation Ψ (ligne 12) comme expliqué dans la section suivante.

La deuxième étape (voir Algo. 2) consiste à calculer les contextes d'identité de niveau supérieur en se basant sur ceux du niveau 1. La boucle (ligne 2) de l'algorithme calcule ces niveaux supérieurs en combinant des contextes de même niveau, et s'arrête lorsqu'il ne peut pas construire de nouveaux contextes d'identité de niveau supérieur. Ce calcul est basé sur un opérateur de treillis d'identité qui est l'ensemble d'inclusion sur les ensembles d'indiscernabilité. Par exemple, un contexte de niveau 2 (deux propriétés dans Π) est construit sur deux contextes de niveau 1 (chacun une propriété dans Π). Là encore, pour réduire le nombre de contextes d'identité possibles à calculer, s'il n'y a pas d'entité similaire à la semence pour un contexte donné C_i, il n'est pas nécessaire de calculer des contextes de niveau supérieur basés sur C_i.

Data: $\mathcal{KG}_1$: the source KG, $\mathcal{KG}_2$: the target KG, $seed$: an entity of $\mathcal{KG}_1$, $\approx$: an alignment procedure between $\mathcal{KG}_1$ and $\mathcal{KG}_2$
Result: $\mathcal{L}$: a lattice of identity contexts between the seed and entities in the target KG

$\mathcal{L} = \emptyset$;
/* Get all explicit and implicit types of the seed */
$\mathcal{T}_{seed} = \{t : \langle seed\ rdf{:}type\ t\rangle \in \mathcal{KG}_1\}$;
/* the following will create all contexts of the lower level (with only one indiscernible property) */
for *each property* p *of* $seed$ **do**
  $candidateEntities = \emptyset$;
  for *each value* o *such as* $\langle seed\ p\ o\rangle \in \mathcal{KG}_1$ **do**
    /* $entities_{p,o}$ is the set of indiscernible entities with $seed$ with respect to the p, o couple */
    $entities_{p,o} = \{e : (\exists(p', o'), p' \approx p, o' \approx o, \langle e\ p'\ o'\rangle \in \mathcal{KG}_2) \wedge (\exists t \in \mathcal{T}_{seed}, t' \approx t, \langle e\ rdf{:}type\ t'\rangle \in \mathcal{KG}_2)\}$;
    if $entities_{p,o} \neq \emptyset$ **then**
      $candidateEntities = candidateEntities \cup \{entities_{p,o}\}$;
    end
  end
  /* $entities_p$ is the set of indiscernible entities with $seed$ with respect to the property p */
  /* intersection of all sets in $candidateEntities$ */
  $entities_p = \bigcap candidateEntities$;
  $\Psi = getPropagationSet(seed, entities_p, \{p\})$;
  if $\Psi \neq \emptyset$ **then**
    $\Pi = \{p\}$;
    $\mathcal{C} = (\Pi, \Psi, \approx)$;
    $\mathcal{L} = \mathcal{L} \cup \mathcal{C}$;
  end
end
/* Now we can combine contexts of the same level */
return $constructUpperLevels(\mathcal{L})$

Algorithme 1: createLattice : calculer le treillis d'identité d'une entité.

4.3 Plongement de phrase

Notre approche étant fondée sur le plongement de phrases ("*sentence embedding*"), nous donnons dans cette section plus de détails sur cette notion. En effet, lors du calcul d'un contexte d'identité, nous calculons son ensemble de propagation correspondant à l'aide de l'algorithme 3.

Le plongement de phrases est une technique qui permet de faire correspondre une phrase à un vecteur numérique. Idéalement, les phrases sémantiquement proches sont représentées par des vecteurs proches dans l'espace vectoriel considéré.

Exemple 3
"Un match de football auquel participent plusieurs hommes" et "Certains hommes pratiquent un sport" sont proches sémantiquement, donc leurs vecteurs devraient être proches en ce qui concerne la distance.

Réciproquement, deux phrases qui ne sont pas apparentées sémantiquement doivent avoir des vecteurs éloignés.

Data: $\mathcal{KG}_1$: the source KG, $\mathcal{KG}_2$: the target KG, $seed$: an entity of $\mathcal{KG}_1$, $\approx$: an alignment procedure between $\mathcal{KG}_1$ and $\mathcal{KG}_2$
Result: $\mathcal{L}$: a lattice of identity contexts between the seed and entities in the target KG
/* lvl is the current level in the lattice */
$lvl = 1$;
while $\emptyset \notin \mathcal{L}$ **do**
  $contexts = \emptyset$;
  for $(\mathcal{C}_1, \mathcal{C}_2) \in \{(\mathcal{C}_i, \mathcal{C}_j) \in \mathcal{L}^2 : |\Pi_{\mathcal{C}_i}| = |\Pi_{\mathcal{C}_j}| = lvl, i > j\}$ **do**
    $\Pi = \Pi_{\mathcal{C}_1} \cup \Pi_{\mathcal{C}_2}$;
    /* *getEntities* function gives the set of entities that are similar under the given identity context in the given KG */
    $entities = getEntities(\mathcal{C}_1, \mathcal{KG}_2) \cap getEntities(\mathcal{C}_2, \mathcal{KG}_2)$;
    if $entities \neq \emptyset$ *and* $\Pi \notin \mathcal{L}$ **then**
      $\Psi = getPropagationSet(seed, entities, \Pi)$;
      /* see Algo. 3 */
      if $\Psi \neq \emptyset$ **then**
        $\mathcal{C} = (\Pi, \Psi, \approx)$;
        $contexts = contexts \cup \mathcal{C}$;
      end
    end
  end
  $\mathcal{L} = \mathcal{L} \cup contexts$;
  $lvl = lvl + 1$;
end
return $\mathcal{L}$

Algorithme 2: constructUpperLevels : calculer les niveaux supérieurs du treillis d'identité d'une entité.

Exemple 4
"Un homme inspecte l'uniforme d'un personnage dans un pays d'Asie de l'Est" et "L'homme dort" doivent avoir des vecteurs éloignés.

Ces vecteurs permettent d'utiliser divers opérateurs mathématiques qui ne sont évidemment pas disponibles avec des chaînes de caractères. L'un des premiers travaux importants dans ce domaine est *Word2Vec* (Mikolov *et al.* (2013)) qui capture la cooccurrence des mots. Chaque mot est traité de manière atomique et fournit un plongement grâce à deux approches distinctes, à savoir *Skip-Gram* et *Continuous Bag of Words* (CBOW). Alors que l'objectif de CBOW est de prédire un mot en fonction de sa fenêtre (c'est-à-dire les mots précédents et suivants dans une phrase), Skip-Gram essaiera de prédire les mots avec lesquels un mot est habituellement vu. De même, *GloVe* (Pennington *et al.* (2014)) fournit des plongements pour des mots uniques et peut utiliser Skip-Gram ou CBOW. Mais *GloVe*, au lieu de capturer la cooccurrence, se concentre (à la fin) sur le nombre d'apparitions parmi les fenêtres (c'est-à-dire les mots précédents et suivants dans une phrase). Ensuite, *fastText* (Bojanowski *et al.* (2017)) est une extension de *Word2Vec* qui traite les mots comme n-grammes de caractères plutôt que comme entité atomique. La taille des N-grammes dépend des paramètres d'entrée. L'utilisation de N-grammes permet une meilleure compréhension des petits mots. Chaque n-gramme est mis en correspondance avec un vecteur et la somme de ces vecteurs est la représentation du mot. Un autre avantage du *fastText* est sa capacité à fournir un plongement même pour les mots inconnus, grâce à l'utilisation de n-grammes. Alors que les trois travaux précédents sont les mieux adaptés pour travailler avec des mots atomiques, le suivant calcule le plongement pour une phrase entière.

Les raisons pour lesquelles on utilise le plongement de phrases plutôt qu'une distance plus classique, par exemple la distance d'édition, le plongement de graphe RDF comme RDF2Vec

Data: $seed$: the entity that generated Π,
$entities$: set of entities similar to $seed$ with respect to Π,
Π : an indiscernibility set
Result: Ψ : a propagation set
/* computation of the embeddings of each property in Π by using one of the encoder */
$indiscernibilityEmbeddings \leftarrow getEmbeddings(\Pi)$;
$meanVector \leftarrow mean(indiscernibilityEmbeddings)$;
/* $getCandidateProperties$ function returns the set of all candidate properties for propagation */
$candidates \leftarrow getCandidateProperties(\Pi, \{seed\} \cup entities)$;
/* then compute their embeddings */
$candidatesEmbeddings \leftarrow getEmbeddings(candidates)$;
$\Psi \leftarrow \emptyset$;
for $candidate$ *in* $candidatesEmbeddings$ **do**
  $similarity \leftarrow cosineSimilarity(candidate, meanVector)$;
  if $similarity \geq threshold$ **then**
    $\Psi \leftarrow \Psi \cup \{candidate\}$;
  end
end
return Ψ

Algorithme 3: getPropagationSet : calculer l'ensemble de propagation.

(Ristoski & Paulheim (2016)), ou une technique d'alignement ontologique sont les suivantes : *(i)* les distances de chaînes de caractères classiques ignorent la sémantique des phrases, *(ii)* les techniques de plongement de graphes RDF ne sont pas encore adaptées à une telle tâche, et *(iii)* les techniques d'alignement ontologique alignent des propriétés par paires et non des ensembles de propriétés.

Une grande attention a été accordée aux plongements de phrases ces derniers temps. Des approches telles que *Universal Sentence Encoder* (Cer *et al.* (2018)), *GenSen* (Subramanian *et al.* (2018)) et *InferSent* (Conneau *et al.* (2017)) font partie des encodeurs de référence pour le plongement de phrases. Nous choisissons d'utiliser ce dernier, mais notre approche pourrait bénéficier de n'importe laquelle de ces approches. *InferSent*, proposé par Conneau *et al.* (2017), est un encodeur de pointe qui s'est révélé efficace pour le plongement de phrases. Pour entraîner leur modèle supervisé de plongement de phrases, les auteurs ont utilisé l'ensemble de données SNLI (Stanford Natural Language Inference) qui consiste en plus de 500K paires de phrases anglaises étiquetées manuellement avec l'une des trois catégories : implication, contradiction et neutre. Ils ont testé plusieurs architectures et ont découvert qu'un réseau BiLSTM avec un *max pooling* offrait les meilleurs résultats. Un réseau BiLSTM est un LSTM bidirectionnel souvent utilisé pour les données séquentielles, c'est-à-dire un réseau de neurones récurrent (avec des boucles). Le *max pooling* est une technique qui permet de réduire le nombre de paramètres du modèle en sélectionnant la valeur maximale d'une "fenêtre" mobile. De plus, le modèle pré-entraîné est basé sur fastText, ce qui permet de calculer des représentations significatives même pour des mots hors vocabulaire, c'est-à-dire des mots qui n'apparaissent pas dans les données d'entraînement. *GenSen* (Subramanian *et al.* (2018)) et *Universal Sentence Encoder* (Cer *et al.* (2018)) sont tous deux basés sur l'apprentissage multitâche (MTL). Le but de MTL est d'apprendre de multiples aspects d'une phrase en alternant entre différentes tâches comme la traduction ou l'inférence en langage naturel. Les premiers utilisent un GRU (Gated Recurrent Units) bidirectionnel, c'est-à-dire un réseau de neurones récurrent comme le LSTM mais avec moins de paramètres. Ce dernier utilise l'architecture *transformer* qui transforme une séquence en une autre, mais sans réseau de neurones récurrent (contrairement à *InferSent* et *GenSen*).

Comme présenté dans la section 1, notre intuition, basée sur la première loi de Tobler,

est qu'un ensemble de propriétés de propagation peut être trouvé étant donné un ensemble d'indiscernabilité, si les vecteurs de description de ces deux ensembles sont suffisamment proches. Dans ce travail, **nous proposons d'utiliser les descriptions textuelles longues en langage naturel des propriétés (par exemple** $rdfs{:}comment$ **ou** $schema{:}description$**) pour trouver des propriétés qui sont sémantiquement liées** et par conséquent de bonnes candidates à la propagation pour un ensemble d'indiscernabilité donné Π. Pour le calcul du plongement, n'importe lequel des encodeurs décrits précédemment peut être utilisé.

L'algorithme 3 présente notre proposition de calculer Ψ étant donné un Π. Il prend comme entrée trois paramètres : une graine (une entité), un ensemble de propriétés construites à partir de la graine (ensemble d'indiscernabilité Π), et un ensemble d'entités qui sont similaires à la graine en ce qui concerne Π. Le calcul de Π est présenté dans la section précédente (voir algorithme 1).

Tout d'abord, pour chaque propriété de l'ensemble d'indiscernabilité Π, nous calculons son vecteur de représentation (voir ligne 1). Ensuite, nous calculons le vecteur moyen qui représente l'ensemble d'indiscernabilité (ligne 2). De même, nous considérons chaque propriété de la graine ou de ses entités similaires, et calculons leurs vecteurs de représentation. Par conséquent, d'une part, nous avons un vecteur qui représente l'ensemble d'indiscernabilité et, d'autre part, nous avons des vecteurs pour les propriétés qui sont candidates à la propagation. Les entités similaires (en ce qui concerne l'ensemble d'indiscernabilité Π) sont également considérées pour obtenir des propriétés candidates, puisque l'une d'entre elles peut éventuellement avoir une propriété de propagation que la graine n'a pas (voir ligne 3).

Ensuite, nous effectuons une boucle sur chaque propriété candidate (ligne 6) pour calculer une similarité cosinus (Singhal (2001)) entre chaque vecteur candidat et le vecteur moyen représentant l'ensemble d'indiscernabilité Π. Si la similarité cosinus est suffisamment élevée (au-dessus d'un seuil spécifié, comme expliqué dans la section suivante), la propriété candidate est considérée comme une propriété de propagation (voir ligne 8).

Notre approche ayant été présentée, nous allons introduire les expériences pour valider notre travail.

5 Résultats expérimentaux

Nous présentons plusieurs requêtes SPARQL qui bénéficient de notre approche pour évaluer cette dernière. Mais tout d'abord, nous présentons succinctement notre implémentation. Il n'est pour l'instant pas possible d'évaluer quantitativement notre approche, puisque notre approche est la première à travailler sur cette problématique d'une part, et, d'autre part, il n'existe pas encore d'étalon d'or auquel nous comparer.

5.1 Implémentation et mise en place

Nous avons implémenté notre approche avec le langage Python. Dans un souci de reproductibilité, le code est mis à disposition sur un dépôt GitHub [4]. Comme mentionné précédemment, nous avons utilisé trois approches de plongement de phrases, à savoir *InferSent* [5], *GenSen* [6] et *Universal Sentence Encoder* [7]. Nous avons utilisé un fichier HDT (voir Martínez-Prieto *et al.* (2012) et Fernández *et al.* (2013)) qui contient un dump de la dernière version de Wikidata [8]. HDT est un format de sérialisation compressé pour les graphes RDF qui permet une meilleure reproductibilité qu'un *endpoint* SPARQL. Contrairement à Turtle ou N-Triples, grâce à la compression, HDT facilite les manipulations nécessaires pour reproduire les expériences. L'ordinateur que nous avons utilisé est doté d'un processeur i7 et de 32 Go de

4. https://github.com/PHParis/ConProKnow
5. https://github.com/facebookresearch/InferSent
6. https://github.com/Maluuba/gensen
7. https://tfhub.dev/google/universal-sentence-encoder/2
8. http://gaia.infor.uva.es/hdt/wikidata/wikidata2018_09_11.hdt.gz

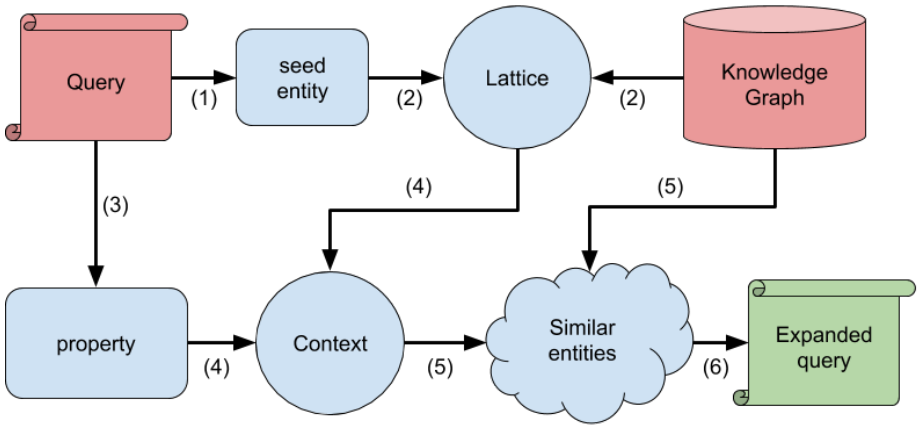


FIGURE *3 – Flux de travail de l'expérience qualitative : les éléments en rouge sont les entrées et l'élément en vert est la sortie. Pour simplifier le diagramme, nous ne considérons qu'une seule entité instanciée liée à une propriété instanciée dans la requête.*

mémoire vive. À titre indicatif, le calcul complet du treillis d'identité pour une entité telle que la ville de Paris prend environ 1 396 ms. Cette entité compte plus de 1 000 couples propriété-valeur et, dans Wikidata, le nombre moyen de couples propriété-valeur est d'environ 60. Il s'agit donc d'une entité assez importante et cette approche semble pouvoir être utilisée à grande échelle.

5.2 Étude qualitative

Dans cette section, nous introduisons trois requêtes différentes qui pourraient bénéficier de notre approche en étendant leurs résultats. Pour atteindre notre objectif, nous avons utilisé *InferSent* et une valeur de seuil égale à 0,9. Toutes ces requêtes sont des requêtes simplifiées testées sur Wikidata (pour la lecture humaine, nous avons adapté les noms d'entités et de propriétés). Les requêtes originales peuvent être trouvées sur le dépôt GitHub.

5.2.1 Objectif de la tâche

Pour chaque requête, l'objectif est de trouver un contexte d'identité qui permettra d'étendre la requête à des entités similaires. De cette façon, les utilisateurs peuvent bénéficier de résultats plus complets. Le flux de travail est le suivant (voir Figure 3) : dans un premier temps, à partir de la requête, nous extrayons l'entité (ou les entités) instanciée(s) qui sera (seront) la (les) graine(s) (étape 1). Ensuite, pour chaque graine, nous calculons son treillis d'identité (étape 2) qui contiendra dans chacun de ses nœuds un ensemble de propriétés indiscernables et de propagation (cf. algorithmes 1 et 3). Troisièmement, avec la propriété (ou l'ensemble de propriétés) instanciée liée à la graine dans la requête, nous sélectionnons dans le treillis le nœud ayant cette propriété dans son ensemble de propagation (étape 4). Ce nœud sera considéré comme le contexte d'identité de la requête. En effet, si plusieurs contextes d'identité sont possibles, l'utilisateur doit choisir celui qui convient le mieux à l'objet de sa tâche. Enfin, sur la base du contexte d'identité sélectionné, nous pouvons obtenir des entités similaires (étape 5) et réécrire la requête avec la graine et les entités similaires (étape 6).

5.2.2 Requêtes

La première requête dans le Listing 2 concerne le médicament "Paracétamol". L'objectif de la requête est de récupérer tous les essais cliniques de ce médicament. Une extension intéressante de cette requête pourrait être de trouver tous les essais de médicaments légaux similaires en termes de conditions médicales traitées et d'interactions physiques.

Requête correspondante	**Listing 2**
Graine	Paracetamol
Ψ	research intervention
Π	condition treated, interacts with, legal status
Entités similaires	Ibuprofen Aspirin
# de résultats sans notre approche	586
# de résultats avec notre approche	**860**

TABLE *1 – Contribution du contexte d'identité à la requête sur les recherches proches du Paracétamol.*

Requête correspondante	**Listing 4**
Graine	France
Ψ	head of
Π	capital, official language
Entités similaires	French 2nd Republic, July Monarchy, French 3rd Republic, Bourbon Restoration, Kingdom of France, 2nd French Empire, French 4th Republic, 1st French Empire, Paris Commune, Vichy France
# de résultats sans notre approche	12
# de résultats avec notre approche	**99** (77)

TABLE *2 – Contribution du contexte d'identité à la requête sur les dirigeants français.*

```
SELECT DISTINCT ?clinicalTrial WHERE {
  ?clinicalTrial :researchIntervention :Paracetamol .
}
```

Listing 2 – Requête SPARQL récupérant toutes les études sur l'analgésique nommé Paracétamol.

Le tableau 1 montre ce que notre approche peut apporter comme résultats supplémentaires. Pour la première requête (et aussi pour les suivantes), il n'y a qu'une seule graine "Paracétamol" (respectivement "France" et "les Républicains" dans les Tables 2 et 3) car c'est la seule entité instanciée dans la requête. Pour remplir ce tableau, nous avons d'abord calculé le treillis de la graine, puis, sélectionné un contexte contenant la propriété "research intervention" dans son Ψ. Nous avons choisi comme contexte des médicaments légaux ayant les mêmes conditions médicales et les mêmes interactions physiques (évidemment, tout autre contexte pourrait être choisi en fonction des besoins des utilisateurs). Enfin, la requête est étendue avec des entités similaires comme indiqué dans Listing 3. Les résultats montrent une augmentation de 47 % du nombre d'essais cliniques pour le contexte considéré.

```
SELECT DISTINCT ?clinicalTrial WHERE {
  VALUES (?drug) {
    (:Paracetamol) (:Ibuprofen) (:Aspirin) }
```

Requête correspondante	**Listing 5**
Graine	Les Républicains
Ψ	member of political party
Π	country, political ideology
Entités similaires	UMP, RPR, UDR, UNR
# de résultats sans notre approche	2
# de résultats avec notre approche	**13**

TABLE *3 – Contribution du contexte d'identité à la requête sur membres des Républicains condamnés.*

```
  ?clinicalTrial :researchIntervention ?drug .
}
```

Listing 3 – Expansion de la requête SPARQL en récupérant toutes les études sur les entités similaires au paracétamol dans le context d'identité choisi.

La deuxième requête, dans Listing 4, vise à retrouver les personnes qui ont autrefois dirigé la France. Cependant, la France a une histoire complexe et a changé de régime politique à plusieurs reprises (par exemple, pendant la Seconde Guerre mondiale, ou pendant la période napoléonienne). Ainsi, même si le territoire français a été presque toujours le même au cours des siècles passés, chaque régime politique a sa propre entité dans Wikidata. Il se peut donc que la requête ne donne pas tous les résultats escomptés. Mais si l'utilisateur choisit le bon contexte d'identité, c'est-à-dire $\mathcal{C}_{(\{capital,officialLanguage\},\{headOf\},\approx)}$ alors toutes les personnes attendues seront récupérées.

```
SELECT DISTINCT ?headOfState WHERE {
  ?headOfState :headOf :France .
}
```

Listing 4 – Requête SPARQL récupérant toutes les personnes qui ont été à la tête de l'État français moderne.

Comme pour la requête sur "Paracétamol", nous avons calculé le treillis et cherché le contexte avec *headOf* dans les propriétés de propagation. Les résultats sont indiqués dans le tableau 2. L'expansion de la requête est réalisée comme pour la précédente. Il est à noter que parmi les 99 résultats, 22 personnes n'étaient pas à la tête de la France. 14 étaient en réalité à la tête du conseil municipal de Paris et 8 étaient Grand Maître d'une obédience maçonnique en France. Cela est dû au fait que le conseil et l'obéissance sont mal placés dans l'ontologie de Wikidata. Ces erreurs ne peuvent donc pas être attribuées à notre approche. Les résultats montrent une augmentation de 542 % du nombre de dirigeants français pour le contexte considéré.

Enfin, dans Listing 5, nous présentons une requête sur les politiciens français du parti Les Républicains qui ont été condamnés. La particularité ici est que cet important parti politique a changé plusieurs fois de nom, soit à cause de scandales politiques, soit à cause de défaites humiliantes. Par conséquent, si le graphe de connaissances n'est pas à jour ou n'est pas complet, certaines personnes qui ont été membres de plusieurs versions de ce parti dans le monde réel pourraient ne pas être effectivement liées à chacune de ces versions dans le graphe de connaissances. C'est le cas de Wikidata qui renvoie, pour la requête de Listing 5, seulement deux politiciens. Cependant, il y a plus d'une douzaine de politiciens de ce parti qui ont été condamnés pour divers crimes. En utilisant notre approche, il est possible de sélectionner un

contexte composé de l'alignement politique et du pays pour lequel la propriété *memberOf* se propage, et, par conséquent, d'obtenir un résultat plus complet (bien sûr en fonction de l'exhaustivité des données sur les hommes politiques dans Wikidata).

```
SELECT DISTINCT ?politician ?crime WHERE {
  ?politician :memberOf :TheRepublicans ;
    :convictedOf ?crime .
}
```

Listing 5 – La requête SPARQL récupère tous les politiciens membres du parti français Les Républicains qui ont été condamnés.

Les mêmes étapes que pour les requêtes concernant le "Paracétamol" et la "France" ont été reproduites. Les résultats sont présentés dans le tableau 3. Les résultats montrent une augmentation de 550 % du nombre de politiciens condamnés pour le contexte considéré.

5.3 Discussion

Comme nous l'avons vu, notre approche permet de découvrir des propriétés de propagation pour un ensemble donné de propriétés indiscernables Π. Un contexte d'identité avec ses ensembles d'indiscernabilité et de propagation peut fournir des réponses plus complètes aux requêtes grâce à l'expansion des requêtes. Les résultats sont très prometteurs, mais il faut les confronter à d'autres types de graphes de connaissances et à des combinaisons de graphes de connaissances distincts. En outre, notre approche ne fonctionne pas bien lorsque la propriété d'une entité manque de propriété la décrivant en langage naturel (comme *rdfs :comment* ou *schema :description*). Il s'agit d'une limitation puisque de nombreuses ontologies ne fournissent pas de descriptions textuelles de leurs propriétés. Par conséquent, une première étape pour les travaux futurs consiste à contourner cette faille par une approche à multiples composantes. De plus, dans une description textuelle, certains mots peuvent ne pas être pertinents (comme un identifiant Wikidata) et dégrader la qualité des résultats.

6 Conclusion et travaux futurs

Dans ce papier, nous avons proposé une approche fondée sur le plongement de phrases pour découvrir les propriétés propageables pour un ensemble de propriétés indiscernables données. Notre approche calcule, pour une entité, un treillis d'identité qui représente tous les contextes d'identité possibles de l'entité, c'est-à-dire les ensembles d'indiscernabilité et leurs ensembles de propagation respectifs.

Quelques limitations de notre approche nécessitent des investigations supplémentaires. En effet, seules les propriétés ayant une description textuelle peuvent être traitées. Utiliser d'autres caractéristiques, par exemple la valeur des propriétés, le nombre d'utilisations des propriétés ou leurs caractéristiques sémantiques, est donc essentiel pour améliorer les résultats. Cependant, capturer les informations ontologiques d'une propriété lors d'un plongement reste un problème ouvert. De plus, utiliser seulement une technique de plongement de phrases combinée avec l'intuition de la première loi de Tobler est peut-être trop naïf dans certains cas. Par conséquent, il est aussi nécessaire de remettre en question notre travail en combinant aussi des graphes de connaissances distincts. Pour l'instant, nous ne considérons dans le treillis que le cas où l'entité est le sujet d'un triplet, il nous faudrait donc essayer aussi de traiter les triplets dans l'autre sens. Pour finir, nous souhaitons proposer un prototype plus complet, automatisant au maximum ce qui peut l'être, pour permettre à l'utilisateur de sélectionner facilement le contexte lui permettant d'obtenir de meilleurs résultats de requête. Par exemple, l'expansion de la requête est réalisée manuellement après le calcul automatique du contexte d'identité. Il serait aussi intéressant d'utiliser RDF* et SPARQL* (Hartig & Thompson (2014)) pour représenter le contexte d'identité tel que défini dans ce papier.

IC 2020

Références

ACHICHI M., BELLAHSENE Z. & TODOROV K. (2015). A survey on web data linking. *Revue des Sciences et Technologies de l'Information-Série ISI : Ingénierie des Systèmes d'Information*.

BEEK W., SCHLOBACH S. & VAN HARMELEN F. (2016). A contextualised semantics for owl : sameas. In *ESWC*.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

CER D., YANG Y., KONG S., HUA N., LIMTIACO N., JOHN R. S., CONSTANT N., GUAJARDO-CESPEDES M., YUAN S., TAR C., SUNG Y., STROPE B. & KURZWEIL R. (2018). Universal sentence encoder. *CoRR*, **abs/1803.11175**.

CONNEAU A., KIELA D., SCHWENK H., BARRAULT L. & BORDES A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, p. 670–680 : Association for Computational Linguistics.

DING L., SHINAVIER J., FININ T. & MCGUINNESS D. L. (2010). owl : sameas and linked data : An empirical study.

DRUMMOND N. & SHEARER R. (2006). The open world assumption. In *eSI Workshop : The Closed World of Databases meets the Open World of the Semantic Web*, volume 15.

FERNÁNDEZ J. D., MARTÍNEZ-PRIETO M. A., GUTIÉRREZ C., POLLERES A. & ARIAS M. (2013). Binary rdf representation for publication and exchange (hdt). *Web Semantics : Science, Services and Agents on the World Wide Web*, **19**, 22–41.

FERRARA A., NIKOLOV A. & SCHARFFE F. (2011). Data linking for the semantic web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, **7**(3), 46–76.

GUARINO N. & WELTY C. A. (2002). Evaluating ontological decisions with ontoclean. *Commun. ACM*, **45**(2), 61–65.

HALPIN H., HAYES P. J., MCCUSKER J. P., MCGUINNESS D. L. & THOMPSON H. S. (2010). When owl : sameas isn't the same : An analysis of identity in linked data. In *International Semantic Web Conference*, p. 305–320 : Springer.

HARTIG O. & THOMPSON B. (2014). Foundations of an alternative approach to reification in rdf. *ArXiv*, **abs/1406.3399**.

HORROCKS I., KUTZ O. & SATTLER U. (2006). The even more irresistible sroiq. *Kr*, **6**, 57–67.

IDRISSOU A. K., HOEKSTRA R., VAN HARMELEN F., KHALILI A. & DEN BESSELAAR P. V. (2017). Is my : sameas the same as your : sameas ? : Lenticular lenses for context-specific identity. In *K-CAP*.

MARTÍNEZ-PRIETO M. A., ARIAS M. & FERNÁNDEZ J. D. (2012). Exchange and consumption of huge rdf data. In *The Semantic Web : Research and Applications*, p. 437–452 : Springer.

MIKOLOV T., CHEN K., CORRADO G. S. & DEAN J. (2013). Efficient estimation of word representations in vector space. *CoRR*, **abs/1301.3781**.

NENTWIG M., HARTUNG M., NGONGA NGOMO A.-C. & RAHM E. (2017). A survey of current link discovery frameworks. *Semantic Web*, **8**(3), 419–436.

NOY N. F., GAO Y., JAIN A., NARAYANAN A., PATTERSON A. & TAYLOR J. (2019). Industry-scale knowledge graphs : lessons and challenges. *Commun. ACM*, **62**(8), 36–43.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *EMNLP*.

RAAD J., PERNELLE N. & SAÏS F. (2017). Detection of contextual identity links in a knowledge base. In *K-CAP*.

RISTOSKI P. & PAULHEIM H. (2016). Rdf2vec : Rdf graph embeddings for data mining. In *International Semantic Web Conference*.

SINGHAL A. (2001). Modern information retrieval : A brief overview. *IEEE Data Eng. Bull.*, **24**(4), 35–43.

SUBRAMANIAN S., TRISCHLER A., BENGIO Y. & PAL C. J. (2018). Learning general purpose distributed sentence representations via large scale multi-task learning. *CoRR*, **abs/1804.00079**.

TOBLER W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography*, **46**(sup1), 234–240.

Vers une démarche ontologique pour la capitalisation des données de l'agroforesterie

Raphaël Conde Salazar[1,3], Fabien Liagre[2], Isabelle Mougenot[3], Jérôme Perez[1], Alexia Stokes[1]

[1] AMAP , Université de Montpellier, CIRAD, CNRS, INRAE, IRD, Montpellier, France
raphael.conde_salazar@cirad.fr, alexia.stokes@cirad.fr, jerome.perez@ird.fr

[2] AGROOF, 30140, Anduze, France
liagre@agroof.net

[3] UNIVERSITÉ DE MONTPELLIER, UMR 228 ESPACE-DEV, Espace pour le Développement, 34093, Montpellier, France
isabelle.mougenot@umontpellier.fr

Résumé : Dans un contexte de développement durable, les recherches menées autour de l'agroforesterie prennent tout leur sens en rendant intelligibles les interactions plante-plante dans un environnement pouvant être dégradé. L'agroforesterie est interdisciplinaire par nature, et mobilise, en particulier, les sciences du vivant et de l'environnement au sein d'approches systémiques. Dans ce cadre, des données d'observation sur le terrain sont acquises en partenariat avec plusieurs catégories d'acteurs (scientifiques naturalistes mais aussi forestiers, agriculteurs, éleveurs, politiques ou encore gestionnaires du territoire). Cependant, ces données sont rendues difficilement exploitables, de par la multiplicité des supports, des formats et des langages de spécialité utilisés par les différents acteurs. De plus, l'agroforesterie est également une discipline qui fait appel à d'autres domaines de connaissances, à l'exemple de la climatologie ou de la pédologie. En conséquence, nous proposons un modèle de gestion des connaissances volontairement générique qui devrait permettre aux communautés œuvrant en agroforesterie de gérer, d'analyser, d'intégrer et de partager des données hétérogènes et de provenances diverses. Ce modèle, qui se veut une base à la capitalisation et au partage des connaissances en agroforesterie, se compose d'une ontologie de domaine nommée "Agroforestry" dont les éléments sont enrichis sémantiquement par des termes issus d'ontologies terminologiques, à l'exemple d'Agrovoc ou encore de Plant Ontology. L'ontologie Agroforestry s'adosse à trois ontologies de plus haut niveau, qui sont : SOSA pour la modélisation d'évènements liés à de l'observation, GeoSPARQL et OWL-Time pour les dimensions spatiale et temporelle. Une telle réutilisation de l'existant nous permet de proposer un système de gestion des connaissances ouvert et flexible à même de refléter la complexité des données collectées jusqu'alors en agroforesterie. Le modèle de connaissances est construit de sorte à pouvoir s'interfacer avec d'autres modèles de connaissances déjà rendus disponibles sur le Web. L'expertise en agroforesterie, en lien avec d'autres domaines d'expertise sur le Web, facilitera la création de services sémantiques et d'outils d'aide à la décision, et fournira ainsi des solutions adaptées aux pratiques actuelles de l'agroforesterie.

Mots-clés : Agroforesterie,Système de gestion des connaissances,Ontologie,OWL,Logiques de Description

1 Introduction

Le développement de l'agriculture intensive en Europe conduit, depuis une cinquantaine d'années, à un épuisement graduel des sols (Calame, 2016). Les effets du réchauffement climatique commencent également, à peser sur l'agriculture Européenne (O'Neill *et al.*, 2017). En parallèle, l'augmentation de la population humaine et l'appauvrissement des ressources nécessaires (eau et énergies fossiles en particulier) au fonctionnement de l'agriculture se poursuivent de façon continue. Ces constats nous conduisent à repenser l'agriculture, à la fois pour préserver les ressources et renforcer l'usage de pratiques respectueuses de l'environnement, dans le but de la rendre à la fois compétitive et durable. Parmi les alternatives proposées, l'agroforesterie semble offrir des solutions efficaces pour relever les défis d'une agriculture raisonnée. L'agroforesterie consiste en la réintroduction des arbres dans les métiers de l'agriculture. Ses formes les plus habituelles sont l'agrisylviculture (arbres destinés à la production de bois avec cultures intercalaires (céréales, maïs, orge...)) et le sylvopastoralisme (association sur un même espace d'activités sylvicoles et pastorales), même si d'autres pratiques agricoles peuvent également s'y associer comme l'apiculture ou l'aquaculture. Il

ne s'agit pas d'une simple plantation d'arbres, mais d'une association étroite entre les arbres et l'agriculture afin d'obtenir une réelle synergie, bénéfique pour les deux parties (Dupraz & Liagre, 2008). Cette synergie permet, entre autres choses, la limitation des intrants (eau, engrais et pesticides) dans les pratiques agricoles dans une perspective de développement durable. Si certaines pratiques agroforestières ont pu résister en Europe, au cours des cent dernières années, à l'exemple des prés-vergers de Normandie (Ducros *et al.*, 2005), face aux politiques de mécanisation (l'arbre y est considéré comme un obstacle) et de remembrement, le savoir-faire en agroforesterie a peu à peu disparu au fil des générations. Nous avons donc besoin aujourd'hui d'indicateurs pour gérer au mieux les ressources dans le cadre d'une pratique moderne et intégrée de l'agroforesterie. Les données issues des observations de terrain, acquises en partenariat avec plusieurs catégories d'acteurs (en particulier naturalistes, forestiers, agriculteurs, éleveurs), dans le cadre d'expériences agroforestières s'accumulent depuis plusieurs années déjà (Labelle, 1987). La gestion et la réutilisation de ces données sont rendues difficiles par la multiplicité des supports et des formats utilisés, et par la diversité des acteurs et de leurs vocabulaires métiers. En outre, les études agroforestières nécessitent des approches systémiques pour comprendre, par exemple, comment mieux gérer un site, en réponse au changement climatique, aux ravageurs, ou à la pollution du sol. Ces réponses se doivent de considérer les liens étroits tissés avec d'autres domaines de connaissance tels que la climatologie, la zoologie ou encore la pédologie. Actuellement, à notre connaissance, il n'existe pas de système de capitalisation des données pour décrire les aménagements agroforestiers. Des classifications (Nair, 1985; Sinclair, 1999) pour les systèmes agroforestiers existent bien, et fournissent un cadre pour l'évaluation des aménagements en vue de leur comparaison et amélioration, mais l'agroforesterie ne dispose pas d'une ontologie terminologique stricto sensu (Roche, 2012) spécifique à son domaine, même si un premier travail de construction d'un thésaurus est en cours de réalisation (Burriel *et al.*, 2017). Pour aider la communauté agroforestière dans l'exploitation et le partage de ses données, et afin de rendre compte de l'évolution et de l'efficacité des aménagements déjà en place, nous proposons de construire un système de gestion des connaissances (Alavi & Leidner, 2001) dédié à l'agroforesterie qui réutilise des ontologies standards empruntées au Web sémantique (Davies *et al.*, 2003). Nous proposons ainsi un modèle de connaissances, ou ontologie cadre, nommé "Agroforestry", présenté en section 2, destiné à l'organisation des différents éléments constitutifs des aménagements agroforestiers les plus habituels. L'originalité de notre travail réside dans le fait que la structure du modèle "Agroforestry" s'appuie en grande partie sur l'interconnexion de différentes ontologies et onto-terminologies existantes. L'objectif est de mobiliser l'existant notamment dans le contexte de l'agriculture et des sciences environnementales, et de pouvoir à terme ré-exploiter des jeux de données ouverts et liés issus de ces domaines. Dans la sous-section 2.5, nous illustrons la mise en œuvre du système de connaissances construit, au travers d'un cas d'usage portant sur une parcelle agroforestière du domaine de Restinclières dans le sud de la France.

2 Une ontologie cadre pour l'agroforesterie

2.1 État de l'art

L'ontologie est à l'origine une notion philosophique dérivée de la pensée aristotélicienne, qui signifie "théorie de l'être". En informatique, nous reprendrons la définition donnée par (Uschold & Gruninger, 1996), qui fait de l'ontologie une compréhension partagée d'un domaine d'intérêt. Une ontologie définit donc de manière explicite, consensuelle et formelle, les termes utilisés pour décrire et représenter un champ de connaissances. Si l'ontologie, le thésaurus et la taxonomie ont en commun de fournir un vocabulaire contrôlé où les termes et concepts sont organisés hiérarchiquement, leur utilisation et leur vocation divergent dans le monde de la représentation du savoir. En effet, les taxonomies sont utilisées pour classer des objets, les thésaurus pour indexer et rechercher des informations au sein de documents, et les ontologies apportent une rupture dans la description des connaissances, en permettant de raisonner sur les différentes connaissances d'un domaine, et d'en dégager collectivement de nouveaux savoirs. En droite ligne avec nos besoins, nous distinguerons les ontologies ter-

minologiques, des ontologies cadres. Les ontologies terminologiques, ou ontoterminologies se concentrent plutôt sur la dénomination des concepts d'un domaine, en en fournissant les définitions et les termes linguistiques associés. De fait, les relations entre les termes désignant les concepts sont principalement des relations de généralisation/spécialisation ou encore de proximité sémantique. Les ontologies cadres sont, de leurs côtés, orientées vers la modélisation des notions mentales associées aux concepts et servent donc à restituer la connaissance du domaine. L'accent est alors mis sur l'expression des relations entre les concepts de haut niveau à des fins d'analyse et de manipulation. L'agroforesterie est définie par (Lundgren & Raintree, 1983), comme désignant un ensemble de pratiques pour lesquelles les arbres sont intégrés aux cultures agricoles et/ou à l'élevage, sur une parcelle présentant une certaine forme d'arrangement spatial (et séquentialité temporelle). Le temps se révèle aussi une dimension importante, puisque des rotations de plantation sont également réalisées au travers du temps sur la parcelle considérée. Il est acté (Liniger *et al.*, 1998) qu'il existe un manque de compréhension quantitative et prévisionnelle au sujet des pratiques agroforestières et de leur importance afin d'en faciliter l'adoption. Nous pensons qu'un système qualitatif de gestion de connaissances associées aux aménagements agroforestiers et à leur temporalité fait également défaut. Ce système peut ensuite servir de socle à de la prévision et de la prise de décision concernant les parcelles agroforestières. Une des premières difficultés est d'arriver à prendre en charge les spécificités de l'agroforesterie (Zschocke, 2011), notamment face aux pratiques agricoles plus usuelles dans les pays occidentaux. En effet, il existe différentes initiatives en agriculture qui proposent des modèles conceptuels et des systèmes d'information ou à base de connaissances, avec des objectifs similaires aux nôtres. Ainsi, le consortium ICASA (International Consortium for Agricultural Systems Applications) (White *et al.*, 2013) fournit modèles et standards de données pour l'intégration de données provenant d'expérimentations conduites dans des parcelles agricoles. Le sol, le climat et la réponse des céréales cultivées aux conditions expérimentales sont au cœur de ces modèles. L'objectif est de faciliter les usages autour de la collecte et de l'échange de données, et in fine d'en faciliter l'analyse au travers de systèmes centrés sur de la simulation ou de l'aide à la décision. Le projet GIEA (Gestion des Informations de l'Exploitation Agricole) (Dufy *et al.*, 2006) s'adosse à l'expertise humaine pour identifier les concepts clés relatifs à la communauté de pratique en agriculture et dégager un langage métier commun à l'ensemble des acteurs qui va permettre la compréhension des données échangées. Trois sous-thèmes ont été investis à cet effet, à savoir le sol, l'élevage et l'exploitation. Le projet FOODIE (Farm-Oriented Open Data in Europe) (Palma *et al.*, 2016) s'appuie sur les directives INSPIRE (Infrastructure pour l'Information Spatiale au niveau Européen) qui portent à la fois sur les standards ISO/OGC visant à cadrer la réprésentation de l'information géographique, et des spécifications qui viennent s'y ajouter pour l'agriculture et l'aquaculture [1], pour en décliner une version en langage OWL à partir des diagrammes de classes UML. L'ontologie nommée FOODIE ainsi construite, a été produite par application des règles de conversion définies dans ISO 19150-2 (ISO/TC & Cox, 2014). Si FOODIE a été exploitée pour des besoins en agriculture de précision, il n'en va pas de même pour l'agroforesterie, et certains des concepts définis dans FOODIE ne sont pas pensés pour la structuration d'aménagements mettant en jeu plusieurs cultures. L'exemple du concept "Plot" qui est défini comme étant une "zone agricole continue plantée d'une espèce végétale cultivée" est un exemple de cette difficulté à prendre en charge toute la complexité de l'agroforesterie, et en particulier des interactions pouvant être bénéfiques entre espèces cultivées. FOODIE reste toutefois un travail qui pose des bases très intéressantes pour tout ce qui concerne l'organisation spatiale des parcelles et des zones aménagées au sein de ces parcelles. D'autres travaux (Arenas *et al.*, 2018; Aubin *et al.*, 2019; Tran *et al.*, 2017) menés dans les sciences environnementales ont également retenu notre attention. Il s'agit de travaux qui s'articulent autour de la notion clé d'observation et qui, pour ce faire, font appel à l'ontologie SOSA (Janowicz *et al.*, 2019). En agroforesterie, tout comme en agriculture, la notion d'observation est fondamentale, et va sous-tendre les analyses conduites à partir de mesures collectées à partir de tout élément d'intérêt : plante cultivée en réponse à son environnement, composition du sol, pluviométrie, aménagement agroforestier à l'exemple d'une culture en

1. https ://inspire.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_AF_v3.0.pdf

allée, ou encore interaction entre deux espèces différentes (plante et/ou animal).

2.2 Positionnement

Pour qu'un tel système sémantique, dédié aux aménagements agroforestiers, soit partagé par l'ensemble des acteurs de l'agroforesterie, voire par les acteurs d'autres domaines si l'on considère la nature systémique de l'agroforesterie, il semble essentiel de maintenir un haut niveau de généricité dans la définition de l'ontologie. La vision retenue est la suivante : une parcelle agroforestière évolue dans le temps et dans l'espace, et a pour sous-parties, des éléments biotiques et abiotiques, qui sont envisagés comme des éléments simples ou décomposables, qui possèdent des propriétés intrinsèques (caractéristiques propres), et qui nouent des inter-relations de différentes natures. Nous nous intéressons, en particulier, aux relations spatiales qui sous-tendent les différents aménagements propres à chaque parcelle. Tout élément, propriété ou relation de l'ontologie est potentiellement enrichi par un ou plusieurs termes issus d'ontologies terminologiques. Ces termes utilisés pour qualifier chaque élément d'un aménagement agroforestier, ainsi que leurs propriétés et leurs inter-relations, impliquent la réutilisation d'ontologies terminologiques existantes (AGROVOC (Caracciolo *et al.*, 2013), PO (Plant Ontology) (Jaiswal *et al.*, 2005), PATO (the Phenotype And Trait Ontology) (Gkoutos *et al.*, 2009), ou encore ENVO (the Environment Ontology) (Buttigieg *et al.*, 2013) et vont dans le sens d'une ouverture vers le LOD (Linked Open Data (Janowicz *et al.*, 2014)). En effet, pour faciliter partage et analyse des données, il est impératif de réutiliser autant que possible, les concepts déjà définis par des communautés d'expertise. L'avantage pour nous est double, d'une part la définition de nouveaux concepts, possiblement redondants avec l'existant, est rendue inutile, et d'autre part la mobilisation de concepts définis de manière consensuelle offre une meilleure visibilité sur les données.

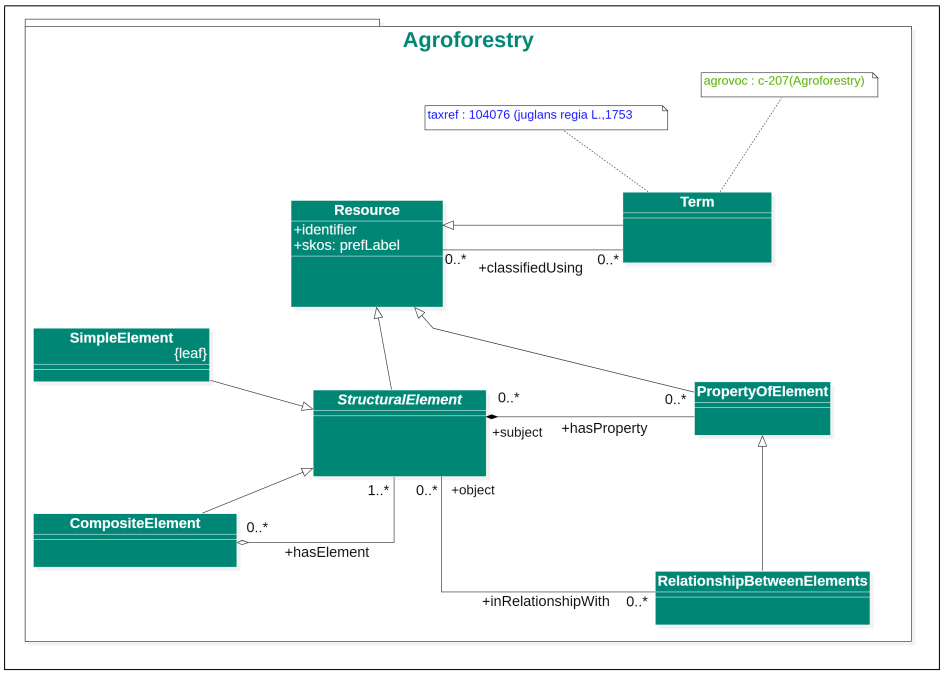


FIGURE *1 – Conceptualisation de l'ontologie "Agroforestry".*

2.3 Une démarche mobilisant l'existant

Pour fournir à notre système les capacités de capitalisation et de partage des connaissances requises, nous avons opté pour une démarche privilégiant l'adossement à différentes ontologies cadres.

D'après Thomas Gruber (Gruber, 1995), l'extensibilité est une qualité importante des ontologies, qui doivent pouvoir se voir ajouter de nouveaux concepts sans que cela ne vienne remettre en question les modélisations antérieures. Dans cette optique, nous définissons différents concepts clés qui sont subsumés par des concepts provenant d'ontologies cadres. Ce mécanisme d'extension nous permet de tirer parti de l'expertise d'autres domaines pour la résolution des problématiques liés à notre modèle de gestion des connaissances en aménagement agroforestier. Nous avons donc mobilisé des ontologies cadres pour résoudre des problématiques déjà résolues dans d'autres domaines de spécialisation. Nous avons fait le choix de présenter les efforts de conceptualisation associés à la construction de l'ontologie Agroforestry, au travers de différents diagrammes UML (Rumbaugh *et al.*, 2004). Un premier diagramme de classes (Fig. 1) décrit les entités clés du modèle Agroforestry, et leurs inter-relations. Les diagrammes suivants restituent graduellement les notions d'importance empruntées aux ontologies cadre GeoSPARQL et SOSA [2]. Un diagramme de paquetages (Fig. 4) montre les liens de dépendance entre composants ontologiques mobilisés. Un dernier diagramme (Fig. 5) explicite l'adossement d'Agroforestry, aux ontologies de plus haut niveau. L'ontologie Agroforestry est ensuite définie dans le formalisme des logiques de description.

Agroforestry est dédiée à une meilleure compréhension des aménagements agroforestiers et à ce titre, se concentre principalement à la description structurelle des entités présentes au sein d'une parcelle. La classe «StructuralElement» occupe en conséquence, une position privilégiée dans cette modélisation. Nous distinguerons les classes filles disjointes «SimpleElement» et «CompositeElement» qui permettent d'énoncer qu'un élément va admettre d'autres éléments structuraux en son sein, ou au contraire être un élément "feuille" du modèle. La classe «CompositeElement» est ainsi enrichie d'une association «hasElement» qui pointe vers la classe «StructuralElement». L'association «hasElement» pourra être affinée dans l'ontologie (en premier lieu via des relations méréo-topologiques). La relation «hasElement» est inspirée des travaux autour de GeoSPARQL et de BOT (Building Ontology Topology) (Rasmussen *et al.*, 2017), et respectivement des différents relations réflexives de geo : SpatialObject vers geo : SpatialObject et de la relation bot : hasElement. Ces éléments de modélisation sont déconnectés de toute considération temporelle. Prenons l'exemple d'un noyer hybride planté dans une parcelle agroforestière donnée. Ce noyer a pour propriété intrinsèque d'être un élément de la parcelle tout au long de son existence. Nous sommes conscients que cette vision simplificatrice peut être sujette à caution dans certains cas de figures. La classe «StructuralElement» se voit associée via l'association «hasProperty» à la classe «PropertyOfElement» qui va permettre d'enrichir les éléments structuraux de diverses caractéristiques amenées à évoluer dans le temps. La propriété «PropertyOfElement» est cette fois-ci réifiée et va être spécialisée en exploitant différents référentiels terminologiques à l'exemple de la taille totale de la plante dans Trait Ontology (TO : 1000012 de nom "whole plant size"). À ce titre, la modélisation s'inspire de SOSA. Nous avons également défini une super-classe nommée «Resource» venant non seulement généraliser «StructuralElement» mais aussi «PropertyOfElement». L'intérêt de cette super-classe est de pouvoir enrichir sémantiquement toute ressource du modèle avec des termes empruntés aux ressources onto-terminologiques rendues disponibles sur le Web. À cet effet, nous introduisons la classe «Term» qui vient également spécialiser la classe «Resource» et dont les instances sont attendues de venir compléter de manière normalisée la description des autres ressources du modèle. Ainsi, les notes, présentes au sein du diagramme de classes UML de la figure 1, font état de termes tirés d'ontologies terminologiques qui peuvent être utilisés pour qualifier les éléments d'un aménagement ainsi que leurs propriétés et inter-relations. Le parti-pris dans

2. Nous exploitons aussi OWL Time, mais nous n'en donnons pas le diagramme UML qui peut être retrouvé à l'adresse `https://www.w3.org/TR/owl-time/`

la définition du modèle de connaissances d'Agroforestry est la réutilisation des ontologies et terminologies existantes. Seul un socle minimum d'organisation de la connaissance vient organiser l'existant afin de rendre cet existant exploitable dans le domaine de l'agroforesterie. Nous détaillons ci-dessous les éléments de modélisation des ontologies cadre mobilisées.

2.3.1 Intégration de l'ontologie SOSA (Sensor, Observation, Sample, and Actuator)

La première nécessité était pour nous de permettre aux agroforestiers de capitaliser les différentes observations faites sur le terrain. Si certaines propriétés de nos éléments agroforestiers possèdent des valeurs immuables (comme par exemple la localisation d'un site), d'autres propriétés (comme l'acidité d'un sol au travers de la mesure de son PH), sont quantifiées ou qualifiées à l'aide d'une procédure d'observation spécifique. Les valeurs collectées au travers de ces observations peuvent varier au cours du temps et sont dépendantes de la méthode de mesure. Cette variabilité doit absolument être prise en compte pour obtenir un résultat rigoureux en cas de comparaison et d'analyse de données issues de multiples observations. À cet effet, nous avons choisi l'ontologie cadre SOSA (Janowicz *et al.*, 2019), qui a pour élément central l'observation (Fig. 2). SOSA est le module central (mais néanmoins autonome) de l'ontologie SSN (Semantic Sensor Network) (Haller *et al.*, 2019) qui permet de décrire les capteurs (y compris humains) et les observations acquises par ces capteurs. Sa simplicité et son autonomie servent de pivot d'interopérabilité pour SSN mais aussi pour d'autres ontologies dédiées à l'observation telles que O&M (Cox, 2017) et OBOE (Madin *et al.*, 2007). SOSA place l'observation («sosa : Observation») au centre de son modèle : Une observation permet ainsi de renseigner la valeur d'une propriété descriptive pour un élément d'intérêt, à un instant ou intervalle de temps donné. L'observation est liée à un individu de la classe «sosa : FeatureOfInterest» par la relation «sosa : hasFeatureOfInterest». La propriété observée de l'élément (classe «sosa : ObservableProperty») est liée à l'observation par la relation «sosa : observedProperty» et la valeur de cette propriété est liée à l'observation par la relation «sosa : hasResult». Dans le modèle Agroforestry, la classe «StructuralElement» vient spécialiser «sosa : FeatureOfInterest», et va naturellement pouvoir bénéficier de toute la modélisation autour de la notion d'observation. Nous pouvons ainsi, par exemple organiser l'information autour de la croissance d'un arbre, en le considérant comme une instance de la classe «sosa : FeatureOfInterest» et renseigner sa taille à intervalles réguliers au travers de plusieurs observations. La classe «ssn : Property» (super-classe de «sosa : ObservableProperty») a également retenu notre attention et nous avons fait le choix d'étendre cette classe par la classe «PropertyOfElement» de manière à pouvoir décrire de manière complémentaire les propriétés engagées dans la description des éléments clés de l'agroforesterie. L'idée est de pouvoir disposer à plus long terme de couples éléments/collection de propriétés pertinents pour l'agroforesterie et ainsi en faciliter la réutilisation par la communauté. Nous avons aussi été motivés par la volonté d'étendre à nouveau «PropertyOfElement» par «RelationshipBetweenElements» pour prendre en charge des relations autres que des relations structurelles entre éléments d'un aménagement agroforestier. Ces relations peuvent être valuées et datées dans le temps, et vont, en particulier, nous permettre de capturer les interactions entre éléments biotiques retrouvés sur les mêmes parcelles. Les sites pilotes en agroforesterie permettrent de mettre en exergue les effets de pratiques agroforestières, au moyen d'expérimentations menées sur des parcelles de test. Les classes «sosa : Sample» et «sosa : Sampling» vont dès lors, nous être également utiles ; une parcelle expérimentale pouvant être vue comme un individu «sosa : Sample». Nous reprenons dans la figure 2, le diagramme de classes UML des principales classes de SOSA dont certaines font l'objet d'extensions dans le modèle Agroforestry.

2.3.2 Extension de l'ontologie GeoSPARQL et de l'ontologie OWL-Time

Une double dimension spatiale et temporelle est nécessaire à toute entité de notre modèle si l'on veut rendre compte de l'évolution d'un système où les relations de proximité dans le temps sont des plus importantes. En effet les aménagements agroforestiers et en particulier ceux de l'agrosylviculture sont de par leur nature des aménagements spatiaux, où la place de

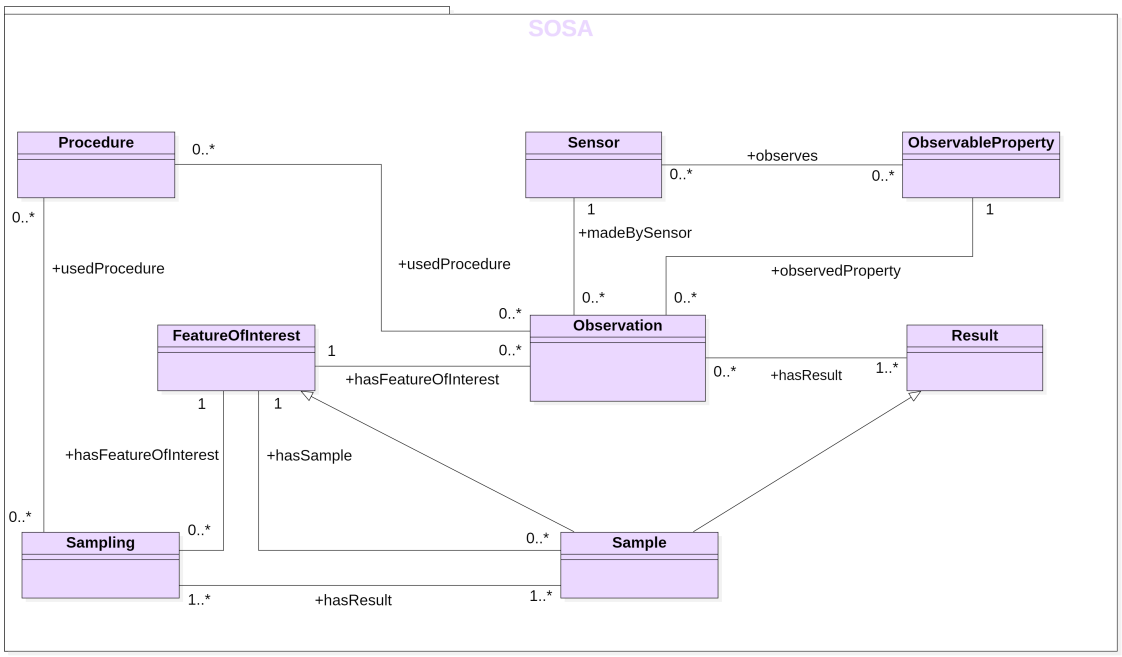


FIGURE 2 – *Diagramme de classes autour du module ontologique SOSA*

chaque élément (arbre, plante céréalière, ligne d'arbres dans son ensemble, ...) a une importance primordiale dans le cadre des interactions plantes-plantes. La localisation géographique des données d'intérêt est à même de rapprocher et de lier des données issues de domaines différents à l'exemple de l'hydrologie ou de la pédologie. Nous prenons l'exemple d'une rivière qui pourrait jouxter une parcelle agroforestière et ainsi limiter grandement la compétition en eau entre arbres et plantes céréalières.

GeoSPARQL (Battle & Kolas, 2012) (Fig.3) est proposé par le consortium international de l'OGC (Open Geospatial Consortium), pour fournir les éléments nécessaires à la représentation et à l'interrogation de données spatialisées dans le contexte du Web sémantique. GeoSPARQL est ainsi organisé de façon modulaire, et peut offrir des services aussi bien aux systèmes basés sur un raisonnement spatial qualitatif, qu'à ceux basés sur des calculs spatiaux quantitatifs. Les calculs spatiaux quantitatifs, à l'exemple d'un calcul de distance, nécessitent de connaître précisément la géométrie des éléments étudiés (géométrie Euclidienne), alors que les systèmes basés sur un raisonnement spatial qualitatif s'appuient sur des relations le plus souvent topologiques entre éléments. Ces dernières relations (à l'exemple de l'adjacence, de l'intersection ou de l'inclusion) sont, par exemple, décrites au travers du formalisme "region connection calculus" (RCC8) (Randell *et al.*, 1992). GeoSPARQL offre également la possibilité d'associer une à plusieurs géométries, à l'exemple de point ou de polygone, à toute entité géographique au travers de la classe «geo : Geometry» qui représente la super classe de toutes les géométries possibles. Dans le modèle Agroforestry, la classe «StructuralElement» est subsumée par «geo : Feature» et nous allons tirer parti de toute la modélisation proposée par GeoSPARQL autour des géométries. La classe «geo : Geometry» du module cœur de GeoSPARQL est par suite étendue par différentes préconisations dont le standard OGC, nommé Simple Features Access (sfa), dont certaines géométries sont illustrées dans le diagramme de classes en figure 3.

Le temps est une notion tout aussi importante dans le contexte des aménagements agroforestiers dans lequel, par exemple, la saisonnalité annuelle des cultures est confrontée à la vie pluriannuelle des arbres. Afin de mieux appréhender les interactions entre arbres et cultures dans le temps, nous devons définir de façon précise les dates de plantation et les périodes

de présence spécifique de chaque élément agroforestier. Ainsi la classe «StructuralElement» subsumée par «sosa : FeatureOfInterest» va pouvoir mobiliser la modélisation retenue dans SOSA, à savoir entretenir une relation de type «sosa : phenomenonTime» avec une entité temporelle «time : TemporalEntity» de l'ontologie OWL-Time (Hobbs & Pan, 2004). La classe «time : TemporalEntity» est ensuite spécialisée en «time : Instant» et «time : Interval», et il est ainsi aisé d'inscrire un élément d'aménagement agroforestier dans une durée basée sur un intervalle de temps précis ; ou bien de définir une estampille temporelle pour une observation spécifique définie pour ce même aménagement agroforestier (Fig.5).
Nous avons présenté à la fois le modèle conceptuel correspondant au socle minimal de struc-

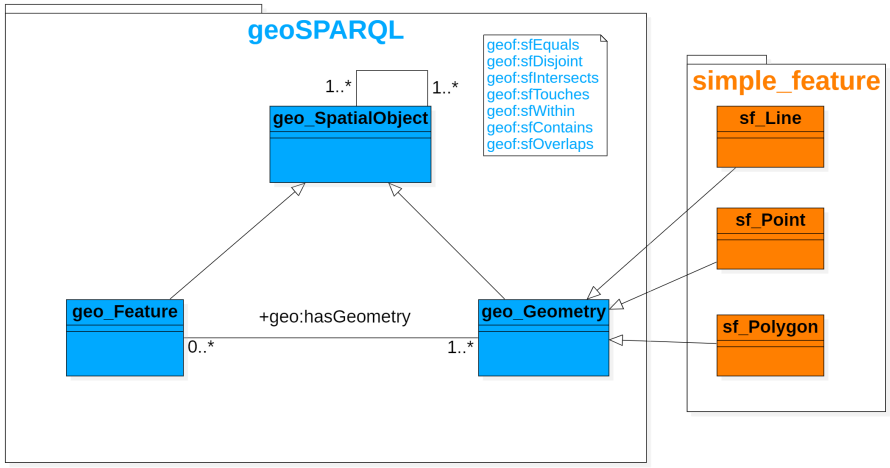


FIGURE *3 – Diagramme de classes autour du module ontologique cœur de GeoSPARQL*

turation des connaissances propres aux aménagements forestiers, et des portions de modèles conceptuels des ontologies cadres véhiculant trois notions qui nous intéressent en tout premier lieu, à savoir l'observation, l'espace et le temps. Nous proposons maintenant d'illustrer les articulations définies entre ces différents modèles au travers de deux nouvelles figures. La figure 5 est un nouveau diagramme de classes intégrant SOSA, GeoSPARQL, OWL-Time et Agroforestry pour les besoins propres des aménagements agroforestiers. Le code couleur permet de repérer visuellement la provenance de chaque élément modélisé. La figure 4 est un diagramme de paquetages qui exprime les liens de dépendance du paquetage «Agroforestry» vis à vis des paquetages «SOSA», «OWL-Time» ou encore «GeoSPARQL». Ce diagramme facilitera la définition des "imports" de composants ontologiques dans Agroforestry.

2.4 Fondements logiques d'Agroforestry

Nous reprenons le socle de connaissances défini spécifiquement pour l'agroforesterie dans le formalisme habituellement dévolu aux logiques de description (Baader *et al.*, 2010; Krötzsch *et al.*, 2014). Le préfixe afy (ontology For AgroForestrY) est le préfixe retenu pour l'ontologie Agroforestry. Les principaux axiomes d'inclusion et d'équivalence de la TBox (boîte terminologique qui contient l'arborescence de concepts de l'ontologie) sont listés en figure Fig.6. Agroforestry se compose essentiellement d'une arborescence de concepts clés, avec «afy : Resource» comme concept le plus général. Les concepts «afy : PropertyOfElement», «afy : Term» et «afy : StructuralElement» viennent raffiner «afy : Resource», et sont disjoints deux à deux. Le concept «afy : Term» découle d'une double motivation : il s'agit d'enrichir les éléments décrits dans une base de connaissances en agroforesterie avec des savoirs terminologiques complémentaires et normalisés, mais aussi et dans un second temps de dégager (en les définissant au besoin) un ensemble de

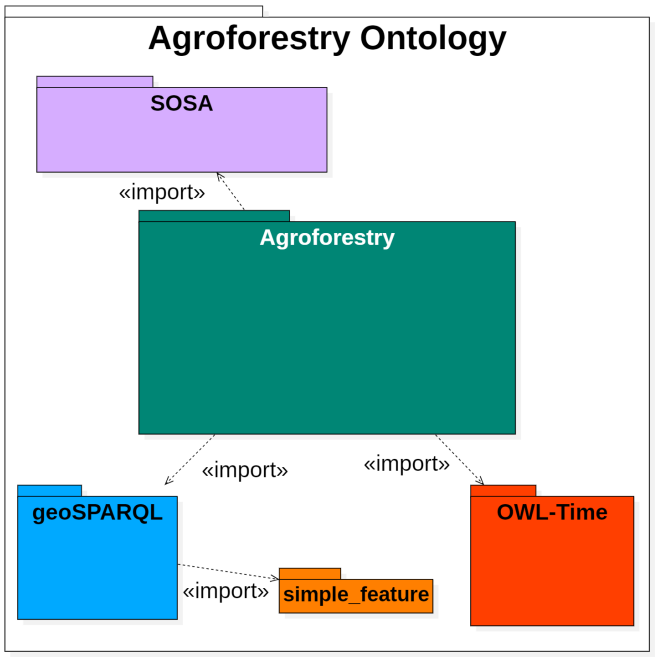


FIGURE *4 – Diagramme de paquetages autour de l'ontologique cadre " Agroforestry ".*

concepts terminologiques pouvant s'appliquer avec succès à l'agroforesterie. Le concept défini «afy : CompositeElement» représente l'ensemble des éléments structuraux comptant au moins une relation de type tout-partie («afy : hasElement») avec tout autre élément structural.
Pour ce qui concerne la RBox (conteneur des rôles de l'ontologie) présentée en figure 7, la propriété «afy : hasElement» a une propriété inverse nommée «afy : elementOf». Par ailleurs, nous posons les axiomes d'inclusion présentés en figure 8 qui viennent adosser Agroforestry aux ontologies GeoSPARQL et SOSA. Ainsi «afy : StructuralElement» , en tant qu'entité spatialisée d'intérêt, spécialise à la fois «geo : Feature» et «sosa : FeatureOfInterest». «afy : PropertyOfElement» dérive de «ssn : Property». «afy : Term» est à rapprocher de «skos : Concept».
L'ontologie ainsi définie, a été exploitée pour tirer parti de premiers mécanismes de raisonnement. La connaissance des arbres qui sont des éléments d'une parcelle peut permettre par exemple d'en calculer la biomasse. À cet effet, l'ontologie a été rendue opérationnelle sous le formalisme OWL2 (Grau *et al.*, 2008; W3C, 2012).

2.5 Instanciation du modèle

Notre objectif est de proposer une démarche qui soit à même de décrire l'organisation d'aménagements agroforestiers actuels ou futurs avec des éléments qui peuvent s'avérer inconnus au moment de la constitution du modèle. Ces nouveaux éléments devront venir compléter le modèle sans le remettre en cause et l'invalider. L'importance est donc de disposer d'une organisation de la connaissance pérenne et extensible, pouvant à la fois être réutilisable dans d'autres contextes d'étude sur le long terme et qui soit facile d'exploitation dans différents cas applicatifs. Mais l'ajout sans contrainte de nouvelles caractéristiques pourrait contribuer à alourdir le modèle avec des problèmes de désignation ou de duplication de ces nouvelles caractéristiques ajoutées au fur et à mesure des besoins. Un manque de clarté dans la définition de ces propriétés pourraient être également un frein à l'analyse des données. Il est indispensable de décrire les entités d'intérêt de l'agroforesterie avec des propriétés décrites au

IC 2020

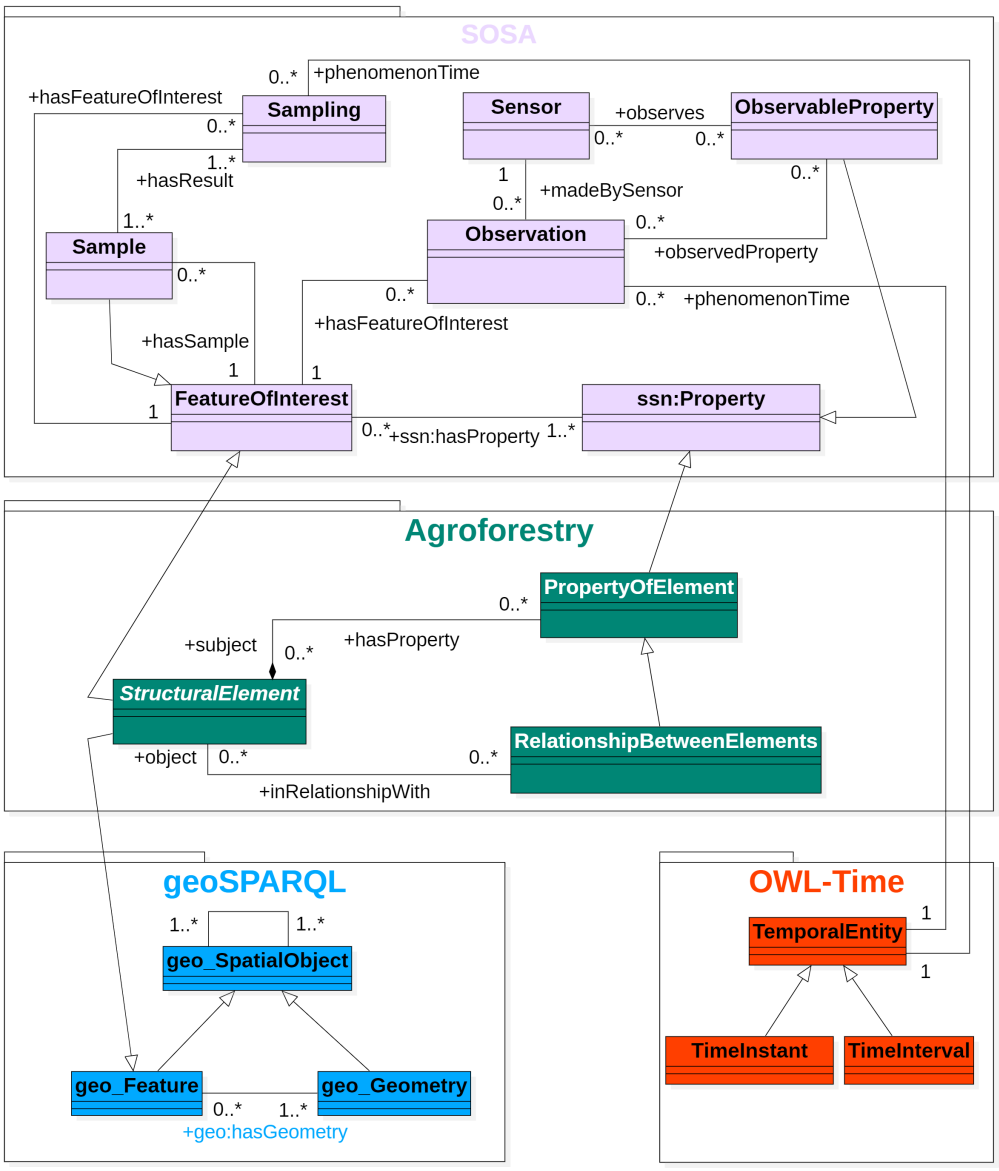


FIGURE *5 – Diagramme de classes intégré*

$\text{afy:Resource} \subseteq \forall\ \text{afy:classifiedUsing.afy:Term} \sqcap \exists\ \text{afy:identifier.xsd:anyURI}$
$\text{afy:PropertyOfElement} \subseteq \text{afy:Resource} \sqcap \neg\ \text{afy:StructuralElement}$
$\text{afy:Term} \subseteq \text{afy:Resource} \sqcap \neg\ \text{afy:StructuralElement}$
$\text{afy:StructuralElement} \subseteq \text{afy:Resource} \sqcap \forall\ \text{afy:hasProperty.afy:PropertyOfElement}$
$\text{afy:CompositeElement} \equiv \text{afy:StructuralElement} \sqcap \exists\ \text{afy:hasElement.afy:StructuralElement}$
$\sqcap \forall\ \text{afy:hasElement.afy:StructuralElement}$
$\text{afy:SimpleElement} \equiv \text{afy:StructuralElement} \sqcap \neg\ \text{afy:CompositeElement}$
$\text{afy:RelationBetweenElements} \equiv \text{afy:PropertyOfElement} \sqcap$
$\exists\ \text{afy:inRelationshipWith.afy:StructuralElement}$

FIGURE *6 – Concepts d'Agroforestry*

$\text{afy:hasElement} \equiv \text{afy:elementOf}^{-}$
$\text{afy:hasProperty} \equiv \text{afy:propertyOf}^{-}$
$\top \subseteq \leq 1\ \text{afy:propertyOf}$
$\text{afy:hasRelationship} \equiv \text{afy:inRelationshipWith}^{-}$
$\top \subseteq \leq 1\ \text{afy:inRelationshipWith}$
$\text{afy:classifiedUsing} \equiv \text{afy:classifiedBy}^{-}$
$\top \subseteq \leq 1\ \text{afy:identifier}$

FIGURE *7 – Propriétés d'Agroforestry*

sein de terminologies normalisées qui soient à la fois décrites de manière non ambiguë, et qui fassent consensus pour l'ensemble des différents acteurs de l'agroforesterie. À cet effet, nous privilégions l'usage de termes provenant de plusieurs ressources onto-terminologiques de domaines connexes, à l'exemple d'Agrovoc, Thesae, AgronomicTaxon, TaxRef ou encore AEO (Agricultural Experiments Ontology). Un travail mené en parallèle concerne la construction d'un thésaurus dédié à l'agroforesterie qui soit rendu disponible au format RDF (Cyganiak *et al.*, 2014) au sein de la sphère des sources de données ouvertes et liées. Certaines notions peuvent en effet être très spécifiques à l'agroforesterie à l'exemple de pratiques agriculturales exclusives à l'agroforesterie, et ne vont pas être retrouvées dans les thésaurus existants. Un premier travail de construction d'un thésaurus dédié à l'agroforesterie (Burriel *et al.*, 2017) a été initié dans le contexte du projet européen AgroFe (Agroforestry Education in Europe) et poursuivi dans le projet Agrof-MM Erasmus+ [3] (Várallyai *et al.*, 2017). Ce thésaurus traite cinq grands volets : services écosystémiques, économie, écologie, systèmes agroforestiers et techniques agroforestières.

2.5.1 Exemples illustratifs

Nous présentons ici des exemples d'individus venant peupler la base de connaissances. Ces individus sont extraits du jeu de données acquis à partir des expérimentations menées sur les parcelles agroforestières, agricoles et forestières du site d'étude de Restinclières (Dufour, 2019). À Restinclières, des parcelles témoins sont également présentes pour comparer la pousse des arbres sans cultures (témoins forestiers) et celle des cultures sans les arbres (témoins agricoles). Les données utilisées pour le cas d'étude ont été collectées entre 2015 et 2018, par des chercheurs de l'INRAE. Les exemples illustrent graduellement les trois axes de modélisation retenus : à savoir représenter la structure des aménagements sur les parcelles, organiser l'ensemble des observations acquises sur les éléments d'intéret présents sur la parcelle et enrichir les connaissances acquises par l'ajout de termes provenant d'onto-terminologies qui vont faciliter la compréhension et donc le partage de la base de connaissances. L'objectif est de transférer la démarche afin de la rendre opérationnelle pour tous les acteurs de l'agroforesterie. Dans le graphe RDF de la figure 9, cinq instances de la classe «afy : StructuralElement» entretiennent des relations tout-partie («obo : BFO_0000051» venant spécialiser «afy : hasElement»). Une parcelle («afy : PA3AF») a pour sous-partie une

3. `http://agrofmm.eu`

afy : StructuralElement ⊆ geo : Feature
afy : StructuralElement ⊆ sosa : FeatureOfInterest
afy : PropertyOfElement ⊆ ssn : Property

FIGURE *8 –* ***"Agroforestry".*** *Adossement aux ontologies de plus haut niveau.*

rangée d'arbres «afy : PA3AFL01» qui a, à son tour, pour sous-parties des arbres dont les arbres numéro 10 et 13 «afy : PA3AFL01A10» et «afy : PA3AFL01A13». Chaque instance se voit associer un label («skos : prefLabel») en anglais. Par exemple, l'arbre 10 a pour label "Tree 10 of line 1"@en. Les individus de la classe «afy : StructuralElement» sont également

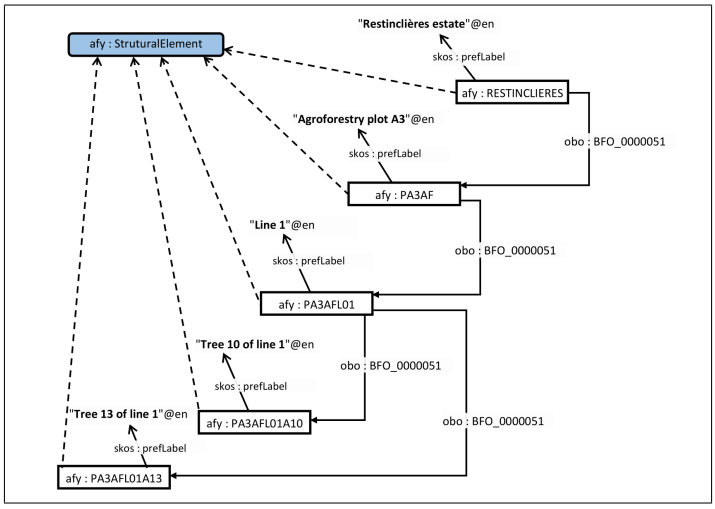


FIGURE *9 – Graphe RDF explicitant l'aménagement de la parcelle agroforestière de Restinclières*

des individus de la classe «geo : Feature» et à ce titre vont admettre des géométries. Dans la figure 10, l'arbre 10 de la rangée 1 possède une géométrie de type «sf : Point», tandis que la rangée possède une géométrie de type «sf : Polygone». Les coordonnées géographiques sont données dans le système WKT et le référentiel géographique WGS 84. La spatialisation des entités permet de considérer tous les liens qui peuvent se nouer entre ces entités. Ainsi les relations de proximité pourront être posées et faciliteront l'étude des interactions plante-plante.

Un dernier exemple porte sur l'acquisition d'observations pouvant être mesurées sur les différentes entités constitutives de la parcelle. Nous prenons l'exemple de la mesure du rendement de la culture de blé en 2015 sur la parcelle agroforestière étudiée. L'estimation de ce rendement se base sur un échantillonnage obtenu à partir de prélèvements de culture avant récolte, réalisés à différents points sur la parcelle. La figure 11 illustre la mise à contribution de la classe Sampling pour traiter les échantillonnages, et le graphe RDF présenté, correspond à un échantillonnage réalisé le 28 Juin 2015 sur la parcelle agroforestière PA3AF. L'observation porte sur les échantillons «afy : PA3AF_Sample_wheat_2017_L1-2_A34_Est» et «afy : PA3AF_Sample_wheat_2017_L1-2_A34_Est» qui sont les résultats de l'échantillonnage «afy : PA3AF_Sampling_wheat_2017». La ressource «agrovoc : c_10176» avec pour label anglais «crop yield» fait office de propriété observée et le protocole de mesure correspond à un contrôle de performance («agrovoc : c_24061 » avec pour label anglais «performance testing»). L'identité du scientifique ayant joué le rôle de capteur humain est consigné ainsi que la date de l'échantillonnage. Les résultats sont donnés en kilogramme par hectare. Les usages concertés du modèle SOSA et de termes empruntés au vocabulaire Agrovoc facilitent le partage de la démarche et permettent d'envisager l'ouverture du jeu de données ainsi structuré à d'autres équipes de recherche. Nous avons exploité ce jeu de données pour étudier les

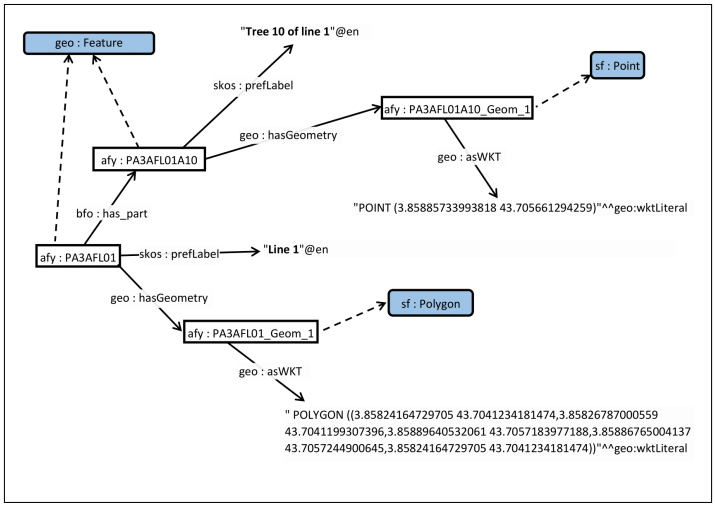


FIGURE *10 – Graphe RDF explicitant les géométries des éléments structuraux de la parcelle agroforestière*

retombées économiques d'une parcelle agroforestière et estimer la rentabilité de cette parcelle lorsque que des arbres, qui ont une valeur marchande en matière de bois d'œuvre, y poussent. La parcelle agroforestière possède une parcelle forestière témoin et des calculs de biomasse moyenne, à partir de la hauteur et du diamètre de chaque arbre modélisé dans la base de connaissances, ont été effectués pour trois années consécutives (2015, 2016 et 2017). Il est montré, après une validation statistique des résultats, que la biomasse dans la parcelle agroforestière y est supérieure à la biomasse calculée dans la parcelle forestière. Cet exemple n'est qu'un premier exemple à mettre à l'actif de l'agroforesterie qui peut apporter des solutions en matière de développement durable face à l'érosion de la biodiversité et aux changements climatiques. Nous avons utilisé, pour le mettre en œuvre, la plateforme pour le Web sémantique Java Jena (The Apache Software Foundation, 2011) et construit des fonctions Jena faisant des appels à des librairies du langage de traitement statistique R (R Core Team, 2012).

3 Conclusion et perspectives

Nous proposons une ontologie nommée "Agroforestry" dédiée à la description spatio-temporelle de structures composites que sont les aménagements agroforestiers. La volonté est de privilégier la réutilisation d'ontologies existantes. À ce titre, nous avons mobilisé des composants ontologiques préconisés par des consortiums œuvrant pour la standardisation des modèles de connaissances. Ainsi GeoSPARQL, pour la dimension spatiale est à l'initiative de l'OGC ; OWL-Time pour la dimension temporelle est à l'initiative du W3C ; de même que SOSA pour tout ce qui découle des observations. Différents travaux (Arenas *et al.*, 2018; Aubin *et al.*, 2019; Tran *et al.*, 2017) ont d'ores et déjà expérimenté tout le potentiel d'un usage commun de SOSA, GeoSPARQL et Time dans les sciences de l'environnement. Nous nous attachons ici à conduire un travail similaire dans la sphère de l'agroforesterie, qui nécessite de modéliser et de capitaliser des connaissances faisant la part belle aux interactions entre éléments biotiques et/ou abiotiques. Nous dégageons également un socle minimal de connaissances qui articule les apports de chaque composant ontologique autour de trois axes : structuration des aménagements, observations et dispositifs expérimentaux autour des éléments clés présents dans les aménagements, et enrichissements terminologiques des éléments décrits. Ce modèle pensé pour l'agroforesterie, reste cependant générique pour intégrer des connaissances en provenance de domaines connexes à l'agroforesterie, à l'exemple de la climatologie, de la pédologie, de l'hydrologie mais aussi de l'agriculture. Nos perspectives

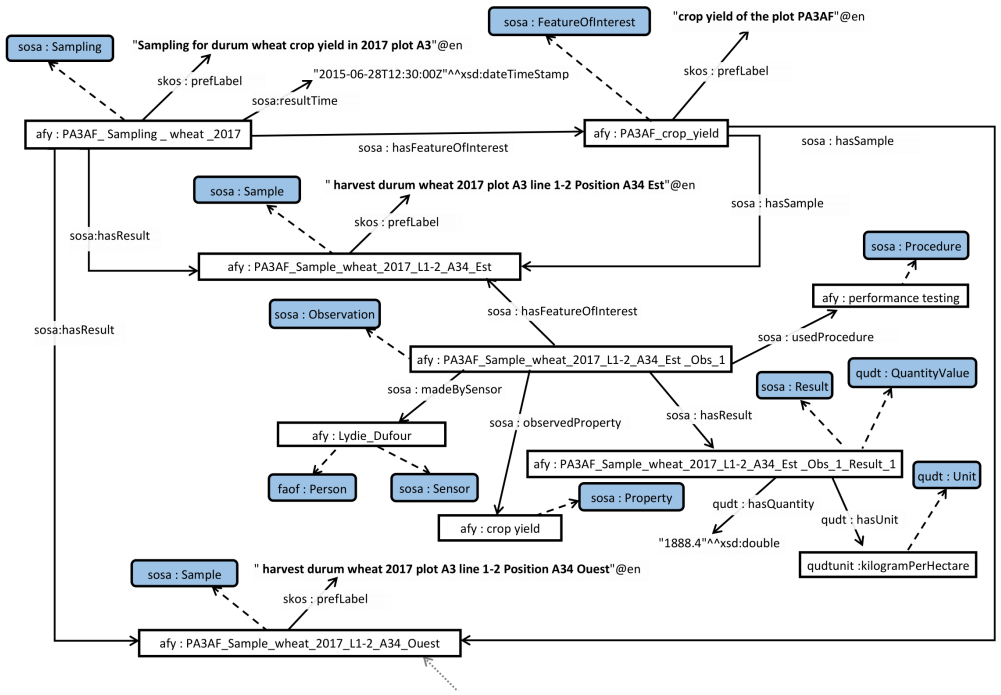


FIGURE *11 – Graphe RDF et définition d'observations s'appliquant aux éléments structuraux*

sont donc d'étendre notre démarche à ces domaines de connaissances connexes afin de permettre une meilleure intégration des données systémiques et ouvrir des perspectives en termes de modélisation des systèmes complexes en agroforesterie. Nous souhaitons également poursuivre dans la définition de cas d'usages pouvant être servis efficacement par l'ontologie Agroforestry, et en particulier traiter de tout ce qui relève des interactions entre les arbres et les cultures venant s'intercaler entre ces arbres. Mieux comprendre ces interactions pourrait permettre de définir des politiques agricoles plus respectueuses de l'environnement. Enfin, il nous semble utile également de poursuivre le travail engagé par d'autres groupes de travail autour de la définition d'un thésaurus dédié à l'agroforesterie, en portant ce thésaurus au sein de la sphère des sources de données ouvertes et liées. Une version de ce thésaurus au format SKOS (Miles & Bechhofer, 2009) déposé sur un portail de ressources onto-terminologiques à l'exemple d'AgroPortal (Jonquet *et al.*, 2018) pourrait être un premier pas dans ce sens.

Références

ALAVI M. & LEIDNER D. E. (2001). Review : Knowledge management and knowledge management systems : Conceptual foundations and research issues. *MIS Quarterly*, **25**(1), 107.

ARENAS H., AUSSENAC-GILLES N., COMPAROT C. & TROJAHN C. (2018). Ontologie pour l'intégration de données d'observation de la terre et contextuelles basée sur les relations topologiques. In S. RANWEZ, Ed., *IC 2018 : 29es Journées francophones d'Ingénierie des Connaissances (Proceedings of the 29th French Knowledge Engineering Conference), Nancy, France, July 4-6, 2018*, p. 5–20.

AUBIN S., BISQUERT P., BUCHE P., DIBIE J., IBANESCU L., JONQUET C. & ROUSSEY C. (2019). Recent progresses in data and knowledge integration for decision support in agri-food chains. In N. HERNANDEZ, Ed., *IC 2019 - Journées francophones d'Ingénierie des Connaissances*, p. 43–59, Toulouse, France : AFIA.

BAADER F., CALVANESE D., MCGUINNESS D. L., NARDI D. & PATEL-SCHNEIDER P. F. (2010). *The Description Logic Handbook : Theory, Implementation and Applications*. USA : Cambridge University Press, 2nd edition.

BATTLE R. & KOLAS D. (2012). Enabling the geospatial Semantic Web with Parliament and GeoSPARQL. *Semantic Web*, **3**(4), 355–370.

BURRIEL C., HERDON M., TAMÁS J. & VÁRALLYAI L. (2017). Knowledge databank and repository service for agroforestry. In *2017 EFITA WCCA Congress*.

BUTTIGIEG P. L., MORRISON N., SMITH B., MUNGALL C. J., LEWIS S. E. & THE ENVO CONSORTIUM (2013). The environment ontology : contextualising biological and biomedical entities. *Journal of Biomedical Semantics*, **4**(1), 43.

CALAME M. (2016). *Comprendre l'agroécologie : Origines, principes et politiques*. Paris : Mayer Charles Leopold Eds.

CARACCIOLO C., STELLATO A., MORSHED A., JOHANNSEN G., RAJBHANDARI S., JAQUES Y. & KEIZER J. (2013). The AGROVOC Linked Dataset. *Semantic Web*, **4**(3), 341–348.

COX S. J. D. (2017). Ontology for observations and sampling features, with alignments to existing models. *Semantic Web*, **8**(3), 453–470.

CYGANIAK R., WOOD D. & LANTHALER M. (2014). RDF 1.1 Concepts and Abstract Syntax. https ://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. [Online ; accessed 19-feb-2020].

DAVIES J., FENSEL D. & VAN HARMELEN F. (2003). *Towards the Semantic Web*. USA.

DUCROS D., KÉPHALIACOS C. & RIDIER A. (2005). Systèmes de production diversifiés. les prés-vergers : une alternative à l'arboriculture intensive face à l'évolution de la pac.

DUFOUR L. (2019). Tree planting and management in agroforestry. In *Agroforestry for sustainable agriculture*, p. 480 p. Burleigh Dodds Science Publishing.

DUFY L., ABT V. & POYET P. (2006). GIEA : gestion des informations de l'exploitation agricole. un projet au service de l'interopérabilité sémantique de la profession agricole. In *Ingénieries eau-agriculture-territoires,Lavoisier*, p. 27–36.

DUPRAZ C. & LIAGRE F. (2008). *Agroforesterie : Des arbres et des cultures*. Paris : Ed. France Agricole, 1.ed. edition.

GKOUTOS G. V., MUNGALL C., DOLKEN S., ASHBURNER M., LEWIS S., HANCOCK J., SCHOFIELD P., KOHLER S. & ROBINSON P. N. (2009). Entity/quality-based logical definitions for the human skeletal phenome using PATO. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, p. 7069–7072.

GRAU B. C., HORROCKS I., MOTIK B., PARSIA B., PATEL-SCHNEIDER P. & SATTLER U. (2008). Owl 2 : The next step for owl. *Journal of Web Semantics*, **6**(4), 309–322.

GRUBER T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing ? *International journal of human-computer studies*, **43**(5-6), 907–928.

HALLER A., JANOWICZ K., COX S. J. D., MAXIME L., TAYLOR K., LE PHUOC D., LIEBERMAN J., GARCÍA-CASTRO R., ATKINSON R. & STADLER C. (2019). The modular SSN ontology : A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, **10**(1), 9–32.

HOBBS J. R. & PAN F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing*, **3**(1), 66–85.

ISO/TC & COX S. (2014). *ISO/DIS 19150-2 :2014 - Geographic information – Ontology – Part 2 : Rules for developing ontologies in the Web Ontology Language (OWL)*.

JAISWAL P., AVRAHAM S., ILIC K., KELLOGG E. A., MCCOUCH S., PUJAR A., REISER L., RHEE S. Y., SACHS M. M., SCHAEFFER M., STEIN L., STEVENS P., VINCENT L., WARE D. & ZAPATA F. (2005). Plant ontology (po) : a controlled vocabulary of plant structures and growth stages. *Comparative and Functional Genomics*, **6**(7-8), 388–397.

JANOWICZ K., HALLER A., COX S. J. D., LE PHUOC D. & LEFRANÇOIS M. (2019). SOSA : A lightweight ontology for sensors, observations, samples, and actuators. *J. Web Semant.*, **56**, 1–10.

JANOWICZ K., HITZLER P., ADAMS B., KOLAS D. & VARDEMAN C. (2014). Five stars of linked data vocabulary use. *Semantic Web*, **5**(3), 173–176.

JONQUET C., TOULET A., ARNAUD E., AUBIN S., DZALÉ YEUMO KABORÉ E., EMONET V., GRAYBEAL J., LAPORTE M., MUSEN M. A., PESCE V. & LARMANDE P. (2018). AgroPortal : A vocabulary and ontology repository for agronomy. *Computers and Electronics in Agriculture*, **144**, 126–143.

KRÖTZSCH M., SIMANCIK F. & HORROCKS I. (2014). Description Logics. *IEEE Intelligent Systems*, **29**(1), 12–19.

LABELLE R. (1987). Ten years of work in agroforestry information and documentation. *Agroforestry systems*, **5**(3), 339–352.

IC 2020

LINIGER H., GICHUKI F. N., KIRONCHI G. & NJERU L. (1998). Pressure on land : the search for sustainable use in a highly diverse environment. *Eastern and Southern Africa Geographical Journal*, (8), 29–44.

LUNDGREN B. & RAINTREE J. (1983). *Sustained agroforestry*. ICRAF Nairobi.

MADIN J., BOWERS S., SCHILDHAUER M., KRIVOV S., PENNINGTON D. & VILLA F. (2007). An ontology for describing and synthesizing ecological observation data. *Ecological Informatics*, **2**(3), 279–296.

MILES A. & BECHHOFER S. (2009). *SKOS Simple Knowledge Organization System : Reference*. Recommendation, W3C. [Online ; accessed 22-feb-2020].

NAIR P. R. (1985). Classification of agroforestry systems. *Agroforestry systems*, **3**(2), 97–128.

O'NEILL B. C., OPPENHEIMER M., WARREN R., HALLEGATTE S., KOPP R. E., PÖRTNER H. O., SCHOLES R., BIRKMANN J., FODEN W., LICKER R., MACH K. J., MARBAIX P., MASTRANDREA M. D., PRICE J., TAKAHASHI K., VAN YPERSELE J.-P. & YOHE G. (2017). IPCC reasons for concern regarding climate change risks. *Nature Climate Change*, **7**(1), 28–37.

PALMA R., REZNIK T., ESBRÍ M., CHARVAT K. & MAZUREK C. (2016). An inspire-based vocabulary for the publication of agricultural linked data. In V. TAMMA, M. DRAGONI, R. GONÇALVES & A. ŁAWRYNOWICZ, Eds., *Ontology Engineering, OWLED 2015*, p. 124–133, Cham : Springer International Publishing.

R CORE TEAM (2012). *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

RANDELL D., CUI Z. & COHN A. (1992). A spatial logic based on regions and connection. p. 165–176.

RASMUSSEN M. H., PAUWELS P., LEFRANÇOIS M., SCHNEIDER G. F., HVIID C. A. & KARLSHØJ J. (2017). Recent changes in the Building Topology Ontology. In *LDAC2017 - 5th Linked Data in Architecture and Construction Workshop*.

ROCHE C. (2012). Ontoterminology : How to unify terminology and ontology into a single paradigm.

RUMBAUGH J., JACOBSON I. & BOOCH G. (2004). *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education.

SINCLAIR F. L. (1999). A general classification of agroforestry practice. *Agroforestry systems*, **46**(2), 161–180.

THE APACHE SOFTWARE FOUNDATION (2011). A free and open source Java framework for building Semantic Web and Linked Data applications. https ://jena.apache.org/. [Online ; accessed 22-feb-2020].

TRAN B., PLUMEJEAUD-PERREAU C., BOUJU A. & BRETAGNOLLE V. (2017). Système d'information spatiotemporel pour l'intégration et l'exploitation de données environnementales. *Revue Internationale de Géomatique*, **27**(3), 423.

USCHOLD M. & GRUNINGER M. (1996). Ontologies : Principles, methods and applications. *The Knowledge Engineering Review, Special Issue on Putting Ontologies to Use*, **11**(2).

VÁRALLYAI L., HERDON M., BURRIEL C. & BOTOS S. (2017). Agroforestry usage, building knowledge databank for agroforestrytraining and education databank and repository service for agroforestry. In *8th International Conference on Information and Communication Technologies in Agriculture, Food & Environment*, p. 54–65.

W3C (2012). OWL 2 Web Ontology Language. https ://www.w3.org/TR/owl2-overview/. [Online ; accessed 19-feb-2020].

WHITE J., HUNT L., BOOTE K., JONES J., KOO J., KIM S., PORTER C., WILKENS P. & HOOGENBOOM G. (2013). Integrated description of agricultural field experiments and production : The ICASA Version 2.0 data standards. *Computers and Electronics in Agriculture*, **96**, 1–12.

ZSCHOCKE T. (2011). Subject Classification with DITA Markup for Agricultural Learning Resources : A Case Example in Agroforestry. In E. GARCÍA-BARRIOCANAL, Z. CEBECI, M. C. OKUR & A. ÖZTÜRK, Eds., *Metadata and Semantic Research*, p. 500–513, Berlin, Heidelberg : Springer Berlin Heidelberg.

DAGOBAH : Un système d'annotation sémantique de données tabulaires indépendant du contexte

Yoan Chabot[1], Thomas Labbé[1], Jixiong Liu[1,2], Raphaël Troncy[2]

[1] Orange Labs, France
yoan.chabot|thomas.labbe@orange.com

[2] EURECOM, Sophia Antipolis, France
raphael.troncy@eurecom.fr

Résumé : Cet article présente le système DAGOBAH permettant d'annoter sémantiquement des tables à l'aide d'entités Wikidata et DBPedia. Le système proposé annote les cellules et les colonnes d'une table et identifie des relations entre ces colonnes. Pour cela, un processus allant du pré-traitement des tables jusqu'à l'enrichissement d'un graphe de connaissances existant en utilisant uniquement les informations contenues dans la table est utilisé. Pour répondre au problème spécifique du typage des colonnes des tables, deux techniques sont introduites pour sélectionner des types suffisamment représentatifs tout en restant à un niveau de spécificité porteur d'informations. Les types ainsi identifiés sont ensuite utilisés dans un processus de désambiguisation des cellules. Le système a été évalué lors du challenge SemTab2019 (Jiménez-Ruiz *et al.*, 2020) de la conférence ISWC 2019 où il a obtenu des résultats prometteurs (Chabot *et al.*, 2019b).

Mots-clés : données tabulaires, graphe de connaissances, annotation sémantique, plongement, DAGOBAH, SemTab2019.

1 Introduction

L'annotation de tables à l'aide de graphes de connaissances est un problème crucial et complexe. Un grand nombre de gisements de données internes aux entreprises ainsi qu'une part importante des données du Web sont représentés sous forme de tables. La capacité à annoter ces dernières en utilisant un graphe de connaissances (encyclopédiques comme DBPedia et Wikidata ou contenant des informations d'entreprises) permet l'intégration de nouveaux services basés sur la sémantique. Cela ouvre notamment la voie à des outils plus efficaces pour l'interrogation (e.g. "aller au delà de simples mots-clefs" pour la recherche de jeux de données (Chapman *et al.*, 2019)), la manipulation et le traitement de corpus de tables hétérogènes (Chabot *et al.*, 2019a). Les données tabulaires sont toutefois difficiles à interpréter automatiquement, principalement à cause du manque de contexte disponible (par opposition aux textes en langage naturel) pour résoudre les ambiguïtés (Table 1). A ce problème s'ajoutent des hétérogénéités syntaxiques souvent mêlées à des artefacts typographiques (Table 2), mais aussi des dispositions et formats de tables parfois difficiles à appréhender.

Pour répondre à ce besoin, nous proposons un système d'annotation de données tabulaires automatique allant du pré-traitement jusqu'à l'identification d'entités et de relations. Afin de garantir sa généricité, ce système utilise uniquement les informations contenues dans les tables et ne prend donc pas en compte les titres et légendes ou d'éventuelles pages englobantes. Les principales contributions du système DAGOBAH présentées dans cet article sont :

— Une nouvelle chaîne de pré-traitement améliorant les résultats du framework DWTC (Eberius *et al.*, 2015).
— Un processus d'annotation automatique en trois étapes (annotation cellule-entité, annotation colonne-type et annotation colonnes-propriétés) utilisant les bases de connaissances Wikidata et DBPedia.
— Une approche alternative utilisant un partitionnement sur des plongements du graphe de connaissances Wikidata.

TABLE *1 – Extrait d'une table métier*

id	name	duration	genre[]	act[].familyName	act[].givenName	price.value
3540	Hector - HD	4980	drame	Solemani , Allen	Sarah , Keith	
296x	Les faussaires	5640	thriller , drame	Markovics , Diehl	Karl , August	5.99
8801	Les bien-aimés - HD	7920	comédie dramatique	Deneuve , Mastroianni	Catherine , Chiara	2.99
01j6	The village green	600	Série-Humour	Pagett , Walker	Daniel , Tim	

TABLE *2 – Extrait de la table* 54719588_0_8417197176086756912 *du challenge SemTab2019*

New ?	Title	Director	Year	Grade
66.66%		class=xl36 style='mso-ignore :colspan'>	1992.82	2.78
1	Big Momma's House	Gosnell	2000	
	Pi	Aronofsky	1998	A
	Fight Club	Fincher	1999	A-
2	Keeping The Faith	Norton ?	2000	
3	Royal Tenenbaums	Anderson	2001	B

La suite de cet article est structurée de la manière suivante. Un état de l'art des travaux dans le domaine de l'annotation de données tabulaires est proposé dans la section 2. Le système DAGOBAH est ensuite présenté dans la section 3. Une évaluation du système sur le challenge SemTab2019 est ensuite exposée dans la section 4. Enfin, une synthèse sur les capacités et les limites de l'approche ainsi que les travaux futurs viennent conclure le papier en section 5.

2 État de l'art

Le problème d'annotation sémantique peut être divisé en trois tâches (Jiménez-Ruiz *et al.*, 2020) : l'annotation cellule-entité (CEA), l'annotation colonne-type (CTA) et l'annotation colonnes-propriété (CPA).

Concernant les correspondances d'entités (CEA), deux approches sont utilisées dans la littérature. La première, largement répandue, consiste à trouver une correspondance pour chaque cellule d'une table donnée (Limaye *et al.*, 2010) (Bhagavatula *et al.*, 2015) (Kilias *et al.*, 2018) (Fernandez *et al.*, 2018). La seconde, quant à elle, tente de faire correspondre une ligne entière d'une table et une entité du graphe de connaissances cible (Efthymiou *et al.*, 2017) en partant du principe qu'une ligne représente une entité principale (colonne clé) et des attributs associés (autres colonnes). Ces correspondances peuvent être réalisées à l'aide de recherches syntaxiques (c'est à dire par comparaison de chaînes de caractères), d'alignement d'ontologies ou en exploitant des plongements lexicaux. La désambigüisation des entités candidates est ensuite traitée comme une tâche de classement et de décision classique, via l'utilisation d'heuristiques ou d'algorithmes de type PageRank (Efthymiou *et al.*, 2017), de mesure de similarité (Kilias *et al.*, 2018) (Fernandez *et al.*, 2018) ou de modèles à base de graphes (Ibrahim *et al.*, 2016).

Les principaux travaux sur le typage des colonnes (CTA) se basent sur l'inférence des classes à partir des entités issues de la tâche de CEA. Différents algorithmes existent, intégrant des heuristiques plus ou moins complexes orientées autour du vote majoritaire (Mulwad *et al.*, 2010). Enfin, l'extraction de relations (CPA) est généralement réalisée par correspondance de paires d'éléments de colonnes au regard des types et entités précédemment déterminés (Ran *et al.*, 2016).

Si les approches précédentes sont intrinsèquement séquentielles, on peut noter quelques travaux qui ambitionnent de réaliser conjointement les trois tâches via des modèles orientés graphes (Limaye *et al.*, 2010) (Zhang, 2014).

En parallèle, des efforts ont été réalisés par la communauté sur la mise à disposition de corpus d'évaluation pour ces différentes tâches. Si le premier corpus portait sur 428 tables Wikipédia (Limaye *et al.*, 2010), les suivants ont ensuite été étoffés soit en volumétrie (WTC (Lehmberg *et al.*, 2016) : 233 millions de tables réparties en trois catégories), soit en précision (T2D Gold Standard (Ritze *et al.*, 2015) : 1748 tables issues du WTC dont les lignes, les attributs et les tables ont été annotés manuellement avec des instances, propriétés et classes DBpedia respectivement). Certains corpus ont également été personnalisés pour répondre à

un besoin particulier (Wikipedia Gold Standard (Efthymiou *et al.*, 2017)) mais ne font pas à ce jour office de référence d'évaluation au sein de la communauté.

Plus récemment, le challenge SemTab2019 proposait à des équipes de comparer les performances de leurs systèmes d'annotation de données tabulaires pour les trois tâches mentionnées précédemment. Le challenge s'est déroulé en quatre manches (Jiménez-Ruiz *et al.*, 2020) proposant des corpus différents par leur taille (respectivement 63, 11925, 2162 et 818 tables) et leur nature avec des tables de plus en plus complexes tant au niveau des formats que des informations à traiter (Hassanzadeh *et al.*, 2019). Dix sept équipes ont participé au challenge avec sept d'entre elles ayant participé à au moins deux des quatre manches du challenge, dont le système DAGOBAH présenté dans cet article. MTab (Nguyen *et al.*, 2019) propose l'utilisation d'une approche probabiliste couplée à des requêtes sur plusieurs services de recherche sur les bases DBpedia, Wikidata et Wikipedia en utilisant des stratégies multilingues. Une majorité des approches (Cremaschi *et al.*, 2019), (Morikawa, 2019), (Oliveira & D'Aquin, 2019), (Thawani *et al.*, 2019), (Steenwinckel *et al.*, 2019) prennent la forme de système à base de recherche de candidats dans les services précédents, de calcul de similarité syntaxique et de votes majoritaires, telle que notre approche de base décrite en section 3.2. IDLab propose une approche itérative où l'annotation de cellules ayant peu d'ambiguïté vient renforcer au fur et à mesure des itérations la désambiguisation des cellules plus complexes (Steenwinckel *et al.*, 2019). Tabularisi utilise les alias des entités définies dans Wikidata et s'assure que les colonnes ont une cohérence sémantique à l'issu du processus d'annotation (Thawani *et al.*, 2019). ADOG créé un index de DBpedia dans la base de données NoSQL ArrangoDB et utilise essentiellement la distance de Levenstein pour mesurer la différence entre les valeurs des cellules et les étiquettes des entités (Oliveira & D'Aquin, 2019). MantisTable se démarque grâce à une interface graphique poussée permettant de configurer le processus d'annotation automatique et et visualiser les résultats (Cremaschi *et al.*, 2019). LOD4ALL utilise essentiellement des requêtes SPARQL ASK pour obtenir des candidats et déduire des contraintes de type sur les colonnes (Morikawa, 2019). Les évaluations montrent que les approches ayant recours à des techniques de recherche élaborées et à des tâches de nettoyage et de pré-traitement optimisées pour les tables considérées obtiennent de meilleures résultats.

L'approche DAGOBAH propose une approche originale, basée sur des plongements de graphes, se démarquant des autres concurrents. Si cette technique a obtenu de très bon résultats sur la première manche, elle n'a pas encore permis d'atteindre les résultats escomptés dans les autres manches. Cela est en partie due à une préparation des données moins poussée que pour les autres approches, l'idée étant de proposer une solution générique pouvant s'appliquer à une grande variété de tables. Cependant, les approches à base de plongements restent une piste intéressante à explorer. Elles permettent de résoudre plus efficacement les ambiguïtés lorsque le contenu de la table est peu explicite. De plus, elles présentent une meilleure robustesse aux changements pouvant survenir dans les jeux de données ou dans les graphes de connaissances.

3 DAGOBAH : Un système d'annotation de données tabulaires de bout en bout

DAGOBAH est implémenté comme un ensemble d'outils utilisés de manière séquentielle pour atteindre les objectifs suivants :

1. L'identification de correspondances entre une table et un graphe de connaissances (i.e. processus d'annotation sémantique). Cet article se concentre uniquement sur les techniques employées et les résultats obtenus pour cette fonctionnalité.
2. L'enrichissement des graphes de connaissances en transformant en triplets les connaissances contenues dans les tables. Le graphe de connaissances cible de DAGOBAH est Wikidata pour plusieurs raisons incluant la fraîcheur des informations qu'il contient, sa large couverture et la qualité des données (Färber *et al.*, 2015). Par conséquent, des adaptations ont dû être réalisées pour le challenge SemTab2019 afin de supporter DBPedia.
3. La production de métadonnées pouvant être utilisées pour le référencement et l'indexation de jeux de données et des processus de recherche et de recommandation sub-

séquents (Chabot *et al.*, 2019a).

Pour fournir ses fonctionnalités d'annotation sémantique, DAGOBAH est structuré de la manière suivante.

Le module de pré-traitement (Section 3.1) réalise le nettoyage des tables et une première caractérisation de leur structure et de leur contenu. Les modules d'annotations accomplissent ensuite les tâches du challenge à proprement parler. Deux méthodes ont été étudiées pour mener à bien ces tâches : une méthode de base exploitant des services de recherche (i.e. "lookup") et des mécanismes de votes (Section 3.2) et une approche géométrique basée sur un partitionnement appliqué à des plongements du graphe Wikidata (Section 3.3).

3.1 Pré-traitement des données tabulaires

Plusieurs informations liées à la disposition de la table et aux types de données en présence sont utiles afin de traiter correctement les informations contenues dans les tables. La chaîne de pré-traitement intégrée dans DAGOBAH génère plusieurs informations dont l'orientation de la table, la présence ou non d'une en-tête, la présence d'une colonne clé et son index ainsi que les types primitifs des colonnes. De plus, elle supporte plusieurs formats de tables en entrée : CSV, TXT, JSON, XLS et XLSX. Dans un contexte d'exploitation réel dans lequel les utilisateurs possèdent peu ou pas de connaissances sur les tables à traiter, les informations produites sont décisives pour la qualité des annotations. Cette chaîne s'est inspirée du DWTC-Extractor [1] en améliorant plusieurs des algorithmes proposés. La précision de la chaîne DAGOBAH a été évaluée sur le corpus de la première manche du challenge (Table 3) et comparée à une version modifiée de l'outil du DWTC (qui n'utilise pas les balises HTML et supporte ainsi davantage de formats de données).

Détection d'orientation de tables. L'algorithme proposé par le DWTC-Extractor est basé sur la taille des chaînes de caractères et l'hypothèse que les valeurs des cellules d'une même colonne ont une longueur similaire. Cependant, la robustesse de cet algorithme peut être améliorée. En effet, deux chaînes de caractères représentant des éléments très différents peuvent avoir la même longueur (e.g. "Paris" et "$10cm^2$').

Pour pallier cette limite, DAGOBAH introduit un nouvel algorithme basé sur un système de typage primitif des colonnes en utilisant 11 types Tp_i (chaîne de caractères, nombres flottants, date, etc.).

Sur la base de ces types, un score d'homogénéité $Hom(x)$ est calculé sur chaque ligne et chaque colonne x (équation 1). La moyenne de l'ensemble des lignes et des colonnes est ensuite comparée et, selon le ratio, la table est dite "HORIZONTAL" ou "VERTICAL" (dans une table horizontale, les entités sont représentées en ligne et les attributs en colonne).

$$Hom(x) = [\frac{1}{|x|} \sum_{Tp_i \in x} (1 - (1 - 2 \times \frac{|Tp_i|}{|x|})^2)]^2 \quad (1)$$

où :

x est une ligne ou une colonne de la table,

$|x|$ est le nombre d'éléments de la ligne ou de la colonne considérée,

Tp_i est le type primitif $i, i \in [1; 11]$,

$|Tp_i|$ est le nombre d'occurrences du type i dans x.

Extraction d'en-tête. L'algorithme utilisé dans DAGOBAH est basé sur les types primitifs évoqués précédemment. L'extracteur d'en-tête utilise les hypothèses suivantes : en règle général, l'en-tête d'une colonne est une chaîne de caractères et possède un type différent du reste de la colonne. L'utilisation de ces deux heuristiques permet d'identifier assez précisément si une table contient ou non des en-têtes (Table 3). Il est à noter que le framework DWTC offre également un outil de détection d'en-tête mais ce dernier contient plusieurs bugs rendant son évaluation impossible.

1. https ://github.com/JulianEberius/dwtc-extractor

Détection de colonne clé. La colonne clé d'une table contient les identifiants des entités décrites par les autres colonnes. Il est à noter que dans la première version de la chaîne de pré-traitement, nos algorithmes font l'hypothèse que la clé d'une table est une colonne seule ce qui n'est pas toujours le cas en réalité (e.g. une table représentant des personnes et ayant pour colonne clé une agrégation des colonnes "nom" et "prénom"). L'algorithme proposé tire partie des informations de typage définies lors de la détection d'orientation et, à l'instar de l'extraction d'en-tête, d'heuristiques. La colonne clé est caractérisée comme une colonne, se situant à gauche dans la table, contenant des chaînes de caractères dont la majorité sont des valeurs uniques (identifiant des entités).

Cette chaîne de pré-traitement a été particulièrement utile durant la première manche du challenge pour identifier automatiquement les informations contenues dans les en-têtes des tables ainsi que la colonne à annoter (la colonne à annoter dans les tables lors du challenge était la colonne clé). L'information de colonne clé peut également être utile pour déterminer le type d'entités décrites par une table. Enfin, lorsque l'objectif est d'enrichir un graphe de connaissances avec les informations des tables, la détection de la colonne clé peut se révéler utile pour déterminer le sujet des triplets RDF générés. La mise à disposition des colonnes à annoter lors des manches suivantes a rendu de fait une partie de la chaîne de pré-traitement moins indispensable. Toutefois, cela ne remet pas en question l'utilité de tels outils pour des applications réelles.

TABLE *3 – Précision des tâches de pré-traitement sur le corpus de la manche 1*

Tâche/Outil	DWTC	DAGOBAH
Détection de l'orientation	0.9	**0.957**
Extraction d'en-tête	Non évalué	**1.0**
Détection de colonne clé	0.857	**0.986**

3.2 Méthode à base de vote majoritaire

Recherche d'entités. Afin d'optimiser les opérations de recherche (i.e. lookups), un premier processus de nettoyage est appliqué aux données. L'objectif de ce dernier n'est pas de corriger tous les problèmes des chaînes de caractères, mais d'avoir une transformation générique couvrant les artefacts les plus courants. Cela inclut l'homogénéisation de l'encodage et la suppression des caractères spéciaux (parenthèses, crochets et caractères non alphanumériques).

Cinq services de recherche sont ensuite utilisés simultanément pour trouver des entités candidates à partir du contenu des cellules : l'API Wikidata, le moteur Cirrus Search de Wikidata, l'API DBPedia, l'API Wikipédia ainsi qu'un index Elasticsearch interne dans lequel ont été stocké les labels, les alias et les types associés aux QIDs Wikidata (étape 1 de la Figure 1). Cette dernière source permet de garder le contrôle des stratégies de recherche et ainsi d'augmenter le nombre de candidats potentiels en élargissant le champ de recherche. A l'inverse, les autres APIs sont des services utiles mais fonctionnent comme des boîtes noires sur lesquelles nous n'avons pas de contrôle quant aux résultats retournés. Dans cette étape, un enrichissement des entités avec les informations issues de l'index Elasticsearch est également réalisé, et ce, afin de disposer, pour chaque entité, de son QID, c'est à dire de son identifiant unique dans Wikidata ce qui nous sera utile pour la suite du processus.

Un ensemble de traitements d'agrégation et d'enrichissement des candidats est ensuite effectué (étape 2 de la Figure 1) :

— Un comptage des occurrences, sur la base du QID, est réalisé à la sortie du processus de recherche afin de sélectionner les entités candidates les plus fréquemment retournées parmi les cinq services ainsi que leurs types.

— DBPedia étant la base de connaissances cible du challenge, les QIDs et les types Wikidata sont traduits en entités DBPedia. Pour cela, des requêtes SPARQL tirant parti des prédicats *owl:sameAs* et *owl:equivalentClass* sont utilisées.

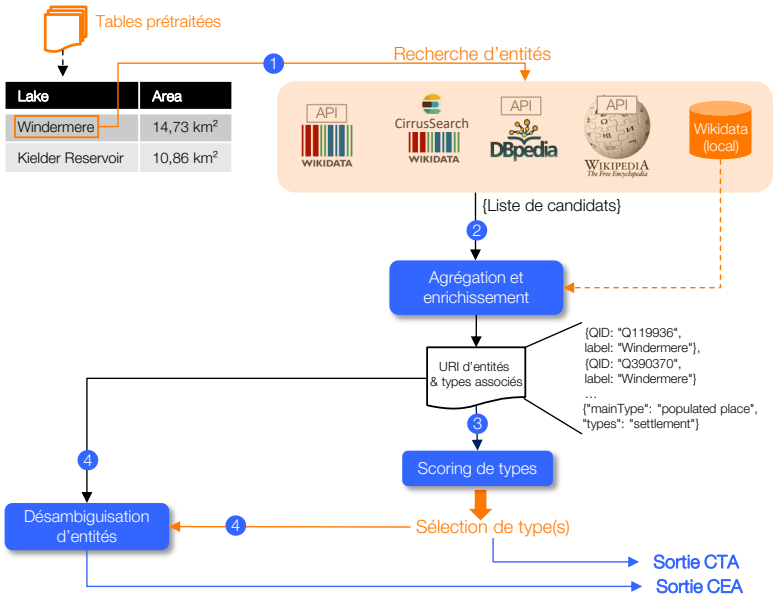


FIGURE *1 – Description du processus utilisé dans l'approche de base*

— En parallèle, le serveur Elasticsearch interne est interrogé afin de récupérer les types Wikidata (à partir des QID ou des labels et alias dans le cas de candidats DBpedia ou Wikipédia) qui sont utilisés comme les types pivots du système.
— Pour chaque entité candidate, la hiérarchie de classes parentes est enfin extraite à l'aide de requêtes SPARQL sur le endpoint DBPedia afin d'augmenter la liste de types candidats.

Typage des colonnes. L'étape suivante consiste à filtrer les types non pertinents, c'est à dire pas suffisamment représentatifs ou trop peu spécifiques. Un comptage basique des occurrences de type ne se révèle pas toujours pertinent étant donné que le type de la colonne considérée peut être très spécifique et pas suffisamment fréquent pour être considéré comme un type candidat. Pour corriger ce problème, un seuil sur les scores relatifs est utilisé. Pour chaque type $t_i \in \{t\}_y$ (liste de l'ensemble des types observés au sein de la colonne y), un score $S_y(t_i)$ est tout d'abord calculé, à partir duquel est déterminé un score relatif $R_y(t_i)$ (équation 2).

$$S_y(t_i) = \frac{|t_i|_y}{|t|_y} \quad , \quad R_y(t_i) = \frac{S_y(t_i)}{max(S_y(t_i))} \tag{2}$$

où :
t_i est le type $i \in \{t\}_y$,
$|t_i|_y$ est le nombre d'occurrences du type t_i dans la colonne y (i.e. le nombre d'entités possédant le type t_i),
$|t|_y$ est le nombre de types distincts dans la colonne y,
$max(S_y(t_i))$ est le score maximal obtenu par un des types i.

Seuls les types ayant un score $R_y(t_i) > 0.7$ (seuil configurable) sont considérés dans les étapes suivantes. Une méthode inspirée de TF-IDF est ensuite utilisée pour calculer la spécificité des types candidats et ainsi sélectionner les types les plus pertinents pour la colonne (étape 3 de la Figure 1). Cette méthode permet de déterminer l'importance d'un type $t_i \in \{t\}_y$

pour la colonne y (équation 3).

$$S_{spec}(t_i, y) = S_y(t_i) \times log(\frac{N_L(y)}{|t_i|_y}) \quad (3)$$

où :

$N_L(y)$ est le nombre d'entités candidates issues du processus de recherche pour la colonne y,

$|t_i|_y$ est le nombre d'occurrences du type t_i dans la colonne y (i.e. le nombre d'entités possédant le type t_i).

L'un des avantages inhérent à cette méthode est qu'elle possède une bonne indépendance vis à vis des bases de connaissances. Une alternative aurait été l'utilisation de la structure hiérarchique du graphe de connaissances pour mesurer la spécificité d'un type. Néanmoins, les performances en auraient été amoindries.

Désambiguisation d'entités. Afin d'améliorer les résultats de la tâche de CEA, les types issus de la sélection précédente sont utilisés afin de désambiguïser les entités candidates de chaque cellule (étape 4 de la Figure 1). Dans le cas où la première entité candidate possède l'un des types issus du CTA, cette entité est choisie comme résultat du CEA pour cette cellule. Dans le cas contraire, la liste des entités candidates est parcourue pour trouver une telle entité (si la liste d'entités candidates est initialement vide ou si aucune entité satisfaisante n'est trouvée, aucune annotation n'est produite pour la tâche de CEA).

Extraction des propriétés entre colonnes. La tâche de CPA a été implémentée de la même manière dans les deux approches (vote majoritaire et plongements). Elle consiste en une recherche syntaxique à partir des en-têtes de la table : les propriétés candidates dans les graphes de connaissances interrogés sont retenues. Une méthode plus élaborée a ensuite été testée sous la forme d'un algorithme récupérant l'ensemble des propriétés observées dans le graphe de connaissances entre les paires d'instances (résultats du CEA) des deux colonnes candidates et réalisant ensuite un vote majoritaire pour conserver uniquement la propriété la plus représentée entre les deux colonnes.

3.3 Méthode à base de plongements

Les méthodes à base de plongements permettent de représenter un ensemble de données textuelles (dans le cas de plongements lexicaux), ou multimodales en toute généralité, sous une forme vectorielle dans un espace de dimension fini. Bien que les dimensions de cet espace ne soient pas interprétables [2], une propriété intéressante est la proximité géométrique (vectorielle) des entités sémantiquement similaires. L'intuition sous-jacente à cette approche est que les entités d'une même colonne partagent a priori les mêmes caractéristiques sémantiques, et devraient donc se trouver géométriquement proches dans l'espace vectoriel des plongements par la nature même de ces derniers. Ainsi, les entités d'une même colonne devraient pouvoir être regroupées au sein de partitions cohérentes. Afin de comparer l'approche par plongements avec l'approche de base, une recherche syntaxique est appliquée à des plongements Wikidata en lieu et place des services de recherche externes précédemment utilisés.

Enrichissement des plongements et recherche d'entités. Les plongements Wikidata pré-entrainés (Han *et al.*, 2018) contiennent uniquement des QIDs Wikidata projetés dans un espace de dimension 200. Nous associons les labels, les alias ainsi que les types des entités aux entités en utilisant l'index Elasticsearch décrit précédemment (étape 1 du processus défini dans la Figure 2). Des opérations de recherche sont ensuite réalisées pour trouver les entités candidates (notées i dans la suite) correspondant au contenu de chaque cellule (étape 2 de la Figure 2). Le système implémente une première stratégie de recherche à base d'expressions régulières et une seconde stratégie à base de distance de Levenshtein.

1. Le label d'une entité candidate ou l'un de ses alias doit inclure tous les termes contenus dans la cellule (sans prise en compte de l'ordre).

2. Des travaux de recherche s'intéressent à la question (Senel *et al.*, 2018), (Templeton, 2020)

IC 2020

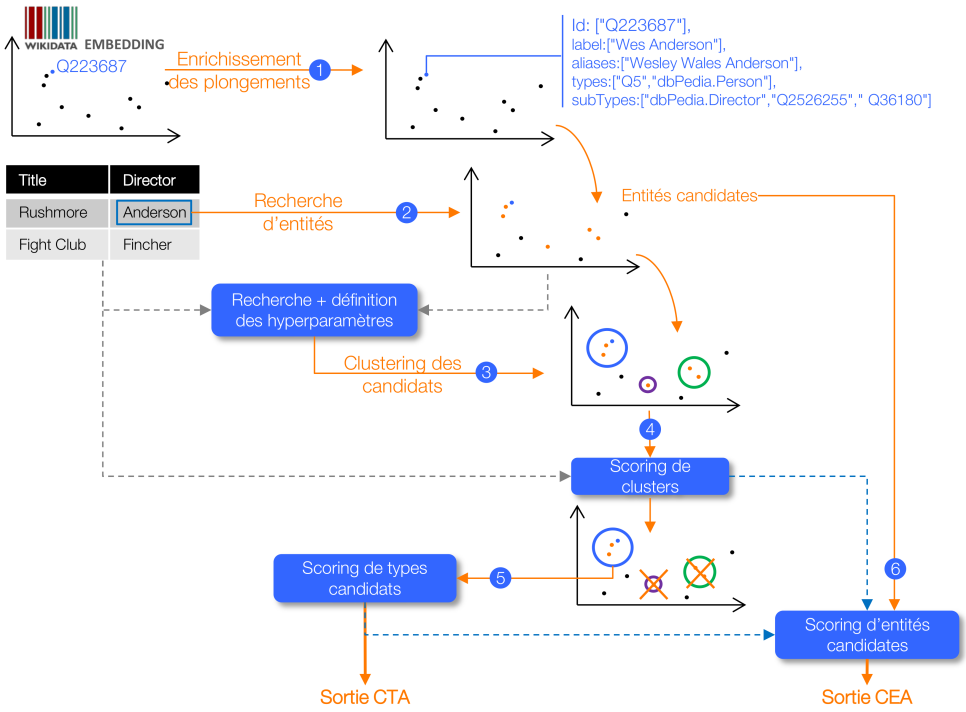


FIGURE 2 – *Description du processus utilisé dans l'approche par plongements*

2. Le ratio de Levenshtein (Sarkar *et al.*, 2016) (qui est fonction de la distance du même nom) entre le label de l'entité ou l'un de ses alias (obtenus suite au lookup) et le contenu de la cellule ($item_{sim}(i)$, où i est l'entité candidate) doit être supérieur ou égal à 0.75.

Si une entité (label ou alias) satisfait l'une des deux conditions précédentes, elle est alors ajoutée à la liste des candidates.

Partitionnement des entités candidates et choix des partitions. Après avoir choisi les candidats, le challenge est ensuite d'obtenir une partition contenant une large portion des candidats attendus pour une colonne donnée. Une stratégie de recherche Grid-search mesurant la capacité de plusieurs algorithmes et leurs hyper-paramètres à regrouper les bons candidats dans une même partition a été implémentée pour déterminer la meilleure configuration possible. Quatre algorithmes de partitionnement ont été évalués : le partitionnement en K-moyennes, le partitionnement spectral, DBSCAN et BIRCH. Cette recherche a permis d'établir que l'algorithme de partitionnement en K-moyennes (utilisant la distance euclidienne par défaut) obtient les meilleurs résultats (c'est-à-dire regroupe le plus de bons candidats d'une colonne tout en excluant les mauvais). Par ailleurs, l'analyse des résultats a permis de mettre en évidence qu'une valeur de K égale au nombre de candidats collectés par les opérations de recherche divisé par le nombre de lignes du tableau constituait une heuristique efficace permettant une automatisation du paramétrage de l'algorithme (étape 3 de la Figure 2). Une fois les partitions identifiées, l'étape suivante vise à choisir la partition représentant le mieux la colonne étudiée (étape 4 de la Figure 2). Les partitions possédant la plus importante couverture de lignes (i.e. le nombre de lignes ayant au moins un candidat dans la partition) sont retenus. Un score de confiance est ensuite associé à chaque candidat i à l'intérieur des partitions sélectionnées (équation 4).

$$S_c(i) = item_{conf}(i) \times item_{sim}(i)^w \tag{4}$$

où :

$item_{conf}(i)$ est le score de co-occurence donné par l'équation 5,

$item_{sim}(i)$ est le ratio de Levenshtein entre l'entité candidate i et la valeur de la cellule considérée,
$w \in \mathbb{N}^+$ permet de donner plus d'importance à la similarité syntaxique.

$$item_{conf}(i) = Fe(i) + 0.5 \times Fh(i) \tag{5}$$

où Fe et Fh sont définis par l'équation 6.

$$Fe(i) = \frac{e(i)+1}{N_C} \quad , \quad Fh = \frac{h(i)+1}{N_C} \tag{6}$$

où :
$e(i)$ est égal au nombre d'entités des autres colonnes de la table similaires aux valeurs de propriétés de l'entité candidate i extraites à partir du graphe de connaissance,
$h(i)$ est égal au nombre d'en-têtes des autres colonnes similaires à un label de propriété Wikidata associé à l'entité candidate i,
N_C est égal au nombre de colonnes de la table.

Le score de confiance normalisé pour une partition n donnée est ensuite calculé par l'équation 7.

$$S_k(n) = \frac{\sum_{i \in n} S_c(i)}{|n|} \tag{7}$$

où $|n|$ est le nombre d'éléments de la partition n.

A partir des candidats se trouvant dans les partitions retenues, un score est associé à chaque type rencontré parmi cette population. Le score d'un type donné est calculé en additionnant les scores de confiance de chaque individu possédant ce type (étape 5 de la Figure 2). Tous les types possédant un score supérieur à un seuil ($Max(score) * 0.75$) sont sélectionnés pour la suite du processus. Le type finalement retenu est le type le plus spécifique dans la hiérarchie DBPedia (la spécificité d'un type est calculée à partir du nombre de ses sous-classes ainsi que la distance le reliant au concept *owl:Thing*). Enfin, les entités candidates correspondant à chaque cellule sont examinées au regard du type retenu (étape 6 de la Figure 2). Le score, défini par l'équation 8, est calculé pour chaque entité candidate i appartenant à la partition n.

$$S_e(i) = w_T \times (0.2 \times S_k(n) + 0.5 \times S_c(i)) \tag{8}$$

où :
$w_T = 1.5$ si l'entité possède le type T résultant de l'étape de CTA,
$w_T = 1.2$ si l'entité a pour type un concept parent de T,
$w_T = 1$ sinon.

Pour une cellule de la table donnée, l'entité ayant le plus grand score est retenue comme résultat du CEA.

4 Résultats

Les différentes approches proposées ont été évaluées à l'aide de deux métriques sur les tâches de CEA et CPA : un score de précision et un score de F1 (équation 9).

$$Precision = \frac{|Labels\ corrects|}{|Labels\ produits|} \quad , \quad Rappel = \frac{|Labels\ corrects|}{|Labels\ cibles|} \quad , \quad F1 = \frac{2 \times P \times R}{P + R} \tag{9}$$

où $labels\ cibles$ fait référence aux cibles fournies par les organisateurs du challenge. Pour la tâche de CTA, un score primaire (AH) et un score secondaire (AP) ont été proposés par les organisateurs (équation 10)

$$AH = \frac{|P| + 0.5 \times |O| - |W|}{|T|} \quad , \quad AP = \frac{|P|}{|P| + |O| + |W|} \tag{10}$$

où P est le nombre d'annotations "parfaites" (le type correct le plus spécifique dans la hiérarchie de concepts), O est le nombre d'annotations "OK" (un concept parent de l'annotation parfaite en excluant la classe *owl:Thing*), W est le nombre d'annotations erronées et T le nombre de colonnes d'une table à annoter. Ces métriques permettent de prendre en compte la hiérarchie de classes du graphe de connaissances cible et ainsi d'évaluer la capacité des approches à produire les types les plus représentatifs mais également les plus spécifiques (et donc, porteur d'informations).

Pour des raisons de temps et de performance, l'approche par plongements a été uniquement testée sur le corpus de la première manche du challenge. Afin de permettre le calcul des scores primaires et secondaires, les données de CTA de cette manche ont été enrichies avec les types parents (excepté *owl:Thing*) du type parfait. Cette manche a permis d'évaluer les performances de l'approche par plongements par rapport à notre approche de base. Toutefois, les autres concurrents n'ont pas pu être comparés à nos approches sur cette manche car les méthodes d'évaluation du challenge ont évoluées entre la première manche et les suivantes. La Table 4 présente les scores respectifs de notre approche de base et de l'approche par plongements et montre la capacité de cette dernière à produire des types plus spécifiques. Seul notre approche de base et les résultats du meilleur concurrent (MTab (Nguyen *et al.*, 2019)) pour chaque tâche sur les trois dernières manches sont donc présentés dans la Table 5.

TABLE *4 – Comparatif de l'approche par plongements et de l'approche de base sur la première manche du challenge SemTab2019 réalisé le 30 novembre 2019 par un algorithme d'évaluation fourni par les organisateurs*

Tâche	CTA		CEA		CPA	
Critère	Prim. Score	Sec. Score	F1	Précision	F1	Précision
Baseline	0.479	0.242	**0.883**	**0.892**	0.415	0.347
Plongement	**1.212**	**0.336**	0.841	0.853	-	-

TABLE *5 – Résultats de l'approche de base et de MTab pour les trois dernières manches du challenge SemTab2019 évalués par la plateforme AICrowd le 30 novembre 2019*

	Tâche	CTA		CEA		CPA	
	Critère	Prim. Score	Sec. Score	F1	Précision	F1	Précision
Manche 2	Baseline	0.641	0.247	0.713	0.816	0.533	0.919
	MTab	1.414	0.276	0.911	0.911	0.881	0.929
Manche 3	Baseline	0.745	0.161	0.725	0.745	0.519	0.826
	MTab	1.956	0.261	0.970	0.970	0.844	0.845
Manche 4	Baseline	0.684	0.206	0.578	0.599	0.398	0.874
	MTab	2.012	0.300	0.983	0.983	0.832	0.832

Les résultats sur la tâche de CEA sont satisfaisants mais la baseline a montré des faiblesses lorsqu'il s'agissait de déduire les types attendus par l'évaluateur de la tâche CTA. Dans la première manche, le type produit par la baseline était soit trop générique, soit trop spécifique.

De plus, la baseline a montré deux limites importantes : une grande dépendance vis à vis des services de recherche (sur lesquels DAGOBAH a peu de contrôle) et des difficultés à paramétrer convenablement les algorithmes (en particulier, lorsqu'il s'agissait de trouver un bon compromis entre la spécificité du type d'une colonne et sa représentativité).

Concernant la tâche de CPA pour la première manche, l'utilisation de la méthode naïve de recherche syntaxique explique les mauvais résultats. Dans les manches suivantes, l'algorithme par vote majoritaire sur les propriétés candidates a été utilisé, ce qui a permis d'améliorer significativement la précision du CPA.

Concernant l'approche par plongement, les performances pour la tâche de CEA sont légèrement inférieures à la baseline. Cela s'explique notamment par l'utilisation de stratégies de recherche volontairement plus rudimentaires (comparativement aux méthodes de recherche très optimisées utilisées par la baseline), l'absence des bonnes entités parmi les partitions retenues ainsi que des problèmes liés aux opérations de correspondances entre Wikidata et DBPedia.

Toutefois, l'approche par plongement a montré son efficacité lorsqu'il s'agissait de déterminer le type d'une colonne (qui est une base de départ pour obtenir des annotations de CEA plus juste, en utilisant des techniques de désambiguisation). De plus, cette approche se révèle être particulièrement intéressante lorsque les cellules de la table étudiée contiennent des mentions incomplètes. Par exemple, dans la table 54719588_0_8417197176086756912 du challenge, l'une des colonnes contient des réalisateurs de films référencés uniquement par leur nom de famille (dans ce type de cas, la baseline obtient de mauvais résultats). Un partitionnement en K-moyennes sur un sous-ensemble de cette table (4 lignes uniquement) donne de très bons résultats comme présenté dans la Figure 3. En effet, la partition verte contient une grande partie des candidats attendus en résultat du CEA (une étape de désambiguisation doit malgré tout être réalisée pour trouver le bon résultat de certaines cellules en utilisant S_e).

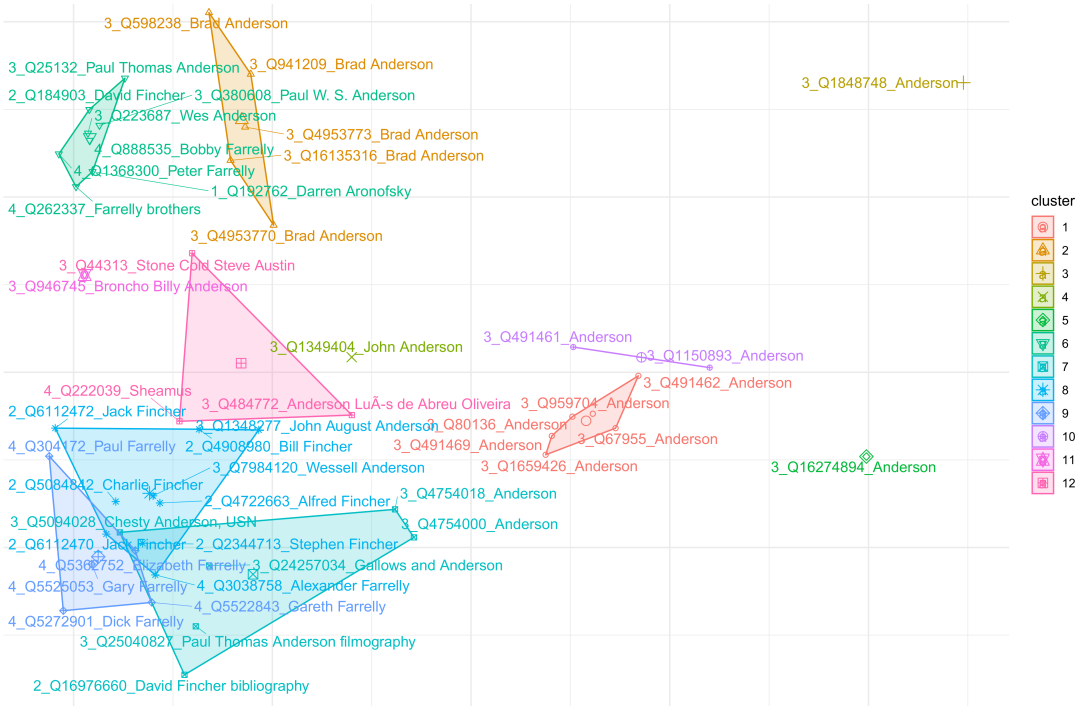


FIGURE *3 – Résultat du partitionnement en K-moyennes sur des plongements Wikidata*

IC 2020

5 Conclusion et travaux futurs

Nous avons présenté le système DAGOBAH, qui propose une approche de base et une approche à base de plongements de graphes pour générer des annotations sémantiques sur des données tabulaires. L'approche par plongement de graphes a montré des résultats encourageants, en particulier sur la tâche de CTA, et une réelle capacité à inférer des candidats à partir d'informations incomplètes, et ce, avec une étape de nettoyage et de préparation des données réduite au minimum. Cependant, l'optimisation des hyper-paramètres reste un problème complexe.

La méthode utilisée par DAGOBAH pour calculer le nombre de partition en sortie de l'algorithme des K-moyennes (basée sur les résultats de recherche et les propriétés de la table) donne de bons résultats mais pourrait ne pas se révéler suffisamment robuste pour l'ensemble des jeux de données (c'est la raison pour laquelle les partitions dont les scores sont les plus élevées sont conservées et pas uniquement la meilleure partition). D'autres algorithmes de partitionnement doivent être testés car ils pourraient être plus précis pour trouver le meilleur compromis entre le fait d'avoir tous les candidats dans une partition unique d'une part, et un nombre suffisant de partitions pour bien discriminer les candidats d'autre part.

La combinaison de Wikipédia et de Wikidata dans un espace de plongements commun conjointement à l'utilisation de plongements lexicaux de type fastText (Bojanowski *et al.*, 2016), ainsi que l'exploitation de la recherche sémantique en plus de la recherche syntaxique, pourraient améliorer considérablement le processus d'annotation sémantique. Enfin, une piste prometteuse est d'avoir recours à une approche complètement vectorielle (Chen *et al.*, 2018) consistant à apprendre des plongements à partir des lignes de la table (de type Poincaré (Nickel & Kiela, 2017) afin de capturer les hiérarchies latentes) combinée à des contraintes géométriques déduites de la structure de la table. Il s'agirait ensuite de trouver des transformations (globales ou locales) entre cet espace vectoriel et un espace de plongements Wikidata/Wikipédia afin d'améliorer la qualité des annotations.

Références

BHAGAVATULA C. S., NORASET T. & DOWNEY D. (2015). TabEL : Entity linking in web tables. In *14^{th} International Semantic Web Conference (ISWC)*, p. 425–441.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv :1607.04606*.

CHABOT Y., GROHAN P., LE CALVEZ G. & TARNEC C. (2019a). Dataforum : Faciliter l'échange, la découverte et la valorisation des données à l'aide de technologies sémantiques. In *Extraction et Gestion des Connaissances (EGC)*, Metz, France.

CHABOT Y., LABBÉ T., LIU J. & TRONCY R. (2019b). DAGOBAH : An End-to-End Context-Free Tabular Data Semantic Annotation System. *SemTab2019, ISWC Challenge*.

CHAPMAN A., SIMPERL E., KOESTEN L., KONSTANTINIDIS G., IBÁÑEZ L.-D., KACPRZAK E. & GROTH P. (2019). Dataset search : a survey. *The VLDB Journal*, p. 1–22.

CHEN J., JIMENEZ-RUIZ E., HORROCKS I. & SUTTON C. (2018). ColNet : Embedding the Semantics of Web Tables for Column Type Prediction. In *33^{rd} AAAI International Conference on Artificial Intelligence*.

CREMASCHI M., AVOGADRO R. & CHIEREGATO D. (2019). Mantistable : an automatic approach for the semantic table interpretation. *SemTab2019, ISWC Challenge*.

EBERIUS J., BRAUNSCHWEIG K., HENTSCH M., THIELE M., AHMADOV A. & LEHNER W. (2015). Building the dresden web table corpus : A classification approach. In *2015 IEEE/ACM 2nd International Symposium on Big Data Computing (BDC)*, p. 41–50 : IEEE.

EFTHYMIOU V., HASSANZADEH O., RODRIGUEZ-MURO M. & CHRISTOPHIDES V. (2017). Matching web tables with knowledge base entities : From entity lookups to entity embeddings. In *16^{th} International Semantic Web Conference (ISWC)*, p. 260–277.

FÄRBER M., ELL B., MENNE C. & RETTINGER A. (2015). A Comparative Survey of DBPedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web Journal*, p. 1–5.

FERNANDEZ R. C., MANSOUR E., QAHTAN A. A., ELMAGARMID A., ILYAS I., MADDEN S., OUZZANI M., STONEBRAKER M. & TANG N. (2018). Seeping semantics : Linking datasets

using word embeddings for data discovery. In *34th International Conference on Data Engineering (ICDE)*, p. 989–1000.

HAN X., CAO S., LV X., LIN Y., LIU Z., SUN M. & LI J. (2018). Openke : An open toolkit for knowledge embedding. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 139–144.

HASSANZADEH O., EFTHYMIOU V., CHEN J., JIMÉNEZ-RUIZ E. & SRINIVAS K. (2019). SemTab2019 : Semantic Web Challenge on Tabular Data to Knowledge Graph Matching - Data Sets. Zenodo.

IBRAHIM Y., RIEDEWALD M. & WEIKUM G. (2016). Making sense of entities and quantities in Web tables. In *International Conference on Information and Knowledge Management*, p. 1703–1712.

JIMÉNEZ-RUIZ E., HASSANZADEH O., EFTHYMIOU V., CHEN J. & SRINIVAS K. (2020). SemTab 2019 : Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. In *17th European Semantic Web Conference (ESWC)*.

KILIAS T., LÖSER A., GERS F. A., KOOPMANSCHAP R., ZHANG Y. & KERSTEN M. (2018). Idel : In-database entity linking with neural embeddings. *arXiv preprint arXiv :1803.04884*.

LEHMBERG O., RITZE D., MEUSEL R. & BIZER C. (2016). A Large Public Corpus of Web Tables containing Time and Context Metadata. In *25th International Conference Companion on World Wide Web (WWW Companion)*, p. 75–76.

LIMAYE G., SARAWAGI S. & CHAKRABARTI S. (2010). Annotating and searching web tables using entities, types and relationships. In *36th International Conference on Very Large Data Bases (VLDB)*, p. 1338–1347.

MORIKAWA H. (2019). Semantic table interpretation using lod4all. *SemTab2019, ISWC Challenge*.

MULWAD V., FININ T., SYED Z. & JOSHI A. (2010). Using linked data to interpret tables. In *1st International Workshop on Consuming Linked Data (COLD)*.

NGUYEN P., KERTKEIDKACHORN N., ICHISE R. & TAKEDA H. (2019). Mtab : Matching tabular data to knowledge graph using probability models. *arXiv preprint arXiv :1910.00246*.

NICKEL M. & KIELA D. (2017). Poincaré embeddings for learning hierarchical representations. *ArXiv*.

OLIVEIRA D. & D'AQUIN M. (2019). Adog - annotating data with ontologies and graphs. *SemTab2019, ISWC Challenge*.

RAN C., SHEN W., WANG J. & ZHU X. (2016). Domain-specific knowledge base enrichment using wikipedia tables. In *IEEE International Conference on Data Mining (ICDM)*, p. 349–358.

RITZE D., LEHMBERG O. & BIZER C. (2015). Matching HTML Tables to DBpedia. In *5th International Conference on Web Intelligence, Mining and Semantics (WIMS)*, p. 1–6.

SARKAR S., PAKRAY P., DAS D. & GELBUKH A. (2016). JUNITMZ at SemEval-2016 Task 1 : Identifying semantic similarity using levenshtein ratio. In *10th International Workshop on Semantic Evaluation (SemEval)*, p. 702–705.

SENEL L. K., UTLU I., CSAHINUCC F., OZAKTAS H. M. & KOCC A. (2018). Imparting interpretability to word embeddings while preserving semantic structure. *arXiv : Computation and Language*.

STEENWINCKEL B., VANDEWIELE G., DE TURCK F. & ONGENAE F. (2019). Csv2kg : Transforming tabular data into semantic knowledge. *SemTab2019, ISWC Challenge*.

TEMPLETON A. (2020). Inherently interpretable sparse word embeddings through sparse coding. *ArXiv*.

THAWANI A., HU M., HU E., ZAFAR H., DIVVALA N. T., SINGH A., QASEMI E., SZEKELY P. & PUJARA J. (2019). Entity linking to knowledge graphs to infer column types and properties. *SemTab2019, ISWC Challenge*.

ZHANG Z. (2014). Towards efficient and effective semantic table interpretation. In *13th International Semantic Web Conference (ISWC)*, p. 487–502.

CASO et IRRIG deux ontologies pour le développement de systèmes contextuels : cas d'usage sur l'automatisation de l'irrigation

Quang-Duy Nguyen[1], Catherine Roussey[1],María Poveda-Villalón[2], Christophe de Vaulx[3], Jean-Pierre Chanet[1], Camille Noûs[4]

[1] UNIVERSITÉ CLERMONT AUVERGNE, INRAE, UR TSCF, 63178 Aubière, France;
quang-duy.nguyen@inrae.fr, catherine.roussey@inrae.fr, jean-pierre.chanet@inrae.fr

[2] ONTOLOGY ENGINEERING GROUP, Universidad Politécnica de Madrid, Spain;
mpoveda@fi.upm.es

[3] LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES (LIMOS), UMR6158 UCA-CNRS, 63170 Aubière, France
christophe.devaulx@isima.fr

[4] LABORATOIRE COGITAMUS
camille.nous@cogitamus.fr

Résumé : Nos travaux portent sur le développement d'un système contextuel automatisant les décisions d'irrigation en fonction des mesures des capteurs. Plus précisément, cet article présente le développement des ontologies intitulées CASO et IRRIG : CASO organise l'ensemble des données manipulées par un système contextuel. L'ontologie IRRIG est une spécialisation de CASO dédiée aux traitements des données nécessaires à l'automatisation d'une méthode d'irrigation manuelle nommée IRRINOV®. Ces ontologies réutilisent des ontologies connues telles que Semantic Sensor Network (SSN) et Smart Appliance REFerence (SAREF). Un ensemble de traitements sur les données concerne les règles permettant de déduire les décisions quotidiennes d'irrigation. Ce projet travaille avec des données d'expérimentation d'irrigation pour évaluer le résultat des règles.

Mots-clés : Ontologies, systèmes contextuels, inférences à base de règles, mesures de capteurs, décision d'irrigation, agriculture numérique.

1 Introduction

Le développement rapide des technologies de l'information et des réseaux de capteurs sans fil a transformé les pratiques agricoles. De nouveaux outils et méthodes sont développées pour faciliter les travaux agricoles. Nos travaux portent sur le développement d'un système contextuel dont l'objectif est d'automatiser les décisions d'irrigation en fonction des mesures des capteurs. L'ensemble de ces travaux a été publié en anglais dans la revue "Applied Science" (Nguyen *et al.*, 2020). L'article français est une traduction partielle de l'article anglais. La version française ne présente que le développement des ontologies intitulées CASO et IRRIG. Dans un système contextuel, les ontologies sont une solution pour résoudre le problème d'intégration des mesures de capteurs hétérogènes. Elles définissent un modèle de données partagé décrivant les mesures pour faciliter leurs interprétations. Ces descriptions sont réutilisables par n'importe quelle machine. Le système contextuel contient également un système d'aide à la décision basé sur un moteur d'inférence à base de règles. Ainsi, nous proposons deux nouvelles ontologies : l'ontologie CASO (Contexte Aware System Ontology) organise l'ensemble des données manipulé par un système contextuel en général. L'ontologie IRRIG est une spécialisation de la précédente. IRRIG modélise l'ensemble des résultats des traitements nécessaire à l'irrigation automatique. Pour ce faire, nous avons automatisé une méthode d'irrigation manuelle nommée IRRINOV®. Ces ontologies réutilisent des éléments d'ontologies connues telles que Semantic Sensor Network (SSN) (Janowicz *et al.*, 2018) et Smart Appliance REFerence (SAREF)(ETSI TS 103 264 - v2.1.1, 2017). L'outil d'aide à la décision que nous avons développé contient un moteur à base de règles pour déduire les

décisions quotidiennes d'irrigation. Ce projet travaille avec des données d'expérimentation d'irrigation. Ainsi, nous avons pu évaluer les résultats des règles à l'aide de données réelles.

L'article est organisé de la manière suivante. Tout d'abord, des connaissances préliminaires sont présentées pour comprendre notre cas d'usage de développement d'un système contextuel. Ensuite un état de l'art présente un aperçu des ontologies existantes dans le domaine de l'irrigation. Puis, nous détaillons les étapes de développement des ontologies et leurs principaux résultats. Enfin, une brève conclusion termine l'article.

2 Connaissances préliminaires

Pour clarifier le discours, nous allons dans un premier temps poser quelques définitions et présenter le cycle de fonctionnement d'un système contextuel. Nous présentons brièvement la méthode d'irrigation IRRINOV®.

2.1 Définitions

Dans nos travaux, nous adoptons la définition suivante d'un système contextuel : "un système contextuel est un système qui utilise le contexte pour fournir des informations et des services appropriés à l'utilisateur. Il convient de noter que la pertinence d'une information ou d'un service dépend de la tâche réalisée par l'utilisateur." (Abowd *et al.*, 1999)

Nous définissons le contexte dans ce type de système comme "l'ensemble des entités caractérisées par leur état, plus toute information permettant de dériver les changements d'états de ces entités"(Sun *et al.*, 2016). Deux types de contexte sont définis (Sun *et al.*, 2016) :

— le contexte de bas niveau contient des données quantitatives telles que les mesures issues de capteurs ou les résultats d'agrégation sur ces mesures.
— le contexte de haut niveau, quant à lui, est constitué des données qualitatives synthétisant la situation d'une entité. Un exemple de contexte de haut niveau est l'état "sol sec" d'une parcelle ou le stade de développement "5 feuilles" d'une culture de maïs.

Un système contextuel pour l'irrigation est composé de trois composantes spécifiques : un Réseau de Capteurs Sans Fil (RCSF) en charge de la surveillance de l'environnement ; un Outil d'Aide à la Décision (OAD) pour déterminer les dates d'irrigation en fonction des mesures de capteurs et enfin un dispositif d'arrosage constituant un actionneur capable d'agir sur l'environnement.

Le cycle de fonctionnement d'un système contextuel se découpe en quatre phases : (1) l'acquisition du contexte, (2) la modélisation du contexte, (3) l'analyse du contexte et (4) l'exploitation du contexte. La figure 1 illustre le cycle de fonctionnement d'un système contextuel dédié à l'irrigation.

— phase d'acquisition du contexte : au cours de cette phase, le système acquiert des données provenant de diverses sources. La principale source de données est le réseau de capteurs sans fil qui mesure, collecte et transmet ses mesures sous forme de flux. Les sorties de cette phase sont les mesures, leurs metadonnées et toutes données nécessaires à la prise de décision.
— phase de modélisation du contexte : les mesures sont annotées pour pouvoir être intégrées dans un modèle de données commun. Les ontologies sont la solution choisie pour définir ce modèle. Ces données stockées et organisées constituent le contexte de bas niveau.
— phase d'analyse : au cours de cette phase, le contexte est enrichi. Tout d'abord des agrégations temporelles et spatiales sont appliquées sur les mesures de capteurs. Ainsi, le contexte de bas niveau s'enrichit de nouvelles données. Ensuite, un raisonnement est appliqué sur le contexte de bas niveau afin de produire le contexte de haut niveau. Dans nos travaux, le raisonnement a été implémenté à l'aide du moteur à base de règles Drools [1]. A la sortie de cette phase, toutes les décisions permettant de produire des actions doivent appartenir au contexte.

1. `https://www.drools.org/`

— phase d'exploitation du contexte : Les décisions sont prises à la phase précédente, il faut maintenant les mettre en application. Dans cette phase, le système exploite le contexte de haut niveau et le distribue à différents agents : des appareils connectés. Par exemple le contexte de haut niveau est transformé pour envoyer un message d'alerte au téléphone de l'agriculteur ou alors il est traduit en commande interprétable par le dispositif d'arrosage.

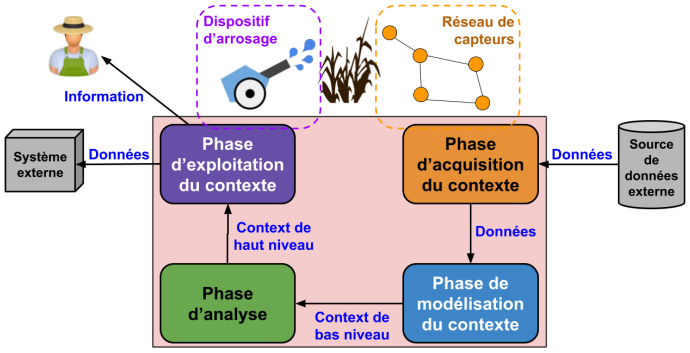


FIGURE *1 – Le cycle de fonctionnement d'un système contextuel dédié à l'irrigation automatique*

2.2 La méthode IRRINOV® pour piloter manuellement l'irrigation

Cette section présente la méthode IRRINOV® [2] développée par l'institut technique Arvalis et ses partenaires. Cette méthode propose un guide de bonnes pratiques aux agriculteurs afin de décider des dates d'irrigation en fonction des mesures des tensiomètres et des pluviomètres. Autrement dit, la méthode fournit des conseils pour répondre à trois questions : (1) quand l'irrigation devrait-elle commencer ? C'est-à-dire quand l'agriculteur doit-il mettre en place son dispositif d'arrosage sur la parcelle (2) quand lancer un arrosage c'est-à-dire quand démarrer un tour d'eau [3] ? (3) quand l'irrigation devrait-elle s'arrêter ? Autrement dit quand l'agriculteur peut retirer son dispositif d'arrosage de la parcelle ?

La méthode IRRINOV®est constituée d'un ensemble de tables de décision et de recommandations pour gérer l'irrigation d'une parcelle. Cette méthode propose de nombreuses variantes dépendant du type de sol de la parcelle et de sa culture. Dans le cadre de nos travaux (Nguyen *et al.*, 2020), la méthode IRRINOV® de la région Limagne pour la culture du maïs grain en sol argilo-calcaire (Arvalis *et al.*, 2007) a été retenue pour être mise en œuvre dans le système d'aide à l'irrigation.

Les équipements nécessaires pour réaliser des mesures avec la méthode IRRINOV® comprennent :

— une station de mesure IRRINOV® composée de 6 sondes Watermark pour mesurer l'humidité du sol (tensiomètre) : 3 sondes sont placées à 30 cm de profondeur et 3 sondes sont placées à 60 cm de profondeur.
— un pluviomètre pour mesurer la quantité d'eau reçue par la culture pendant un tour d'eau.
— une station météorologique comprenant un pluviomètre pour mesurer la quantité d'eau reçue par la culture pendant une pluie, et un thermomètre pour mesurer chaque jour la température minimale et la température maximale de l'air.

Des informations supplémentaires sur la méthode IRRINOV sont disponibles dans l'article (Poveda Villalón *et al.*, 2018).

2. `http://www.irrinov.arvalisinstitutduvegetal.fr/irrinov.asp`
3. temps entre deux arrosages d'une même parcelle

La méthode IRRINOV® (Arvalis *et al.*, 2007) propose plusieurs tables de décision pour déterminer le démarrage d'un tour d'eau. Ces tables dépendent du type de sol, de la durée du tour d'eau, de la culture et de son stade de développement.

Nous définissons la variable Probe30 (et Probe60) qui représente la mesure atteinte par deux sondes sur les trois sondes situées à 30 cm de profondeur (et respectivement à 60 cm de profondeur). Cette variable correspond à la médiane des mesures des 3 sondes.

La table de décision 1 présente le seuil utilisé pour commencer un tour d'eau pour une parcelle de maïs grain cultivé dans un sol argilo-calcaire. Cette table s'applique à la culture de maïs lorsque son stade de développement est entre "5 feuilles" et "10 feuilles". Les cellules de cette table présentent les seuils à atteindre par les mesures des tensiomètres en fonction de la durée des tours d'eau.

La première colonne d'une table de décision doit se lire comme "si la durée du tour d'eau propre à la parcelle est entre 9 à 10 jours" et " lorsque la variable Probe30 est supérieure à 30 cbar et la variable Probe60 est supérieure à 10 cbar " OU " lorsque la somme (Probe30 + Probe60) est supérieure à 40 cbar ", alors le tour d'eau peut commencer. Il convient de noter que dans cette table de décision, si les deux premières contraintes sont vraies simultanément alors la troisième contrainte est forcément vraie, car si $Probe30 > 30$ et $Probe60 > 10$ alors $Probe30 + Probe60 > 40$.

TABLE *1 – Valeurs de seuil en fonction de la durée des tours d'eau*

	9 à 10 jours	6 à 8 jours	inférieure ou égale à 5 jours
$Probe30$	30 cbar	50 cbar	60 cbar
$Probe60$	10 cbar	20 cbar	20 cbar
total	40 cbar	70 cbar	80 cbar

Nos travaux (Nguyen *et al.*, 2020) ont pour objectif de traduire ces tables de décision en règles. Ces traitements sont réalisés dans la phase d'analyse du système contextuel. La phase d'exploitation traduit les décisions d'irriguer en commandes exécutables par le dispositif d'arrosage.

3 Etat de l'art

D'après nos travaux sur le développement de système contextuel, une ontologie est un modèle de données partagé pour faciliter l'interprétation des données issues de différentes sources. Ainsi la base de connaissances structurée par cette ontologie intègre la description des mesures issues de capteurs hétérogènes. La description des mesures doit être réutilisable par n'importe quelle machine et compréhensible par l'homme. Pour atteindre cet objectif, nous avons étudié les ontologies utilisées dans les systèmes d'aide à la décision d'irrigation. Notre objectif est de réutiliser ces modélisations autant que faire se peut.

Plusieurs systèmes experts ont été développés pour déterminer les besoins en eau d'une culture. Par exemple, (Nada *et al.*, 2014) décrit un système expert développé à l'aide de la méthode Common-KADS pour irriguer les manguiers. Ce système expert n'est pas associé à un réseau de capteurs ou d'actionneurs. Son objectif est de déterminer la meilleure date d'irrigation. Dans ces travaux, l'ontologie développée n'a pas été publiée sur le Web.

Les technologies du Web sémantique ont déjà été utilisées pour le développement de systèmes contextuels dédié à l'irrigation. Par exemple, le projet SWAMP a pour objectif de développer une plateforme de gestion de l'eau pour l'irrigation de précision (Kamienski *et al.*, 2019). Ce projet a plusieurs sites pilotes afin d'expérimenter plusieurs solutions d'irrigation de précision dans des environnements différents. La plateforme IoT FIWARE qui permet de connecter des appareils au cloud est un des composants de l'architecture développée au sein du projet SWAMP. La plateforme FIWARE intègre une solution Semantic Web of Thing intitulée SPARQL Event Processing Architecture (SEPA). Les mesures de capteurs sont échangées au format JSON-LD pour mettre à jour une base de connaissances. Ces mesures sont ensuite interrogées à l'aide d'un moteur SPARQL. À notre connaissance, le projet SWAMP n'a pas encore publié les ontologies utilisées dans ces expérimentations.

Les travaux de (Wang *et al.*, 2015; Wang & Wang, 2018) présentent quelques ontologies dédiées à la culture des agrumes. Une des ontologies concerne l'irrigation. Cette ontologie permet de stocker des mesures de l'humidité du sol en fonction du type de sol et du stade de développement de la culture. Lorsque l'humidité du sol atteint un certain seuil, les agriculteurs reçoivent une alerte. Ces travaux n'ont pas publié leur ontologie sur le Web.

Selon nous, le système contextuel le plus abouti dédié à l'irrigation est celui développé lors du projet PLANTS (Goumopoulos *et al.*, 2009). Ce système concerne la culture sous serre de plants de framboisiers. Il utilise plusieurs types de capteurs pour observer le développement des plants. Ce système contextuel contrôle des dispositifs d'arrosage dans quatre zones de la serre. L'ontologie développée dans PLANTS décrit des e-entités et leurs interactions (Goumopoulos *et al.*, 2009, 2014). Une e-entité est une représentation virtuelle d'un objet physique : un plant ou un appareil. Chaque mesure de capteur est définie comme le paramètre d'une e-entité. Par exemple, l'e-entité représentant un plant a comme paramètre "LeafTemperature". Ce paramètre est mis à jour par les mesures d'un thermomètre. La dernière mesure du capteur est utilisée pour mettre à jour le paramètre correspondant. Par conséquent, ces travaux ne traitent pas d'agrégation de flux de mesures de capteurs. Des règles sont utilisées pour inférer les états des plants de framboisiers en fonction des valeurs de certains paramètres. Le dispositif d'arrosage est contrôlé par la détection de l'état "stress hydrique" dans une zone spécifique. Ainsi, la seule partie de l'ontologie dont nous pourrions nous inspirer est la hiérarchie des états des plants : les différents types de stress de la plante. Malheureusement, cette ontologie n'est pas publiée sur le Web car il s'agit du schéma de la base de faits implémenté dans le moteur d'inférences Jess.

Pour conclure, certains travaux concernent la gestion des flux de mesures de capteurs à l'aide des technologies Web sémantique ou le calcul d'agrégation sur ces flux. Malheureusement, leurs résultats ne sont pas directement réutilisables. Notre approche doit gérer à la fois l'agrégation temporelle et spatiale des flux de mesures de capteurs. De plus, les ontologies mentionnées dans les travaux précédents ne sont pas disponibles sur le Web ou leur domaine ne correspond pas exactement à notre cas d'usage.

4 Développement d'ontologies

LOT est une méthodologie utilisée pour le développement d'ontologies. Elle a été proposée pour la première fois dans (Poveda-Villalón, 2012) et développée dans (García-Castro *et al.*, 2017). LOT s'inspire des techniques agiles pour aligner le développement des ontologies avec le développement du système d'information. Cette méthodologie se concentre sur (1) la réutilisation d'éléments (classes, propriétés et attributs) existant dans des vocabulaires ou ontologies déjà publiés et (2) la publication de l'ontologie selon les principes du Web de données liées. Elle réutilise certaines activités d'ingénierie des connaissances définies dans la méthodologie NeOn (Suárez-Figueroa *et al.*, 2015). Cette méthodologie définit les itérations sur les quatre activités suivantes : (1) spécification des besoins ontologiques, (2) implémentation de l'ontologie, (3) publication de l'ontologie et (4) maintenance de l'ontologie.

4.1 Spécification des besoins ontologiques

L'objectif de l'ontologie à développer est de stocker toutes les données manipulées par le système contextuel : les données capteurs, les résultats des traitements (agrégation ou inférence) jusqu'à la décision finale d'irrigation. Tout d'abord, nous avons étudié la documentation de la méthode IRRINOV® Arvalis *et al.* (2007) pour identifier les éléments suivants :

- les types de capteurs et leurs mesures,
- les données nécessaires à la prise de décisions : type de sol, variété de culture, date de semences, localisation des capteurs et leur profondeur, quantité de pluie, évolution des stades de développement de la culture, mesures d'humidité du sol à différentes profondeurs,
- les différents traitements appliqués sur les flux de mesures de capteurs, par exemple le nettoyage, l'agrégation temporelle et spatiale, la comparaison à l'aide de seuils,

Specification Competency Questions document

Version 0.1

Code	Compentency Questions	Answer/Example	
CQ1	Description of measurement equiments		
CQ1.1	What is the identifier of the probe/sensor?	2013F13 03	
CQ1.2	Which type of measurement provides the probe/sensor? Which phenomenon and which associated property is measured by the probe/sensor?	Example: soil moisture at 30 cm-depth	
CQ1.3	What is the correction value of the probe/sensor? Does the raw measurement produced by a probe/sensor need a transformation before being considered as a valid measurement?	1	Tl va re
CQ1.4	What is the range of value provided by the probe/sensor? Is the measurement of the probe/sensor valid?	0 to 200	
CQ1.5	What is the unit of the probe/sensor?	ohar, mm, degree Celsius	

FIGURE 2 – *Quelques exemples de question de compétences*

— les différentes décisions prises par la méthode.

Dans un second temps, nous avons discuté avec les experts en irrigation d'Arvalis et d'INRAE pour clarifier la méthode IRRINOV®. Par exemple, nous avons posé les questions suivantes aux experts : "Quelle est la différence entre une irrigation et un tour d'eau ? Quelles sont les valeurs possibles des stades de développement du maïs ? Est-ce-que tous les stades vont apparaître ? Comment observer un stade de développement du maïs ?"

Troisièmement, nous avons essayé de généraliser les connaissances décrites lors de ces discussions en proposant des définitions consensuelles des notions nécessaires à la compréhension d'IRRINOV®. Ce travail a été effectué en interrogeant des sources disponibles sur le Web. Les sources sélectionnées sont des articles de références produites par des organisations reconnues. Par exemple, les différents stades de développement du maïs sont définis dans une publication de l'Université de l'Iowa (Lori J. Abendroth *et al.*, 2011).

Quatrièmement, l'analyse de l'ensemble de données fournies par Arvalis nous a permis de mieux comprendre les calculs effectués sur les flux de mesures des capteurs. Cette analyse nous a permis de définir plusieurs chaînes de traitements : chacune correspondant à un type de flux de mesures du capteur. Une chaîne commence par valider les mesures de capteurs, ensuite elle agrège ces mesures (agrégations temporelle ou spatiale) puis elle compare les valeurs agrégées à des seuils et enfin des déductions sont appliquées sur les états. Un exemple d'agrégation temporelle est le calcul de la moyenne des mesures d'humidité du sol produites par un tensiomètre sur une fenêtre de 24 heures. Ces agrégats sont modélisés sous la forme d'une nouvelle propriété du phénomène observé par la sonde. Notre système contextuel comprend quatre principaux flux de mesures de capteurs. Chacun de ces flux est décrit par un phénomène observé (sol, pluie, culture) et ses propriétés associées : (1) l'humidité du sol, (2) la quantité de pluie, (3) le stade de développement de la culture et (4) le besoin en eau de la culture. A partir de cette analyse, nous avons défini les spécifications ontologiques de notre système contextuel. Ces spécifications sont écrites sous la forme de questions de compétences. L'ensemble des questions de compétences a été décrit dans un tableur disponible dans le répertoire gitlab associé à ce projet [4]. En résumé, l'ontologie doit être capable de décrire tous les traitements appliqués aux flux de mesures de capteurs jusqu'à obtenir la décision journalière d'irrigation.

La figure 2 présente un extrait des questions de compétences. Par exemple, la question CQ1.1 définit le besoin de connaître le type et la marque de chaque sonde.

4.2 Conceptualisation de l'ontologie

La spécification des besoins ontologiques était dans un premier temps orientée vers le développement d'une seule ontologie. En cours de développement, nous avons noté que cer-

4. https://gitlab.irstea.fr/irrig/public/tree/master/CompetencyQuestions

tains éléments de l'ontologie pouvaient être réutilisés dans la conception de tout système contextuel. Nous avons donc décidé de définir deux ontologies : CASO et IRRIG. L'ontologie CASO (Contexte Aware System Ontology) est dédiée au développement de tout système contextuel. L'ontologie IRRIG est dédiée au développement d'un système contextuel automatisant l'irrigation.

A noter qu'avant de commencer le développement de notre ontologie, nous avons étudié plusieurs ontologies de référence sur les mesures de capteurs (Poveda Villalón *et al.*, 2018). Ainsi, nous avons choisi de réutiliser SOSA/SSN car cette ontologie propose plusieurs patrons de conception ontologiques décrivant précisément les mesures de capteurs.

La première étape de la conceptualisation définit les composants de l'ontologie à partir des questions de compétences. Cette étape se compose de différentes phases exécutées dans l'ordre de présentation ci-dessous :

— identification des classes : Nous voulons modéliser tous les traitements appliqués aux flux de mesures de capteurs. Ce sont principalement des agrégations et des comparaisons à partir de seuils. Le but d'une comparaison est de déduire un état à partir d'une valeur agrégée. Pour différencier les inférences des autres calculs, nous devons définir la classe "Déduction". Le résultat d'une déduction est un état (une valeur symbolique). Nous définissons les seuils comme des bornes d'un état donné.
— identification des propriétés d'objets : Pour caractériser une déduction, nous avons besoin de deux nouvelles propriétés d'objet : (1) une propriété qui lie une déduction à son résultat (un état) et (2) une propriété qui lie une déduction à l'intervalle temporel où cette déduction est valide, c'est-à-dire la durée pendant laquelle le phénomène observé est dans cet état.
— identification des contraintes de cardinalité : Cette étape détermine si une propriété d'objet est fonctionnelle. Par exemple, la propriété qui lie une déduction à son résultat est fonctionnelle. Une déduction n'a qu'un seul résultat : l'état du phénomène.
— identification des individus : D'après les patrons de conception de SOSA/SSN, les phénomènes observés sont modélisés sous la forme d'instances de la classe **sosa :FeatureOfInterest** (Janowicz *et al.*, 2018). Les types de mesures caractérisant ces phénomènes sont modélisés sous la forme d'instances de la classe **ssn :Property**, chacune étant liée à l'instance correspondante de la classe **sosa :FeatureOfInterest** par la propriété d'objet `ssn:hasProperty`. Par exemple, un individu, instance de la classe **ssn :Property**, représente la quantité journalière de pluie et est lié à une instance de la classe **sosa :FeatureOfInterest** représentant la pluie.

Ensuite, nous avons identifié des éléments d'ontologies existantes pour les réutiliser. Les ontologies sélectionnées ont été développées et sont maintenues par des organisations de normalisation.

— la nouvelle version de l'ontologie Semantic Sensor Network (SOSA/SSN) qui est une recommandation du W3C et de l'OGC. Cette ontologie décrit les capteurs, leurs mesures, les échantillons et les actionneurs (Janowicz *et al.*, 2018).
— l'ontologie Smart Appliances REFerence (SAREF) [5] est une recommandation de l'ETSI pour résoudre les problèmes d'interopérabilité dans le domaine de l'IoT (ETSI TS 103 264 - v2.1.1, 2017). Cette ontologie modélise les appareils connectés ainsi que leurs fonctions, commandes, services, états et profils. Nous avons réutilisé des éléments de SAREF4AGRI qui est une extension de SAREF pour l'agriculture.
— l'ontologie PROV du W3C est utilisée pour modéliser la provenance des données (Lebo *et al.*, 2013).
— l'ontologie SKOS du W3C est préconisée pour représenter les thésaurus et tout système d'organisation des connaissances (Bechhofer & Miles, 2009). Un thésaurus peut être utilisé pour identifier les types de mesures ou de phénomènes observés.
— l'ontologie Time du W3C [6] permet de décrire tous les éléments temporels.
— l'ontologie Units of Measure and Related Concepts (OM) (Rijgersberg *et al.*, 2013) identifie un ensemble d'unités de mesures.

5. `http://www.w3id.org/saref`
6. `http://www.w3.org/2006/time`

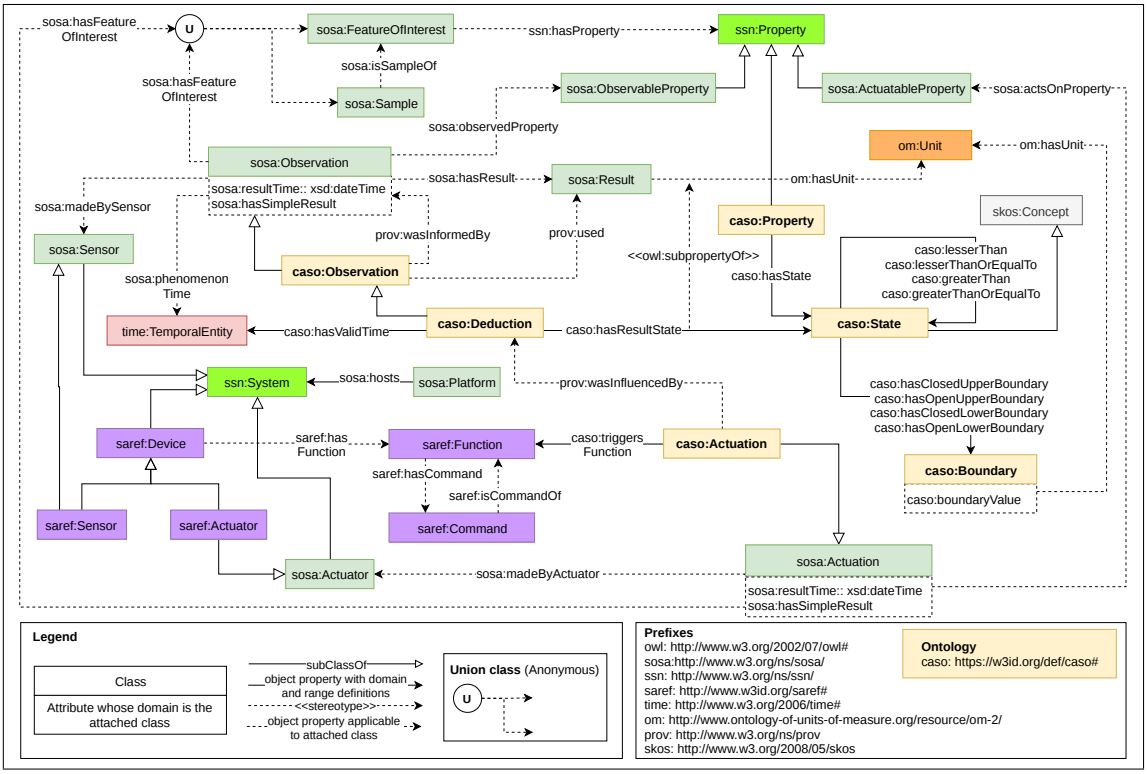


FIGURE *3 – L'ontologie CASO.*

Dans un cas, les éléments des ontologies existantes correspondaient exactement à l'utilisation prévue dans notre ontologie. Par conséquent, ils ont été intégrés sans aucune modification. Quelques exemples de ces réutilisations sont les classes **sosa :Platform**, **sosa :FeatureOfInterest** ou **om :Unit**.

Dans un autre cas, certaines caractéristiques ou contraintes supplémentaires ont dû être ajoutées aux éléments des ontologies existantes. Pour ce faire, nous avons spécialisé des classes existantes en définissant de nouvelles sous classes. Par exemple, la classe **caso :Observation** est une spécialisation de la classe **sosa :Observation**. Elle représente tout processus de calcul. Toutes les instances de cette nouvelle classe doivent être liées par la propriété objet `prov:wasInformedBy` à leurs données d'entrée. La classe **caso :Deduction** est une spécialisation de **caso :Observation**. Une déduction est également un processus de calcul qui a pour résultat un état (une valeur symbolique et non numérique). Pour forcer la modélisation des états sous forme de valeur symbolique qu'il faut documenter correctement, la classe **caso :State** est définie comme une sous classe de **skos :Concept**. Les états (des plantes ou des dispositifs) sont réutilisables dans différentes applications agricoles, leurs définitions dans un thésaurus faciliteront leurs réutilisations.

Finalement, la conceptualisation a abouti à deux ontologies : CASO et IRRIG. L'ontologie IRRIG importe et étend l'ontologie CASO. L'ontologie CASO est illustrée à la figure 3. L'ontologie IRRIG est représentée sur la figure 4. Les entités définies dans les ontologies CASO et IRRIG sont respectivement coloriées en jaune et en bleu.

L'objectif de CASO est de modéliser tout traitement du contexte. IRRIG est une spécialisation de CASO pour décrire les traitements du contexte de la méthode IRRINOV®. Pour cette raison, elle spécialise certains éléments de CASO, comme le montre la figure 4. Par exemple, IRRIG spécialise la classe **ssn :Property** en différentes sous-classes liées à l'humidité, le stress et la croissance. Plusieurs sous-classes de **caso :Observation** spécifient différents types d'observations. Chaque sous-classe est dédiée aux observations d'une instance spécifique de **ssn :Property**.

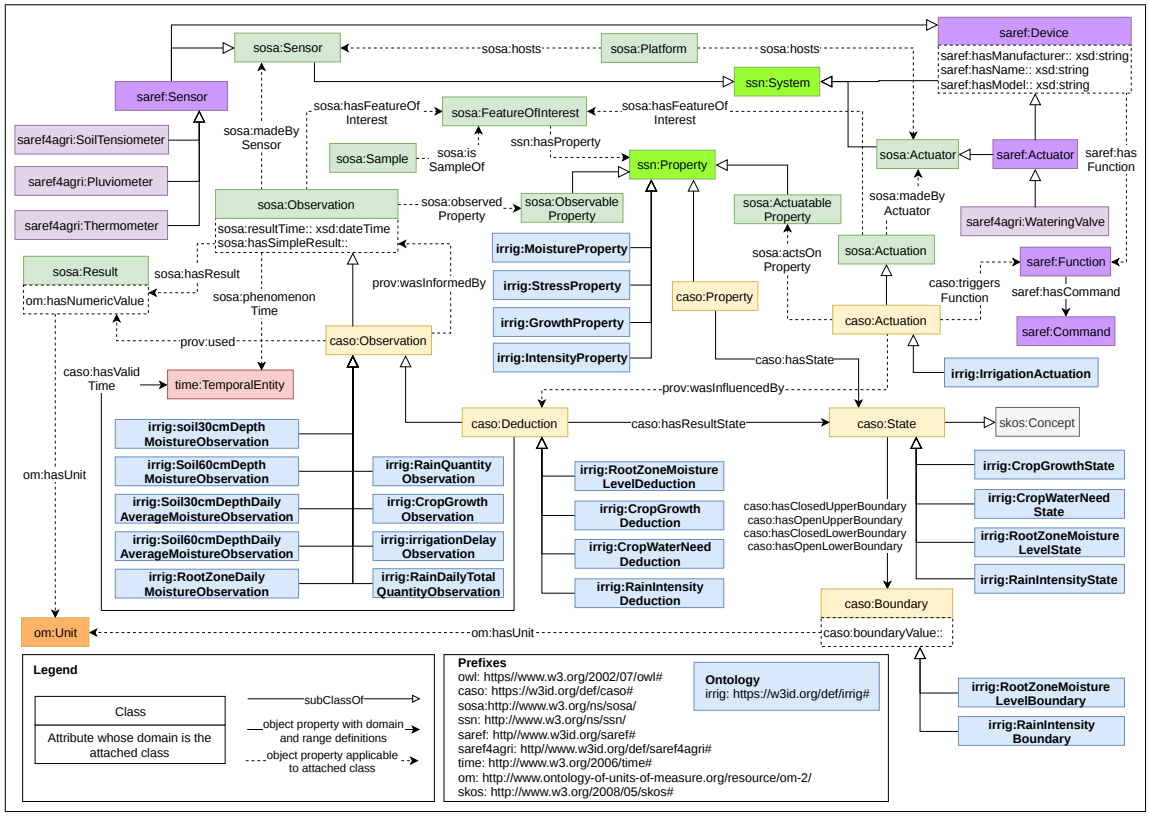


FIGURE *4 – L'ontologie IRRIG*

IC 2020

4.3 Encodage des ontologies

Une fois les modèles de données finalisés, les deux ontologies ont été implémentées dans le langage OWL (Group, 2012) en utilisant Protégé (Musen, 2015). La mise en œuvre des ontologies comprenait la déclaration de metadonnées telles que les auteurs, les dates et les licences, selon (Garijo & Poveda Villalon, 2017). Le code OWL des ontologies est disponible dans deux répertoires GitHub distincts. L'implémentation actuelle de CASO contient 27 classes, 40 propriétés d'objet et 4 propriétés de type de données. Elle réutilise les ontologies SOSA/SSN, SKOS, SAREF, PROV, Time et OM. L'ontologie IRRIG importe CASO et l'étend avec 31 classes, 4 propriétés de type de données et 71 individus pour modéliser les états, les types de mesures et les phénomènes naturels liés à l'irrigation. Parmi eux, quatre classes et trois propriétés de type de données sont importées de l'ontologie SAREF4AGRI, une extension de SAREF pour agriculture.

4.4 Évaluation de l'ontologie

Les deux ontologies ont été évaluées de trois manières différentes : vérification de l'absence de mauvaises pratiques, vérification de la couverture des besoins ontologiques et validation de l'ontologie par l'implémentation des règles avec le moteur Drools.

Pour vérifier l'absence de mauvaises pratiques ou d'erreurs dans CASO et IRRIG, le logiciel de vérification OOPS ! a été utilisé (Poveda-Villalón *et al.*, 2014). Le rapport fournit par ce logiciel nous a permis d'améliorer les aspects suivants des ontologies : (1) définir les propriétés inverses, (2) enrichir la description des composants des ontologies en complétant les propriétés d'annotation `rdfs:label` et `rdfs:comment`, et (3) saisir correctement certains éléments de l'ontologie en spécifiant la classe ou le type de propriétés OWL (type de données, objet ou annotation).

En ce qui concerne la vérification de la couverture des besoins, l'ontologie IRRIG, qui contient déjà CASO, a été comparée aux questions de compétences précédemment définies. Pour chaque question, les éléments de l'ontologie répondant à la question ont été identifiés.

Enfin, nous avons vérifié que les données fournies par Arvalis pouvaient être annotées avec nos ontologies. Nous avons utilisé le jeu de données d'Arvalis produit pendant une expérimentation d'irrigation en 2013. Pour ce faire, plusieurs fichiers RDF ont été générés. Chaque fichier décrit les mesures produites par les capteurs pendant une journée. L'ensemble des résultats des traitements effectués pour aboutir à la décision finale ont aussi été stockés dans ces fichiers. Ainsi nous avons pu tracer tous les résultats des traitements effectués par notre outil d'aide à la décision. Des exemples du contenu de ces fichiers sont décrits en détail dans (Nguyen *et al.*, 2020).

4.5 Publication de l'ontologie et maintenance

Pour maintenir ces deux ontologies et les faire évoluer, nous avons besoin d'obtenir les commentaires de la communauté d'utilisateurs potentiels. Tout d'abord, les ontologies CASO et IRRIG ont été publiées sur le Web, chacune ayant reçu un identifiant pérenne : `https://w3id.org/def/caso#` et `https://w3id.org/def/irrig#`. La publication de ces ontologies sur le Web a été réalisée à l'aide de la suite logicielle OnToology (Alobaid *et al.*, 2018). OnToology s'appuie sur un répertoire GitHub contenant le code et la documentation HTML. Les pages HTML ont été générées automatiquement par l'outil Widoco (Garijo, 2017). Puis, elles ont été mises à jour manuellement pour inclure des diagrammes et des informations manquantes.

Le répertoire GitHub est rendu public, ainsi les utilisateurs peuvent écrire des commentaires et rendre compte des problèmes rencontrés. La documentation du répertoire fournit aussi les adresses e-mail des responsables des ontologies. Ces ontologies ont aussi été publiées sur l'Agroportal du Lirmm.

Nous avons prévu de maintenir les ontologies IRRIG et CASO au moins pendant 5 ans.

5 Synthèse et Conclusion

Les activités d'ingénierie des connaissances sont maintenant soutenues par plusieurs méthodologies. Mais il existe encore des décisions importantes à prendre pendant la conceptualisation de l'ontologie. Une de nos décisions la plus controversée et la plus longue était de savoir s'il fallait créer la classe Déduction. Représente-t-elle vraiment une nouvelle notion ou est-ce simplement une observation au sens de SOSA/SSN ? Nous avons choisi de mettre en avant les déductions en créant cette classe car elle permet d'identifier le contexte de haut niveau avec le raisonnement symbolique associé. Une déduction aboutie à un état. Une autre remarque importante lors du développement de l'ontologie est de terminer le cycle entre les propriétés observables par des capteurs et les propriétés actionnables par des actionneurs. C'est pour cette raison que nous avons combiné les ontologies SOSA/SSN et SAREF.

Avant de commencer le développement de CASO et IRRIG, nous avions décidé de réutiliser des ontologies existantes proposées par des organismes de normalisation pour accroître l'interopérabilité des données. Cependant, ce choix a considérablement augmenté le temps de développement des ontologies. Quand on réutilise un élément existant, il faut comprendre précisément la sémantique et les conséquences de choix de modélisation des auteurs. Cette tâche s'avère encore plus compliquée lorsque l'on combine plusieurs ontologies. D'après notre expérience, il est plus facile de construire un modèle de données à partir de zéro que d'intégrer des éléments existants. Par conséquent, la décision de réutiliser les ontologies repose entièrement sur les exigences demandées en matière d'interopérabilité des données.

Cet article est une traduction partielle de l'article (Nguyen *et al.*, 2020). Nous avons présenté les étapes de développement des ontologies intitulées CASO et IRRIG. CASO est une ontologie générique qui a pour but de modéliser tous traitements effectués par un système contextuel sur les flux de mesures de capteurs. IRRIG importe et étend CASO pour modéliser les traitements nécessaires à la prise de décision d'irrigation en suivant la méthode IRRINOV. CASO et IRRIG réutilisent des éléments des ontologies SOSA/SSN, PROV, SAREF, SAREF4AGRI, OM, Time, SKOS. Ces ontologies sont publiées sur le Web. IRRIG a permis de définir la base de faits utilisée par le moteur à base de règles Drools. Ainsi nous avons pu valider la modelisation proposée pour décrire les traitements nécessaires à la prise de décision d'irrigation.

Acknowledgements

Cette recherche est financée au travers de différents projets : (1) le projet CPER "ConnecSens", (2) le projet "Programme I-Site Clermont WOW ! Wide Open to the World - CAP20-25" (16-IDEX-0001) et (3) le projet ETSI STF534 "SAREF extensions". Le projet CPER "ConnecSens" est cofinancé par la région Auvergne-Rhône-Alpes en France, ainsi que par l'Union européenne via le fond FEDER.

Références

ABOWD G. D., DEY A. K., BROWN P. J., DAVIES N., SMITH M. & STEGGLES P. (1999). Towards a Better Understanding of Context and Context-Awareness. In *Handheld and Ubiquitous Computing*, volume 1707, p. 304–307. Springer Berlin Heidelberg.

ALOBAID A., GARIJO D., POVEDA-VILLALÓN M., SANTANA-PEREZ I., FERNÁNDEZ-IZQUIERDO A. & CORCHO O. (2018). Automating ontology engineering support activities with OnToology. *Journal of Web Semantics*.

ARVALIS, INRA & CHAMBRE D'AGRICULTURE (2007). *Guide de l'utilisateur, Carnet de terrain : Piloter l'irrigation avec la méthode IRRINOV*. Technical report 07X04, Arvalis, Midy-Pyrénées, France.

BECHHOFER S. & MILES A. (2009). Skos simple knowledge organization system reference. *W3C recommendation, W3C*.

ETSI TS 103 264 - v2.1.1 (2017). *SmartM2M ; Smart Appliances ; Reference Ontology and oneM2M Mapping*. Rapport interne, ETSI.

IC 2020

GARCÍA-CASTRO R., FERNÁNDEZ-IZQUIERDO A., HEINZ C., KOSTELNIK P., POVEDA-VILLALÓN M. & SERENA F. (2017). *D2.2 Detailed Specification of the Semantic Model*. Rapport interne, Universidad Politécnica de Madrid (UPM).

GARIJO D. (2017). Widoco : a wizard for documenting ontologies. In *International Semantic Web Conference*, p. 94–102 : Springer.

GARIJO D. & POVEDA VILLALON M. (2017). *A checklist for complete vocabulary metadata*. Rapport interne, WIDOCO.

GOUMOPOULOS C., KAMEAS A. D. & CASSELLS A. (2009). An Ontology-Driven System Architecture for Precision Agriculture Applications. *Int. J. Metadata Semant. Ontologies*, **4**(1/2), 72–84.

GOUMOPOULOS C., O'FLYNN B. & KAMEAS A. (2014). Automated zone-specific irrigation with wireless sensor/actuator network and adaptable decision support. *Computers and Electronics in Agriculture*, **105**, 20–33.

GROUP W. O. W. (2012). *OWL 2 Web Ontology Language Document Overview (Second Edition)*. W3c Recommendation, W3C.

JANOWICZ K., HALLER A., COX S. J., LE PHUOC D. & LEFRANÇOIS M. (2018). SOSA : A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*.

KAMIENSKI C., SOININEN J.-P., TAUMBERGER M., DANTAS R., TOSCANO A., SALMON CINOTTI T., FILEV MAIA R. & TORRE NETO A. (2019). Smart Water Management Platform : IoT-Based Precision Irrigation for Agriculture. *Sensors*, **19**(2), 276.

LEBO T., SAHOO S., MCGUINNESS D., BELHAJJAME K., CHENEY J., CORSAR D., GARIJO D., SOILAND-REYES S., ZEDNIK S. & ZHAO J. (2013). Prov-o : The prov ontology. *W3C recommendation*, **30**.

LORI J. ABENDROTH, ROGER W. ELMORE, MATTHEW J. BOYER & STEPHANIE K. MARLAY (2011). *Corn Growth and Development*. Iowa State University.

MUSEN M. A. (2015). The protégé project : a look back and a look forward. *AI Matters*, **1**(4), 4–12.

NADA A., NASR M. & HAZMAN M. (2014). Irrigation expert system for trees. *International Journal of Engineering and Innovative Technology (IJEIT)*, **3**(8), 170–175.

NGUYEN Q.-D., ROUSSEY C., POVEDA-VILLALÓN M., DE VAULX C. & CHANET J.-P. (2020). Development Experience of a Context-Aware System for Smart Irrigation Using CASO and IRRIG Ontologies. *Applied Sciences*, **10**(5), 1803.

POVEDA-VILLALÓN M. (2012). A reuse-based lightweight method for developing linked data ontologies and vocabularies. In E. SIMPERL, P. CIMIANO, A. POLLERES, O. CORCHO & V. PRESUTTI, Eds., *The Semantic Web : Research and Applications*, p. 833–837, Berlin, Heidelberg : Springer Berlin Heidelberg.

POVEDA-VILLALÓN M., GÓMEZ-PÉREZ A. & SUÁREZ-FIGUEROA M. C. (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.

POVEDA VILLALÓN M., NGUYEN Q.-D., ROUSSEY C., DE VAULX C. & CHANET J.-P. (2018). Ontological Requirement Specification for Smart Irrigation Systems : A SOSA/SSN and SAREF Comparison. In *Proceedings of the 9th International Semantic Sensor Networks Workshop, International Semantic Web Conference*, volume 2213 of *CEUR Workshop Proceedings*, p. 1–16, Monterey, United States.

RIJGERSBERG H., VAN ASSEM M. & TOP J. (2013). Ontology of units of measure and related concepts. *Semantic Web*, **4**(1), 3–13.

SUÁREZ-FIGUEROA M. C., GÓMEZ-PÉREZ A. & FERNANDEZ-LOPEZ M. (2015). The neon methodology framework : A scenario-based methodology for ontology development. *Applied ontology*, **10**(2), 107–145.

SUN J., DE SOUSA G., ROUSSEY C., CHANET J.-P., PINET F. & HOU K. M. (2016). Intelligent Flood Adaptive Context-aware System : How Wireless Sensors Adapt their Configuration based on Environmental Phenomenon Events. *IFSA*, **206**(11), 68–81.

WANG Y. & WANG Y. (2018). Citrus ontology development based on the eight-point charter of agriculture. *Computers and Electronics in Agriculture*, **155**, 359–370.

WANG Y., WANG Y., WANG J., YUAN Y. & ZHANG Z. (2015). An ontology-based approach to integration of hilly citrus production knowledge. *Computers and Electronics in Agriculture*, **113**(C), 24–43.

Traitement des requêtes d'agrégation sur un serveur SPARQL préemptif ⋆

Arnaud Grall[1,2], Thomas Minier[1], Hala Skaf-Molli[1], and Pascal Molli[1]

[1] LS2N – UNIVERSITY OF NANTES, FRANCE
{arnaud.grall,thomas.minier,hala.skaf,pascal.molli}@univ-nantes.fr

[2] GFI Informatique - IS/CIE, Nantes, France arnaud.grall@gfi.fr

1 Introduction

En suivant les principes du web des données, les fournisseurs de données ont publié des milliards de triples en format RDF (Bizer *et al.*, 2009; Schmachtenberg *et al.*, 2014). Il est possible de calculer des statistiques sur ces données en exécutant des requêtes SPARQL d'agrégation ; par exemple le nombre de propriétés par classe (Hasnain *et al.*, 2016), ou encore la durée de vie moyenne de scientifiques célèbres par pays. Cependant, le traitement des requêtes d'agrégation sur les serveurs SPARQL public reste difficile. En effet, la durée d'exécution des requêtes d'agrégation dépasse généralement les limites de temps autorisées par les serveurs SPARQL. On obtient alors des résultats partiels inutilisables dans le cadre de requêtes d'agrégation. (Polleres *et al.*, 2018; Soulet & Suchanek, 2019; Hasnain *et al.*, 2016).

Pour surmonter les limitations de quotas, les fournisseurs de données fournissent, en plus des serveurs SPARQL en ligne, des fichiers de sauvegarde contenant l'ensemble des données. Cependant, la ré-ingestion de milliards de faits RDF sur des ressources locales est extrêmement coûteuse et pose des problèmes de fraîcheur des données.

Récemment, des travaux de recherche ont été menés afin de construire des serveurs SPARQL fonctionnant sans limite de temps comme TPF (Verborgh *et al.*, 2016) ou SaGe (Minier *et al.*, 2019). Cependant, dans ces approches, les données à agréger sont d'abord transférées du serveur de données à un client intelligent qui effectue le calcul d'agrégation. La requête d'agrégation termine, mais le transfert de données est prohibitif.

Dans (Grall *et al.*, 2020), nous montrons comment il est possible d'étendre un serveur SPARQL préemptif avec un opérateur d'agrégation. Ce résultat est basé sur les propriétés de décomposition des fonctions d'agrégation. Dans notre approche, le serveur SPARQL préemptif est capable de calculer des agrégats partiels côté serveur pendant que le client intelligent combine ces agrégats partiels de manière incrémentale pour calculer les résultats finaux. Cette stratégie permet de réduire considérablement le trafic réseau lors des calculs d'agrégation.

2 Agrégation partielle et préemption web

Un serveur web préemptif permet de suspendre l'exécution d'une requête SPARQL après un quantum de temps afin d'en continuer l'exécution plus tard (Minier *et al.*, 2019). Malheureusement, il n'est pas possible de suspendre un opérateur d'agrégation. En effet, un calcul d'agrégation nécessite une structure de données dont la taille est proportionnelle à la taille de l'agrégat i.e. dans le pire cas, des données. Il faudrait donc au moment de la suspension, transmettre cet état du serveur web au client, puis dans le sens inverse au moment de la reprise.

Pour dépasser ce problème, nous calculons des agrégats partiels par quantum. En effet, la préemption web crée naturellement des partitions dans les résultats d'une requête. Comme les

⋆. Cet article est un résumé en français de notre article publié à ESWC2020 (Grall *et al.*, 2020)

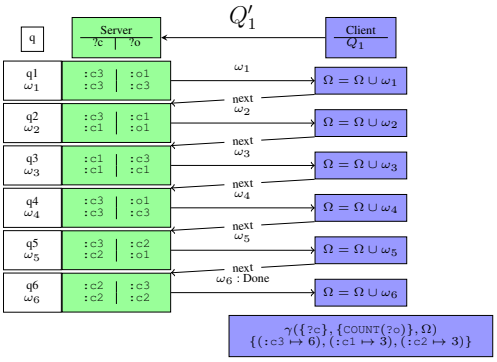
IC 2020

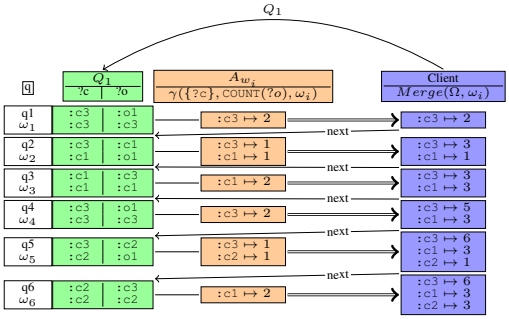
```
:s1 :p1 :o1 .
:s1 :a :c2, :c3.
:s2 :p1 :o1 .
:s2 :a :c1, :c3.
```

(a) $\mathcal{G}_1$

```
SELECT ?c
  (COUNT(?o) AS ?z)
WHERE { ?s :a ?c .
  ?s ?p ?o . ?s :p1 :o1}
GROUP BY ?c
```

(b) Requête SPARQL Q_1





(c) Evaluation de Q_1 sur $\mathcal{G}_1$ avec la préemption(d) Evaluation de Q_1avec l'agrégation partielle Web originale (Minier *et al.*, 2019)

FIGURE 1 – Evaluation des requêtes d'agrégation SPARQL avec la préemption Web

TABLE 1 – Statistics of RDF datasets used in the experimental study

RDF Dataset	# Triples	# Subjects	# Predicates	# Objects	# Classes
BSBM-10	4 987	614	40	1 920	11
BSBM-100	40 177	4 174	40	11 012	22
BSBM-1k	371 911	36433	40	86202	103
DBpedia 3.5.1	153M	6 085 631	35 631	35 201 955	243

fonctions d'agrégation sont décomposables, nous calculons les agrégats partiels par quantum coté serveur et nous les fusionnons ensuite sur le client.

La figure (c) montre l'exécution de $Q1$ sur $\mathcal{G}_1$ sans agrégation partielle. Le corps Q'_1 de la requête est envoyé au serveur, qui renvoie les solutions (ω_i) en fin de chaque quantum (q_i). Tous les couples $<?c, ?o>$ sont donc transférés du serveur au client. Avec l'agrégation partielle (figure (d)), la requête $Q1$ est transmise au serveur, qui renvoie désormais les comptage partiels au lieu des couples. Le transfert de données est considérablement réduit.

3 Etude expérimentale

Nous voulons répondre empiriquement aux questions suivantes : (i) Quel est l'impact de l'agrégation partielle sur le transfert de données ? (ii) Quel est l'impact de l'agrégation partielle sur le temps d'exécution ?

Nous avons implémenté l'agrégation partielle comme une extension du moteur de requêtes SAGE [1]. Toutes les extensions et les résultats expérimentaux sont disponibles sur `https://github.com/folkvir/sage-sparql-void`.

Données et requêtes : nous avons construit un workload (*SP*) de 18 requêtes SPARQL d'agrégation extraites des requêtes de SPORTAL (Hasnain *et al.*, 2016) (requêtes sans ASK ni FILTER). La plupart des requêtes extraites ont le modificateur DISTINCT. Pour étudier l'impact de DISTINCT sur les performances des requêtes agrégées, nous avons défini un

1. `https://sage.univ-nantes.fr`

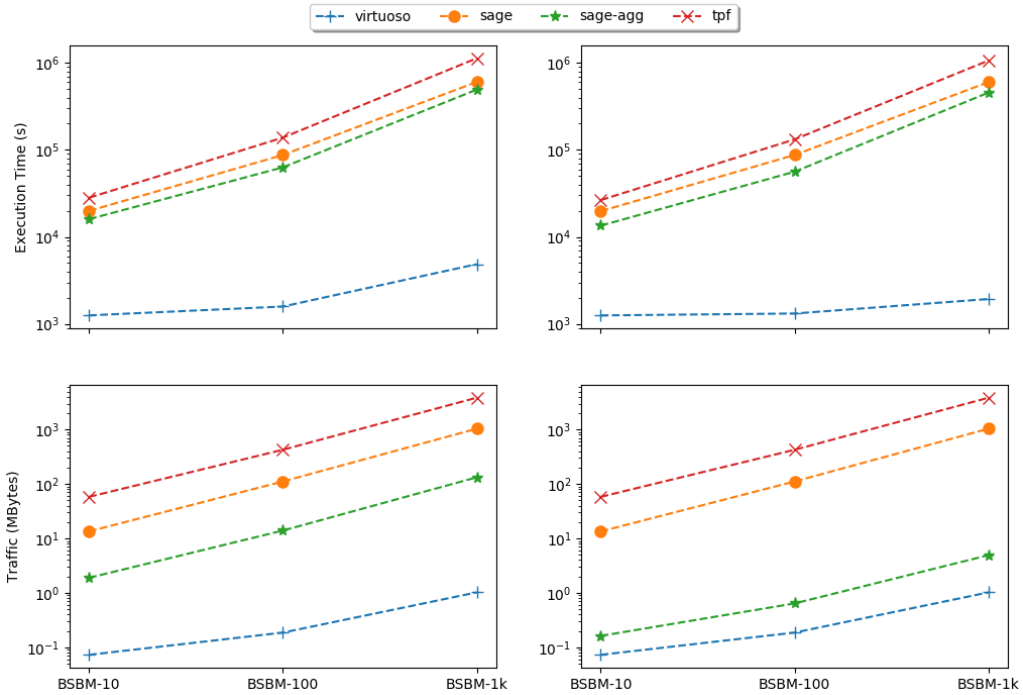


FIGURE 2 – Données transférées et temps d'exécution pour BSBM-10, BSBM-100 et BSBM-1k, lors de l'exécution des workloads *SP* (à gauche) et *SP-ND* (à droite)

nouveau workload, notée *SP-ND*, en supprimant le modificateur DISTINCT des requêtes de *SP*. Nous exécutons les workloads *SP* et *SP-ND* sur Berlin SPARQL Benchmark (BSBM) avec différents tailles et le fragment de DBpedia *v3.5.1*. Les statistiques de jeux de données utilisées sont détaillées dans le tableau 1.

Approches : Nous comparons les approches suivantes :

— SAGE : nous exécutons le moteur de requêtes SAGE query (Minier *et al.*, 2019) avec un quantum de temps de 150 ms et une page de taille maximale de 5000 résultats. Les données sont stockées sur un serveur PostgreSQL, avec des index sur *(SPO)*, *(POS)* et *(OSP)*.

— SAGE-AGG : est notre extension de SAGE avec les opérateurs d'agrégations partielles. Il fonctionne avec la même configuration que SAGE.

— *TPF* : nous exécutons le serveur TPF (Verborgh *et al.*, 2016) (sans cache Web) et le client Communica, en utilisant la taille de page standard de 100 triplets. Les données sont stockées au format HDT.

— *Virtuoso :* nous exécutons Virtuoso SPARQL endpoint (Erling & Mikhailov, 2009) (v7.2.4) **sans quotas** afin de fournir des résultats complets et un transfert de données optimal. Virtuoso est configuré avec *un seul thread* pour une juste comparaison.

Configurations des serveurs : les expérimentations sont menées sur Google Cloud Platform, avec un vCPU standard n1 2 : 2, 7,5 Go de mémoire avec un disque local SSD.

Métriques d'évaluation : Les résultats présentés correspondent à la moyenne obtenue sur trois exécutions successives de workloads. (i) *Données transférées* : est le nombre d'octets transférés au client lors de l'évaluation d'une requête. (ii) *Temps d'exécution* : est le temps entre le début de la requête et la production des résultats finaux par le client.

Résultats expérimentaux

Dans cette section, nous présentons seulement les résultats sur les jeux de données synthétiques (BSBM). Les résultats expérimentaux complets sont détaillés dans (Grall *et al.*, 2020).

IC 2020

La figure 1 présente les résultats en matière de transfert de données et de temps d'exécution pour BSBM-10, BSBM-100 et BSBM-1k. Les graphiques de gauche détaillent les résultats pour SP et les graphiques de droite, les résultats pour SP-ND. Virtuoso sans quota est présenté comme optimal en termes de données transférées et de temps d'exécution. Comme prévu, on observe les pires performances pour TPF. En effet, TPF ne prend pas en charge les projections et les jointures côté serveur. Par conséquent, le transfert de données est énorme même pour de petits ensembles de données. SaGe offre de meilleures performances que TPF principalement parce qu'il prend en charge la projection et les jointures côté serveur. SaGe améliore considérablement le transfert de données mais pas les temps d'exécution. En effet, les agrégations partielles permettent de réduire le transfert de données mais ne permettent pas d'accélérer le scan des données sur disque. En comparant les 2 charges de travail, nous pouvons voir que le traitement des requêtes sans `DISTINCT` (à droite) est beaucoup plus efficace dans le transfert de données qu'avec `DISTINCT` (à gauche). Pour les requêtes `DISTINCT`, les agrégations partielles ne peuvent supprimer que les doublons observés pendant un temps quantique uniquement et non ceux observées lors de l'exécution de la requête.

4 Conclusion

Dans (Grall *et al.*, 2020), nous avons montré qu'il est possible d'exécuter des requêtes d'agrégation sur des serveurs publics sans quotas et avec des transfers de données raisonnables. En perspective, nous projetons d'améliorer les temps d'exécution de requêtes d'agrégation en parallélisant les calculs d'agrégat partiels.

Références

Bizer C., Heath T. & Berners-Lee T. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, **5**(3), 1–22.

Erling O. & Mikhailov I. (2009). RDF support in the virtuoso DBMS. In *Networked Knowledge - Networked Media - Integrating Knowledge Management, New Media Technologies and Semantic Systems*, p. 7–24.

Grall A., Minier T., Skaf-Molli H. & Molli P. (2020). Processing SPARQL Aggregate Queries with Web Preemption. In *17th Extended Semantic Web Conference (ESWC 2020)*, The Semantic Web : ESWC 2020, Herkalion, Greece : Springer, Cham.

Hasnain A., Mehmood Q. & e Zainab ang Aidan Hogan S. S. (2016). SPORTAL : profiling the content of public SPARQL endpoints. *Int. J. Semantic Web Inf. Syst.*, **12**(3), 134–163.

Minier T., Skaf-Molli H. & Molli P. (2019). Sage : Web preemption for public SPARQL query services. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, p. 1268–1278.

Polleres A., Kamdar M. R., Fernández J. D., Tudorache T. & Musen M. A. (2018). A more decentralized vision for linked data. In *Proceedings of the 2nd Workshop on Decentralizing the Semantic Web (DeSemWeb 2018) co-located with ISWC 2018*.

Schmachtenberg M., Bizer C. & Paulheim H. (2014). Adoption of the linked data best practices in different topical domains. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference*, volume 8796, p. 245–260.

Soulet A. & Suchanek F. M. (2019). Anytime large-scale analytics of Linked Open Data. In *18th International Semantic Web Conference, ISWC*.

Verborgh R., Sande M. V., Hartig O., Herwegen J. V., Vocht L. D., Meester B. D., Haesendonck G. & Colpaert P. (2016). Triple pattern fragments : A low-cost knowledge graph interface for the web. *J. Web Sem.*, **37-38**, 184–206.

Modéliser la Compatibilité Entre les Licences

Benjamin Moreau[1,2], Patricia Serrano-Alvarado[2], Matthieu Perrin[2], Emmanuel Desmontils[2]

[1] Opendatasoft
{Name.Lastname}@opendatasoft.com

[2] UMR6004 – Université de Nantes, Nantes, France
{Name.Lastname}@opendatasoft.com

Résumé : Le web facilite la création de ressources par combinaison (code source, services web, données liées, etc.). Pour un producteur de données, choisir une licence adaptée qui protégera sa combinaison de ressources n'est pas facile. En effet, la licence doit être assez restrictive pour se conformer aux licences des ressources combinées, mais assez permissive pour ne pas limiter son usage. Trouver le bon compromis entre restrictivité et usage n'est pas évident. La possibilité d'ordonner automatiquement les licences selon leur compatibilité faciliterait ce choix. Nous proposons donc *CaLi*, un modèle capable d'ordonner partiellement les licences. Notre approche utilise la restrictivité entre les licences pour définir la compatibilité. Notre travail a pour objectif de faciliter et encourager la publication et la réutilisation de ressources protégées par des licences sur le Web.
Mots-clés : Licences, Combinaison, Compatibilité, Web des Données, RDF

1 Introduction

Afin de faciliter la réutilisation des ressources sur le Web, les producteurs de ressources doivent définir leur licence avant de les partager ou de les publier (Seneviratne *et al.*, 2009). Une licence indique précisément les conditions d'utilisation de la ressource, i.e., quelles actions sont autorisées, obligatoires, interdites.

Creative Commons [1] (CC) est une organisation ayant pour objectif de faciliter la diffusion et et le partage d'informations. Elle propose un ensemble de licences [2] fréquemment utilisées sur le web. Cependant, un grand nombre d'autres licences sont mis à disposition et de nombreux producteurs de données créent leur propre licences.

Pour un producteur de données, choisir une licence adaptée qui protégera sa ressource n'est pas facile. En effet, les licences des ressources combinées doivent être compatibles avec la licence, mais aussi permettre la réutilisation de la ressource. Le risque est de choisir une licence trop restrictive qui empêchera la réutilisation de la ressource ou, au contraire, de choisir une licence trop permissive qui ne la protégera pas assez.

Nous pensons qu'un modèle capable d'ordonner les licences selon leur compatibilité faciliterait ce choix. Notre définition de la compatibilité est la suivante : *Une licence l_i est compatible avec une licence l_j si une ressource protégée par l_i peut être protégée par l_j sans violer les conditions d'utilisation de l_i*. Si une licence l_i est compatible avec une licence l_j alors, les ressources protégées par l_i sont réutilisables avec des ressources protégées par l_j. En général, quand une licence l_i est compatible avec une licence l_j, l_j est plus (ou autant) restrictive que l_i. Nous considérons qu'*une licence l_j est plus (ou autant) restrictive qu'une licence l_i si l_j autorise au plus les mêmes permissions et impose au moins les mêmes obligations et interdictions*.

Dans la majorité des cas, quand l_i est moins restrictive que l_j alors, l_i est compatible avec l_j. La Figure 1 représente trois licences Creative Commons en RDF décrites à l'aide

1. https://creativecommons.org/
2. https://creativecommons.org/licenses/

de l'ontologie ODRL [3]. Dans cet exemple, (a) est moins restrictive que (b), (b) est moins restrictive que (c) et, par transitivité, (a) est moins restrictive que (c). Nous remarquons que (a) est compatible avec (b) et (c), mais (b) n'est pas compatible avec (c). Dans cet exemple, la restrictivité entre (b) et (c) n'est pas accompagnée d'une relation de compatibilité. Ceci est dû à la sémantique de l'action *DerivativeWorks* qui interdit la distribution d'une modification de la ressource sous une autre licence [4]. En effet, selon la sémantique de leurs actions, la relation de restrictivité entre deux licences n'implique pas forcement une relation de compatibilité.

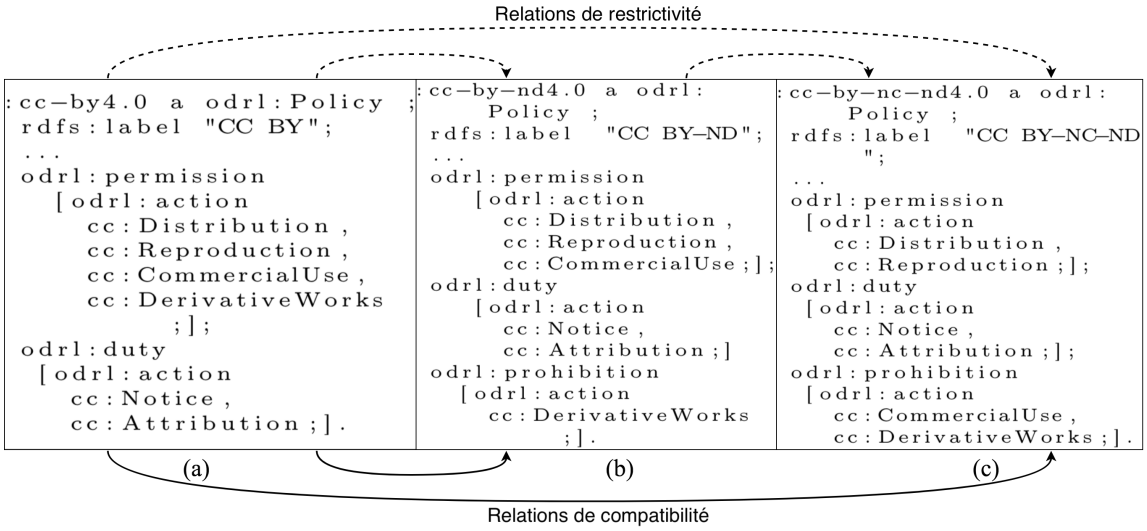


FIGURE *1: Trois licences Creative Commons décrites en RDF.*

Notre question de recherche est : *Soit une licence l_i, comment positionner automatiquement l_i dans un ensemble de licences selon leur compatibilité ?* Notre défi est de généraliser la relation d'ordre entre les licences tout en prenant en compte l'influence de la sémantique des actions.

Nous proposons CaLi (ClAssification of LIcenses), un modèle basé sur un treillis permettant d'ordonner des licences. CaLi utilise la relation de restrictivité et des contraintes pour définir la compatibilité entre les licences.

L'article complet (Moreau *et al.*, 2019b) a été publié dans les actes de l'Extended Semantic Web Conférence (ESWC 2019). Il contient les algorithmes complets et une évaluation expérimentale des algorithmes implémentés. Un papier de démonstration a aussi été publié (Moreau *et al.*, 2019a) et illustre l'utilisation de notre modèle dans un moteur de recherche basé sur un ordre CaLi. Ce dernier est capable de trouver les ressources dont les licences sont compatibles avec une licence particulière.

Dans la suite, nous présentons brièvement comment ordonner des licences à l'aide du modèle CaLi, ainsi que nos expérimentations.

2 Ordonner des licences avec le modèle CaLi

Dans une licence, les actions (e.g., lecture, modification, etc.) peuvent être réparties entre ce que nous appelons des *statuts*, e.g., permissions, obligations, interdictions, etc. Dans l'exemple d'introduction, nous considérons que les permissions sont moins restrictives que les obligations lesquelles sont moins restrictives que les interdictions. L'ordre de restrictivité entre les statuts est subjectif et peut varier en fonction du contexte.

3. https://www.w3.org/TR/odrl-model/
4. https://wiki.creativecommons.org/wiki/Wiki/cc_license_compatibility

Nous avons remarqué que si deux licences ont une relation de restrictivité alors il est possible qu'elles aient aussi une relation de compatibilité. L'avantage de la relation de restrictivité entre les licences est qu'elle peut être obtenue automatiquement selon les statuts des actions. Inspiré des travaux sur le controle d'accès basé sur des treillis Davey & Priestley (2002), nous définissons une relation de restrictivité entre les licences.

Pour identifier la compatibilité entre les licences, nous affinons la relation de restrictivité avec des contraintes. L'objectif est de prendre en compte la sémantique des actions. Les contraintes distinguent aussi les licences valides des non-valides. Nous considérons qu'une licence est non-valide si aucune ressource ne peut être protégée par cette dernière., e.g., une licence qui, simultanément, autorise l'action *Derive* [5] et interdit l'action *DerivativeWorks* [6].

2.1 Modèle CaLi

Inspiré par les modèles de contrôle d'accès basés sur des treillis, le modèle CaLi est un tuple $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ capable d'ordonner partiellement des licences, tel que :

1. $\mathcal{A}$ est un ensemble d'*actions* (e.g., *lire, modifier, distribuer*, etc.) ;
2. $\mathcal{LS}$ est *un treillis de restrictivité des statuts* définissant (i) l'ensemble des statuts possibles (e.g., permission, obligation, interdiction, etc.) d'une action dans une licence et (ii) la relation de restrictivité $\leqslant_S$ entre ces statuts. La Figure 2 montre plusieurs exemples de treillis de restrictivité de statuts ;
3. $C_{\rightarrow}$ est un ensemble de *contraintes sur la compatibilité* permettant d'identifier les relations de restricitivité entre deux licences qui sont aussi des relations de compatibilité ;
4. Enfin, $C_{\mathcal{L}}$ est un ensemble de *contraintes sur les licences* permettant d'identifier les licences qui ne sont pas valides.

$\mathcal{L}_{\mathcal{A},\mathcal{LS}}$ défini l'ensemble exhaustif des licences exprimables avec $\mathcal{A}$ et $\mathcal{LS}$. $(\mathcal{L}_{\mathcal{A},\mathcal{LS}}, \leqslant_{\mathcal{R}})$ est le treillis de restrictivité des licences définissant la relation de restrictivité $\leqslant_{\mathcal{R}}$ sur l'ensemble de licences $\mathcal{L}_{\mathcal{A},\mathcal{LS}}$. L'ensemble de contraintes $C_{\mathcal{L}}$ identifient les licences non-valides. Si deux licences valides ont une relation de restrictivité alors, il est possible qu'elles aient une relation de compatibilité. L'ensemble de contraintes $C_{\rightarrow}$ identifient les relations de compatibilité parmi les relations de restrictivité.

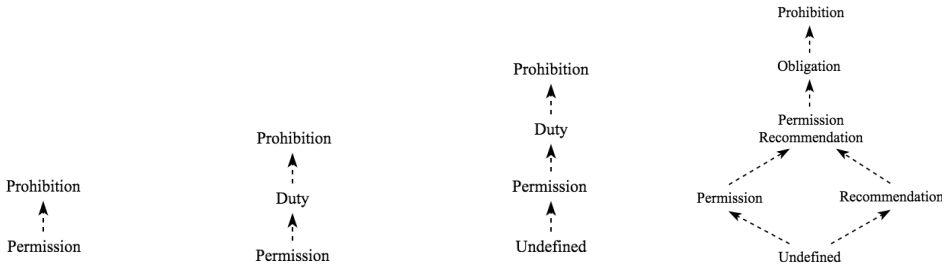


FIGURE *2: Exemples de treillis de restrictivité de statuts ($\mathcal{LS}$). Les flèches pointillées représentent la restrictivité. 2a est inspiré par les systèmes de fichiers où les actions peuvent être soit autorisées, soit interdites et où l'interdiction de lire un fichier est plus restrictive que la permission de le lire. 2b est basé sur les licences CC. 2c est inspiré par le vocabulaire ODRL où les actions peuvent être autorisées, obligées, interdites ou non spécifiées (i.e.,* undefined*). Fig. 2d montre un treillis de restrictivité où une action recommandée ou autorisée est moins restrictive que la même action lors qu'elle est autorisée et recommandée.*

5. `https://www.w3.org/TR/odrl-vocab/#term-derive`
6. `https://www.w3.org/TR/odrl-vocab/#term-DerivativeWorks`

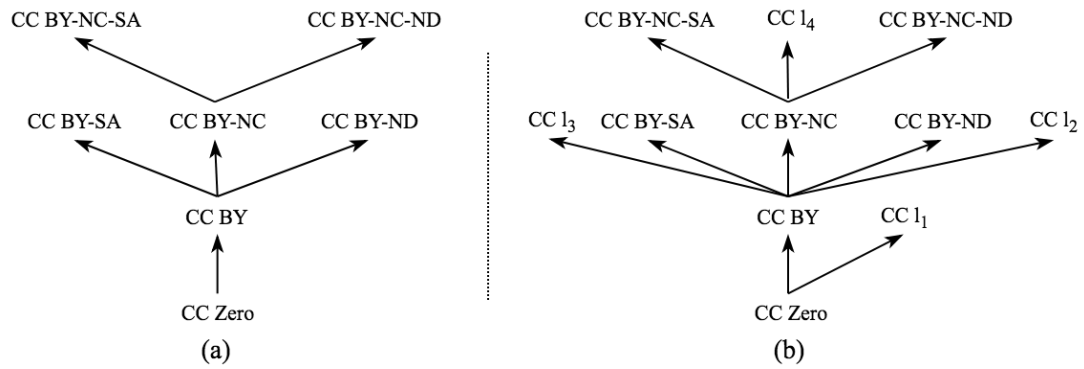


FIGURE *3: Sous-graphes de compatibilité de CC_CaLi : (a) contient les 7 licences officielles CC et (b) contient en plus d'autres licences valides.*

2.2 Un ordre CaLi pour licences de Creative Commons

Creative Commons propose 7 licences juridiquement vérifiées, gratuites et faciles à comprendre. Elles sont largement utilisées sur le Web. Ces licences utilisent 7 actions qui peuvent être autorisées, obligées ou interdites (cf. Figure 2b). Dans notre modèle CC_CaLi, nous modélisons un ordre de compatibilité de licences complet pour toutes les licences possibles et valides utilisant les 7 actions de Creative Commons.

Ainsi, CC_CaLi est un modèle CaLi $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ tel que :

— $\mathcal{A}$ est l'ensemble des 7 actions *{cc :Distribution, cc :Reproduction, cc :DerivativeWorks, cc :CommercialUse, cc :Notice, cc :Attribution, cc :ShareAlike}* ;

— $\mathcal{LS}$ est le treillis de restrictivité des statuts de la Figure 2b ;

— $C_{\mathcal{L}}, C_{\rightarrow}$ sont les ensembles de contraintes issues du vocabulaire CC et réinterprétées sous forme de fonctions.

$C_{\mathcal{L}} = \{\omega_{\mathcal{L}_1}, \omega_{\mathcal{L}_2}\}$ invalide une licence (1) quand *cc :CommercialUse* est obligée, ou (2) quand *cc :ShareAlike* est interdit :

$$\omega_{\mathcal{L}_1}(l_i) = \begin{cases} Faux & \text{si } l_i(\textit{cc :CommercialUse}) = \text{Duty;} \\ Vrai & \text{sinon.} \end{cases}$$

$$\omega_{\mathcal{L}_2}(l_i) = \begin{cases} Faux & \text{si } l_i(\textit{cc :ShareAlike}) = \text{Prohibition;} \\ Vrai & \text{sinon.} \end{cases}$$

$C_{\rightarrow} = \{\omega_{\rightarrow_1}, \omega_{\rightarrow_2}\}$ identifie les relations de restrictivité qui ne sont pas des relations de compatibilité (1) quand *cc :ShareAlike* est obligée ou (2) quand *cc :DerivativeWorks* est interdit. En effet, *cc :ShareAlike* exige que la ressource redistribuée soit protégée par la même licence. L'interdiction de *cc :DerivativeWorks*, n'autorise pas la distribution d'une ressource ou licence modifiée. Ces deux actions sont donc en contradiction avec la définition de la compatibilité.

$$\omega_{\rightarrow_1}(l_i, l_j) = \begin{cases} Faux & \text{si } l_i(\textit{cc :ShareAlike}) = \text{Duty;} \\ Vrai & \text{sinon.} \end{cases}$$

$$\omega_{\rightarrow_2}(l_i, l_j) = \begin{cases} Faux & \text{si } l_i(\textit{cc :DerivativeWorks}) = \text{Prohibition;} \\ Vrai & \text{sinon.} \end{cases}$$

D'autres contraintes peuvent être définies pour être encore plus proche des propriétés du langage CC REL [7]. Mais pour notre exemple, ces contraintes sont suffisantes.

7. `https://creativecommons.org/ns`

Nous représentons les licences sous forme d'une fonction $l : \mathcal{A} \rightarrow S$, où $\mathcal{A}$ est un ensemble d'actions et $\mathcal{LS} = (S, \leqslant_S)$ est un treillis de restrictivité de statuts. Par exemple, la licence CC BY-NC-SA peut être définie comme :

$$CCBYNC(a) = \begin{cases} \text{Permission} & \text{if } a \in \{\textit{cc :Distribution, cc :Reproduction} \\ & \quad \textit{cc :DerivativeWorks, cc :ShareAlike}\}; \\ \text{Duty} & \text{if } a \in \{\textit{cc :ShareAlike, cc :Notice, cc :Attribution}\}; \\ \text{Prohibition} & \text{if } a \in \{\textit{cc :CommercialUse}\}. \end{cases}$$

La taille du treillis de restrictivité CC_CaLi est de 3^7 licences, mais le nombre de licences valides est de 972. Ceci est dû aux contraintes $C_{\mathcal{L}}$. En effet, il est possible d'avoir 5 actions dans n'importe quel statuts et 2 actions (*cc :CommercialUse* et *cc :ShareAlike*) dans uniquement 2 statuts : $3^5 * 2^2$.

La Figure 3 montre deux sous-graphes de CC_CaLi. 3a montre uniquement les 7 licences officielles de CC tandis que 3b montre en plus d'autres licences valides. Grâce à $\omega_{\rightarrow_1}$, la relation de restrictivité entre CC BY-SA et CC BY-NC-SA n'est pas identifiée comme une relation de compatibilité. Et grâce à $\omega_{\rightarrow_2}$, la relation de restrictivité entre CC BY-ND et CC BY-NC-ND n'est pas identifié comme une relation de compatibilité non plus. A noter qu'une licence interdisant *cc :DerivativeWorks* n'est pas compatible avec elle même.

La relation de compatibilité de la Figure 3a est conforme au graphe de compatibilité qu'il est possible de générer à partir de la table de compatibilité des licences Creative Commons [8].

3 Implementation et expérimentation

Notons que la taille d'un ordre CaLi est exponentiel $|\mathcal{LS}|^{|\mathcal{A}|}$. Cependant, il n'est pas nécessaire de générer le treillis complet pour l'utiliser. Dans l'article (Moreau *et al.*, 2019b), nous proposons un algorithme de tri pouvant ordonner n'importe quel sous-ensemble de $\mathcal{L}_{\mathcal{A},\mathcal{LS}}$ en approximativement $n^2/2$ comparaisons, n étant le nombre de licences dans notre sous-ensemble i.e., $O(n^2)$. Cet algorithme est capable d'insérer une licence dans un ensemble de licences partiellement ordonné en temps linéaire $O(n)$.

L'implémentation en Python de notre algorithme et les détails de nos expériences sont disponibles sur GitHub [9].

Nous montrons l'utilité de notre modèle avec un moteur de recherche de jeux de données basé sur des licences [10]. Il utilise l'ordre $ODRL\_CaLi$ capable d'ordonner par compatibilité des licences décrites à l'aide de l'ontologie ODRL.

$ODRL\_CaLi$, est un modèle CaLi $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ tel que :

— $\mathcal{A}$ est l'ensemble des 72 actions proposées dans ODRL [11] ;

— $\mathcal{LS}$ est le treillis de restrictivité des statuts où (i) les statuts possibles sont les trois règles ODRL : Permission, Duty (i.e. obligation), Prohibition (i.e. interdiction) [12] ou Undefined (pour les actions n'apparaissant pas dans la licence), et où (ii) les relations de restrictivité entre les statuts es celui de la Figure 2c ;

— Enfin, $C_{\mathcal{L}}, C_{\rightarrow}$ sont les ensembles de contraintes issues du vocabulaire ODRL.

$C_{\mathcal{L}} = \{\omega_{\mathcal{L}_1}, \omega_{\mathcal{L}_2}\, \omega_{\mathcal{L}_3}\}$ invalide une licence de la même manière que dans l'odre CC_CaLi avec en plus (3) quand la sémantique d'une action autorisée ou obligée est incluse (*odrl :includedIn*) dans une action interdite (e.g., si *CommercialUse* est permis alors, l'action *use* ne

8. https://wiki.creativecommons.org/wiki/Wiki/cc_license_compatibility
9. https://github.com/benjimor/CaLi-Search-Engine
10. http://cali.priloo.univ-nantes.fr/ld/
11. https://www.w3.org/TR/odrl-vocab/#actionConcepts
12. https://www.w3.org/TR/odrl-model/#rule

doit pas être interdite, car l'usage commerciale d'une ressource implique son utilisation) :

$$\omega_{\mathcal{L}_3}(l_i) = \begin{cases} Faux & \text{si } a_i \text{ odrl :includedIn } a_j \\ & \text{et } (l_i(a_i) = \text{Permission ou } l_i(a_i) = \text{Duty}) \\ & \text{et } l_i(a_j) = \text{Prohibition;} \\ Vrai & \text{sinon.} \end{cases}$$

Un exemple de sous-ensemble de licences partiellement ordonné selon l'ordre $ODRL\_CaLi$ [13] alimente notre moteur de recherche de jeux de données. Il contient un ordre partiel avec les licences les plus utilisées sur les plateformes DataHub [14] et Opendatasoft Data Network [15]. Le même moteur de recherche est mis à disposition pour de dépôts GitHub et contient les licences les plus utilisées sur cette plateforme [16].

4 Conclusion

Nous proposons un modèle basé sur un treillis pour définir la relation de compatibilité entre licences. Notre approche se base sur la relation de restrictivité affinée avec des contraintes pour prendre en compte la sémantique des actions. Nous avons montré la faisabilité de notre approche à travers deux ordres CaLi, le premier utilisant le vocabulaire Creative Commons et le second utilisant le vocabulaire ODRL. Nous avons expérimenté la production d'ordres CaLi avec l'implementation d'un algorithme de tri par insertion dont la complexité est de $n^2/2$ comparaisons. Nous avons implementé le prototype d'un moteur de recherche de jeux de données basé sur les licences montrant l'utilité de notre contribution. Notre modèle de compatibilité n'a pas l'intention de fournir un avis juridique mais il permet d'exclure les licences qui violerait une licence particulière.

Remerciements

Les auteurs remercient Margo Bernelin et Sonia Desmoulin-Canselier (laboratoire de Droit et Changement Social - UMR CNRS 6297) pour nos discussions sur ce travail.

Références

DAVEY B. A. & PRIESTLEY H. A. (2002). *Introduction to Lattices and Order*. Cambridge university press.

MOREAU B., SERRANO-ALVARADO P., PERRIN M. & DESMONTILS E. (2019a). A License-Based Search Engine. In *Extended Semantic Web Conference (ESWC), Poster&Demo*.

MOREAU B., SERRANO-ALVARADO P., PERRIN M. & DESMONTILS E. (2019b). Modelling the Compatibility of Licenses. In *Extended Semantic Web Conference (ESWC)*.

SENEVIRATNE O., KAGAL L. & BERNERS-LEE T. (2009). Policy-Aware Content Reuse on the Web. In *International Semantic Web Conference (ISWC)*.

13. http://cali.priloo.univ-nantes.fr/ld/graph
14. https://old.datahub.io/
15. https://data.opendatasoft.com/pages/home/
16. http://cali.priloo.univ-nantes.fr/rep/graph

Interoperabilité et raisonnement dans le Web Sémantique des objets: le projet CoSWoT

Francesco Antoniazzi[2], Ghislain Atemezing[6], Fabien Badeig[2], Mahdi Bennara[2], Stephan Bernard[4], Pierre-Antoine Champin[3], Jean-Pierre Chanet[4], Christophe Gravier[5], Yann Gripay[1], Frédérique Laforest ⋆,[1], Maxime Lefrançois[2], Lionel Médini[3], Laure Moiroux[4], Catherine Roussey[4], Sylvie Servigne[1], Kamal Singh[5], Julien Subercaze[5], Antoine Zimmermann[2]

[1] Univ Lyon, INSA Lyon, LIRIS, CNRS UMR5205, F-69622, Villeurbanne, France
{firstname.lastname}@insa-lyon.fr,

[2] Mines Saint-Etienne, Univ Lyon, Univ Jean Monnet, IOGS, CNRS, UMR 5516, LaHC, Institut Henri Fayol, F - 42023 Saint-Etienne, France
{firstname.lastname}@emse.fr,

[3] Univ Lyon, Univ Lyon 1, LIRIS, CNRS UMR5205, F-69622, Villeurbanne, France
{firstname.lastname}@univ-lyon1.fr,

[4] Univ Clermont Auvergne, INRAe, UR TSCF, F-63178 Aubière, France
{firstname.lastname}@inrae.fr,

[5] Univ Jean Monnet, IOGS, CNRS, UMR 5516, LaHC, F - 42023 Saint-Etienne, France
{firstname.lastname}@univ-st-etienne.fr,

[6] Mondeca, Paris, France
ghislain.atemezing@mondeca.com

Résumé : Cet article présente le contexte et les objectifs du projet ANR nommé Constrained Semantic Web of Things (CoSWoT). CoSWoT a pour objectif de proposer une architecture logicielle distribuée compatible WoT et embarquée sur des dispositifs contraints. Cette architecture a deux caractéristiques principales : (1) elle utilisera des modèles de connaissances à base de graphes pour déclarer la logique applicative des dispositifs et la sémantique des messages échangés ; (2) les dispositifs auront des capacités de raisonnement afin de répartir les tâches de traitement entre eux. Le développement d'applications WoT sera simplifié : notre plateforme permettra le développement et l'exécution d'applications intelligentes et décentralisées du WoT malgré l'hétérogénéité des dispositifs connectés. La plateforme proposée sera testée sur plusieurs cas d'utilisation dans le bâtiment intelligent et en agriculture numérique.

Mots-clés : Web des Objets sémantique, ontologie, description de service , capteur, actionneur, raisonnement incrémental, raisonnement distribué, objet contraint

1 Contexte

Le Web des Objets (WoT) est le résultat de l'intégration dans le Web d'objets communicants hétérogènes, potentiellement mobiles, connectés par intermittence et présentant des capacités limitées. De nouveaux services applicatifs innovants peuvent être envisagés pour l'utilisateur pour peu que ces objets puissent se découvrir, inter-opérer, et prendre des décisions collectivement. Les applications du WoT concernent l'agriculture numérique, le bâtiment intelligent, les villes intelligentes, la gestion de l'énergie et de l'eau, la santé, etc. Dans le domaine de l'agriculture numérique, des objets hétérogènes fixes ou mobiles sur

⋆. Contact author, coordinator of the ANR CoSWoT project

les parcelles cultivées peuvent capter et échanger des informations puis mener des raisonnements pour construire une vue analytique d'un champ et prendre des décisions, par exemple comment irriguer le champ de manière optimale. Les graphes de connaissances sont des représentations formelles obtenues par l'unification de données hétérogènes et distribuées qui ont été enrichies de leur contexte d'acquisition, et liées (Hogan *et al.*, 2020). Ces graphes permettent aussi de raisonner et prendre des décisions. Les modèles et technologies du Web sémantique forment un socle théorique privilégié pour les graphes de connaissances émergeant de l'échange, du stockage, du traitement et du raisonnement sur des données dans le Web des Objets.

2 Objectifs

L'objectif scientifique du projet CoSWoT est de proposer une architecture logicielle embarquée sur des objets communicants contraints, avec deux caractéristiques principales : (1) elle utilisera des modèles de connaissances à base de graphes pour spécifier de manière déclarative la logique applicative des objets contraints ainsi que les messages qu'ils échangent (2) elle donnera aux objets, malgré leurs contraintes, une capacité de raisonnement sur ces graphes de connaissances pour déporter les traitements de données au bord de l'architecture du système. Ainsi, notre prototype permettra la construction et l'exécution d'applications WoT intelligentes et décentralisées malgré l'hétérogénéité des objets. Ces applications WoT reposeront sur une plateforme hébergeant les services nécessaires. Nous évaluerons donc les solutions existantes (plateforme du projet ANR ASAWoO (Médini *et al.*, 2017; Mrissa *et al.*, 2015), Servient du groupe WoT du W3C (Kovatsch *et al.*, 2020), Interworking proxy de la spécification OneM2M de l'ETSI (ETSI, 2017), etc.). Nous sélectionnerons la solution la plus adaptée à nos contraintes techniques, et développerons une version étendue aux verrous scientifiques de notre projet.

Le premier verrou scientifique concerne l'utilisation des graphes de connaissances comme modèle de données généralisé dans les échanges entre objets hétérogènes. Les objets consomment et transmettent des messages avec des syntaxes et modèles de données variés. Chaque constructeur ou consortium développe sa norme d'échange des données. Nous participons à la convergence de certaines initiatives notamment au sein du W3C et de l'ETSI (Sun *et al.*, 2016; Meddeb, 2016; Lefrançois, 2017; Roussey *et al.*, 2020). Nous avons initié des travaux pour étudier comment des objets peuvent être rendus interopérables sémantiquement malgré leurs hétérogénéités, justement en se basant sur l'utilisation généralisée de modèles à base de graphes de connaissances (Lefrançois *et al.*, 2017; Lefrançois, 2017). Ces travaux sont notamment issus du projet ANR OpenSensingCity et du projet ITEA2 Smart Energy Aware Systems, primé *ITEA Award of Excellence 2017*. Des questions de recherche subsistent notamment concernant (i) l'adéquation des modèles de graphes de connaissances existants pour le domaine d'application envisagé ; (ii) l'applicabilité des principes théoriques proposés à une variété de protocoles et standards existants, potentiellement basés sur la génération ou la consommation de flux de données ; (iii) la découverte des objets hétérogènes, des services qu'ils exposent, et de comment les solliciter.

Le second verrou concerne le raisonnement incrémental embarqué et distribué, qui permettra de doter les objets connectés de compétences de raisonnement compatibles avec leurs capacités, et intégrant les nouvelles données au fur et à mesure (Motik *et al.*, 2012; Barbieri *et al.*, 2010; Kazakov & Klinov, 2013; Chevalier *et al.*, 2015, 2016). Nous avons une première expérience pour embarquer des descriptions sémantiques sur des objets contraints avec Hydra (Rojas *et al.*, 2016) mais les principes de distribution restent à définir. Il faut aussi développer des approches et des outils incluant les apports du Web sémantique dans les nouvelles architectures décentralisées où les données sont traitées à la source. Dans (Terdjimi *et al.*, 2015, 2016), notre raisonneur incrémental HyLAR issu du projet ANR ASAWoO déploie les tâches de raisonnement indifféremment côté serveur ou côté client, et permet la découverte de fonctionnalités exposées (Médini, 2016). Dans le projet FSN OpenCloudWare, nous avons défini Slider (Chevalier *et al.*, 2015, 2016) pour optimiser l'empreinte système du raisonnement incrémental en mémoire et en calcul centralisé. De nombreux travaux dans la littérature proposent des optimisations du raisonnement proches du matériel, comme (Munoz *et al.*, 2007;

Neumann & Weikum, 2008; Goodman & Mizell, 2010; Hoeksema & Kotoulas, 2011; Gurajada *et al.*, 2014) ou notre raisonneur Inferray (Subercaze *et al.*, 2016), mais elles doivent être adaptées au raisonnement distribué pour le WoT, ajoutant de nouvelles contraintes comme la limitation des volumes échangés. La solution envisagée n'est pas unique, mais devra être adaptable aux compétences de chaque objet.

Côté applications, l'agriculture fait face à plusieurs transitions. La France a ouvert et échange des données d'agriculture (Bournigal, 2017). Les projets européens Igreen, Foodie ou SmartAgriFood standardisent l'échange de données agricoles avec des modèles de graphes de connaissances. Des plateformes WoT émergent dans le domaine agricole (Jayaraman *et al.*, 2016; Lehmann *et al.*, 2012), avec des systèmes sensibles au contexte et capables de raisonner. L'INRAe a démontré l'apport des graphes de connaissances (Sun *et al.*, 2016) et propose des jeux de données en agriculture (Roussey *et al.*, 2020) grâce à l'acquisition de données de capteurs sur leur site expérimental au sein de leur AgroTechnoPôle. Le projet CoSWoT passe d'une architecture centralisée à une architecture distribuée, où des objets hétérogènes pourront, en fonction de leurs capacités, participer à l'interprétation des données. Dès la fin de l'année 2, nous réaliserons des expérimentations, de durées et complexités croissantes : communication machine mobile/capteur en champ pour la prise de décision de l'accès à la parcelle en fonction de l'état du sol, communication machine mobile/capteur en champ pour déterminer la vitesse de progression de la machine en fonction de la compaction du sol et de l'état des cultures, jusqu'à la prise de décision et l'automatisation de l'irrigation d'une parcelle cultivée sur une période de plusieurs mois, en fonction de l'état du sol, de la culture et des prévisions météorologiques.

Pour plus d'information sur l'avancée de ce projet, vous pouvez consulter le site web `https://coswot.gitlab.io/`

3 Remerciements

Le projet CoSWoT est financé par l'agence nationale de la recherche sous la référence ANR-19-CE23-0012.

Références

BARBIERI D. F., BRAGA D., CERI S., DELLA VALLE E. & GROSSNIKLAUS M. (2010). Incremental Reasoning on Streams and Rich Background Knowledge. In *The Semantic Web : Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part I*, p. 1–15 : Springer.

BOURNIGAL J.-M. (2017). *AgGate - Portail de données pour l'innovation en agriculture*. Rapport interne, Ministère de l'agriculture, de l'agroalimentaire et de la forêt.

CHEVALIER J., SUBERCAZE J., GRAVIER C. & LAFOREST F. (2015). Slider : An Efficient Incremental Reasoner. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, p. 1081–1086.

CHEVALIER J., SUBERCAZE J., GRAVIER C. & LAFOREST F. (2016). Incremental and Directed Rule-Based Inference on RDFS. In *Database and Expert Systems Applications - 27th International Conference, DEXA 2016, Porto, Portugal, September 5-8, 2016, Proceedings, Part II*, p. 287–294 : Springer.

ETSI (2017). *SmartM2M ; Smart Appliances ; Reference Ontology and oneM2M Mapping*. Technical Specification 103 264 V2.1.1, ETSI.

GOODMAN E. L. & MIZELL D. (2010). Scalable in-memory RDFS closure on billions of triples. In *Proceedings of the 6th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2010)*, p. 17–31 : CEUR.

GURAJADA S., SEUFERT S., MILIARAKI I. & THEOBALD M. (2014). TriAD : a distributed shared-nothing RDF engine based on asynchronous message passing. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, p. 289–300 : ACM.

HOEKSEMA J. & KOTOULAS S. (2011). High-performance Distributed Stream Reasoning using S4. In *Ordring Workshop at ISWC*.

HOGAN A., BLOMQVIST E., COCHEZ M., D'AMATO C., DE MELO G., GUTIERREZ C., LABRA GAYO J. E., KIRRANE S., NEUMAIER S., POLLERES A., NAVIGLI R., NGONGA NGOMO

IC 2020

A.-C., RASHID S. M., RULA A., SCHMELZEISEN L., SEQUEDA J., STAAB S. & ZIMMERMANN A. (2020). *Knowledge Graphs*. ArXiv technical report.

JAYARAMAN P. P., YAVARI A., GEORGAKOPOULOS D., MORSHED A. & ZASLAVSKY A. (2016). Internet of Things Platform for Smart Farming : Experiences and Lessons Learnt. *Sensors*, **16**(11), 1884.

KAZAKOV Y. & KLINOV P. (2013). Incremental Reasoning in OWL EL without Bookkeeping. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, p. 232–247 : Springer.

KOVATSCH M., MATSUKURA R., LAGALLY M., KAWAGUCHI T., TOUMURA K. & KAJIMOTO K. (2020). *Web of Things (WoT) Architecture*. W3c recommendation, W3C.

LEFRANÇOIS M. (2017). Planned ETSI SAREF Extensions based on the W3C&OGC SOSA/SSN-compatible SEAS Ontology Pattern. In *Joint Proceedings of SEMANTiCS 2017 Workshops* : CEUR.

LEFRANÇOIS M., ZIMMERMANN A. & BAKERALLY N. (2017). A SPARQL Extension for Generating RDF from Heterogeneous Formats. In *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, p. 35–50 : Springer.

LEHMANN R. J., REICHE R. & SCHIEFER G. (2012). Future internet and the agri-food sector : State-of-the-art in literature and research. *Computers and Electronics in Agriculture*, **89**, 158–174.

MEDDEB A. (2016). Internet of things standards : who stands out from the crowd ? *IEEE Communications Magazine*, **54**(7), 40–47.

MÉDINI L. (2016). An Avatar-based Workflow for the Semantic Web of Things. In *W3C Track @WWW 2016*.

MÉDINI L., MRISSA M., KHALFI E.-M., TERDJIMI M., LE SOMMER N., CAPDEPUY P., JAMONT J.-P., OCCELLO M. & TOUSEAU L. (2017). Building a Web of Things with Avatars : A comprehensive approach for concern management in WoT applications. In *Managing the Web of Things : Linking the Real World to the Web*, p. 151–180. Morgan Kaufmann.

MOTIK B., HORROCKS I. & KIM S. M. (2012). Delta-reasoner : a semantic web reasoner for an intelligent mobile platform. In *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, p. 63–72 : ACM.

MRISSA M., MÉDINI L., JAMONT J.-P., LE SOMMER N. & LAPLACE J. (2015). An Avatar Architecture for the Web of Things. *IEEE Internet Computing*, **19**(2), 30–38.

MUNOZ S., PÉREZ J. & GUTIERREZ C. (2007). Minimal Deductive Systems for RDF. In *The Semantic Web : Research and Applications, 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007, Proceedings*, p. 53–67 : Springer.

NEUMANN T. & WEIKUM G. (2008). RDF-3X : a RISC-style engine for RDF. *PVLDB*, **1**(1), 647–659.

ROJAS R., MÉDINI L. & CORDIER A. (2016). Toward Constrained Semantic WoT. In *Proceedings of the Seventh International Workshop on the Web of Things, WoT 2016, Stuttgart, Germany, November 7, 2016*, p. 31–37.

ROUSSEY C., BERNARD S., ANDRÉ G. & BOFFETY D. (2020). Weather Data Publication on the LOD using SOSA/SSN Ontology. *Semantic Web journal*. To appear in 2020.

SUBERCAZE J., GRAVIER C., CHEVALIER J. & LAFOREST F. (2016). Inferray : fast in-memory RDF inference. *PVLDB*, **9**(6), 468–479.

SUN J., DE SOUSA G., ROUSSEY C., CHANET J.-P., PINET F. & HOU K.-M. (2016). Intelligent Flood Adaptive Context-aware System : How Wireless Sensors Adapt their Configuration based on Environmental Phenomenon Events. *Sensors & Transducers*, **206**(11), 68.

TERDJIMI M., MÉDINI L. & MRISSA M. (2015). HyLAR : Hybrid Location-Agnostic Reasoning. In *Proceedings of the ESWC Developers Workshop 2015 co-located with the 12th Extended Semantic Web Conference (ESWC 2015), Portorož, Slovenia, May 31, 2015*, p. 1–6 : CEUR.

TERDJIMI M., MÉDINI L. & MRISSA M. (2016). HyLAR+ : Improving Hybrid Location-Agnostic Reasoning with Incremental Rule-based Update. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, p. 259–262.

Catégorisation des méthodes de classification fondées sur l'Analyse de Concepts Formels

Hayfa Azibi[1], Nida Meddouri[1,2], Mondher Maddouri[1,3]

[1] LIPAH, Faculté des Sciences de Tunis, Université El-Manar , Tunisie
hayfa.azibi@fst.utm.tn

[2] GREYC-CNRS UMR 6072, Université Caen Normandie, France
nida.meddouri@unicaen.fr

[3] CoB, Université de Jeddah, Arabie Saoudite
maddourimondher@yahoo.fr

Résumé : Les deux dernières décennies ont vu le développement de plusieurs méthodes de classification basées sur l'analyse de concepts formels (ACF). Dans cet article, nous présentons trois catégories de méthodes de classification basées sur l'ACF : des approches basées sur un classifieur unique, des approches basées sur une combinaison de classifieurs générés par une méthode d'ensemble et des approches basées sur un classifieur distribué.
Mots-clés : Intelligence artificielle, Fouille de données, Apprentissage automatique, Classification supervisée, Analyse de concepts formels, Méthodes d'ensemble, Big data.

1 Introduction

L'explosion du volume et la rapidité de la croissance des données ont introduit plusieurs défis dans de nombreux problèmes d'apprentissage du monde réel. La classification supervisée est une tâche de l'apprentissage automatique. L'objectif d'un problème de classification est de déterminer la classe avec laquelle seront étiquetées les nouvelles données.

L'analyse de concepts formels (ACF) (Ganter & Wille, 1999) est largement utilisée dans l'apprentissage automatique. L'ACF est une théorie mathématique basée sur les hiérarchies d'un treillis de concepts formels. C'est un cadre théorique qui structure un ensemble d'objet (appelé extension) et un ensemble d'attributs (appelé intention). L'extension couvre tous les objets appartenant au concept. L'intention est l'ensemble d'attributs qui caractérisent un objets.

Une approche de classification supervisée se fait en deux phases : une phase d'apprentissage et une phase de classement. Dans la phase d'apprentissage, un classifieur est généré à partir d'un modèle de classification ; en analysant des objets qui sont décrits par des attributs dans l'ensemble de données d'apprentissage. Chaque objet est censé appartenir à une classe prédéfinie et représentée par une étiquette précise dans l'ensemble de données d'apprentissage. Dans la phase de classement, le modèle construit précédemment est utilisé pour classer/étiqueter les nouveaux objets.

Dans littérature, plusieurs études comparatives ont été réalisées concernant les méthodes de classification par l'ACF. (Fu *et al.*, 2004) ont réalisé une étude comparative théorique et expérimentale sur quelques méthodes de classification par l'ACF. D'autres méthodes sont présentées dans (Meddouri & Maddouri, 2008) et qui se basent sur un classifieur unique. Les auteurs ont présenté les méthodes de classification par l'ACF en évoquant les notions de treillis complet, de demi-treillis et de couverture. (Trabelsi *et al.*, 2016) ont présenté une taxonomie des méthodes de classification supervisée existantes. Cette taxonomie propose deux catégories : des méthodes exhaustives et des méthodes combinatoires. La première catégorie se caractérise par l'utilisation d'un seul classifieur. La deuxième catégorie contient les méthodes combinatoires qui exploitent les paradigmes d'apprentissage à partir d'ensembles de classifieurs générés séquentiellement ou parallèlement. Par conséquent, le travail présenté dans cet article est de mettre à jour une catégorisation des méthodes en introduisant, entre autres, une nouvelle catégorie fondée sur un classifieur distribué.

2 Méthodes de classification basées sur l'ACF

Nous présentons trois catégories de méthodes de classification par l'ACF. Cependant, la principale différence entre ces catégories réside dans la façon que le classifieur est généré. En fait, ces méthodes se reposent sur l'utilisation d'un classifieur unique, une combinaison de classifieurs générés par une méthode d'ensemble ou d'un classifieur distribué.

2.1 Les méthodes fondées sur un classifieur unique

Les méthodes fondées sur un classifieur unique s'appuient sur la génération d'un treillis de concepts. Le treillis de concepts est une structure mathématique regroupant l'ensemble des concepts formels d'un contexte d'apprentissage ; et qui sont hiérarchiquement organisées par des relations de sous-concept/super-concept. Nous allons présenter dans la suite les méthodes de classification basées sur un classifieur unique selon le mode de construction du treillis.

La génération d'un treillis complet consiste à ajouter des concepts formels à ce treillis et mettre à jour les liaisons hiérarchiques qui se trouvent entre eux. De nombreux algorithmes de classification par l'ACF qui construisent un treillis complet, ont été développés. Nous citons GRAND (Oosthuizen, 1996), RULEARNER (Sahami, 1995) et NAVIGALA (Visani *et al.*, 2011).

Un demi-treillis est une structure mathématique qui représente une partie du treillis de manière sélective. Le processus de classification est le même pour les méthodes citées précédemment. Mais la principale différence entre elles est le nombre de concepts formels généré et à partir de quel demi-treillis (supérieur ou inférieur). Des méthodes comme LEGAL (Nguifo & Njiwoua, 2005) et CLANN (Tsopzé *et al.*, 2007) construisent un demi-treillis supérieur en réduisant considérablement leurs complexités théoriques et leurs temps d'exécution.

Une couverture de concepts est définie comme étant une partie du treillis qui ne contient que quelques concepts générés. IPR (Maddouri, 2004) et CITREC (Douar *et al.*, 2008) sont deux méthodes qui génèrent une couverture de concepts. IPR permet de générer une couverture de concepts pertinents. Cependant, IPR peut induire des concepts redondants. CITREC propose de réduire le contexte d'apprentissage et dans la suite générer un treillis complet à partir de ce contexte réduit. Une représentation condensée de données peut causer une perte d'information pour CITREC.

L'inconvénient majeur des méthodes citées précédemment demeure dans l'utilisation d'un seul classifieur, en outre une complexité importante et le type de données traitées qui sont binaires pour la plupart des méthodes. En conséquence, de nombreuses recherches dans la littérature se sont orientées vers la combinaison de méthodes de classification basées sur les méthodes d'ensemble par Boosting ou Bagging.

2.2 Les méthodes fondées sur les ensembles de classifieurs

Les méthodes d'ensemble Boosting (Freund, 1995) et Bagging (Breiman, 1996), sont des modèles d'apprentissage qui combinent les sorties de plusieurs classifieurs pour améliorer les performances. Le principe de Boosting fait référence à la combinaison d'un ensemble de classifieurs à travers un processus en cascade pour améliorer les décisions de classification du modèle produit. En revanche, le Bagging consiste à sous-échantillonner l'ensemble des données d'apprentissages et générer un classifieur pour chaque sous-échantillon. Il existe deux catégories de méthodes de classifications ensemblistes : les méthodes fondées sur le Boosting comme BFC (Meddouri & Maddouri, 2009) et BNC (Meddouri & Maddouri, 2010) et les méthodes fondées sur le Bagging comme DNC (Meddouri *et al.*, 2014) et B-RCL (Ali, 2018).

2.3 Les méthodes fondées sur un classifieur distribué

Au cours des dernières décennies, le volume de données générées à partir de diverses sources n'a cessé d'exploser. En effet, les algorithmes existants ne sont pas extensibles aux

nouveaux ensembles de données énormes pour l'extraction et la représentation des connaissances. Une nouvelle catégorie de méthodes de classification par l'ACF, comme Dist-CNC (Fray *et al.*, 2019), est proposée. Cette méthode permet l'extraction des connaissances à partir de grand volume de données dans un environnement distribué (cloud computing).

3 Discussion

La table 1 montre une comparaison des méthodes de classification fondées sur un classifieur unique à savoir : GRAND, LEGAL et IPR. Ces méthodes génèrent respectivement un treillis complet, un demi-treillis et une couverture de concepts à partir des données binaires. Ces méthodes traitent des données multiclasses à l'exception de LEGAL qui se limite à deux classes. Pour la construction de treillis, ces méthodes utilisent des algorithmes pour générer les treillis de concepts ; et qui peuvent être incrémentaux ou non-incrémentaux. Ces méthodes choisissent de représenter les connaissances apprises par des concepts pertinents ou des règles. Dans la phase de classement, chaque méthode utilise sa stratégie appropriée afin de prédire une classe pour chaque nouvel objet.

TABLE *1 – Comparaison des méthodes de classification basées sur un classifieur unique*

Méthode	**GRAND**	**LEGAL**	**IPR**
Type de données	Binaire	Binaire	Binaire
Numbre de classes	Multi-classe	2 classes	Multi-classe
Strusture de concepts	Treillis complet	Demi-treillis	Couverture
Algorithme de construction de treillis	Ossthuizen	Bordat	Approche heuristique
Incrémental	Oui	Non	Oui
Sélection de concepts	Cohérence maximalité	Cohérence maximalité	Entropie de Shanoon
Connaissance apprise	Règles	Concepts pertinents	Règles
Classification	Vote	Vote	Règles pondérées
Complexité	O($2^l \times l^4$ avec l=min(n,m)	O(($\|L\| \times$ n (1-α)) avec $\|L\|$ = nombre de concepts, α = critère de validité	O($n^2 \times m^2 \times$ (m+n))

La table 2 présente une comparaison des méthodes de classification fondées sur les ensembles de classifieurs. Les méthodes présentées varient en fonction de l'approche d'apprentissage : séquentiel ou parallèle.

TABLE *2 – Comparison des méthodes de classification basées sur les méthodes d'ensemble*

Méthode	**BFC**	**BNC**	**DNC**	**B-RCL**
Structure de concepts	Couverture	Couverture	Couverture	Demi-treillis
Type de données	Binaire	Nominal	Nominal	Nominal
Sélection de concept	Entropie	Gain informationnel	Gain informationnel	Couverture conceptuelle aléatoire
Connaissance apprise	Règle	Règles	Règles	Règles
Classification	Vote pondéré	Vote pondéré	Vote majoritaire	Vote majoritaire
Ensemble	Séquentiel	Séquentiel	Parallèle	Parallèle
Complexité	O(nlog(n)+ nm)	O(nlog(n)+ nm') avec m' attributs nominaux	O(n') avec n' taille du sous-échantillon stratifié	O(N^3) avec N est le nombre de classifieurs

Les tables 1 et 2 montrent également une comparaison des complexités théoriques où *n* est le nombre d'objets et *m* est le nombre d'attributs. GRAND a une complexité exponentielle, car il navigue dans la totalité de l'espace de recherche (le treillis de concepts). LEGAL construit un demi-treillis ce qui réduit considérablement cette complexité. IPR a la complexité minimale parmi ces méthodes grâce à la génération des concepts les plus pertinents. Comme l'illustre le tableau 2, les méthodes basées sur l'apprentissage parallèle comme DNC et B-RCL atteignent respectivement une complexité linéaire et une complexité polynomiale. L'extraction de connaissances à partir de grands ensembles de données reste un défi et une tâche difficiles pour l'outil traditionnel d'exploration de données. Les classifieurs distribués deviennent une solution pour répondre à ce problème.

IC 2020

4 Conclusion

Dans cet article, nous avons présenté trois catégories de méthodes de classification fondées sur l'ACF. Ces méthodes se divisent en méthodes fondées sur un classifieur unique, méthodes fondées sur les ensembles de classifieurs et une nouvelle catégorie de méthodes fondées sur un classifieur distribué.

Références

ALI M. A. T. (2018). Bagged randomized conceptual machine learning method. Master's thesis, College of Engineering, Qatar.

BREIMAN L. (1996). Bagging predictors. *Machine learning*, **24**(2), 123–140.

DOUAR B., LATIRI C. & SLIMANI Y. (2008). Approche hybride de classification supervisée à base de treillis de galois : application à la reconnaissance de visages. In *Actes des 8èmes Journées Francophones en Extraction et Gestion des Connaissances*, volume E-11 of *Revue des Nouvelles Technologies de l'Information*, p. 309–320 : Cépaduès-Éditions.

FRAY R., MEDDOURI N. & MADDOURI M. (2019). Cloud implementation of classier nominal concepts using distributedwekaspark. In *Supplementary Proceedings of ICFCA 2019 Conference and Workshops*, volume 2378 of *CEUR Workshop Proceedings*, p. 125–136 : CEUR-WS.org.

FREUND Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, **121**(2), 256–285.

FU H., FU H., NJIWOUA P. & NGUIFO E. M. (2004). A comparative study of fca-based supervised classification algorithms. In *Proceeding of Second International Conference on Formal Concept Analysis*, p. 313–320.

GANTER B. & WILLE R. (1999). Formal concept analysis, mathematical foundation. Springer.

MADDOURI M. (2004). Towards a machine learning approach based on incremental concept formation. *Journal of Intelligent Data Analysis*, **8**(3), 267–280.

MEDDOURI N., KHOUFI H. & MADDOURI M. (2014). Parallel learning and classification for rules based on formal concepts. In *Proceedings of the 18th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Procedia Computer Science, p. 358–367 : Elsevier.

MEDDOURI N. & MADDOURI M. (2008). Classification methods based on formal concept analysis. In *Proceedings of the 6th International Conference on Concept Lattices and Their Applications*, p. 9–16.

MEDDOURI N. & MADDOURI M. (2009). Boosting formal concepts to discover classification rules. In *Proceeding of the 22rd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, volume 5579 of *Lecture Notes in Computer Science*, p. 501–510 : Springer.

MEDDOURI N. & MADDOURI M. (2010). Adaptive learning of nominal concepts for supervised classification. In *Proceedings of the 14th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6276 of *Lecture Notes in Computer Science*, p. 121–130 : Springer.

NGUIFO E. M. & NJIWOUA P. (2005). Treillis de concepts et classification supervisée. *Journal of Technique et Science Informatiques*, **24**(4), 449–488.

OOSTHUIZEN G. (1996). The application of concept lattice to machine learning. *Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, Tech. Rep. CSTR*, **94**(01).

SAHAMI M. (1995). Learning classification rules using lattices (extended abstract). In *Proceedings of the 8th European Conference on Machine Learning*, volume 912 of *Lecture Notes in Computer Science*, p. 343–346 : Springer.

TRABELSI M., MEDDOURI N. & MADDOURI M. (2016). New taxonomy of classification methods based on formal concepts analysis. In *Proceedings of the 5th International Workshop "What can FCA do for Artificial Intelligence" ? co-located with the European Conference on Artificial Intelligence*, volume 1703, p. 113–120.

TSOPZÉ N., MEPHU NGUIFO E. & TINDO G. (2007). Clann : Concept lattice-based artificial neural network for supervised classification. In *Proceedings of the 5th International Conference on Concept Lattices and Their Applications*, volume 331.

VISANI M., BERTET K. & OGIER J. (2011). Navigala : an original symbol classifier based on navigation through a galois lattice. *International Journal of Pattern Recognition and Artificial Intelligence*, **25**(4), 449–473.

Travail du futur, futur du travail : Portails e-collaboratifs intrinsèquement motivants personnalisables

Mohamed El Fatri[1,2], Francis Rousseaux[1,], Pierre Dreux[2]

[1] CRESTIC, Université de Reims Champagne-Ardenne, Reims, France
mohamed.elfatri@experconnect.com
francis.rousseaux@univ-reims.fr

[2] EXPERCONNECT – 72 Rue de Faubourg Saint-Honoré 75008 Paris, France
pierre.dreux@experconnect.com

Résumé : Les plateformes de travail e-collaboratif permettent aux personnes travaillant sur des sujets similaires de partager un espace de travail, d'échanger du contenu et de faire des recherches « intelligentes ». Cela nécessite un investissement personnel important de la part des utilisateurs qui seuls pourraient freiner les usages individualistes. Cet article propose une première réflexion pour solutionner le problème de manque de la motivation des utilisateurs dans le travail collaboratif numérique et sur ce que l'intelligence artificielle peut apporter au travail du futur soutenu par des portails e-collaboratifs. Après un état de l'art des théories de la psychologie positive, nous mettons l'accent sur les facteurs motivationnels à prendre en compte dans la personnalisation d'un écosystème digital intrinsèquement motivants pour un collectif d'utilisateur afin de promouvoir la création participative de connaissances et l'innovation collaborative.

Mots-clés : Personnalisation, interaction homme-machine, transfert des connaissances, travail collaboratif, psychologie positive, Experconnect.

1 Introduction

Le travail collaboratif propose aux utilisateurs de s'inscrire dans un principe d'amélioration continue, ainsi qu'un engagement dans une création participative des connaissances, dans cet objectif des plateformes numériques collaborative ont vu le jour, s'inspirant des travaux de (Bhattacharya & Chatterjee, 2000) pour proposer des systèmes modulaires répondant à la plupart des besoins fonctionnels des collaborateurs. Malheureusement, ces environnements ne sont pas très appréciés des utilisateurs et sont confrontés à la non-contribution des individus ce qui représente une limitation handicapante au partage des connaissances. Les travaux sur la psychologie positive, doivent être prises en compte dans la conception de ces plateformes du futur. Nous mettrons l'accent sur les limites des environnements e-collaboratifs actuels, avant d'effectuer un état de l'art des théories de la motivation à réussir, et finir par proposer une première réflexion sur les facteurs psychologique à intégrer dans la conception d'une plateforme e-collaborative personnalisable et intrinsèquement motivant pour différents types d'utilisateur.

2 Constat « KEEN » et hypothèse

PME parisienne Fondée par Gilles Effront et Caroline Young. Experconnect est « Le créateur de la collaboration post-retraite ». L'entreprise intervient dans le domaine du management et du conseil en ressources humaines depuis 2005, en France et à l'international. En 2016, dans ce contexte de collaboration post-retraite avec un véritable collectif professionnel constitué à la fois de salariés et de retraités. Des experts « partis à la retraite » pouvant s'approprier des cas soumis par des salariés « en poste » et organiser leurs connaissances expertes pour résoudre ces cas de manière collaborative dans un environnement numérique appelé « KEEN » pour « Knowledge and Expertise Exchange Network ».

IC 2020

KEEN est basée sur « JPlatform » de la société (Jalios, 2018). Elle permet de répondre aux besoins des utilisateurs en mettant à leur disposition différents outils qui sont censés favoriser le partage des connaissances, et stimuler l'intelligence collective. Cependant, l'insertion de KEEN dans la pratique s'est faite difficilement. Le taux de participation des effectifs aux sujets de discussion et aux problématiques remontées est très faible en termes de contenu, voir une participation nulle pour la majorité des collaborateurs.

Une hypothèse place le problème au niveau des utilisateurs : les plateformes e-collaboratives actuelles ne sont pas en mesure de les motiver correctement. Beaucoup de ces systèmes produisent une réaction émotionnelle particulièrement négative chez leurs utilisateurs (O'regan, 2003). Cependant, les émotions détectées sont souvent liées à un manque de motivation (Weiner, 1985) qui a pour effet de susciter la peur et la résistance au changement chez les utilisateurs qui manquent de culture et de partage de pratiques plus habitués au développement individuel que collectif. Selon Clifton (2002), » la psychologie positive est une révolution de la pensée car elle nous fait passer d'un mode d'expression en termes de déficit vers un mode d'expression en termes positifs «

Cela nous amène à associer travaux de psychologie cognitive et informatique dans le sens de l'ingénierie cognitive telle que développée par (Hollnagel & Woods, 1983) ou (Woods & Roth, 1988) pour concevoir des futurs modèles des plateformes collaboratifs digitales. Nous détaillerons par la suite les caractéristiques des théories de la motivation à réussir à prendre en compte lors de la modélisation d'une Plateforme e-collaborative pour différents types de profils utilisateurs.

3 La motivation à réussir

(Ryan & Deci, 2000a) nous précisent qu'être motivé signifie qu'il y a un engagement pour faire une tâche, si une personne ne ressent ni impulsion ni inspiration pour agir peut ainsi être qualifiée de non motivée, alors que celle qui est stimulée ou activée dans un but est considérée comme motivée. La motivation à réussir peut répondre à nos attentes en matière d'aboutissement de partage des connaissances soutenues par des plateforme e-collaboratives. Nous listerons quelques théories qui ont tenté de répondre aux questions : Est-ce que je peux faire cette tâche ? Est-ce que je veux faire cette tâche ? Que dois-je faire pour réussir dans cette tâche ?

3.1 Théorie de l'auto-efficacité : sentiment d'efficacité personnelle

Elaborée par Bandura (Bandura, 1994) dans le cadre de sa théorie sociale cognitive (Bandura, 1986). Elle met l'accent sur le sentiment qu'une personne peut produire sachant qu'elle peut faire les choses bien et de manière efficace dans n'importe quel contexte. Bandura estime que l'efficacité personnelle puise à quatre sources : les performances antécédentes, le plaisir retiré de l'apprentissage, les encouragements verbaux des autres personnes, et les réactions physiologiques et émotionnelles. La théorie de l'auto-efficacité a été appliquée, essentiellement dans l'apprentissage humain qui ne diffère guère du processus de partage des connaissances.

3.2 Théorie de l'auto-détermination : auto-motivation

La Théorie de l'Autodétermination (TAD) (Ryan & Deci, 2000b) stipule que différents types de motivation existent (Motivation intrinsèque, Motivation extrinsèque, et Amotivation). Ces types provoquent un comportement plus ou moins efficace en se référant aux besoins de compétence, de relationnel et d'autonomie. Ces besoins partagés universellement indépendamment des cultures (Chirkov & Ryan, 2001), (Deci & Ryan, 2002), (Chirkov *et al.*, 2003).Conformément à (Ryan & Deci, 2001) plus le type de la motivation pour une tâche tend vers la motivation intrinsèque, meilleure est la qualité de la motivation et par conséquence l'engagement dans l'activité. La théorie de l'auto-détermination a été appliquée et validée dans les domaines de la santé, le sport, et l'apprentissage humain.

3.3 La théorie de l'expérience optimale : Sentiment de fluidité

(Csikszentmihalyi, 1997) dans le cadre de ses études sur la créativité et le bonheur a désigné sous le terme « Flow » une expérience consciente, positive et complexe qui exprimait le sentiment de fluidité, de continuité et de concentration que décrivaient les personnes interrogées lorsqu'elles exerçaient leur activité préférée. Parmi les circonstances qui sont supposées conduire à l'expérience optimale, on trouve l'équilibre compétences/défi, la clarté des objectifs et un retour instantané. Le « Flow » se caractérise par : 1. un but clair, 2. une rétroaction immédiate, 3. l'intégration de la conscience et de l'action, 4. la concentration sur la tâche, 5. la perception de contrôle de la situation, 6. l'absence de préoccupation à propos de soi, 7. un sens altéré du temps, 8. la motivation intrinsèque, 9. l'équilibre entre les exigences de la tâche et les capacités de l'individu (Fullagar *et al.*, 2013), (Kawabata & Mallett, 2011), et (Moneta & Csikszentmihalyi, 1996).

Cet état de l'art nous a permis de mettre en évidence les facteurs de chacune des théories menant à un état psychologique positive, nous les listerons dans le paragraphe qui suit.

4 Personnalisation des plateformes e-collaboratives

L'augmentation de l'engagement d'un utilisateur lors d'un processus d'acquisition ou/et de partage des connaissances dépend d'une personnalisation des interfaces et des contenus des plateformes e-collaboratives.

Dans un objectif purement motivationnel, nous utiliserons un algorithme d'apprentissage automatique qui permettra de prédire le comportement ou l'état psychologique à travers des questionnaires et des traces pertinentes cueillies et stockées pour chaque utilisateur. Cela permettra de déduire les besoins psychologiques de chaque utilisateur à partir des trois théories précédentes : Le contenu sera personnalisé pour chaque utilisateur à l'aide d'un système de

TABLE *1 – Besoins pour la motivation à réussir.*

	Auto-efficacité	Auto-détermination	Théorie du Flow
Compétences	X	X	X
Plaisir	X		X
Relationnel	X	X	
Réactions	X		
Autonomie		X	
Clarté des objectifs			X

recommandation basé sur du filtrage collaboratif.

Les préférences en termes de fonctionnalités numériques seront aussi prises en compte. Une des classifications qui nous paraissent pertinentes dans notre cas d'étude KEEN est celle avancée par(Prensky, 2001) dans son article « On the Horizon » , décrivent deux types d'utilisateurs numériques. Les « digital natives » nés après 1980, jeunes hyperconnectés, dont la maîtrise des environnements numériques du quotidien (ordinateurs, jeux vidéo, internet ...) est presque intuitive. Contrairement aux « natifs du numérique », nous trouvons les « immigrants numériques », nés dans un univers à prédominance papier, une génération qui a dû apprendre à vivre avec le monde numérique.

Inspiré de (Vodanovich *et al.*, 2010), Nous pouvons résumer les préférences en termes de fonctionnalités numériques :

le « natif du numérique » : utilisateur actif, producteur des contenus. Il préfère les Conversations en ligne, les messages instantanés. Son Partage est illimité voir très fréquent,

le « immigrant numérique » : utilisateur passif, Consommateur des contenus. Il préfère les appels téléphoniques. Son partage est limité voir occasionnel.

IC 2020

5 Conclusion

L'intégration des besoins psychologiques peut rendre une plateforme numérique plus attractive pour l'utilisateur. La prochaine étape sera de concevoir un modèle de personnalisation et recommandation motivationnelle. Inspiré des trois théories citées ci-dessus, une fois intégré à notre environnement numérique collaboratif « KEEN », ce dernier doit être capable de : Prendre en compte les compétences des utilisateurs. Offrir du plaisir durant l'utilisation. Avoir une dimension relationnelle humaine entre différents utilisateur (encouragements, félicitations...). Prendre en compte les réactions psychologique et émotionnelle. offrir une grande autonomie aux utilisateurs. Avoir des objectifs (Tâche) très clairs et précis. S'adapter en fonction des utilisateurs en offrant des Défis en lien avec leurs compétences.

L'ensemble sera testé en collaboration avec un collectif constitué des experts retraités, des salariés en poste des sociétés Framatome et EDF, ainsi que des salariés de la société Experconnect.

Références

BANDURA A. (1986). Social foundations of thought and action. volume 1986.

BANDURA A. (1994). : New York : Academic Press.(Reprinted in H. Friedman [Ed.], Encyclopedia of ....

BHATTACHARYA M. & CHATTERJEE R. (2000). Collaborative innovation as a process for cognitive development. volume 11, p. 295–312 : Association for the Advancement of Computing in Education (AACE).

CHIRKOV V., RYAN R. M., KIM Y. & KAPLAN U. (2003). Differentiating autonomy from individualism and independence : A self-determination theory perspective on internalization of cultural orientations and well-being. volume 84, p.̃97 : American Psychological Association.

CHIRKOV V. I. & RYAN R. M. (2001). Parent and teacher autonomy-support in russian and us adolescents : Common effects on well-being and academic motivation. volume 32, p. 618–635 : Sage Publications Sage CA : Thousand Oaks, CA.

CSIKSZENTMIHALYI M. (1997). Flow and education. volume 22, p. 2–35.

DECI E. L. & RYAN R. M. (2002). Overview of self-determination theory : An organismic dialectical perspective. p. 3–33.

FULLAGAR C. J., KNIGHT P. A. & SOVERN H. S. (2013). Challenge/skill balance, flow, and performance anxiety. volume 62, p. 236–259 : Wiley Online Library.

HOLLNAGEL E. & WOODS D. D. (1983). Cognitive systems engineering : New wine in new bottles. volume 18, p. 583–600 : Elsevier.

JALIOS (2018). Jalios jplatform 10.0 le socle digital de l'organisation.

KAWABATA M. & MALLETT C. J. (2011). Flow experience in physical activity : Examination of the internal structure of flow from a process-related perspective. volume 35, p. 393–402 : Springer.

MONETA G. B. & CSIKSZENTMIHALYI M. (1996). The effect of perceived challenges and skills on the quality of subjective experience. volume 64, p. 275–310 : Wiley Online Library.

O'REGAN K. (2003). Emotion and e-learning. volume 7, p. 78–92.

PRENSKY M. (2001). Digital natives, digital immigrants. from on the horizon. volume 9, p. 1–6 : 5, 9.

RYAN R. M. & DECI E. L. (2000a). Intrinsic and extrinsic motivations : Classic definitions and new directions. volume 25, p. 54–67 : Elsevier.

RYAN R. M. & DECI E. L. (2000b). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. volume 55, p.̃68 : American Psychological Association.

RYAN R. M. & DECI E. L. (2001). On happiness and human potentials : A review of research on hedonic and eudaimonic well-being. volume 52, p. 141–166 : Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA.

VODANOVICH S., SUNDARAM D. & MYERS M. (2010). Research commentary—digital natives and ubiquitous information systems. volume 21, p. 711–723 : INFORMS.

WEINER B. (1985). An attributional theory of achievement motivation and emotion. volume 92, p. 548 : American Psychological Association.

WOODS D. D. & ROTH E. M. (1988). Cognitive systems engineering. In *Handbook of human-computer interaction*, p. 3–43 : Elsevier.

CubicWeb : vers un outil pour des applications clé en main dans le Web Sémantique

Fabien Amarger[1], Simon Chabot[1], Nicolas Chauvat[1], Elodie Thiéblin[1]

LOGILAB, 104 boulevard Louis-Auguste Blanqui Paris, France
prénom.nom@logilab.fr

Résumé : CubicWeb est un cadriciel pour le développement d'applications Web, qui permet d'exposer des ressources en RDF et gère la négociation de contenu pour contribuer au Web de données liées.

Mots-clés : Web de données liées, Déréférencement, Cadriciel, Python, RDF, Développement d'applications

1 Introduction et motivations

Le développement de CubicWeb, un cadriciel libre écrit en Python, a commencé en 2001, année de publication de l'article fondateur (Berners-Lee *et al.*, 2001) et a été mené en ayant connaissance des bases de données relationnelles, des standards objets tels que UML (*Unified Modeling Language*) et MOF (*Meta-Object Facility*) et des travaux tels que DAML+OIL.

CubicWeb ayant été conçu pour faciliter la réalisation d'applications déployées pour les clients de Logilab, un choix initial a été de s'appuyer sur des fondations solides en matière de gestion de données, à savoir des bases de données relationnelles, plutôt que sur des *TripleStores*, jugés trop instables à l'époque. Dans le but de se rapprocher d'une part des modélisations de type UML et OWL et d'autre part des langages d'interrogation de graphes, deux langages spécifiques ont été créés : YAMS (*Yet Another Magic Schema* [1]) pour décrire les modèles et RQL (*Relation Query Language* [2]) pour interroger les données. CubicWeb traduit le modèle YAMS en un modèle SQL, compile le RQL en des requêtes SQL et fournit un moteur de génération semi-automatique de l'interface utilisateur. Sur ces bases, il est possible d'initialiser une application web complète à partir d'un simple modèle YAMS.

Pour mettre les applications en production, il a été nécessaire dès le début de traiter les questions relatives à l'authentification des utilisateurs, la gestion des droits, la personnalisation des interfaces, la migration des données au fur et à mesure de l'évolution des modèles...

Fort de ces fonctionnalités CubicWeb a pu répondre aux besoins de projets conséquents tels que DataBnf (`https://data.bnf.fr`) (Simon *et al.*, 2013), FranceArchives (`https://francearchives.fr/`) ou DataPOC (`https://datapoc.mnhn.fr/`).

Même si les idées et principes du Web sémantique étaient présents dès la création de CubicWeb, il a été difficile de suivre les processus de normalisation menés au W3C, car il aurait fallu dégager les ressources suffisantes pour faire évoluer le cœur de CubicWeb et les applications déjà déployées chez les clients. En 2009, une première tentative a été faite de permettre l'interrogation de CubicWeb en SPARQL [3] et de convertir des modèles de OWL vers YAMS. Nous avons repris cet effort en 2018 en y ajoutant l'interrogation en GraphQL, puis la négociation de contenu RDF.

Nous présentons ici l'architecture de CubicWeb et les derniers travaux en terme de négociation de contenu, puis nous détaillons un scénario d'utilisation et concluons en mettant CubicWeb en perspective avec d'autres travaux et en présentant la suite des travaux prévus.

1. `https://cubicweb.readthedocs.io/en/3.27/book/devrepo/datamodel/definition/`
2. `https://cubicweb.readthedocs.io/en/3.27/book/annexes/rql/language/`
3. `https://www.cubicweb.org/blogentry/344822`

IC 2020

2 Architecture de CubicWeb

CubicWeb fonctionne par composants, appelés *cubes* (`https://www.cubicweb.org/project`), exploitables par plusieurs applications. Il est possible de combiner plusieurs cubes pour créer une application (qui est elle-même un cube réutilisable). Un cube est composé :

i) d'un schéma (ou modèle données) exprimé en YAMS, un langage qui permet de représenter un schéma entité-association et les permissions associées en python ;

ii) d'une logique applicative ;

iii) de vues (interfaces graphiques, ou types d'export de données).

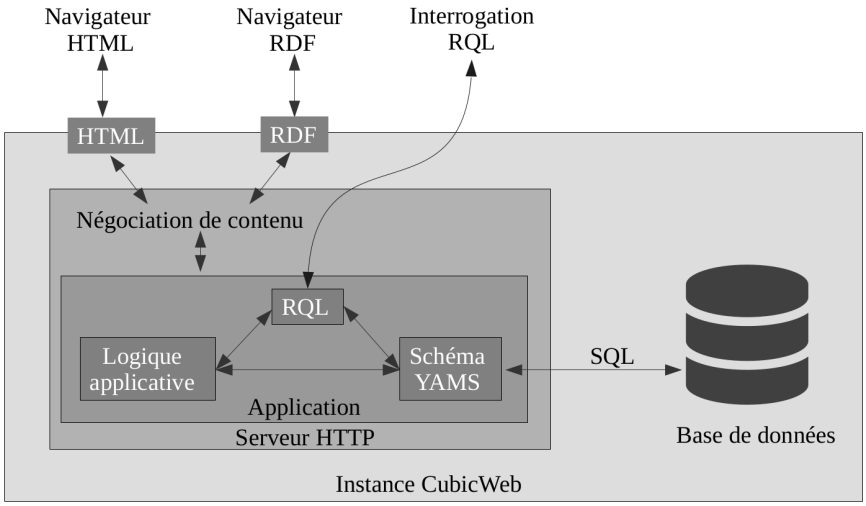


FIGURE *1 – Schéma d'une instance CubicWeb en fonctionnement*

Une instance applicative, appelée instance CubicWeb dans la suite de l'article, peut-être créée à partir d'un ou plusieurs cubes. À sa création, le modèle de données en YAMS est utilisé pour générer une base de données SQL qui sera alimentée au cours de l'utilisation de l'application. Les données dans CubicWeb sont gérées en monde fermé. La Figure 1 représente l'architecture d'une instance CubicWeb à l'utilisation. L'instance CubicWeb contient également un serveur HTTP qui gère (depuis peu) la négociation de contenu orientée serveur [4]. L'en-tête `Accept` d'une requête HTTP permet à un agent de préciser au serveur le format attendu lors de l'interrogation d'une ressource. La logique de l'application interagit avec la base de données par le biais du schéma YAMS et/ou du langage de requête RQL. Les vues de l'instance sont générées automatiquement mais un mécanisme de sélection de vues permet de dissocier données et interfaces pour faciliter leur gestion et leur personnalisation. Les vues contiennent également des interfaces d'administration permettant aux utilisateurs autorisés d'ajouter, éditer, supprimer les entités (instance d'un type YAMS). C'est la logique applicative qui se charge de générer la description en RDF d'une entité lorsque celle-ci est demandée. Les équivalences entre le schéma YAMS et la ou les ontologie(s) OWL utilisées dans l'export RDF sont spécifiées dans le cube ou les cubes sur lesquels repose l'instance CubicWeb.

4. `https://www.w3.org/TR/dwbp/#dataAccess`

3 Scénario d'utilisation

Prenons pour exemple le cube Blog [5], dans lequel les concepts clés sont les blogs et les billets de blogs, décrit en YAMS par :

```
class Blog(EntityType):
    title = String(maxsize=50, required=True)
    description = RichString()

class BlogEntry(WorkflowableEntityType):
    __permissions__ = {
        'read': (
            'managers',
            'users',
            ERQLExpression('X in_state S, S name "published"'),
        ),
        'add': ('managers', 'users'),
        'update': ('managers', 'owners'),
        'delete': ('managers', 'owners')
    }
    title = String(required=True, fulltextindexed=True)
    content = RichString(required=True, fulltextindexed=True)
    entry_of = SubjectRelation('Blog')
```

La classe `Blog` contient un titre et une description. La classe `BlogEntry`, qui décrit un billet de blog, contient un titre et un contenu. Une relation `entry_of` relie un billet de blog à un blog. La classe `BlogEntry` hérite de la classe `WorkflowableEntityType` : elle est liée à un `State` (par défaut *draft* ou *published*) par une relation `in_state`. Les billets de blog ont des permissions particulières définies grâce à l'attribut `__permissions__`. Ces permissions spécifient qu'un billet de blog est accessible en lecture par le groupe *managers*, le groupe *users* (un utilisateur connecté) et à tout le monde si le billet de blog est publié. Le groupe *owners* est un groupe virtuel, qui contient le propriétaire (généralement le créateur) de l'entité. Cet exemple simple illustre que CubicWeb permet la gestion des permissions sous la forme d'*ACL* (*Access Control List*) et d'utiliser des requêtes *RQL* pour exprimer les permissions aussi finement que souhaité. CubicWeb offre un fonctionnement similaire concernant les relations.

Une instance CubicWeb basée sur le schéma YAMS ci-dessus, est publiquement accessible en ligne à cette adresse : `https://pfia.demo.logilab.fr/`. CubicWeb propose nativement une visualisation du schéma, servi sur le chemin relatif `/schema`, ainsi qu'une traduction en OWL de ce schéma YAMS, sur `/view?vid=owl`. Une interface d'aide à l'écriture de requêtes RQL est disponible, *via* un cube spécifique, sur `/browse`.

Un blog contenant un billet de Blog ont été créés par l'administrateur de l'instance CubicWeb. Le billet de blog exemple se trouve à l'adresse `https://pfia.demo.logilab.fr/blogentry/992`. Le cube "Blog" implémente la génération de triplets RDF en fonction de son schéma et permet donc la négociation de contenu. Nous souhaitons récupérer la description en RDF de ce billet de blog au format turtle. Pour cela, nous pouvons exécuter la commande suivante :

```
curl -H "Accept: text/turtle" https://pfia.demo.logilab.fr/blogentry/992
```

Nous obtenons alors la réponse suivante :

```
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix sioc: <http://rdfs.org/sioc/types#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://pfia.demo.logilab.fr/992> a sioc:BlogPost ;
    dcterms:date "2020-02-12T16:02:00.439309+00:00"^^xsd:dateTime ;
    dcterms:modified "2020-02-12T16:02:18.030359+00:00"^^xsd:dateTime ;
    dcterms:title "un premier billet de blog" ;
    sioc:container <https://pfia.demo.logilab.fr/989> ;
    sioc:content "voici le contenu de mon premier billet de blog" .
```

5. `https://www.cubicweb.org/project/cubicweb-blog`

IC 2020

4 Conclusion et perspectives

Plusieurs outils permettent la gestion et la publication de données RDF sur le Web (Heath & Bizer, 2011). Par exemple, les plateformes de données liées telles que Apache Marmotta [6] ou CarbonLDP [7] supportent la négociation de contenu et fournissent une interface d'administration des données mais ne permettent pas de créer une application avec interfaces graphiques de manière directe. Pubby [8] permet, à l'inverse, de partir d'un endpoint SPARQL pour générer des interfaces de consultation des ressources. PSPS [9] se base sur des données RDF existantes et des vues personnalisables pour générer un site Web, sans toutefois offrir d'interface d'administration. Enfin, Virtuoso [10] dans sa version payante offre des interfaces d'administration et de consultation.

CubicWeb permet de publier des données sur le Web et fournit des interfaces d'administration et de consultation de ces ressources dont les permissions sont gérées finement.

Dans la notion de « *clé en main* », il nous semble important qu'un créateur d'application puisse importer directement des données formalisées en OWL et RDF dans CubicWeb. Des travaux pour extraire un schéma YAMS d'une ontologie ont été initiés et il sera important de déterminer quel fragment de OWL peut être traduit sans perte. Ensuite, le peuplement de la base de données CubicWeb à partir de ressources RDF pourra être effectué automatiquement. Afin de conserver l'esprit de mutualisation des fonctionnalités de CubicWeb, un alignement entre l'ontologie en entrée et les classes définies dans les cubes existants peut être envisagé. Par exemple si l'ontologie fait référence à la classe `sioc:Blog`, utiliser le cube `Blog` plutôt que de recréer une classe du même nom.

Toujours dans la même dynamique d'alignement vers les standards du Web Sémantique, nous avons commencé à rendre possible l'interrogation en SPARQL des données CubicWeb. Pour l'instant il n'est possible que d'effectuer des requêtes simples de projection avec uniquement des triplets sans variable pour les relations. Néanmoins nous envisageons de continuer ces travaux pour prendre en compte un maximum d'éléments de SPARQL sans prétendre arriver à couvrir toute la recommandation.

En suivant ces perspectives, nous espérons faciliter la publication de données sur le Web de données liées, leur administration et le développement d'applications les utilisant. Une première étape a été de proposer la négociation de contenu nativement dans CubicWeb. La gestion des requêtes SPARQL, l'import d'ontologies OWL et de données RDF permettront une intégration complète de la chaîne de publication. Nous pourrons alors répondre à la question « *Une fois que nous avons nos données en RDF, qu'en faisons nous ?* » par la réponse « *Mettons les dans CubicWeb pour en faciliter la découverte, la consultation, le partage et* in fine *l'accès à de nouveaux savoirs* ».

Références

BERNERS-LEE T., HENDLER J. & LASSILA O. (2001). The semantic web. *Scientific american*, **284**(5), 34–43.

HEATH T. & BIZER C. (2011). *Linked Data : Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers.

SIMON A., WENZ R., MICHEL V. & MASCIO A. D. (2013). Publishing bibliographic records on the web of data : Opportunities for the bnf (french national library). In *ESWC*, volume 7882 of *Lecture Notes in Computer Science*, p. 563–577 : Springer.

6. `https://marmotta.apache.org`
7. `https://carbonldp.com/`
8. `http://wifo5-03.informatik.uni-mannheim.de/pubby`
9. `https://github.com/factsmission/psps`
10. `https://virtuoso.openlinksw.com`