Actes des 32es journées francophones d'Ingénierie des Connaissances

Maxime Lefrançois

AfIA

Association française
pour l'Intelligence Artificielle

IC

Journées francophones d'Ingénierie des Connaissances

PFIA 2021

Crédit photo : Flicr/xlibber

Table des matières

Éditorial

Journées francophones d'Ingénierie des Connaissances

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.

Pour cette année 2021 de la conférence, nous avons eu l'honneur de recevoir Elena Simperl – professeure au King's College London –, qui a donné une conférence invitée intitulée « What Wikidata teaches us about knowledge engineering ? ».

Concernant les contributions scientifiques, 22 articles ont été soumis. Au total 15 articles ont été acceptés et constituent le contenu de ces actes. Ces articles sont de plusieurs types. 10 articles longs présentent des contributions originales dans les thèmes de la conférence. 4 articles sont dérivés d'une publication dans les conférences et journaux internationaux du domaine, et contribuent ainsi à leur diffusion et à leur discussion dans la communauté francophone. Enfin, 1 article est un papier court.

Pour la seconde fois consécutive, l'édition PFIA 2021 se déroule de manière virtuelle sur fond de pandémie de Covid-19. Nous constatons néanmoins un maintien de l'intérêt des auteurs pour diffuser leurs contributions originales à la communauté, et un maintien également de l'implication des membres du comité de programme et du comité de pilotage IC. Le résultat est une collection dans ces actes d'articles scientifiques de haute qualité, qui intègrent les commentaires de leurs quatre évaluations constructives et bienveillantes, de haute qualité également. Il nous reste à remercier chaleureusement l'ensemble des acteurs de la communauté francophone d'Ingénierie des Connaissances, et à nous souhaiter de pouvoir se réunir à nouveau en présentiel dès l'année prochaine pour l'édition 2022 de PFIA à Saint-Étienne.

Maxime Lefrançois

Comité de programme

Président

— Maxime Lefrançois - MINES Saint-Étienne

Membres

— Marie-Hélène Abel - Université de technologie de Compiègne
— Xavier Aimé - Cogsonomy
— Yamine Ait-Ameur - Toulouse INP/IRIT
— Nathalie Aussenac-Gilles - CNRS/IRIT
— Bruno Bachimont - University de technologie de Compiègne
— Jean-Paul Barthès - Université de technologie de Compiègne
— Nacera Bennacer - CentraleSupélec
— Mahdi Bennara - MINES Saint-Étienne
— Sandra Bringay - LIRMM
— Patrice Buche - INRAe
— Davide Buscaldi - École Polytechnique
— Sylvie Calabretto - INSA de Lyon
— Gaoussou Camara - University Alioune Diop Bambey in Senegal
— Pierre-Antoine Champin - ERCIM
— Jean Charlet - Assistance Publique hôpitaux de Paris
— Victor Charpenay - MINES Saint-Étienne
— Olivier Corby - Université Côte d'Azur
— Sylvie Despres - Paris 13
— Gilles Falquet - University of Geneva, Switzerland
— Catherine Faron - Université Côte d'Azur
— Cécile Favre - Université Lyon 2
— Béatrice Fuchs - Université Lyon 3
— Frederic Furst - Université de Picardie
— Alban Gaignard - CNRS
— Jean-Gabriel Ganascia - LIP6
— Alain Giboin - Inria
— Ollivier Haemmerlé - Université Toulouse 2 Jean Jaurès/IRIT
— Mounira Harzallah - Université de Nantes
— Nathalie Hernandez - Université Toulouse 2 Jean Jaurès/IRIT
— Liliana Ibanescu - Agro Paris Tech
— Sébastien Iksal - Le Mans Université
— Antoine Isaac - Europeana
— Clement Jonquet - UNi
— Mouna Kamel - Institut de Recherche en Informatique de Toulouse
— Gilles Kassel - Université de Picardie Jules Verne
— Pascale Kuntz - Université de Nantes
— Michel Leclère - LIRMM
— Marie Lefèvre - Université Lyon 1
— Dominique Lenne - Université de technologie de Compiègne
— Cedric Lopez - emvista
— Pascal Molli - Université de Nantes
— Isabelle Mougenot - Université de Montpellier
— Fleur Mougin - Université de Bordeaux
— Amedeo Napoli - LORIA Nancy
— Jérôme Nobécourt - Université Paris 13
— Nathalie Pernelle - LIPN, Université Sorbonne Paris Nord
— Yannick Prié - Université de Nantes
— Cedric Pruski - Université Paris Sud
— Sylvie RANWEZ - IMT Mines Alès
— Catherine Roussey - INRAe

— Fatiha Saïs - LRI
— Pascal Salembier - UTT
— Karim Sehaba - LIRIS, CNRS
— Nathalie Souf - Université Toulouse 3 Paul Sabatier/IRIT
— Konstantin Todorov - LIRMM, Université de Montpellier, CNRS
— Raphael Troncy - Eurecom
— Haïfa Zargayouna - Université Sorbonne Paris Nord

Le dictionnaire des francophones, une plateforme lexicographique contributive et sémantique.

F. Limpens[1], Nicolas Delaforge[1], P-R. Lhérisson[1], N. Gasparini[2]

[1]SCIC Mnémotix, https://mnemotix.com
[2]Institut international pour la Francophonie, Univ. Jean Moulin Lyon 3 (2IF)

freddy.limpens@mnemotix.com

Résumé

Le Dictionnaire des francophones (DDF) est un projet inédit de plateforme articulant ressources lexicographiques savantes et issues de contributions de la communauté. Ces différentes ressources prennent la forme d'un graphe de connaissances rendu accessible en lecture et/ou écriture par une architecture innovante alliant l'état de l'art des technologies du Web sémantique et du big data. Cet article propose une visite guidée de cette plateforme aujourd'hui accessible en ligne et qui illustre certaines des problématiques typiques de l'Ingénierie des Connaissances: l'intégration et l'interopérabilité de sources de données hétérogènes, et la gestion des cycles de vie de ces mêmes ressources, incluant la gestion des contributions et des droits d'accès sur un graphe RDF. L'article se conclut sur les perspectives et les prochaines étapes de ce projet s'inscrivant dans l'écosystème de l'Open Linked Data et des communs logiciels.

Mots-clés

Lexicographie, Dictionnaires, Web Sémantique, Ontolex, Approche Contributive

Abstract

The Dictionnaire des francophones (DDF) is a novel project that articulates lexicographic resources coming from scholars and from community contributions. These different resources take the form of a knowledge graph that evolves thanks to an innovative platform architecture combining state of the arts semantic web and big data technologies. This article proposes a guided tour of this platform, available online, and which illustrates some of the typical problems of Knowledge Engineering: the integration and interoperability of heterogeneous data sources, and the management of the life cycles of these same resources, including the management of contributions and access rights on an RDF graph. The article concludes with the prospects and next steps for this project set in the Open Linked Data and Software Commons ecosystem.

Keywords

Lexicography, Dictionaries, Semantic Web, Ontolex, Contributive approaches

Introduction

Le Dictionnaire des francophones (DDF) est une base d'informations sur les mots du français dont l'interface principale est celle d'un dictionnaire de définitions avec leurs aires d'usage et bien d'autres informations. Mais le DDF se pose avant tout comme une nouvelle ressource lexicographique contributive visant à compléter et prolonger des approches comme le Wiktionnaire[1] en ouvrant davantage encore le champ des contributions possibles. Le DDF décrit la richesse et la diversité du français parlé au sein de l'ensemble de l'espace francophone. C'est un projet institutionnel et académique: impulsé par le gouvernement français en mars 2018, ce projet a été transformé en actes par différentes institutions[2] dont l'Institut international pour la Francophonie (2IF) qui en est l'opérateur. La société Mnémotix[3] s'en est vu quant à elle confier le développement. Le contenu initial du DDF est issu de travaux existants et il s'enrichira grâce à l'implication du lectorat dans la description des usages.

Dans cet article nous nous attachons à montrer en quoi l'approche de l'Ingénierie des Connaissances combinées aux standards du Web Sémantique et à une architecture réactive et taillée pour traiter de gros flux de données ont permis de répondre aux défis posés par le projet de la plateforme DDF. Cette plateforme sera lancée officiellement et accessible au public le 16 mars 2021[4].

Modèle de données et approche contributive

Une des contributions majeures du DDF est d'offrir un accès unique, sous formes de données liées ouvertes[5] à un ensemble

[1] https://fr.wiktionary.org/
[2] La Délégation générale à la langue française et aux langues de France (DGLFLF), l'Institut international pour la Francophonie composante de l'Université Jean Moulin Lyon 3 (2IF), l'Organisation internationale de la Francophonie (OIF) et l'Agence universitaire de la Francophonie (AUF)
[3] https://mnemotix.com
[4] https://www.dictionnairedesfrancophones.org/
[5] i.e. adoptant des préceptes du Linked Open Data, et offrant notamment un accès SPARQL aux données du DDF.

de ressources lexicographiques déjà existantes et en construction. Le Dictionnaire des francophones intègre ainsi: l'*Inventaire des particularités lexicales du français en Afrique noire (Inventaire),* le *Wiktionnaire* francophone, le *Dictionnaire des synonymes, des mots et expression du français parlé dans le monde (ASOM),* Le *Grand Dictionnaire terminologique (GDT)*, l'ouvrage *Belgicismes - Inventaire des particularités lexicales du français en Belgique,* le *Dictionnaire des régionalismes de France (DRF)* et la *Base de données lexicographiques panfrancophone (BDLP)*. *FranceTerme* est en cours d'intégration. Les trois premières ressources sont diffusées sous licence libre CC BY-SA 3.0 tandis que les quatre suivantes sont diffusées sous licence CC BY-SA-ND 4.0.

Toutes ces ressources ont pu être agrégées et liées entre elles en un graphe homogène de connaissances grâce à un modèle RDFS pivot, l'ontologie DDF[6] (Steffens et. al. 2020), basée sur l'ontologie Ontolex Lemon (McCrae et al. 2017). Dans ce modèle, les entrées d'un dictionnaire s'articulent autour d'unités lexicales (`ontolex:LexicalEntry`) pouvant avoir plusieurs formes et plusieurs définitions (sens). Quelques ajustements ont dû être apportés à ce modèle pour affiner la granularité de description des propriétés lexicales. Par exemple, la propriété `ddf:place` permet de lier une forme ou une définition à une localité représentée par une URI geonames[7], et un ensemble de propriétés spécifiques permettent d'ajouter de nombreux marqueurs sémantiques aux définitions comme la connotation, le domaine, etc. Un ensemble de vocabulaires contrôlés (au format SKOS) permettent par ailleurs de structurer tous ces descripteurs. Enfin les relations sémantiques entre formes et (ou entre) définitions sont également décrites de manière complète, via une réification RDF et un vocabulaire contrôlé dédié.

Aux dictionnaires importés s'ajoute une ressource lexicographique contributive qui suit le même modèle de données. Le DDF s'inscrit ainsi dans l'approche de la lexicographie contributive, riche en initiatives variées (Dolar, 2017). Le modèle contributif du DDF repose sur une approche additive et non agrégative, c'est-à-dire que chaque contributrice ou contributeur peut soit ajouter une nouvelle forme, ou une nouvelle définition à une forme existante, ou ajouter une nouvelle information (exemple, marqueur de définition, etc.) à une définition ou forme existante. Les contributions se complètent et complètent également les ressources importées, et n'ont pas le devoir de fusionner. Cette approche résolument ouverte pose certaines contraintes et a orienté profondément la conception de la plateforme, que nous décrivons dans ce qui suit.

Conception et implémentation de la plateforme

Intégration des dictionnaires existants

Les données du DDF proviennent de dictionnaires sérialisés dans des formats hétérogènes, et pour la plupart sans point d'accès (API). Pour chacune de ces différentes ressources nous avons établis une correspondance vers le modèle pivot (cf supra). En fonction de leur nature nous avons traité ces données en adoptant des stratégies différentes. Nous avons mis en place un traitement génériques pour les dictionnaires qui comptent moins de dix mille entrées et un traitement spécifique pour le wiktionnaire français qui compte plus de 1844360 entrées[8] (en incluant les flexions).

Pour les dictionnaires qui contiennent peu d'entrées, nous avons utilisé un modèle pivot que nous avons décliné en XML et en CSV. Le modèle XML a été utilisé principalement dans le cas de l'Inventaire. Nous avons extrait le contenu du dictionnaire d'un fichier PDF et nous l'avons converti en XML en suivant le modèle OWL. Nous avions choisi cette stratégie afin de permettre aux experts linguistes d'intervenir sur la source et avoir ainsi une conversion RDF optimale.

Pour l'insertion des autres dictionnaires (GDT, BDLP, ASOM, DRF) nous avons opté pour le format CSV beaucoup plus facilement lisible et modifiable à l'aide d'un tableur. Les dictionnaires à insérer sont convertis en un fichier CSV en suivant un modèle strict ou chaque colonne du fichier correspond à une classe ou une propriété OWL du modèle DDF. Un parseur lit ce fichier et produit le RDF. Ce modèle pivot permet de rendre générique la création des triplets et d'être agnostique aux formats sources des dictionnaires.

Pour convertir le Wiktionnaire en RDF nous avons adopté une stratégie différente. Des dumps du Wiktionnaire sont disponibles dans un format XML, ce qui facilite l'accès à la donnée. Le contenu de chaque page du Wiktionnaire est quant à lui encodé dans un format texte appelé wikicode dont la syntaxe est définie par une documentation consultable sur le site de mediawiki. De par la nature collaborative du Wiktionnaire, cette syntaxe n'est pas toujours suivie de manière rigoureuse, ce qui rend difficile l'écriture d'un parseur universel permettant de lire le wikicode (Navarro et al., 2009; Sajous et al., 2010, 2011). Nous avons écrit un parseur pour sélectionner les entrées françaises, découper le wikicode de ces entrées selon des sections, des définitions, des exemples et des citations, et pour parser les modèles du wikicode afin qu'ils aient un rendu textuel lisible. Nous avons pu par la suite transformer en RDF les données du Wiktionnaire, ce qui constitue une des contributions notables de ce projet, pour le contenu en français du Wiktionnaire, complétant ainsi d'autres rares approches similaires comme DBnary (Serasset, 2015) pour le contenu multilingue.

Chaque Dictionnaire transformé en RDF est intégré dans un graphe nommé. Cela a pour but de faciliter la gestion des licences des dictionnaires et de permettre des mises à jour séparées. Le schéma de nommage des URIs prend ici une importance capitale et doit être déterministe afin que les URIs des ressources importées ne changent pas entre les différents imports tant que leur contenu reste identique. Ceci permet en retour que les liens pointant vers ces URIs ne soient pas cassés à chaque nouvel import. Ainsi les URIs des `lexicog:Entry` ou des `ontolex:LexicalEntry` ne changent pas, en

[6] https://gitlab.com/mnemotix/ddf/ddf-models
[7] http://geonames.org/
[8] https://fr.wiktionary.org/wiki/Wiktionnaire:Statistiques

revanche les URIs des définitions changent si le contenu de la définition change ou si des exemples et des citations sont ajoutés à la définition. Des routines calculant les différences seront par la suite développées afin de permettre un réalignement entre contenus ayant peu évolué, et ainsi préserver les contributions liées à ces contenus.

Conception de la plateforme

Ces dictionnaires importés sont stockés dans un triple store (GraphDB[9]) et servis par une plateforme se présentant comme une interface de navigation et de contribution dans un corpus de définitions. Afin de permettre un accès rapide, ces données sont indexées dans un moteur de recherche (Elasticsearch). La page d'accueil de la plateforme permet de saisir une forme et de voir toutes les définitions associés à cette forme. Si une forme ne retourne aucun résultat, par exemple si elle a été mal orthographiée, le moteur de recherche suggère des formes similaires.

La contribution au DDF peut se faire pour l'instant par l'ajout de définitions avec ou sans nouvelle forme. Ainsi, une des premières difficultés qui se pose est de permettre un tri parmi l'ensemble des définitions proposées, contributives ou importées. Par exemple, le mot "faire" en compte plus de 83 sans compter les définitions contributives. L'approche ici adoptée est d'établir un tri dans la liste des résultats selon de multiples critères:

- la localisation choisie par l'utilisatrice depuis la page d'accueil: sont affichés en premières les définitions liées à une localité la plus proche.
- score issu de la validation ou du signalement par le lectorat: sur la page de chaque définition, les usagers peuvent voter ("valider") pour cette définition, ou bien également la signaler (en pouvant préciser le motif de signalement). Le comptage des validations (score positif) et des signalements (score négatif) permet d'établir un score global et les définitions ayant le score le plus haut remonteront dans la liste.
- score issu de la modération: la modération du contenu est assurée par une communauté de personnes membres du DDF et ayant le statut "opérateur"; celles-ci peuvent marquer certaines définitions comme "à supprimer". L'action de suppression est quant à elle effectuée par les administrateurs ou administratrices du DDF qui se voient suggérer automatiquement les définitions relevées par les membres "opérateurs" dans l'espace d'administration de la plateforme. Le statut "à supprimer" pour une définition contribue à faire baisser son score et donc sa place dans la liste des résultats.

La localisation des définitions est un critère de différenciation important dans le DDF, et celui-ci est mis en avant de manière graphique. Un code couleur correspondant aux continents d'origine des différentes localisations possibles permet de rapidement identifier la provenance d'une définition et oriente la lectrice ou le lecteur.

Le système de curation collective, via le vote ou le signalement de tous les contributeurs, et le système de modération *a posteriori,* via les actions des "opérateurs" et des administrateurs, sont les 2 aspects de la stratégie adoptée par les concepteurs du DDF pour permettre à la fois une grande liberté de contribution, et en même temps de mettre en avant les contenus de plus grande qualité et au contraire identifier et éliminer les contenus inappropriés.

D'un point de vue technique, la dimension contributive repose également sur une structuration des données basée sur les standards de Web Sémantique. L'ontologie pivot a donc été étendue[10] pour permettre d'outiller tous les aspects de la contribution (édition, droits d'accès), de la modération (profils d'usagers, signalements) et de la curation collaborative (vote). La modélisation des contributions du DDF est basée sur le modèle d'actions et de provenances PROV[11]. Ce modèle très simple mais très puissant repose sur une triade Entité-Action-Agent sur laquelle nous avons greffé le modèle de contribution du DDF. Les "Entités" sont ici toutes les contributions possibles du DDF (définition, forme, exemple, etc); les "Actions" réifiées sur ces entités (`:Creation`, `:Update`, `:Deletion`) permettent de versionner ces contributions (et par exemple d'annuler une suppression). Enfin les différents profils des contributeurs, et leurs droits d'accès correspondants, sont modélisés à partir d'une extension de SIOC[12] comme des groupes d'utilisateurs (`sioc:UserGroup`) auxquels sont rattachés les "Agents" (`sioc:UserAccount`).

Une architecture réactive et résiliente

Comme nous venons de le voir précédemment, l'ensemble des données importées et produites par les usagers est stocké au format RDF dans un triple store. Afin d'assurer le niveau de réactivité et de performance requis par une plateforme grand public telle que le DDF, nous avons mis au point une architecture reposant sur l'état de l'art des technologies du Web Sémantique et Big Data.

Le triple store choisi est GraphDB, proposé par Ontotext, et structuré ici en cluster afin d'assurer la résilience et une disponibilité optimale du système. De plus, l'accès en lecture des données est assuré par un moteur de recherche, en l'occurrence ElasticSearch, sur lequel les données sont indexées au chargement des données et de manière incrémentale; le graphe et les indexes sont ainsi synchronisés en permanence.

L'usage conjoint de ces 2 types de datastores (index et graphe) requiert cependant une architecture réactive que nous avons mise au point au sein de Mnémotix sous la forme d'un middleware générique, Synaptix[13]. Ainsi, chaque requête est traitée de manière asynchrone et non bloquante par un système de micro-services fédérés par un système de bus de messages implémentant le standard AMQP[14]. Chaque datastore est connecté au bus de messages par des "guichets" AMQP de type "Producer" ou "Consumer". Les "Consumers" ont des

[9] https://graphdb.ontotext.com/

[10] voir https://mnemotix.gitlab.io/ddf/ddf-models/
[11] https://www.w3.org/TR/prov-o/
[12] https://www.w3.org/Submission/sioc-spec/
[13] https://gitlab.com/mnemotix/synaptix
[14] https://www.amqp.org/

files où sont stockés les messages à traiter. Chaque message est aiguillé et traité individuellement jusqu'à ce que la file de messages soit vidée. Synaptix s'appuie sur ce système de messages pour synchroniser les datastores entre eux et pour ingérer des sources de données externes sans ralentir le fonctionnement global de la plate-forme.

Un autre aspect permettant des gains substantiels en termes de réactivité est que l'API publique n'est plus une API REST classique mais une API GraphQL qui permet de combiner plusieurs requêtes en un seul appel HTTP. Cette technologie développée par Facebook a été inventée pour répondre au problème du nombre important de requêtes qu'il fallait pour construire les murs de données des utilisateurs et qui saturait les serveurs Facebook. Avec cette technologie, ils ont pu charger un mur en une seule requête.

Enfin la partie cliente tire elle-aussi parti de la réactivité du backend servant les données du DDF en adoptant un paradigme SPA (Single Page Application). L'application web DDF consiste donc en un serveur web codé en NodeJS[15] et utilisant des composants d'interfaces basés sur la librairie ReactJS[16] et s'interfaçant avec l'API GraphQL grâce au client Apollo[17]. L'assemblage de ces technologies permet ainsi un chargement progressif des différents types de données nécessaires pour afficher par exemple tous les détails liés à une définition.

Conclusion

Le DDF a l'ambition de devenir un projet contributif incontournable dans le paysage de la francophonie en proposant un ensemble de ressources lexicographiques reflétant la grande diversité du français tel qu'il est parlé à travers le monde. A la fois base de données liées ouvertes, et plateforme contributive, le DDF se présente également comme un commun logiciel, dont le code source et les données sont disponibles sous licences libres[18] et exploitables à des fins de recherche. Son architecture repose sur le middleware générique et open-source Synaptix, développé par Mnémotix[19].

Le DDF est appelé à évoluer très prochainement en offrant de nouvelles fonctionnalités contributives qui permettront d'ajouter des localisations, des exemples, des marqueurs d'usages, et des relations sémantiques entre formes et ou définitions. Il sera également possible d'engager des discussions sur l'usage et l'étymologie des mots. Des espaces de discussion dédiés à la communauté des contributeurs seront également bientôt proposés pour soutenir son développement. De par sa nature résolument ouverte et inclusive, le DDF pose également des défis techniques et scientifiques quant à la curation et la modération des données contributives. Enfin une version du DDF capable de fonctionner hors-connexion est à l'étude et permettra d'élargir encore l'accès à ce nouveau bien commun des francophones.

[15] https://nodejs.org/
[16] https://fr.reactjs.org/
[17] https://www.apollographql.com/docs/react/
[18] voir https://gitlab.com/mnemotix/ddf pour le code source et https://www.dictionnairedesfrancophones.org/sparql pour le endpoint SPARQL
[19] middleware sémantique par ailleur au coeur d'autres communs logiciels reposant sur les technologies du web sémantique, comme notamment les outils développés par la coopérative https://www.elzeard.co/

Références

Dolar, K. (2017). Les dictionnaires collaboratifs en tant qu'objets discursifs, linguistiques et sociaux. PhD thesis. Université Paris Nanterre, Paris, France

Dolar, K., Steffens, M. & Gasparini, N. (2020). Dictionnaire des Francophones: A New Paradigm in Francophone Lexicography. In: Proceedings of XIX EURALEX Congress: Lexicography for Inclusion, Vol. I. Thrace: Democritus University of Thrace, pp. 23-30. https://euralex2020.gr/wp-content/uploads/2020/11/EURALEX2020_ProceedingsBook-p023-030.pdf

McCrae, J., Bosque-Gil, J., Garcia, J., Buitelaar, P. & Cimiano, P. (2017). The OntoLex-Lemon Model: development and applications. In Proceedings of eLex 2017, pp. 587-597. Accessed at https://elex.link/elex2017/wp-content/uploads/2017/09/paper36.pdf [30/05/2020]

Navarro, E., Sajous, F., Gaume, B., Prévot, L., Hsieh, S., Kuo, I., Magistry, P., et Huang, C.-R. (2009). Wiktionary and NLP: Improving synonymy networks. In Proceedings of the 2009 ACL-IJCNLP Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, 19–27, Suntec, Singapore. Association for Computational Linguistics

Steffens, M.., Dolar, K., & Gasparini, N. (2020). Structuration de données pour un dictionnaire collaboratif hybride. In: Terminologie & Ontologie: Théories et Applications. Actes de la conférence TOTh 2019. Chambéry: Presses Universitaires Savoie Mont Blanc, pp. 413-426.

Sajous, F., Hathout, N., et Calderone, B. (2013). GLÀFF, un Gros Lexique À tout Faire du Français. In Actes de la 20e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2013), 285–298, Les Sables d'Olonne, France.

Sajous, F., Navarro, E., et Gaume, B. (2011). Enrichissement de lexiques sémantiques approvisionnés par les foules :le système WISIGOTH appliqué à Wiktionary. TAL, 52(1):11–35.

Sajous, F., Navarro, E., Gaume, B., Prévot, L., et Chudy, Y. (2010). Semi-automatic Endogenous Enrichment of Collaboratively Constructed Lexical Resources: Piggybacking onto Wiktionary. In Loftsson, H., Rögnvaldsson, E., et Helgadóttir, S. (eds), Advances in Natural Language Processing, vol. 6233 of LNCS, 332–344. Springer Berlin /eHeidelberg

Sérasset G. (2015). DBnary: Wiktionary as a Lemon-Based Multilingual Lexical Resource in RDF. Semantic Web – Interoperability, Usability, Applicability, IOS Press, 2015, Multilingual Linked Open Data, 6 (4), pp.355-361.

Une méthodologie et un outil d'évaluation du niveau de " FAIRness " pour les ressources sémantiques : le cas d'AgroPortal

Emna Amdouni[1] et Clément Jonquet[1,2]

[1] LIRMM, Univ. de Montpellier, CNRS, France

[2] MISTEA, Univ. de Montpellier, INRAE, Institut Agro, France

emna.amdouni@lirmm.fr et jonquet@lirmm.fr

Résumé

Les principes "FAIR" définissent un ensemble de caractéristiques que les données et leurs métadonnées devraient présenter pour être Faciles à trouver, Accessibles, Interopérables et Réutilisables. Également, suivant le principe I2, une ontologie, et plus généralement une ressource sémantique, devrait aussi être "FAIR". Des critères spécifiques aux ontologies commencent à apparaître, mais il n'existe toujours pas de mécanismes pour évaluer le degré de mise en œuvre de ces principes. Nous proposons une approche d'évaluation automatique du niveau de "FAIRness" d'une ontologie qui se base sur une représentation riche et structurée des métadonnées. Nous nous reposons sur le modèle de métadonnées MOD et avons développé un prototype pour AgroPortal, un portail de ressources sémantiques et d'ontologies en agronomie.

Mots-clés

Science ouverte, principes FAIR, FAIRness, ontologies, ressources sémantiques, portail d'ontologies, AgroPortal

Abstract

The "FAIR" principles define a set of characteristics data and their metadata should respect in order to be Findable, Accessible, Interoperable, and Reusable. Per principle I2, an ontology, and more generally a semantic resource, should also be FAIR. Ontology-specific criteria begin to emerge, but there is still no mechanism to assess the degree of implementation of these principles. We propose an automatic evaluation method of the level of "FAIRness" of an ontology which is based on a rich and structured representation of the metadata. We rely on the MOD metadata model and have developed a prototype for AgroPortal, a repository of semantic resources and ontologies in agronomy.

Keywords

Open science, FAIR principles, FAIRness, ontologies, semantic resources, ontology repository, AgroPortal

1 Introduction

En 2014, un groupe de chercheurs et d'éditeurs (appelé FORCE 11) a établi des principes de création et de partage des données scientifiques afin de favoriser leur exploitation et leur réutilisation d'une manière automatique entre autres via le Web. Ces principes sont introduits sous le nom de FAIR (acronyme de Findable, Accessible, Interoperable et Reusable). Chacun des quatre principes FAIR décrit un ensemble de caractéristiques génériques que les données (et leurs métadonnées) devraient avoir mais ils ne précisent pas, ni ne préconisent, les mécanismes de mise en œuvre [1], [2].

Ci-après, nous présentons brièvement la signification de chaque aspect F-A-I-R : Premièrement, nous parlons de données *faciles à trouver* lorsqu'elles sont suffisamment décrites avec des métadonnées et hébergées ou indexées dans une librairie ou un portail accessible ouvertement. La mise en œuvre de ce principe implique que les données, métadonnées et autres ressources doivent avoir un identifiant unique et persistant qui les rend trouvables et référençables par les humains et les machines. Deuxièmement, les données sont considérées comme *accessibles* lorsque les utilisateurs peuvent les consulter à l'aide d'un protocole ouvert et universellement implémentable. Mais cela ne signifie pas que les données doivent être librement ouvertes sans restrictions. Parfois, les données peuvent être "FAIR" et non ouvertes. En d'autres termes, les données FAIR doivent être associées à des métadonnées qui spécifient les conditions dans lesquelles les données sont accessibles. Troisièmement, les données sont *interopérables* lorsque n'importe quel utilisateur peut facilement et d'une manière standardisée, les traiter sans avoir recours à une chaîne de traitement spécifique. Les sous-principes I peuvent être considérés comme les plus difficiles à réaliser et les plus importants pour être FAIR. Plus spécifiquement, ils indiquent que les données et les métadonnées doivent être représentées dans un langage formel, accessible, partagé et générique pour la représentation des connaissances. De plus, ces données doivent elles-mêmes utiliser des vocabulaires ou ontologies FAIR et inclure des références qualifiées à d'autres données et métadonnées. Il est clair que le Web sémantique et les technologies de données liées ont été identifiés parmi les meilleurs candidats à utiliser pour la représentation des connaissances, la lisibilité par machine et l'interopérabilité sur le Web, mais les principes FAIR ne peuvent pas être réduits au Web sémantique [3]. Enfin, les données sont *réutilisables* lorsqu'elles sont fournies avec des informations claires sur la licence et l'utilisation des données pour les humains et les machines. Elles doivent également être associées à des métadonnées et à une documentation riche qui détaillent leur provenance (spécifications des données, financement, cas d'utilisation, versions, processus expérimentaux, etc.).

Les principes FAIR sont décrits à un niveau générique et leur

mise en œuvre a volontairement été mise de côté au début. Cependant, avec le succès de la démarche, cette question d'implémentation des principes FAIR est devenue très importante, et elle doit se décliner au cas par cas pour chaque type d'objet digital. En Europe, par exemple, la mise en œuvre des principes FAIR est réalisée, en partie, par le programme European Open Science Cloud (https://eosc-portal.eu) de la Commission européenne, en particulier par le groupe d'experts sur les données FAIR. L'initiative GO FAIR (www.go-fair.org) a également pour objectif de favoriser l'adoption des principes FAIR, notamment via la description et l'élaboration de profils de mise en œuvre ("FAIR Implementation Profile") et le déploiement de serveur d'accès aux données ("FAIR Data Points"). Parmi d'autres initiatives internationales qui encouragent l'adoption des principes FAIR, nous pouvons citer : le programme américain NIH Data Commons (https://commonfund.nih.gov/commons), la Research Data Alliance (www.rd-alliance.org), plusieurs projets H2020 dont FAIRsFAIR (www.fairsfair.eu), et des initiatives communautaires telles que le Food System GO FAIR Implementation Network (www.foosin.fr).

Bien que la vision FAIR soit largement reconnue par plusieurs initiatives internationales et que les ontologies soient considérées comme un élément clé pour rendre les données FAIR, à ce jour il n'existe toujours pas de mécanisme, ni d'outil d'évaluation du niveau de "FAIRness" pour les ressources sémantiques. En effet, aucun des travaux existants dans la littérature n'a réussi à définir une approche de FAIRness claire et complète qui couvre à la fois les aspects méthodologiques et techniques associés aux ressources sémantiques. Pour toutes ces raisons, nous présentons ici le développement d'une méthodologie et d'un outil, très fortement inspiré de l'état de l'art actuel, pour l'évaluation du degré de mise en œuvre – nous parlerons aussi de niveau de "*FAIRness*" ou "*FAIRness assessment*" – des principes FAIR pour les ressources sémantiques. Notre approche se base sur une représentation riche et structurée des métadonnées d'une ontologie.[1] Nous nous basons sur le modèle de métadonnées MOD (voir Section 3) qui en amont a identifié les vocabulaires de métadonnées (et les propriétés définies dans ces vocabulaires) pour les ontologies. Nous avons développé un prototype de FAIRness assessment pour AgroPortal (http://agroportal.lirmm.fr) un portail pour les ressources sémantiques et les ontologies en agronomie [7].

L'article est structuré comme suit : la Section 2 présente l'état de l'art actuel sur l'évaluation du niveau de FAIRness. Ensuite, la Section 3 décrit les étapes de conception de notre méthodologie d'évaluation du niveau de FAIRness des ressources sémantiques basée sur les métadonnées, détaille notre approche en présentant une projection de chaque principe FAIR pour les ressources sémantiques, et propose des indicateurs de mesure présentés sous forme de questions. Puis, la Section 4 présente notre prototype implémenté dans AgroPortal, explique son fonctionnement, et fournit des résultats préliminaires pour quelques ressources sémantiques dans le domaine de l'agriculture. Enfin, la Section 5 résume notre contribution et cite des perspectives.

2 Etat de l'art

Avant l'apparition des principes FAIR, en 2011, Berners-Lee a présenté les principes fondamentaux du Linked Open Data [8] pour rendre les données disponibles, partageables et interconnectées sur le Web. Les principes FAIR ont été proposés pour des raisons similaires en mettant davantage l'accent sur la réutilisabilité des données. Les principes LOD 5-star ont été spécialisés en 2014 pour les vocabulaires [9] et présentés sous la forme de cinq règles à suivre pour créer et publier des "bons" vocabulaires sur le Web. Dans ce schéma, les étoiles indiquent la qualité des données menant à une meilleure structure (e.g., l'utilisation des recommandations du W3C) et une meilleure interopérabilité pour la réutilisation (c'est-à-dire la représentation des métadonnées, la réutilisation des vocabulaires et l'alignement). Bien que le système de notation 5-star proposé pour les vocabulaires soit simple, à ce jour aucun outil de mise en œuvre n'a été développé autour de ces principes; ces principes ne sont pas non plus bien référencés dans la littérature. Une première étude d'alignement des principes LOD et FAIR a été réalisée dans [10], une deuxième étude plus approfondie a été proposée par Poveda et al.[11]; nous avons intégré ces alignements dans notre méthodologie.

En 2017, l'initiative *Minimum Information for Reporting an Ontology* (MIRO) a publié des directives destinées aux développeurs, pour la description d'une ontologie dans des rapports scientifiques [12]. Les directives MIRO visent à améliorer la qualité et la cohérence des descriptions du contenu de l'information; y compris la méthodologie de développement, la provenance et le contexte des informations de réutilisation. Ces directives définissent des éléments d'information (tels que nom, licence, URL) et spécifient leur niveau d'importance : 'must', 'should', 'optional'. Ce travail était significatif mais, jusqu'à présent, il n'existe aucune étude sur la façon dont les directives MIRO s'alignent avec ou complètent les principes FAIR. Le modèle de métadonnées MOD 1.4[2] a cependant fourni un alignement entre chaque directive MIRO et les propriétés de métadonnées correspondantes dans MOD. Nous avons donc utilisé cet alignement dans notre méthodologie, pour influencer le score de FAIRness avec les directives MIRO. Par exemple, la 'guideline' A.3 de MIRO recommande de préciser la licence de l'ontologie, et MOD suggère d'utiliser `dct:license` pour cela. Ici, nous ajoutons que cela implémente le principe R1.1.

Depuis 2018, plusieurs approches et outils génériques pour l'auto-évaluation du niveau de FAIRness d'une ressource sont apparues; nous pouvons citer : SHARC[13] , FDMM [14], FAIR Metrics [15], [16] (devenu FAIR evaluator), FAIRdat [17] et FAIR-Aware [18]. Une analyse synthétique de comparaison de 12 outils d'évaluation faite par la Research Data Alliance est disponible sur ce lien.[3] Par manque d'espace, nous ne présentons que l'approche FDMM et SHARC; les

[1] Dans cet article, nous utilisons parfois le terme "ontologie" pour faire référence à des ressources sémantiques [4], des systèmes d'organisation des connaissances [5], ou des artefacts sémantiques [6]. Typiquement, des terminologies, thésaurus et vocabulaires.

[2] Metadata for Ontology Description and Publication Ontology: https://github.com/sifrproject/MOD-Ontology

[3] https://github.com/rd-alliance/FAIR-data-maturity-model-WG

approches génériques intégrées dans notre méthodologie.

En 2017, le groupe d'intérêt SHARC (SHARing Rewards and Credit) de la Research Data Alliance (RDA) a proposé une grille d'évaluation du niveau de FAIRness dédiée aux données. L'objectif de ce travail est de guider les chercheurs et les autres parties prenantes à adopter les principes FAIR. La grille FAIR définit un ensemble de 45 critères génériques avec trois niveaux d'importance ('essential', 'recommended', 'desirable'); les critères sont formulés en questions qui sont parfois dépendantes les unes des autres comme dans un arbre de décision.

En 2018, le groupe de travail RDA FAIR Data Maturity Model (FDMM) a publié une liste de recommandations génériques visant à normaliser la méthodologie d'évaluation du niveau de FAIRness afin de permettre la comparabilité des résultats. Plus concrètement, le travail est présenté sous forme d'une grille qui définit 47 critères génériques avec des priorités ('essential', 'important', 'useful'); ces critères sont dérivés de chaque principe FAIR.

En résumé, la grille SHARC et FDMM considèrent que certains principes FAIR sont plus importants que d'autres. Nous avons suivi cette vision dans notre méthodologie et avons décidé d'inclure les résultats de SHARC et FDMM pour nous aider à déterminer l'ordre d'importance des principes FAIR pour les ressources digitales.

Ces deux dernières années, des approches plus spécifiques autour de l'évaluation des ressources sémantiques selon les principes FAIR, ont été publiées, nous les détaillons ci-dessous.

En mars 2020, le projet H2020 FAIRsFAIR a introduit 17 recommandations pour le partage des ressources sémantiques avec les principes FAIR [6]. Pour chaque recommandation, les auteurs fournissent une description détaillée et soulignent les technologies du Web sémantique qui peuvent être utilisées pour la mise en œuvre de cette recommandation pour n'importe quelle ressource sémantique. Ce travail, très pertinent pour nous, et en cours de révision par la communauté scientifique, reste préliminaire, et devra être complété par des indicateurs concrets de FAIRness qui pourront être utilisés pour évaluer automatiquement le degré de mise en œuvre des recommandations proposées pour les ressources sémantiques.

Également, Garijo et Poveda [11] proposent une liste de bonnes pratiques pour des ontologies FAIR. Dans un autre article [19], les auteurs complètent leur travail par une analyse qualitative du degré d'alignement des principales approches existantes sur le partage des ressources sémantiques (5-stars LOD [8] , 5-stars V [9], et FAIRsFAIR) avec les principes FAIR. Cette analyse démontre qu'aucune approche existante n'est complètement alignée aux principes FAIR et soulève aussi des enjeux scientifiques importants. L'approche de Poveda et al. est significative mais elle est limitée à l'étude de certaines approches sémantiques et ne propose pas des métriques de mesure du niveau de FAIRness. Nous pensons que ce travail d'analyse réalisée est important mais il devrait être amélioré pour inclure d'autres approches plus génériques notamment FDMM et SHARC. C'est également une contribution de notre travail présenté ici.

Finalement, l'outil DBPedia Archivo lancé par Frey et al. [20] est le premier outil public pour les ontologies FAIR. Archivo propose l'archivage et l'évaluation du niveau de FAIRness de toute ontologie hébergée dans sa propre librairie publique. L'inconvénient de cet outil est qu'il ne détaille pas ses résultats d'évaluation et ne propose pas de recommandations pour guider ses utilisateurs dans l'amélioration du score de FAIRness obtenu pour leurs ontologies. D'un point de vue méthodologique, Archivo n'a pas encore dévoilé l'approche qu'il applique dans le processus d'évaluation de ses ressources. Par conséquent, leur approche reste inconnue pour la communauté Web sémantique.

Notre analyse de l'état de l'art montre clairement qu'aucune *approche spécifique* pour les ontologies n'est strictement alignée aux 15 sous-principes FAIR (voir Tableau 1); également qu'aucune *approche générique* des principes FAIR n'est strictement applicable aux ontologies. Ainsi, nous proposons ici une méthodologie qui fait converger les deux visions. Nous constatons aussi que les approches génériques tiennent fortement compte de tous les sous-principes FAIR sauf F2, F3 et A2. Cependant, les approches spécifiques accordent plus d'importance à F1, A1, I1, I2, R1 et R.1.2. Les résultats obtenus soulignent aussi la nécessité d'établir un ensemble minimum de représentation des métadonnées de l'ontologie (en cours de discussion dans le cadre du projet FAIRsFAIR), une meilleure fédération et interopération des portails et services pour les ontologies, des stratégies de maintenance long terme pour les ontologies et leurs métadonnées au sein de ces portails/services, et des meilleures pratiques pour documenter et communiquer sur les ontologies.

Pour construire notre approche, nous avons considéré SHARC, FDMM, LOD 5-star, MIRO, FAIRsFAIR et Poveda et al. Nous avons étudié chaque approche (i) en analysant comment elle s'aligne avec chaque sous-principe FAIR (voir Tableau 1) et (ii) en pondérant (à l'aide de crédits) les sous-principes en fonction de l'importance que lui a donné une approche. Plus précisément, nous avons fixé une valeur "de départ" par défaut à 10 crédits égale pour chaque sous-principe. Dans notre calcul, nous sommes partis de cette valeur de départ et avons ajouté des crédits aux sous-principes en fonction de l'importance que leur donnent les approches étudiées. Le calcul des crédits obtenus pour chaque sous principe n'est pas détaillé ici mais peut être consulté dans [21]. La Figure 1 montre les valeurs de crédits obtenus pour chaque sous-principe. Le vert clair est la valeur de départ; le vert moyen représente les ajouts de crédits obtenus avec les approches génériques, et le vert foncé représente les ajouts de crédits obtenus avec les approches spécifiques aux ontologies. Ainsi, on peut constater que les trois principes qui ressortent le plus sont F1, A1 et I1 et ceux qui ressortent le moins sont F3, A1.2 et A2.

	SHARC	**FDMM**	**MIRO**	**5*V**	**FsF**	**Poveda et al.**
F1	X	X	X	-	X	X
F2	X	X	X	-	X	X
F3	X	X	-	-	X	-
F4	X	X	-	-	X	X
A1	X	X	X	X	X	X
A1.1	X	X	-	-	X	-

A1.2	X	X	-	-	X	-
A2	X	X	-	-	X	-
I1	X	X	X	X	X	X
I2	X	X	X	-	X	-
I3	X	X	X	X	X	X
R1	X	X	X	X	-	X
R1.1	X	X	X	-	X	X
R1.2	X	X	X	-	X	X
R1.3	X	X	-	-	X	X

Tableau 1. Alignement des approches existantes aux 15 sous-principes FAIR. Le symbole (X) indique que l'approche étudie le principe concerné, en revanche le symbole (-) indique le contraire.

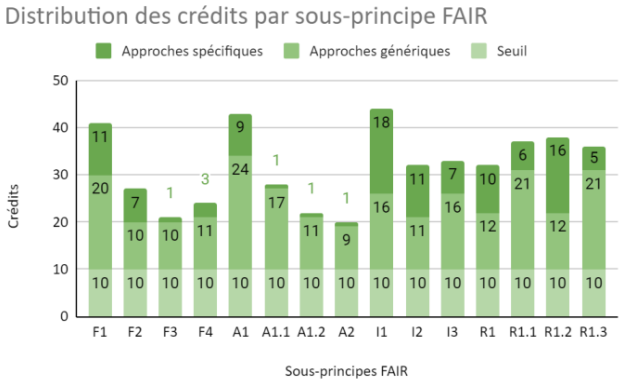

Figure 1. Distribution des crédits obtenus pour chaque sous-principe FAIR; le calcul est issu de l'intégration des approches spécifiques, génériques, et du seuil.

3 Méthodologie

Dans ce travail, nous considérons que l'évaluation du niveau de FAIRness des ontologies et des ressources sémantiques devrait autant que possible être basée sur l'évaluation de descriptions formelles de leurs métadonnées, idéalement indexées, partagées et normalisées par des portails d'ontologies tels que le NCBO BioPortal [22] ou Ontology Lookup Service de EBI [23] en biomédecine, ou AgroPortal en agronomie. Dans des travaux antérieurs, nous avons construit un nouveau modèle de métadonnées harmonisé pour AgroPortal et démontré qu'il améliore l'identification et la réutilisation des ressources sémantiques en agronomie [7], [24]. Ici, nous démontrons que les portails d'ontologies sont également importants pour l'évaluation du niveau de FAIRness.

Dans cette section, nous interprétons trois sous-principes (F1, A1, I1 et R1.1) pour les ressources sémantiques et listons les propriétés de métadonnées standardisées utilisées pour décrire les informations nécessaires liées à l'évaluation de ces sous-principes. Nous traitons dans cet article trois sous principes sur 15, le reste de la méthodologie est documentée sur GitHub, dans le dépôt du code open source qui implémente notre approche: https://github.com/agroportal/fairness. Nous avons utilisé le modèle MOD comme référence, il regroupe 346 propriétés tirées de 23 vocabulaires de métadonnées (tels que Dublin Core, DCAT, VoID, ADMS, VOAF, Schema.org, etc.) [24]. MOD n'est donc pas un standard, mais un listing de standards. Ici, nous utilisons le modèle de métadonnées MOD pour identifier sans ambiguïté quelle propriété peut être utilisée; cependant, notre méthodologie est indépendante de ce modèle et exige seulement que l'information soit bien représentée et harmonisée dans les métadonnées d'une ontologie.

Nous avons distribué les crédits obtenus précédemment sur des séries de questions proposées pour l'évaluation de chaque sous-principes FAIR. La distribution des crédits d'un sous-principe sur les questions associées est basée sur une distribution équilibrée et pertinente qui pourrait être modifiée ou ajustée si des questions étaient supprimées ou ajoutées à l'avenir. Toutes nos questions d'évaluation, sauf celles du sous-principe R1.3, sont génériques (c'est-à-dire non spécifiques à l'agronomie ou à AgroPortal) et pourront être appliquées sur n'importe quelle ontologie par d'autres communautés scientifiques. Nous recommandons donc aux futurs utilisateurs de notre travail de spécifier eux même les questions de R1.3 en s'adressant aux experts de leur domaine d'intérêt.

Concrètement, le résultat de notre travail est une grille d'évaluation composée de 68 questions couvrant la totalité des sous-principes FAIR; nous avons identifié 55 propriétés de métadonnées nécessaires (correspondantes à 309 crédits) pour répondre à toutes ces questions sauf celles de F2; 73 propriétés additionnelles sont proposées dans MOD pour enrichir une ontologie avec des métadonnées et augmenter son niveau de FAIRness. En d'autres termes, nous proposons dans F2 d'évaluer toutes les propriétés de métadonnées qui ne sont pas considérées dans le reste des sous-principes. Pour chaque sous-principe, nous avons attribué à chaque question un certain nombre de crédits dans la limite des valeurs identifiées dans la Figure 1. Il faut noter que les questions sont majoritairement binaires (oui ou non) ainsi, pour une question donnée une ressource sémantique se verra en général attribuer soit tous les crédits, soit aucun; mais rarement une valeur entre les deux.

Dans la suite, nous listons les questions d'évaluation de F1, A1, I1 et R1.1 (c'est-à-dire, une illustration d'un sous-principe pour un groupe de principe donné) et précisons le nombre de crédits que notre algorithme attribuera si le test de conformité sémantique est validé pour chaque sous-principe. Le score final de FAIRness est ensuite une simple somme de l'ensemble des crédits obtenus aux questions d'évaluation. Ce score peut être normalisé sur 100 pour faciliter sa compréhension et la comparaison des ontologies entre elles.

Sous-principe F1. Les ontologies et les métadonnées d'ontologie reçoivent un identifiant global unique et persistant. F1 concerne les *identifiants*; bien que cela n'est pas explicitement mentionné dans l'intitulé du sous-principe, plusieurs travaux demandent que les identifiants soient résolvables, c'est-à-dire un identifiant qui permettent d'avoir accès à la ressource ou à la description de la ressource (déréférencement). FAIRsFAIR recommande l'utilisation d'un identifiant [25] *globalement unique, persistant et résolvable* (GUPRI) pour les ontologies et les métadonnées d'ontologies.

Dans la plupart des cas, les ontologies décrites avec les langages du Web sémantique possèdent un *identifiant de ressource unique* (URI) et les métadonnées sont soit représentées au

niveau du fichier de l'ontologie (le plus fréquent) soit dans un fichier externe. Les URIs sont généralement uniques sur le Web, mais ils ne sont pas toujours pérennes et résolvables. Dans certains cas, les ontologies peuvent avoir un identifiant supplémentaire attribué par une organisation externe telle qu'un DOI. Parfois, les URIs prennent la forme de PURL qui sont supposés persistants mais qui ne sont pas certifiés comme le sont les DOI. Idéalement, les URIs d'ontologies devraient pouvoir être résolvables pour garantir un degré de conformité plus élevé pour F1. Lorsqu'elles sont sauvegardées dans un fichier séparé, les mêmes règles concernant l'identification doivent s'appliquer au fichier de métadonnées de l'ontologie. En complément, les communautés qui développent des ontologies essaient parfois de maintenir une utilisation cohérente des acronymes pour identifier les ontologies. Par exemple, dans le cadre de l'OBO Foundry, un nom court est obligatoire demandé et sera utilisé pour identifier l'ontologie et pour l'attribution du PURL (par exemple, l'acronym de l'*Agronomy Ontology* est AGRO et son PURL est http://purl.obolibrary.org/obo/agro.owl)

F1 peut être évalué en vérifiant la valeur affectée à la propriété `owl:ontologyIRI`, utilisée pour coder l'URI de l'ontologie et la propriété `dct:identifier`, utilisée pour coder un autre identifiant "externe". De plus, la propriété `owl:versionIRI`, qui sauvegarde un URI spécifique à la version, peut également être utilisée pour évaluer si l'ontologie distingue clairement des identifiants de versions. La Liste 1 résume les questions d'évaluation pour ce sous-principe.

Liste 1. Questions d'évaluation de F1 (41 crédits)

Q1. Une ontologie a-t-elle un identifiant "local", c'est-à-dire un identifiant globalement unique et potentiellement persistant mais attribué par le développeur (ou l'organisation de qui développe l'ontologie) ? **9 cts**
Q2. Une ontologie fournit-elle un identifiant "externe" supplémentaire, c'est-à-dire un identifiant globalement unique et persistant attribué par un organisme accrédité ? **6 cts**
Q3. Si oui, cet identifiant externe est-il un DOI ? **5 cts**
Q4. Les métadonnées de l'ontologie sont-elles incluses dans le fichier d'ontologie et partagent-elles par conséquent les mêmes identifiants ? **6 cts**
Q5. Sinon, l'ensemble des métadonnées est-il clairement identifié par son propre GUPRI ? **6 cts**
Q6. Une ontologie fournit-elle un URI spécifique à la version ? **4 cts**
Q7. Si oui, cet URI est-il résolvable/ déréférençable ? **5 cts**

Sous-principe A1. Les ontologies et les métadonnées d'ontologie sont accessibles par leur identifiant à l'aide d'un protocole de communication normalisé. A1 exprime l'importance des identifiants pour rendre une ressource accessible sur le Web. En Web sémantique, une ontologie hébergée sur un serveur Web devient accessible via le protocole standard HTTP. Ainsi, les objets d'une ontologie, (c'est-à-dire, classes, relations et métadonnées) peuvent être facilement récupérées via des services Web et dans le meilleur des cas elles peuvent aussi être disponibles sous différents formats (par exemples, JSON, HTML, texte, etc.) en utilisant la négociation de contenu.

A1 peut-être évalué en vérifiant la résolvabilité (et la prise en charge de la négociation du contenu) via HTTP des URIs d'ontologie et des métadonnées d'ontologie. Voir les métadonnées F1 liées aux identifiants pour l'évaluation de l'accessibilité sous HTTP. Dans le schéma MOD, d'autres propriétés existent pour évaluer si une ontologie et les métadonnées d'ontologie sont accessibles à travers d'autres protocoles de communication comme les requêtes SPARQL : la propriété `sd:endpoint` peut être utilisée pour stocker le point d'accès SPARQL qui peut être utilisé pour récupérer le contenu et les métadonnées de l'ontologie. De plus, les propriétés `void:openSearchDescription` et `void:uriLookupEndpoint` peuvent également être utilisées pour évaluer l'existence d'un moteur de recherche en text libre ou par URI sur l'ontologie; cependant nous définissons ces propriétés comme des métadonnées optionnelles dans F2.

Liste 2. Questions d'évaluation de A1 (43 crédits)

Q1. Est-ce que l'URI (ou d'autres identifiants) de l'ontologie se déréférence vers l'ontologie ? **6 cts**
Q2. Est-ce que l'URI de l'ontologie ou l'URI de métadonnées externes se déréférence vers les valeurs de métadonnées? **7 cts**
Q3. L'ontologie et ses métadonnées prennent-elles en charge la négociation de contenu? **24 cts**
Q4. Une ontologie et ses métadonnées sont-elles accessibles via un autre protocole standard tel que SPARQL? **6 cts**

Sous-principe I1. Les ontologies et les métadonnées d'ontologie utilisent un langage formel, accessible, partagé et largement applicable pour la représentation des connaissances. I1 met l'accent sur l'importance du langage de représentation des connaissances. Une ontologie est généralement une ressource conçue pour être compréhensible par la machine et qui repose donc sur un *langage formel*. Cependant, certaines ontologies ou ressources sémantiques peuvent être décrites sous forme textuelle ou graphique qui ne sont pas directement exploitables par une machine.

Théoriquement, une ontologie ou une ressource sémantique est sauvegardée dans un fichier en utilisant une *syntaxe* dédiée (RDF/XML, Turtle, JSON-LD) et un *langage de représentation* (OWL, SKOS, RDFS, OBO). Une ressource sémantique peut avoir différents niveaux de formalité (par exemples, ontologie, terminologie, thésaurus, vocabulaire). Les métadonnées d'ontologies sont généralement représentées en utilisant la même syntaxe et le même langage de représentation que l'ontologie elle-même. Lorsque les métadonnées sont sauvegardées dans un fichier externe, le langage de représentation des connaissances doit être évalué indépendamment.

Le sous-principe I1 peut être évalué en regardant le niveau de formalité et d'accessibilité du langage de représentation utilisé pour encoder l'ontologie ainsi que dans quelle mesure ce langage est partagé/adopté par une communauté, s'il est générique (c'est-à-dire "largement applicable" et non spécifique au domaine) et s'il est recommandé par des organismes de normalisation (dans notre cas le W3C principalement). Dans notre méthodologie, nous avons considéré que les ressources sémantiques et leurs métadonnées

peuvent être décrites en utilisant le langage OWL, OBO, RDFS et SKOS (indépendamment de leur syntaxe associée) ainsi que dans les formats CSV, XML, PDF ou TEXT. Pour l'évaluation de I1, nous pouvons regarder le langage de représentation, et le niveau de formalité utilisé et enregistré en tant que métadonnée (déclaré par les auteurs ou le portail d'hébergement). Nous pouvons aussi regarder la disponibilité de la ressource dans d'autres formats/syntaxes. Dans le schéma MOD, le langage de représentation d'une ontologie est représenté via la propriété `omv:hasOntologyLanguage`, son niveau de formalité est décrit avec la propriété `omv:hasFormalityLevel`, sa syntaxe est décrite avec la propriété `omv:hasOntologySyntax`. Si une ressource sémantique est disponible dans un autre format ou une autre syntaxe, ces informations peuvent être décrites avec les propriétés `dct:hasFormat` et `dct:isFormatOf`. La Liste 3 résume les questions d'évaluation pour ce sous-principe.

Liste 3. Questions d'évaluation de I1 (44 crédits)

Q1. Quel est le langage de représentation utilisé pour l'ontologie et les métadonnées de l'ontologie ? **20 cts***
Q2. Le langage de représentation utilisé est-il une recommandation du W3C ? **10 cts**
Q3. La syntaxe de l'ontologie est-elle déclarée ? **5 cts**
Q4. Le niveau de formalité de l'ontologie est-il déclaré ? **5 cts**
Q5. La disponibilité d'autres formats est-elle déclarée ? **4 cts**
(*) Nous proposons l'échelle suivante pour la notation de chaque langage de représentation : (owl, 20 pts) - (skos, 18 pts) - (rdfs, 16 pts) - (obo, 14 pts) - (xml, 12 pts) - (csv, 11 pts)) - (pdf, 5 pts) - (txt, 5 pts). Nous donnons une petite avance a OWL qui d'après nous est le langage le plus "formal and broadly applicable".

Sous-principe R1.1. Les ontologies et les métadonnées d'ontologie sont publiées avec une licence d'utilisation claire et accessible. Bien que l'ouverture ne soit pas un critère obligatoire pour rendre les données FAIR, il est évident que le fait de rendre les ontologies et les métadonnées d'ontologies ouvertement et librement disponibles va améliorer leur réutilisation. Quel que soit le type de licence choisie, R1.1 exige une représentation compréhensible par la machine pour la licence. En fait, l'absence d'une description explicite de la licence pourrait empêcher des personnes de réutiliser l'ontologie, même si elle était à l'origine ouverte et destinée à être partagée. Actuellement, le site RDF License[4] offre des URIs et des descriptions RDF pour la plupart des licences ; également le vocabulaire Creative Commons[5] fournit plusieurs propriétés pour garantir une description compréhensible par la machine des droits d'accès et des licences.

Le sous-principe R1.1 peut être évalué en vérifiant si les informations de licence et de droits d'accès sont fournies et résolvables (notamment la licence). Le modèle de métadonnées MOD suggère les propriétés `dct:license` pour décrire les informations de licence, et `dct:accessRights` pour détailler les droits d'accès (qui a accès à quoi). Le modèle MOD propose aussi des propriétés pour décrire les informations sur les permissions et les conditions d'utilisation associées à l'ontologie (`cc:morePermissions`, `cc:useGuidelines`) ainsi que le détenteur du droit d'auteur (`dct:rightsHolder`). Dans notre méthodologie, nous supposons que l'ontologie et ses métadonnées sont régies par la même licence – par défaut lorsque les métadonnées sont effectivement décrites dans le même fichier que l'ontologie – mais bien sûr, si ce n'est pas le cas, deux licences doivent être spécifiées et les crédits doivent être divisés en fonction. La Liste 4 résume les questions d'évaluation pour ce sous-principe.

Liste 4. Questions d'évaluation de R1.1 (37 crédits)

Q1. La licence de l'ontologie est-elle clairement spécifiée (c'est-à-dire avec un identifiant unique et persistant) ? **8 cts**
Q2. Si oui, la description de la licence est-elle accessible et résolvable par une machine ? **7 cts**
Q3. Les droits d'accès à l'ontologie sont-ils clairement spécifiés / déclarés ? **7 cts**
Q4. Les autorisations, les conditions d'utilisation et le détenteur des droits d'auteur sont-ils clairement documentés ? **15 cts**

4 Résultats

Nous avons implémenté un prototype d'évaluation du niveau de FAIRness sous la forme d'un service Web indépendant qui utilise – via l'API REST – les métadonnées des ressources sémantiques dans AgroPortal et évalue automatiquement 59 questions sur les 68 définies dans notre approche. Seulement 10 questions ne peuvent pas encore être évaluées dans AgroPortal : 1 question liée à F1, 3 questions liées à F3, 3 questions I2, et 3 questions I3. L'extension du modèle MOD avec des nouvelles propriétés devrait aider à couvrir trois de ces questions e.g., évaluer le lien entre l'ontologie et les métadonnées (F1-Q5), décrire l'état de curation (I2-Q6) ou la qualification des alignements (I3). AgroPortal étant un portail d'hébergement des ressources sémantiques donc ici, il est important de noter que notre service Web traite les métadonnées de l'objet stocké chez AgroPortal et non pas celles du fichier d'origine.

[4] http://rdflicense.appspot.com/

[5] https://creativecommons.org/licenses/

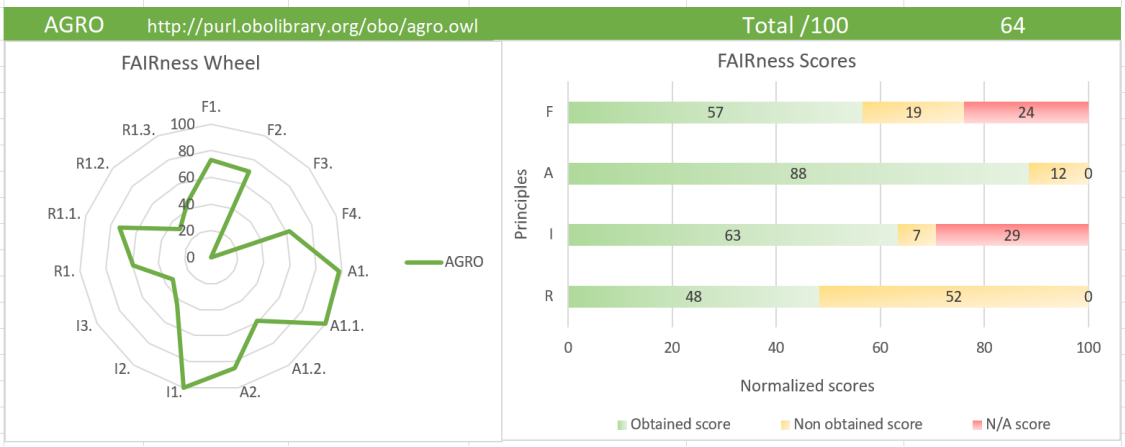

Figure 2. Synthèse graphique obtenue pour l'évaluation du niveau de FAIRness de l'Agronomy Ontology (AGRO).

Le service Web prend comme paramètre un acronyme[6] d'une ressource sémantique, c'est-à-dire un identifiant local dans AgroPortal, et renvoie en sortie un fichier JSON qui contient le score FAIR obtenu pour chaque question et chaque sous-principe (tableau 'scores'), ainsi que les scores par principe (champ numérique 'total score'). Chaque score de question est justifié par une petite phrase de justification (tableau 'explanations'), afin que l'utilisateur puisse être informé de la raison pour laquelle ce score a été obtenu. Ci-après, un exemple d'appel au service Web pour l'évaluation de l'ontologie AGRO (Agronomy Ontology) :

http://services.agroportal.lirmm.fr/fairness_assessment/?portal=agroportal&ontology=AGRO

Le temps de traitement du service Web est non linéaire, l'évaluation des questions ne dépend pas de la taille de l'ontologie (classes/relations) ; elle repose uniquement sur une liste déterminée de métadonnées décrivant l'ontologie. Un exemple de résultat pour le critère F1 est illustré dans la Figure 3.

A court terme, nous prévoyons d'enrichir notre sortie JSON avec des scores normalisés sur 100 (tel que rapportés sur la Figure 2) et de permettre la visualisation des résultats de FAIRness sur la page 'Summary' d'AgroPortal : (e.g., http://agroportal.lirmm.fr/ontologies/AGRO). Sur du plus long terme, il s'agira de représenter nos résultats dans un format structuré standard pour la représentation de FAIRness assessment; de tels formats sont actuellement en cours de discussion.

Également, le service Web prend comme paramètre le portail à utiliser pour l'évaluation du niveau de FAIRness (paramètre `portal`) car notre objectif est d'offrir un service générique pour tout portail d'ontologies mettant en œuvre le modèle de métadonnées d'ontologie MOD et/ou offrant un modèle de métadonnées riche et harmonisé pour les ontologies. Le code du prototype actuel est basé sur la version 3 d'OntoPortal (https://ontoportal.org) dans lequel nous avons changé le modèle de métadonnées [24].[7] Ce code est utilisé pour AgroPortal et le SIFR BioPortal (un portail d'ontologies et de terminologies biomédicale francaises) [26]; ainsi, pour l'instant, seul AgroPortal et le SIFR BioPortal supporte le service Web de FAIRness assessment implémenté. Dans le cadre de l'Alliance OntoPortal, nous envisageons d'étendre notre modèle de métadonnées aux autres portails d'ontologies basés sur cette technologie (i.e., NCBO BioPortal, LifeWatch EcoPortal, MedPortal et MatPortal).

```
{
  "AGRO": {
    "Findable": {
      "F1": {
        "resultSet": {
          "explanations": [
            "Valid ontology URI",
            "Resolvable ontlogy URI",
            "Valid GUID",
            "GUID is not a DOI",
            "Metadata are not included in the ontology file",
            "Metadata are identified by a resolvable URI",
            "Valid URI version",
            "Resolvable ontology URI version"
          ],
          "scores": [
            3,
            6,
            6,
            0,
            0,
            6,
            4,
            5
          ],
          "totalScore": 30
        }
```

(a) (b) (c)

Figure 3. Résultat d'évaluation de F1 pour l'ontologie AGRO. La sortie JSON montre (a) les explications données aux scores, (b) les détails des scores et (c) le score total du sous-principe concerné.

Pour analyser les scores de FAIRness par ontologie ou groupe

[6] La liste des acronymes des ontologies hébergées dans AgroPortal est disponible via le lien : http://agroportal.lirmm.fr/ontologies/

[7] https://github.com/ontoportal-lirmm

d'ontologies, nous avons produit dans une feuille Excel des graphiques synthétiques pour visualiser les résultats. La Figure 2 détaille à titre d'exemple le score obtenu pour l'ontologie AGRO. La "roue de FAIRnes" de la Figure 2 indique la répartition du score obtenu par l'ontologie ou par un groupe d'ontologies sur l'ensemble des 15 sous-principes. Le score global normalisé pour AGRO est de 64/100). L'histogramme de la Figure 2 affiche pour chaque principe : le score total obtenu (série en vert), les credits non obtenu dans AgroPortal (série jaune), et les credits qui ne peuvent pas encore être affecté/calculé au sein de AgroPortal (série en rouge). A titre d'exemple, l'ontologie AGRO a un score normalisé de 57 sur les 76 points évaluables sur AgroPortal pour le principe F.

Les moyennes que nous avons pu obtenir sur l'ensemble des ontologies d'AgroPortal nous indique qu'un score au-dessus de 65 est relativement un "bon" score de FAIRness. En effet, la moyenne des scores pour les 134 ontologies d'AgroPortal que nous avons testées en mars 2021 est de 50. La médiane est de 49. Etant donné qu'aucun des travaux existants ne définissent de valeur de référence pour définir les niveaux de FAIRness, nous nous reposons ici sur nos statistiques expérimentales comme la valeur moyenne de FAIRness. Nous expliquons dans [21] qu'une métrique est indispensable pour justement indiquer à partir de quand une ontologie n'est pas FAIR, est FAIR ou même FAIRer.

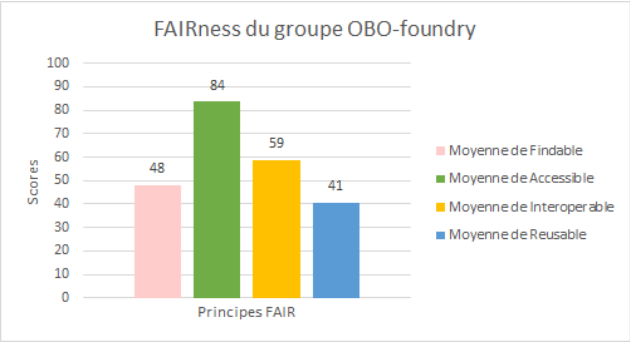

Figure 4. Synthèse de l'évaluation du degré de mise en œuvre des principes FAIR au sein du groupe 'OBO-foundry'.

La Figure 4 montre le résultat d'évaluation du degré de mise en œuvre des principes FAIR pour 24 ontologies appartenant au groupe "OBO-foundry" dans AgroPortal. Le score de FAIRness pour tout le groupe est de 58 (c'est la moyenne des scores décrits en Figure 4) : l'ontologie qui obtient le meilleure score de FAIRness est la *Phenotype And Trait Ontology* (PATO) avec un score de 65 et un des score le moins haut est obtenu pour l' *Agriculture and Forestry Ontology* (AFO). La valeur de FAIRness du groupe OBO est plutôt satisfaisante; cette valeur, au dessus de la moyenne est essentiellement obtenue grâce: (i) aux principes de conception que l'OBO Foundry demande aux ontologies qu'elle héberge (incluant des aspects sur l'utilisation des PURLs, la maintenance des fichiers de métadonnées, la clarté sur les conditions d'utilisation, et le support de négociation du contenu sous différents formats); et (ii) à l'hébergement de ces ontologies dans un 'repository' comme AgroPortal qui aide à implémenter certains principes FAIR pour n'importe laquelle des ontologies hébergées (e.g., accessibilité du contenu avec le protocole HTTP, la description riche des métadonnées, les fonctionnalités de recherche, l'alignement avec d'autres ressources, et l'archivage des versions, etc.). Néanmoins, ces ontologies peuvent devenir encore plus FAIR en améliorant la mise en œuvre des sous-principes tels que F2 et R1.2 à travers les propriétés de métadonnées MOD relatives à ces sous-principes.

Une analyse détaillée du niveau de FAIRness de toutes les ontologies de AgroPortal fera l'objet d'une autre communication.

5 Discussion et conclusions

Ce travail aborde la problématique d'évaluation de la mise en œuvre des 15 principes FAIR pour les ressources sémantiques et apporte des solutions concrètes qui faciliteraient l'adoption de ces principes par la communauté sémantique. Nous avons présenté une méthodologie et un outil d'évaluation du niveau de FAIRness. La méthodologie proposée est : (i) alignée aux approches de l'état-de-l'art, (ii) basée sur un modèle de métadonnées, (iii) générique et peut donc être appliquée pour tout type de ressource sémantique quel que soit le domaine d'application. L'implémentation que nous avons produite peut être subjective sur certains aspects (e.g., nombre de crédits par questions, liste des questions), c'est pourquoi nous avons voulu la méthodologie sous-jacente aussi générique que possible de façon à ce que chacun puisse la déployer avec ses spécificités. A termes, d'autres indicateurs (vote, usage, sondage) interviendront pour "évaluer" les outils de FAIRness assessment.Les résultats préliminaires de notre prototype montrent l'intérêt d'une évaluation par crédits pour l'ensemble des sous-principes et l'étude donne également une idée sur les analyses par ontologie ou groupe d'ontologies que nous pourrions générer à partir des scores obtenus.

Plusieurs enjeux scientifiques liés à l'évaluation du niveau de FAIRness des ressources sémantiques nécessitent d'être traités par la communauté nous citons à titre d'exemples : le besoin d'établir un consensus pour garantir la persistance des URIs (par exemple, un service d'enregistrement d'identifiants), de proposer un ensemble de métadonnées à évaluer pour chaque sous-principe, de fournir des mécanismes de standardisation et d'échange des métadonnées afin de faciliter leur récupération par les moteurs de recherche.

Prochainement, (i) nous effectuerons une analyse détaillée des scores de FAIRness pour l'ensemble des ontologies d'AgroPortal, (ii) nous réaliserons une enquête pour déterminer comment notre approche a aidé nos utilisateurs dans la sélection et l'amélioration des ressources sémantiques, (iii) nous continuerons nos efforts pour la standardisation et l'interopérabilité des métadonnées des ressources sémantiques au sein des initiatives internationales de la communauté FAIR (RDA, GO FAIR et projet FAIRsFAIR).

Remerciements

Ce travail a été réalisé dans le cadre du projet ANR *Des Données aux Connaissances en Agronomie et Biodiversité* (D2KAB – www.d2kab.org – ANR-18-CE23-0017) et du projet ANR *Participation française au GO FAIR Food Systems Implementation Network* (FooSIN – www.foosin.fr – ANR 19-DATA-0019). Nous remercions également le groupe de travail VSSIG (*Vocabulary and Semantic Services Interest Group*) de

la Research Data Alliance ainsi que le projet H2020 FAIRsFAIR pour les discussions sur les ontologies et les vocabulaires FAIR.

Références

[1] M. D. Wilkinson *et al.*, « The FAIR Guiding Principles for scientific data management and stewardship », *Sci. Data*, vol. 3, nº 1, Art. nº 1, mars 2016, doi: 10.1038/sdata.2016.18.

[2] B. Mons, *Data Stewardship for Open Science: Implementing FAIR Principles*. CRC Press, 2018.

[3] B. Mons, C. Neylon, J. Velterop, M. Dumontier, L. O. B. da Silva Santos, et M. D. Wilkinson, « Cloudy, increasingly FAIR; revisiting the FAIR Data guiding principles for the European Open Science Cloud », *Inf. Serv. Use*, vol. 37, nº 1, p. 49-56, janv. 2017, doi: 10.3233/ISU-170824.

[4] C. Caracciolo *et al.*, « 39 Hints to Facilitate the Use of Semantics for Data on Agriculture and Nutrition », *Data Sci. J.*, vol. 19, nº 1, Art. nº 1, déc. 2020, doi: 10.5334/dsj-2020-047.

[5] M. L. Zeng et P. Mayr, « Knowledge Organization Systems (KOS) in the Semantic Web: a multi-dimensional review », *Int. J. Digit. Libr.*, vol. 20, nº 3, p. 209-230, sept. 2019, doi: 10.1007/s00799-018-0241-2.

[6] Y. Le Franc, J. Parland-von Essen, L. Bonino, H. Lehväslaiho, G. Coen, et C. Staiger, « D2.2 FAIR Semantics: First recommendations », mars 2020, doi: 10.5281/zenodo.3707985.

[7] C. Jonquet *et al.*, « AgroPortal: A vocabulary and ontology repository for agronomy », *Comput. Electron. Agric.*, vol. 144, p. 126-143, janv. 2018, doi: 10.1016/j.compag.2017.10.012.

[8] C. Bizer, T. Heath, et T. Berners-Lee, « Linked Data: The Story so Far », *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 2011. www.igi-global.com/chapter/linked-data-story-far/55046 (consulté le mars 14, 2021).

[9] K. Janowicz, P. Hitzler, B. Adams, D. Kolas, et C. Vardeman II, « Five stars of Linked Data vocabulary use », *Semantic Web*, vol. 5, nº 3, p. 173-176, 2014, doi: 10.3233/SW-140135.

[10] A. Hasnain et D. Rebholz-Schuhmann, « Assessing FAIR Data Principles Against the 5-Star Open Data Principles », in *The Semantic Web: ESWC 2018 Satellite Events*, Cham, 2018, p. 469-477. doi: 10.1007/978-3-319-98192-5_60.

[11] D. Garijo et M. Poveda-Villalón, « Best Practices for Implementing FAIR Vocabularies and Ontologies on the Web », *ArXiv*, 2020, doi: 10.3233/ssw200034.

[12] N. Matentzoglu, J. Malone, C. Mungall, et R. Stevens, « MIRO: guidelines for minimum information for the reporting of an ontology », *J. Biomed. Semant.*, vol. 9, nº 1, p. 6, janv. 2018, doi: 10.1186/s13326-017-0172-7.

[13] R. David *et al.*, « FAIRness Literacy: The Achilles' Heel of Applying FAIR Principles », *Data Sci. J.*, vol. 19, nº 1, Art. nº 1, août 2020, doi: 10.5334/dsj-2020-032.

[14] C. Bahim *et al.*, « The FAIR Data Maturity Model: An Approach to Harmonise FAIR Assessments », *Data Sci. J.*, vol. 19, nº 1, Art. nº 1, oct. 2020, doi: 10.5334/dsj-2020-041.

[15] M. D. Wilkinson, S.-A. Sansone, E. Schultes, P. Doorn, L. O. Bonino da Silva Santos, et M. Dumontier, « A design framework and exemplar metrics for FAIRness », *Sci. Data*, vol. 5, juin 2018, doi: 10.1038/sdata.2018.118.

[16] M. D. Wilkinson *et al.*, « Evaluating FAIR maturity through a scalable, automated, community-governed framework », *Sci. Data*, vol. 6, nº 1, Art. nº 1, sept. 2019, doi: 10.1038/s41597-019-0184-5.

[17] « SurveyMonkey Powered Online Survey ». https://www.surveymonkey.com/r/fairdat (consulté le avr. 19, 2021).

[18] M. Mokrane, L. Cepinskas, V. Åkerman, J. de Vries, et I. von Stein, « FAIR-Aware », 2020, Consulté le: mars 14, 2021. [En ligne]. Disponible sur: https://pure.knaw.nl/portal/en/publications/fair-aware

[19] M. Poveda-Villalón, P. Espinoza-Arias, D. Garijo, et O. Corcho, « Coming to Terms with FAIR Ontologies », in *Knowledge Engineering and Knowledge Management*, Cham, 2020, p. 255-270. doi: 10.1007/978-3-030-61244-3_18.

[20] J. Frey, D. Streitmatter, F. Götz, S. Hellmann, et N. Arndt, « DBpedia Archivo: A Web-Scale Interface for Ontology Archiving Under Consumer-Oriented Aspects », in *Semantic Systems. In the Era of Knowledge Graphs*, vol. 12378, E. Blomqvist, P. Groth, V. de Boer, T. Pellegrini, M. Alam, T. Käfer, P. Kieseberg, S. Kirrane, A. Meroño-Peñuela, et H. J. Pandit, Éd. Cham: Springer International Publishing, 2020, p. 19-35. doi: 10.1007/978-3-030-59833-4_2.

[21] E. Amdouni et C. Jonquet, « FAIR or FAIRer? An integrated quantitative FAIRness assessment grid for semantic resources and ontologies », avr. 2021. Consulté le: mai 20, 2021. [En ligne]. Disponible sur: https://hal-lirmm.ccsd.cnrs.fr/lirmm-03208544

[22] N. F. Noy *et al.*, « BioPortal: ontologies and integrated data resources at the click of a mouse », *Nucleic Acids Res.*, vol. 37, nº Web Server issue, p. W170-W173, juill. 2009, doi: 10.1093/nar/gkp440.

[23] S. Jupp, T. Burdett, C. Leroy, et H. Parkinson, « A new Ontology Lookup Service at EMBL-EBI », 2015.

[24] C. Jonquet, A. Toulet, B. Dutta, et V. Emonet, « Harnessing the Power of Unified Metadata in an Ontology Repository: The Case of AgroPortal », *J. Data Semant.*, vol. 7, nº 4, p. 191-221, déc. 2018, doi: 10.1007/s13740-018-0091-5.

[25] N. Juty, S. M. Wimalaratne, S. Soiland-Reyes, J. Kunze, C. A. Goble, et T. Clark, « Unique, Persistent, Resolvable: Identifiers as the Foundation of FAIR », *Data Intell.*, vol. 2, nº 1-2, p. 30-39, janv. 2020, doi: 10.1162/dint_a_00025.

[26] C. Jonquet, A. Annane, K. Bouarech, V. Emonet, et S. Melzi, « SIFR BioPortal : Un portail ouvert et générique d'ontologies et de terminologies biomédicales françaises au service de l'annotation sémantique », Genève, Switzerland, juin 2016. Consulté le: mars 22, 2021. [En ligne]. Disponible sur: https://hal.archives-ouvertes.fr/hal-01398250

Un modèle sémantique en vue d'améliorer la FAIRisation des données météorologiques

Amina Annane[1], Mouna Kamel[1], Nathalie Aussenac-Gilles[1], Cassia Trojahn[1], Catherine Comparot[1], Christophe Baehr[2]

[1] Université de Toulouse, IRIT
[2] Météo-France, CNRM

Résumé

Rendre les données météorologiques FAIR pour faciliter leur réutilisation est un enjeu stratégique car ce sont des données essentielles à la recherche scientifique dans de nombreux domaines. Cet article propose un modèle sémantique associant un modèle de métadonnées et un modèle de données pour décrire les données météorologiques d'observation. En effet, la modélisation des (méta)données est une étape essentielle vers leur FAIRisation. Nous utilisons le jeu de données "SYNOP" de Météo-France pour illustrer les difficultés liées à l'accès et à la compréhension de ce type de données, et pour montrer comment le modèle proposé améliore leur adhésion aux principes "F", "I", et "R".

Mots-clés

Données météorologiques, principes FAIR, métadonnées sémantiques.

Abstract

Making meteorological data FAIR in order to ease its reuse is a strategic issue because this data is essential to advance research in many fields. This work proposes a semantic model which combines a metadata model and a data model for describing meteorological observation data. Indeed, modeling (meta)data is an essential step towards their FAIRification. We use the SYNOP open dataset made available by Météo-France to illustrate how difficult data access and understanding can be, and how the use of the proposed model to represent meteorological data improves their compliance with the "F", "I" and "R" principles.

Keywords

FAIR principles, meteorological data, semantic metadata, ontology.

1 Introduction

La météorologie s'appuie sur des modèles mathématiques qui agrègent des données provenant de nombreuses sources, essentiellement de capteurs disposés sur les stations, de satellites ou de radars météorologiques. Les données météorologiques sont nécessaires au développement de bon nombre d'applications, dans différents domaines tels que la météorologie, les transports, l'agriculture, la médecine, etc. Partager ces données est donc devenu un enjeu majeur pour faire des avancées scientifiques dans tous ces domaines.
Or la réutilisation des données météorologiques est difficile car initialement, ces données n'étaient utilisées que par les services météorologiques qui les produisaient : leur codification et interprétation étaient destinées aux météorologues, les usages restant contraints et limités à leurs pratiques. Si l'on veut désormais que des utilisateurs non experts en météorologie puissent les réutiliser, il faut partager, en plus des données elles-mêmes, toutes les métadonnées nécessaires pour les retrouver, y accéder, interpréter correctement, intégrer et analyser (e.g., format, signification, droits d'accès).
Pour répondre à cet enjeu, des initiatives européennes importantes ont été entreprises ces dernières années telles que la directive INSPIRE [1] ou le programme Copernicus [2]. La directive INSPIRE [6], élaborée par la Direction générale de l'environnement de la Commission européenne, impose aux autorités publiques produisant des données géolocalisées, y compris météorologiques, de les rendre publiques. Le programme Copernicus, quant à lui, met à disposition un catalogue de jeux de données d'observation de la terre et météorologiques de différentes origines : satellites, capteurs ou au sol, modèles de prévision météorologiques, etc.
Les principes **FAIR** ont été proposés pour répondre de façon plus globale à la problématique de partage des données en vue de leur réutilisation [21]. Ils consistent en un ensemble de 15 recommandations pour rendre les données faciles à (re)trouver (**F**indable), accessibles (**A**ccessible), intéropérables (**I**nteroperable) et réutilisables (**R**eusable) (Fig. 1). Selon [10], le processus de FAIRisation des données comprend trois phases (1) Pré-FAIRisation : identifier l'objectif de FAIRisation, et analyser les (méta)données (i.e., données et métadonnées), (2) FAIRisation comportant trois étapes (i) développer un modèle sémantique pour représenter les (méta)données , (ii) transformer les (méta)données en une représentation exploitable par la machine (i.e., machine-readable) en utilisant le modèle sémantique développé précédemment, et (iii) rendre les (méta)données disponibles pour les humains et les machines. Enfin, (3) Post-FAIRisation : évaluer si l'objectif fixé dans la phase (1) a été atteint.

1. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32008R1205&from=EN
2. https://atmosphere.copernicus.eu/catalogue#/

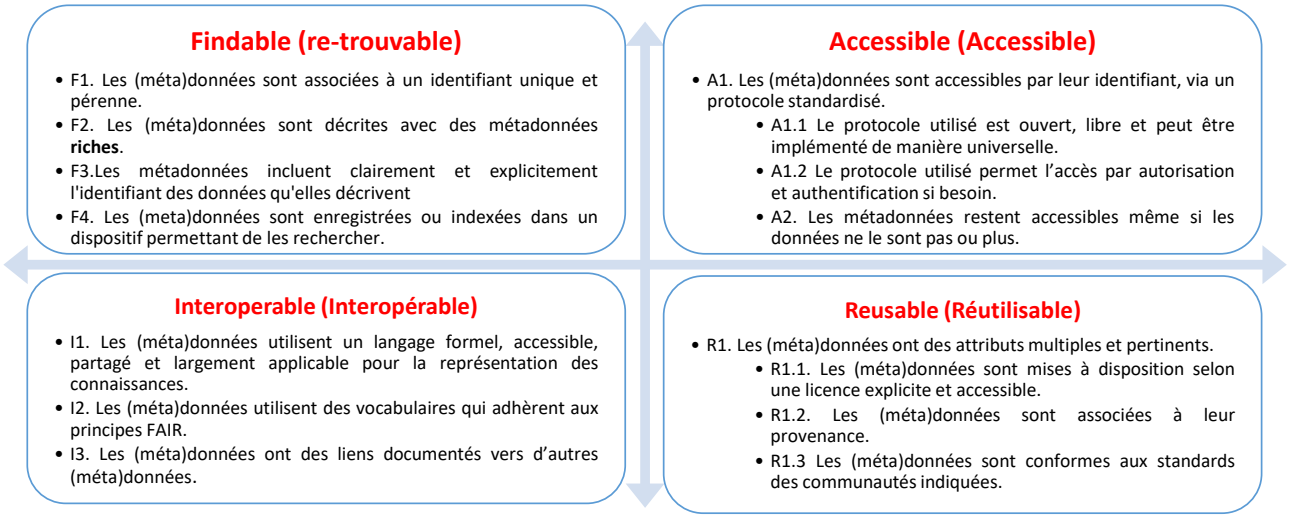

FIGURE 1 – Les principes FAIR (selon [21]).

Dans ce travail, nous nous intéressons à la FAIRisation des données météorologiques d'observation dites "in situ". Il s'agit de mesures directes de différents paramètres (température, vent, humidité, rayonnement, etc.) effectuées par des instruments au sol ou en altitude à partir de lieux prédéfinis (stations d'observation). Après avoir analysé ce type de données (phase 1), nous avons développé un modèle sémantique générique pour représenter les (méta)données (étape (i) de la phase (2)) du processus de FAIRisation. Comme souligné par les auteurs [10], le développement du modèle est l'étape la plus difficile qui prend le plus de temps. Cela vient du fait que cette modélisation nécessite à la fois une expertise métier (expertise en météorologie dans notre cas) et une expertise en modélisation sémantique de données. D'où l'intérêt d'avoir des modèles génériques pour accélérer le processus de FAIRisation. Pour développer notre modèle, nous avons bénéficié de l'expertise métier des météorologues travaillant chez Météo-France, le service officiel de météorologie en France.

Comme de nombreux travaux qui ont traité la FAIRisation des données [22], nous avons opté pour les technologies du web sémantique pour implémenter notre modèle et représenter les (méta)données. Les principes FAIR n'imposent pas l'utilisation de RDF ou de tout autre technologie du web sémantique. Néanmoins, RDF, ainsi que les ontologies formelles, constituent actuellement une solution populaire au problème du partage des connaissances qui répond également aux exigences de FAIR [17].

Le modèle proposé (section 2) est composé de deux sous-modèles pour la représentation des métadonnées et des données, respectivement. Pour que notre modèle soit à son tour FAIR, nous avons réutilisé des vocabulaires FAIR existants. Contrairement aux travaux de [16] et [19], et vu les caractéristiques des données météorologiques (voir section 2.1), nous proposons de ne pas transformer toutes les données en RDF, mais de décrire finement le schéma et la structure des données, et d'expliciter les entités sémantiques incluses dans ces données à l'aide des ontologies de domaines pour permettre une recherche de données plus efficace sans avoir à manipuler un graphe RDF immense [12].
En collaboration avec Météo-France et dans le cadre du projet ANR Semantics4FAIR [3], nous montrons comment la description du jeu de données "DONNÉES SYNOP ESSENTIELLES OMM" (dit SYNOP) par des métadonnées conformes au modèle proposé améliore son adéquation aux principes FAIR. Les données SYNOP et leur évaluation sont présentées en section 3. La section 4 situe notre contribution par rapport aux travaux similaires, avant de conclure par des perspectives en section 5.

2 Modèle Sémantique

Une des implémentations possibles du principes I2 est de développer des ontologies de représentation des métadonnées et de données [10]. Le développement de ces ontologies doit se baser autant que possible sur des ontologies FAIR existantes pour favoriser l'intéropérabilité.
Nous avons développé notre ontologie en suivant quatre étapes principales : (i) Spécification : identifier les besoins qui doivent être couverts par l'ontologie à développer. Pour ce faire, nous avons interviewé trois profils d'utilisateurs de données météorologiques (chercheur en météorologie, biologiste et chercheur en informatique) afin d'identifier un ensemble de questions (i.e., competency questions) auxquelles l'ontologie doit répondre. (ii) Sélection d'ontologies existantes : étudier les ontologies déjà disponibles, les comparer et les confronter aux besoins de représentation identifiés afin de sélectionner celles qui sont les plus pertinentes à réutiliser. (iii) Intégration d'ontologies : préciser les fragments d'ontologies à réutiliser et à combiner afin d'obtenir le modèle final, définir la manière de réutiliser ces fragments (i.e., directe ou indirecte), et vérifier le besoin d'enrichir ou non avec de nouvelles entités. (iv) Evaluation et maintenance : vérifier si le modèle obtenu répond

3. https ://www.irit.fr/semantics4fair/

aux besoin identifiés et l'entretenir. Pour plus d'informations sur la méthodologie adoptée, un rapport détaillé est disponible [4].

Dans ce qui suit, nous commençons par décrire brièvement les caractéristiques des données météorologiques d'observation (un des résultats de l'étape spécification), ensuite nous présentons notre modèle qui est une combinaison de vocabulaires/ontologies de référence complémentaires, en précisant les besoins couverts par chacun d'eux.

2.1 Propriétés des données météorologiques

Données géospatiales. Pour être exploitables, les valeurs de mesures météorologiques doivent être localisées dans l'espace. La localisation est généralement renseignée à l'aide de coordonnées géospatiales (latitude, longitude et altitude). L'interprétation de ces coordonnées dépend du système de coordonnées de référence utilisé (CRS).

Données temporelles. Chaque mesure est effectuée à un moment précis qui doit être noté avec le résultat de la mesure (c'est-à-dire la valeur de la mesure). Comme pour la localisation géospatiale, la localisation temporelle est essentielle à la bonne interprétation des mesures.

Données volumineuses. Les données météorologiques sont produites en continu. Au sein de chaque station, plusieurs capteurs sont installés (thermomètre, baromètre, etc.). Chaque capteur génère plusieurs valeurs de mesure avec une fréquence qui diffère d'une mesure à l'autre selon le besoin (horaire, journalière, trihoraire, etc.).

Données tabulaires. Les données d'observation sont généralement publiées sous forme de tableaux dans lesquels les valeurs de mesure sont organisées selon des dimensions spatio-temporelles. Selon une étude récente de Google, le format tabulaire est le format le plus répandu pour publier des données sur le Web (37 % des jeux de données indexés par Google sont en CSV ou XLS) [1].

2.2 Représentation des métadonnées

Rendre les données FAIR passe par la génération de métadonnées sémantiques qui les décrivent. En effet, 12 des 15 principes FAIR font référence aux métadonnées (Fig. 1) qui doivent de surcroît rester accessibles même si les données ne le sont plus (A2). Ces principes donnent des indications sur les catégories de métadonnées requises : (i) métadonnées descriptives pour l'indexation et la découverte des données (titre, mots-clé, etc.) ; (ii) métadonnées sur la provenance des données (R1.2) ; (iii) métadonnées sur les droits d'accès et les licences d'usage (R1.1).

Notre objectif est donc d'avoir un vocabulaire de métadonnées qui couvre ces différentes catégories, assurant ainsi l'adhésion au principe F2 (métadonnées riches), bien qu'aucune mesure ne permette de quantifier la richesse d'un ensemble de métadonnées.

GeoDCAT-AP. Nous avons choisi le vocabulaire GeoDCAT-AP [5] pour représenter les métadonnées. GeoDCAT-AP est un vocabulaire RDF permettant de décrire les catalogues de données géospatiales sur le WEB. Il a fait l'objet d'une spécification de la recommandation W3C DCAT qui a été développée par Joinup (une plateforme collaborative créée par la Commission européenne et financée par l'Union européenne dans le but de promouvoir l'intéropérabilité des données). Le choix de GeoDCAT-AP est motivé par la richesse des éléments de vocabulaire qu'il inclut. Cette richesse vient du fait que GeoDCAT-AP combine plusieurs vocabulaires/ontologies de référence PROV-O pour représenter la provenance, DCAT, Time, GeoSPARQL, DQV pour représenter des mesures sur la qualité des données, etc. Ainsi, GeoDCAT-AP couvre toutes les catégories de métadonnées citées précédemment. De plus, il offre des propriétés spécifiques et requises pour interpréter correctement des données spatiales [20] comme `dct:spatial` pour décrire la zone géographique concernée par les données, `dct:conformsTo` pour spécifier le CRS utilisé et à choisir dans une liste définie par l'OGC [6], ainsi que `dcat:spatialResolutionInMeters` pour préciser la résolution spatiale des données. De plus, GeoDCAT-AP permet de distinguer la description du jeu de données de la description de ses distributions [7] grâce à la propriété et la classe `dcat:distribution` et `dcat:Distribution`, respectivement. Ce qui permet de préciser par exemple la licence d'utilisation ou de décrire la structure interne d'une distribution spécifique.

2.3 Représentation des données

Comme discuté précédemment, les données météorologiques sont volumineuses et ne cessent de croître. Transformer toutes les archives en RDF ne nous semble pas pertinent pour deux raisons principales :

1. Coût important : transformer toutes les archives de données météorologiques en RDF nécessiterait des moyens conséquents (humains et matériels), ce qui générerait un coût important. Or, selon les principes FAIR, les données doivent être accessibles gratuitement, autant que faire se peut.
2. Efficacité : transformer les archives des données d'observation en RDF générerait un immense graphe RDF qui ne favoriserait pas l'interrogation et l'accès aux données [12].

Cependant, il est essentiel de décrire finement et de manière sémantique le schéma des données et la structure de leurs distributions, pour permettre aux humains d'interpréter et d'explorer correctement les données, et aux machines de les traiter et les interroger automatiquement [15].

Nous avons choisi le vocabulaire RDF data cube pour représenter le schéma des données indépendemment de tout format physique, et le vocabulaire csvw pour représenter la structure des distributions tabulaires car c'est le format le plus populaire. Le modèle peut être enrichi avec d'autres vocabulaires pour d'autres formats tels que JSON-LD [8] et

4. https ://www.irit.fr/semantics4fair/files/onto_report.pdf
5. `https://semiceu.github.io/GeoDCAT-AP/releases/2.0.0/`
6. http ://www.opengis.net/def/crs/EPSG/
7. Une distribution est une représentation spécifique ou une sérialisation du jeu de données
8. https ://www.w3.org/TR/json-ld/

XML [9] si besoin.
De plus, nous réutilisons des ontologies de domaine afin d'expliciter les entités incluses dans les données.

RDF Data Cube (qb). Les données météorologiques sont des données multidimensionnelles, organisées selon les dimensions espace et temps. qb est un vocabulaire dédié à la représentation de ce type de données. Il est une recommandation W3C depuis 2014. Il peut être vu comme un méta-modèle qui permet dans un premier temps de représenter le schéma des données principalement à l'aide des trois sous-classes de `qb:ComponentProperty` : (i) `qb:DimensionProperty` pour spécifier les dimensions, (ii) `qb:MeasureProperty` pour les mesures, et (iii) `qb:AttributeProperty` pour documenter les artefacts comme l'unité de mesure d'une `qb:MeasureProperty` par exemple. La représentation des données se fait dans un deuxième temps par l'instanciation du concept `qb:Observation` en affectant des valeurs aux différents dimensions, mesures, et éventuellement attributs. Comme discuté précédemment, dans notre travail, on se limite à la représentation du schéma de données, d'ailleurs la classe `qb:Observation` n'appartient pas à notre modèle. qb offre la possibilité de représenter les fragments (`qb:Slice`) appartenant au même jeu de données. Par ailleurs, la propriété `qb:concept` permet d'expliciter la sémantique des composants (mesure ou dimension) en les associant aux concepts qui leur correspondent. Ces concepts appartiennent aux ontologies de domaine. En effet, le codomaine de cette propriété est un `skos:Concept`, un type générique que peut avoir tout concept.
L'extension qb4st [10] de qb permet de décrire plus finement les aspects spatiaux et temporels, en spécialisant les classes de qb par exemple `qb4st:SpatialDimension` et `qb4st:TemporalProperty` sont des sous-classes de `qb:ComponentProperty`.

csvw. Comme souligné dans [14], il est essentiel, notamment pour la réutilisation des données, de représenter la structure interne des jeux de données. Le vocabulaire `csvw` [11] proposé par le W3C répond à ce besoin pour les données tabulaires. Dans notre modèle, une distribution dans un format tabulaire sera ainsi instance de `csvw:Table`, dont les colonnes `csvw:Column` seront spécifiées à partir du schéma (`csvw:Schema`) de la table. csvw permet aussi de représenter les relations pouvant exister entre deux distributions (une distribution correspond à un seul fichier), notamment grâce aux notions de clés primaire et étrangère (`csvw:Foreignkey`).

Ontologies de domaine. Les données météorologiques font référence à des concepts du domaine météorologique tels que la température, la vitesse du vent, l'humidité ou tout autre paramètre atmosphérique, les capteurs (e.g., thermomètre, baromètre, etc.), etc. Pour décrire au mieux les jeux de données, nous utilisons les ontologies suivantes :

- **SOSA** (Sensor, Observation, Sample, and Actuator) : ontologie de référence pour la représentation des données issues de capteurs.
- **ENVO** (ontologie de l'environnement) [5] et SWEET (Semantic Web Earth and Environment Technology ontology) [18] : elles incluent les concepts qui représentent les paramètres atmosphériques. Nous les utilisons pour les associer aux mesures représentées avec RDF Data Cube.
- **aws** : ontologie représentant les types de capteurs météorologiques selon les paramètres atmosphériques.
- **QUDT** : ontologie représentant les unités de mesure.

La Fig. 5 liste les acronymes des vocabulaire utilisés.

2.4 Nouvelles propriétés

Lorsqu'un jeu de données X dépend d'un autre jeu de données Y pour être exploité, il est essentiel pour la réutilisation de X d'expliciter cette dépendance. GeoDCAT-AP n'offrant pas cette possibilité, nous avons défini une nouvelle propriété récursive `:requires` concernant le concept `dcat:dataset` pour représenter cette relation.

Par ailleurs, il nous a semblé important de lier la représentation de la structure des données avec csvw à la représentation du schéma de données avec qb. Pour cela, nous avons rajouté la relation `:references` qui permet d'associer à chaque colonne du vocabulaire csvw, une dimension ou une mesure représentée avec qb.

La Fig. 2 présente une vue globale du modèle proposé, sans toutefois, pour des questions de lisibilité, reporter toutes les classes et propriétés utilisées.

3 Cas d'étude : données SYNOP

Aujourd'hui, l'adhésion aux principes FAIR devient un enjeu stratégique pour tout producteur de données voulant promouvoir la réutilisation de ses données. C'est le cas de Météo-France, dont les données sont difficiles à trouver et à réutiliser. En effet, il est surprenant de constater (début 2021) qu'à la requête "normales climatiques en France", les trois premiers résultats fournis par Google ne pointent pas vers le portail de Météo-France, mais vers des sites web concurrents tels que *lameteo.org*, *meteocontact.fr*, qui re-publient les données de Météo-France. Pourtant Météo-France a mis en place un portail web de données.

Afin d'évaluer le degré de FAIRisation de ces jeux de données, nous avons choisi un jeu de données représentatif : le jeu de données SYNOP ("SYNOP ESSENTIELLES OMM"). Il s'agit de données d'observations issues des messages internationaux d'observation en surface circulant sur le Système Mondial de Télécommunication de l'Organisation Météorologique Mondiale (OMM). Ce choix se justifie par le fait que ces données sont ouvertes et gratuites, et concernent plusieurs paramètres atmosphériques mesurés (température, humidité, direction et force du vent, etc.). Ces paramètres sont importants pour de nombreuses études scientifiques. À titre d'exemple et dans le cadre du projet Semantics4FAIR, des chercheurs biologistes du laboratoire

9. https ://www.w3.org/XML/Schema
10. RDF Data Cube extensions for spatio-temporal components
11. https ://www.w3.org/ns/csvw

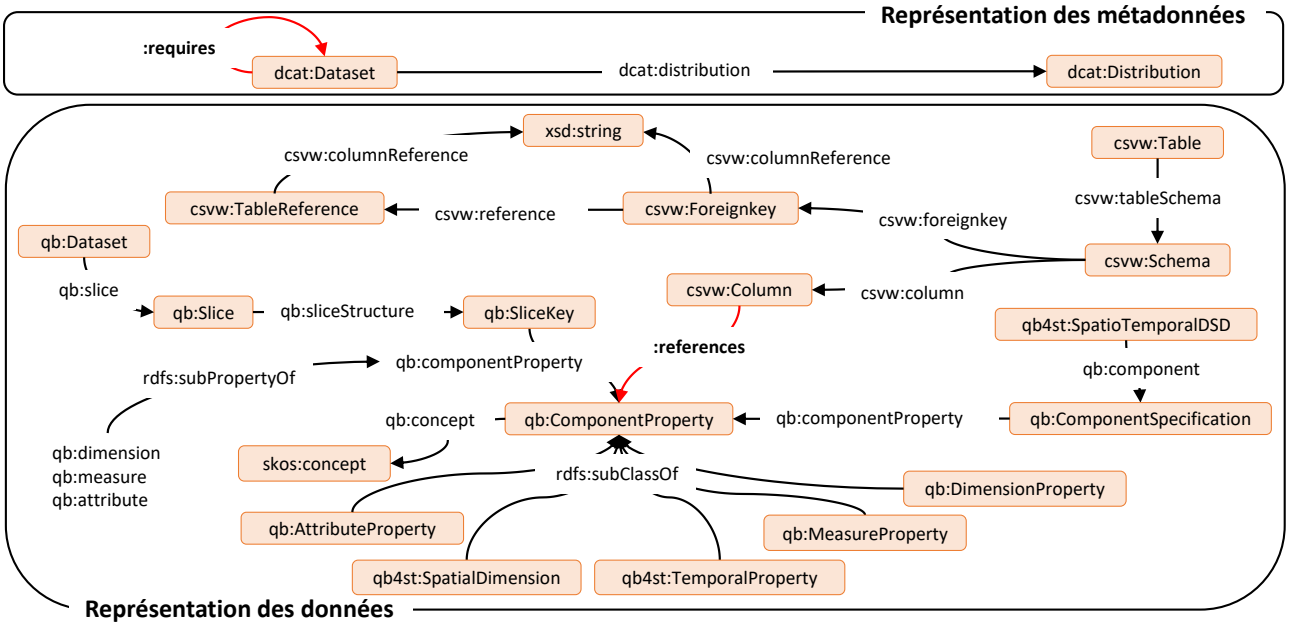

FIGURE 2 – Extrait du modèle montrant les concepts principaux.

numer_sta	date	pmer	ff	t	...
7005	20200201000000	100710	3.200000	285.450000	...
7015	20200201000000	100710	7.700000	284.950000	...
7020	20200201000000	100630	8.400000	284.150000	...
7027	20200201000000	100770	5.500000	285.650000	...
...	...	...	...	...	...

FIGURE 3 – Extrait d'un fichier de données SYNOP.

GET de l'OMP [12] qui étudient la corrélation entre les conditions météorologiques (température et humidité), et la propagation (germination et floraison) d'une plante très allergisante "l'ambroisie", affirment que les données SYNOP auraient pu répondre à leurs besoins, s'ils en avaient eu connaissance et avaient pu accéder à ces données et à leur documentation. Ces données présentent d'autant plus d'intérêt qu'elles sont publiées par tous les états membres de l'OMM, ce qui permettrait d'élargir leur étude.

3.1 Description des données SYNOP

Les données SYNOP publiées en open data [13] sont décrites par sept items : (i) *description* : description résumée du contenu en langage naturel, (ii) *conditions d'accès* : licence Etalab [14], (iii) *moyens d'accès* : téléchargement direct, (iv) *téléchargement* : téléchargement au format csv pour une date donnée, (v) *téléchargement de données archivées* : semblable à l'item précédent, mais pour un mois donné, (vi) *Informations sur les stations* : liste des stations (id_station, nom) accompagnée d'une carte affichant la localisation de ces stations, et (vii) *documentation* : trois liens qui référencent respectivement (a) un fichier *pdf* qui explicite les acronymes (libellé, type, unité de mesure) présents dans l'entête des fichiers de données SYNOP, (b) un fichier *csv* qui fournit les localisations des différentes stations météorologiques de Météo-France (id_station, nom, latitude, longitude, altitude), et (c) un fichier *json* contenant les mêmes informations que le fichier *csv* précédent. Fig. 3 montre un extrait des données SYNOP. Le fichier contient 59 colonnes, les deux premières correspondent au numéro de la station et à la date des mesures effectuées, les 57 autres colonnes aux mesures météorologiques.

Plusieurs problèmes sont alors constatés au vu de tous ces fichiers pourtant disponibles. Chaque observation, pour être exploitable, doit être localisée ; or la localisation de la station est enregistrée dans un fichier de documentation. Les noms des colonnes (acronymes) du fichier de données (Fig. 3) ne sont pas significatifs pour un utilisateur non expert en météorologie. De plus, le fichier de documentation se limite à éclater les acronymes sans donner aucune définition ou information sur la manière dont la mesure a été effectuée (type de capteur ou méthode de calcul utilisée), ou sur le type précis de la mesure (e.g., pour les températures mesurées, s'il s'agit de température de l'air, au sol, à l'abri, etc.). Enfin, la documentation en langage naturel et au format *pdf* (format difficile à traiter automatiquement) ne comporte pas les métadonnées sémantiques conformes exploitables par la machine, pour que ce jeu de données puisse être indexé par des moteurs de recherche de données comme Google dataset search [4].

12. Observatoire Midi-Pyrénées
13. https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=90&id_rubrique=32
14. https://www.etalab.gouv.fr/wp-content/uploads/2014/05/Licence_Ouverte.pdf

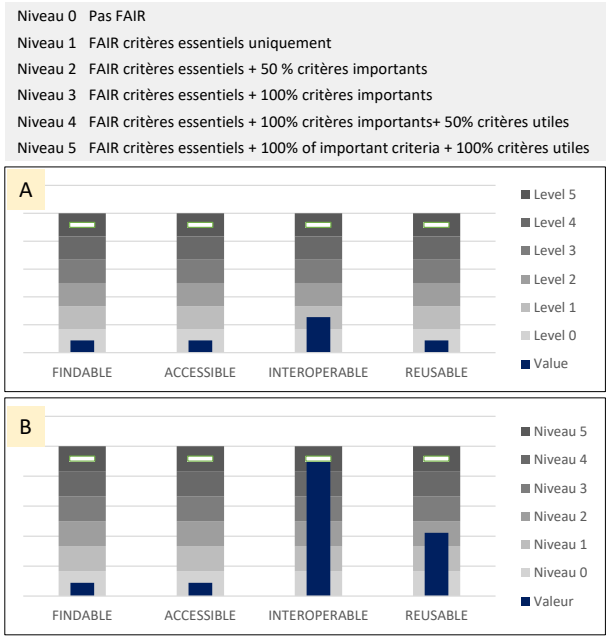

FIGURE 4 – Évaluation du jeu de données SYNOP selon le modèle de maturité de la RDA : (A) avant l'ajout des métadonnées sémantiques, (B) après l'ajout de ces métadonnées.

3.2 Évaluation du degré de FAIRisation des données SYNOP

Plusieurs travaux se sont intéressés à l'évaluation du degré de FAIRisation d'un jeu de données. Parmi ceux-ci, nous avons choisi le modèle *FAIR data maturity model* (Modèle de maturité des données FAIR) proposé par la RDA [7]. Ce modèle d'évaluation comporte : i) 41 indicateurs qui permettent de mesurer l'état ou le niveau d'une ressource digitale selon les principes FAIR ; ii) des priorités (*essentiel*, *important*, *utile*) associées aux indicateurs ; iii) 2 méthodes d'évaluation : la première dédiée aux fournisseurs de données consiste à attribuer à chaque indicateur un niveau de maturité compris entre 0 et 4 (indication permettant d'améliorer le degré de FAIRisation des données côté producteur), la seconde dédiée aux évaluateurs externes, consiste à vérifier si le critère porté par l'indicateur est vrai ou faux.

L'évaluation du jeu de données SYNOP selon ce modèle et en utilisant la seconde méthode d'évaluation a fourni les résultats suivants : (i) le niveau 0 pour les principes F, A et R car au moins un indicateur essentiel n'est pas satisfait pour ces 3 principes, et (ii) le niveau 1 pour le principe I car aucun indicateur n'est essentiel pour ce principe, le niveau 1 est le niveau minimum. Ces résultats nous permettent de conclure que, bien qu'ouvertes et publiées, les données SYNOP ne sont pas FAIR (voir Fig. 4 (A), ainsi que le rapport détaillé de l'évaluation [15]).

15. `https://hal.archives-ouvertes.fr/hal-03197115`

3.3 Instanciation du modèle proposé

Nous avons implémenté le modèle proposé [16] et vérifié sa consistance grâce aux différents raisonneurs implémentés dans Protégé (Hermit, ELK, et Pellet). De plus, en guise d'une première évaluation de sa capacité à représenter les métadonnées de jeux de données météorologiques, nous l'avons instancié avec le jeu de données SYNOP [17]. Pour plus de lisibilité, les figures 6, 7 et 8 ne décrivent qu'une partie des métadonnées qui y sont associées. De plus, les types des instances sont représentés entre parenthèse au dessous de leurs identifiants.

L'archive des données SYNOP se compose d'un ensemble de fichiers mensuels (le plus ancien date de janvier 1996), chaque fichier mensuel ne contenant que les observations réalisées durant ce mois. Nous représentons ici le jeu de données SYNOP du mois de février 2020 (voir Fig. 3).

Fig. 6 montre les différentes métadonnées associées à `:synop_dataset_feb20`, instance des deux classes `dcat:Dataset` et `qb:Slice`. En effet, chaque jeu de données mensuel est représenté par une instance de `dcat:Dataset` conformément aux bonnes pratiques de la publication des données sur le web *"...each dataset covers a different set of observations about the world should be treated as a new dataset."* [18]. On observe que le producteur de données est Météo-France (`dct:Creator`), les données proviennent des Stations Météo-France (`dct:provenance`), les données couvrent la période du 1er février 2020 au 29 février 2020 (`dcat:startDate` et `dcat:endDate`), le système de coordonnées de référence est le `crs:4326` (`dct:conformsTo`), la couverture spatiale est la France (`dct:spatial`), etc. Notons l'utilisation de la nouvelle propriété `:requires` pour représenter que `:synop_dataset_feb20` nécessite le jeu de données `:station_dataset` pour être exploité. Le jeu de données `:synop_dataset_feb20` possède une distribution au format csv représenté par `:synop_distribution_feb20`, instance de `dcat:distribution`. Les métadonnées associées à cette distribution sont décrites sur la Fig. 8. Notons l'utilisation des vocabulaires contrôlés (rectangles bleus) spécifiés par GeoDCAT-AP pour une meilleure intéropérabilité et intégration de données.

Fig. 7 décrit la représentation de la structure de données SYNOP avec qb et les ontologies de domaines. Les rectangles vides correspondent à des noeuds anonymes "blank nodes". Les données SYNOP (tous les fichiers mensuels) ont exactement la même structure, d'où la définition d'une seule structure de données `:synop_dataset_structure` instance de `qb4st:SpatioTemporalDSD`.

Toutes les données SYNOP sont alors représentées par une instance de `qb:Dataset`, et chaque fragment mensuel par une instance de `qb:Slice`, en plus d'être une instance de `dcat:Dataset`. Cette modélisation permet de regrouper toutes les parties d'un même jeu de données.

16. https ://www.irit.fr/semantics4fair/files/MeteOnto.owl
17. https ://www.irit.fr/semantics4fair/files/synop_feb_2020.ttl
18. https ://www.w3.org/TR/dwbp/#dataVersioning

@prefix	URI
base	https://synop-example.ttl#
aws	http://purl.oclc.org/NET/ssnx/meteo/aws#
cat	http://inspire.ec.europa.eu/metadata-codelist/TopicCategory/
country	http://publications.europa.eu/resource/authority/country/
crs	http://www.opengis.net/def/crs/EPSG/0/
csvw	http://www.w3.org/ns/csvw
dcat	http://www.w3.org/ns/dcat#
dct	http://purl.org/dc/terms/
foaf	http://xmlns.com/foaf/0.1/
freq	http://publications.europa.eu/resource/authority/frequency/
geodcatap	http://data.europa.eu/930/
gsp	http://www.opengis.net/ont/geosparql#
lang	http://publications.europa.eu/resource/authority/language/
owl	http://www.w3.org/2002/07/owl#
prov	http://www.w3.org/ns/prov#
qb	http://purl.org/linked-data/cube#
qb4st	http://www.w3.org/ns/qb4st/
qudt	http://qudt.org/1.1/vocab/unit#
qudts	http://qudt.org/1.1/schema/qudt#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
sosa	http://www.w3.org/ns/sosa/
sweet	http://sweetontology.net/propPressure/
time	http://www.w3.org/2006/time#
type	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/
vcard	http://www.w3.org/2006/vcard/ns#
xsd	http://www.w3.org/2001/XMLSchema#
timePeriod	http://publications.europa.eu/resource/authority/timeperiod/
year	http://reference.data.gov.uk/id/year/

FIGURE 5 – Liste des vocabulaires réutilisés

Nous avons défini une dimension spatiale `:station_dimension` et trois dimensions temporelles : `:year_dimension`, `:month_dimension`, et `:date_dimension`. La nature spatiale ou temporelle d'une dimension est précisée à l'aide des concepts du vocabulaire qb4st, respectivement `qb4st:SpatialDimension` et `qb4st:TemporalProperty`. Il est à noter que les dimensions `:year_dimension` et `:month_dimension` qui ne correspondent pas à des colonnes à proprement parler mais qui sont contenues dans la dimension `:date_dimension`, ont été rajoutées pour pouvoir instancier des `qb:Slices`. En effet, selon le vocabulaire qb, chaque instance de `qb:Slice` doit être associé à une instance de `qb:SliceKey` qui définit un sous-ensemble de dimensions à valeurs fixes via la propriété `qb:componentProperty`. Dans notre cas, pour l'instance `:synop_dataset_feb20` qui est un jeu de données mensuelles, les dimensions fixes sont l'année `:year_dimension` et le mois `:month_dimension` qui ont les valeurs `month:FEB` et `:2020`. Bien que la dimension `:station_dimension` ne soit pas directement une coordonnée géographique, elle est définie comme instance de `qb4st:spatialDimension` car elle permet d'accéder aux coordonnées géospatiales contenues dans le jeu de données des stations.

Chaque dimension ou mesure est associée à un concept de domaine, grâce à la propriété `qb:concept`. Ainsi, la mesure `:pmer_measure` (seule représentée ici alors que les 57 mesures ont été instanciées) est liée aux concepts `sosa:observableProperty` et `sweet:SeaLevelPressure` pour en expliciter le sens. L'attribut `:unit_of_measure_attribute`, correspondant à `qudts:physicalUnit` permet de documenter les unités de mesure des `qb:Measure`. Cela permet de spécifier que l'unité de mesure de `pmer_measure` est `qudt:Pascal`. Ainsi le contenu de ce jeu de données peut être indexé par ces concepts de domaines et pas uniquement des mots clés (chaîne de caractères).

En plus de la description du schéma de données avec qb, notre modèle permet de décrire d'une manière sémantique la structure de la distribution synop à l'aide des entités venant de csvw (Fig. 8). `:synop_distribution_feb20` est une instance à la fois de `dcat:distribution` et de `csvw:Table`. Cette distribution est accessible et téléchargeable via les URL (`dcat:accessURL` et `dcat:downloadURL`) reportées en Fig. 8; elle est soumise à une licence (`dct:license`). Enfin, les colonnes (ici `num_sta`, `date`, et `pmer`) de ce fichier sont caractérisées par leur nom (`csvw:name`), leur libellé (`csvw:title`), leur type (`csvw:datatype`) à partir du schéma `:synop_file_schema` (`csvw:tableSchema`), etc. Notons également la représentation de la clé étrangère `:fk` qui relie la colonne "num_sta" du fichier des données SYNOP, à la colonne "ID" du fichier des stations (`:station_distribution`) en passant par l'instance `:tr` de `csvw:TableReference`. Enfin, la nouvelle propriété `:references` est utilisée pour expliciter la sémantique de chaque colonne, en lui associant une mesure ou une dimension. Grâce à la représentation fine qui combine qb et csvw, les données peuvent être traitées et interrogées automatiquement sans être transformées en RDF.

Le jeu de données SYNOP a été réévalué une fois l'ensemble de ces métadonnées sémantiques générées. L'amélioration du degré de FAIRisation peut être observé Fig.4(B), particulièrement pour les principes "I" et "R" . Bien que l'évaluation du principe "F" n'ait pas montré d'amélioration, le modèle permet de représenter des métadonnées d'indexation "riches" qui sont essentielles pour le "F". Pour ce qui est du processus de FAIRisation dans sa totalité pour ce jeu de données SYNOP, les principes "F" et "A" nécessitent de générer des identifiants pérennes et uniques, et de publier les métadonnées générées sur le web. Ce sera la prochaine étape de notre travail.

3.4 Discussion

Dans ce travail nous nous sommes intéressés à une description sémantique des (méta)données météorologiques en vu de les rendre réutilisables, le but ultime des principes FAIR. Cette description est essentielle mais pas suffisante, elle doit être accompagnée d'autres efforts tels que la mise en place d'API d'accès à distance aux données, l'affectation d'identifiants pérennes, la publication et l'indexation des métadonnées générées sur le web, etc.

Certaines métadonnées pertinentes ne sont pas disponibles même si le modèle permet de les représenter, comme les instruments de mesure, les mesures de qualité, etc. Aussi,

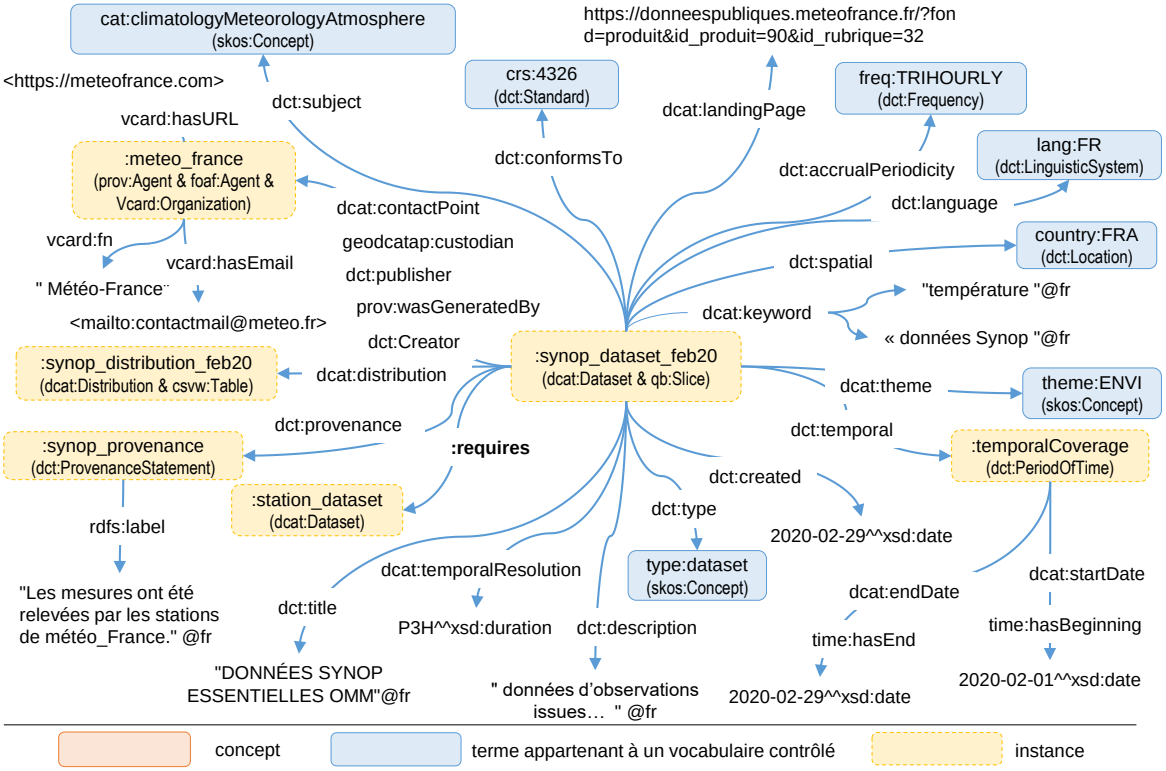

FIGURE 6 – Représentation des données SYNOP pour le mois de février 2020 avec GeoDCAT-AP.

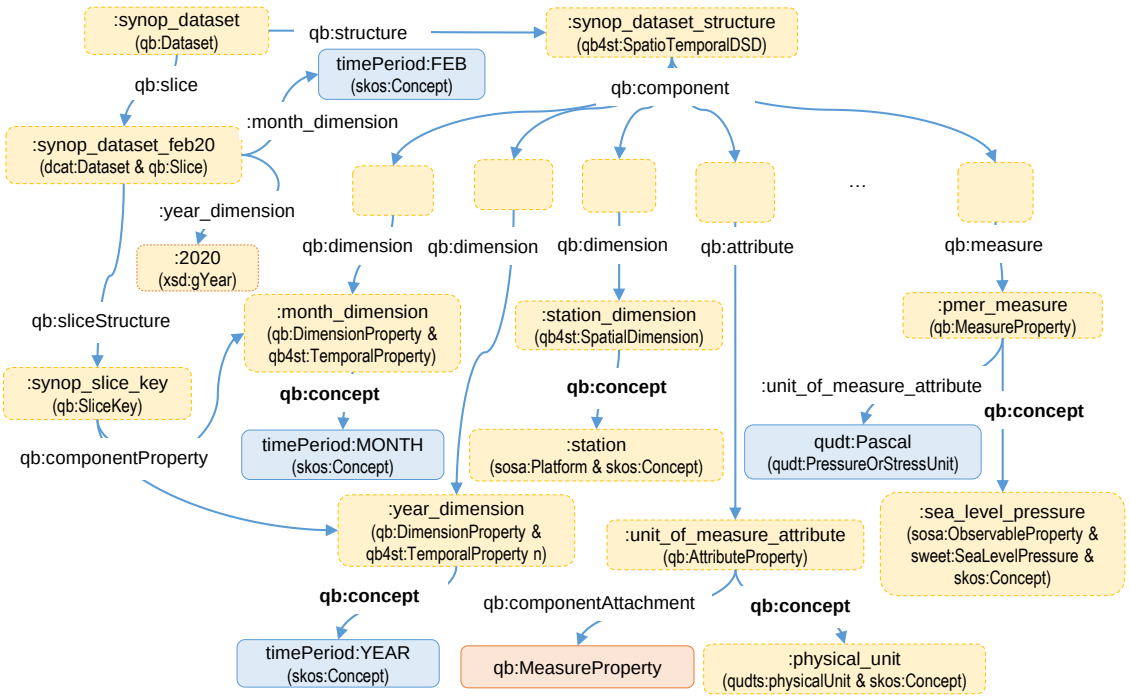

FIGURE 7 – Représentation sémantique des données SYNOP avec RDF data cube, QB4ST et les ontologies du domaine.

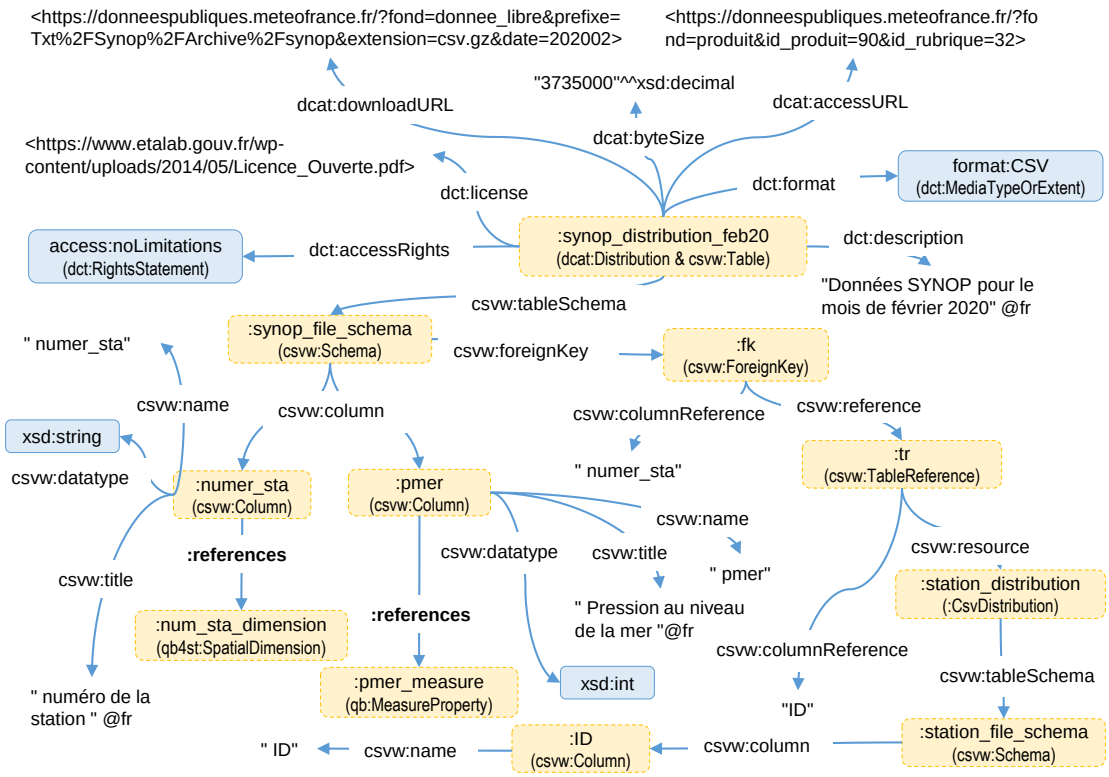

FIGURE 8 – Représentation de la distribution des données SYNOP de février 2020 avec GeoDCAT-AP et CSVW.

seule la dernière localisation (latitude, longitude, et altitude) de chaque station est fournie en documentation, alors que ces stations sont amenées à changer de localisation au fil du temps. Il est important de partager l'historique des localisations des stations dans un format facilement exploitable, pour permettre à l'utilisateur de connaître la localisation exacte de la station au moment de la mesure.
Il est aussi à noter qu'en météorologie, les procédures de mesure, les types de capteurs à utiliser, les normes de qualité, qui sont des métadonnées importantes pour la réutilisation, sont définis par l'OMM. Cette dernière fournit des guides détaillés, tel que "le guide des instruments et des méthodes d'observation météorologiques" [19]. Néanmoins, à notre connaissance, aucune version sémantique de ces guides n'existe. Nous pensons qu'il serait très intéressant de sémantiser ces guides pour une meilleure documentation des données météorologiques en faveur de leur réutilisation.

4 Travaux liés

Approches sémantiques et critères FAIR. Grâce à leur capacité à rendre explicites les types des données, dans un format manipulable par des services, et à produire des métadonnées riches, les ontologies contribuent doublement à rendre les données FAIR, les principes I et R étant sans doute ceux pour lesquels cette contribution est la plus immédiate. Les autres recommandations du I peuvent être assurées en liant les données à d'autres données formalisées (identifiants uniques d'auteurs, de journaux scientifiques, de lieux ou d'organismes) par des liens d'identité [8] ou grâce à l'alignement d'ontologies [23] [2]. Guizzardi redéfinit l'interopérabilité sémantique entre deux systèmes comme l'interconnexion non seulement des données, mais aussi de leur conceptualisation [9]. La réutilisabilité (R) fait référence non seulement à la richesse des métadonnées, aux standards utilisés pour les représenter et à leur accessibilité, mais également à la connaissance de la provenance des données. Pour cela, [8] utilisent l'ontologie PROV-O [20].
Comme recommandé par [11] et mis en oeuvre dans [3], nous enrichissons des vocabulaires standards selon les besoins du cas d'utilisation afin de décrire les métadonnées, y compris la provenance des données, et les types des données selon des concepts du domaine. L'originalité de notre contribution est d'y ajouter des métadonnées décrivant la structure de chaque jeu de données, et les liens entre les éléments de structure et les types de données.

Représentation de données météorologiques. En ce qui concerne le partage des données géolocalisées (comme les données météorologiques), plusieurs schémas de métadonnées existent. L'article [13] compare huit de ces schémas, et ainsi identifie sept critères cruciaux pour la description des données spatiales. Parmi les nombreux vocabulaires possibles pour représenter les données météorologiques et atmosphériques, SWEET [21] et SOSA [22] ainsi que les

19. https://library.wmo.int/doc_num.php?explnum_id=4148

20. https ://www.w3.org/TR/prov-o/
21. https ://www.github.com/ESIPFed/sweet
22. https ://www.w3.org/TR/vocab-ssn/

standards pour représenter des données spatio-temporelles (OWL-Time et Geo-SPARQL) sont parmi les plus utilisés.

5 Conclusion

Nous avons présenté un modèle sémantique générique pour décrire des jeux de données météorologiques d'observation. Ce modèle permet de représenter explicitement et formellement des informations jusque là non disponibles au sujet des jeux de données, à la fois sur la structure et leur contenu en des termes du domaine. Ainsi, les jeux de données vont être plus faciles à trouver et répondent mieux (mais encore incomplètement) aux critères FAIR. La prochaine étape consistera à étudier les spécificités des données issues des modèles statistiques pour enrichir le modèle actuel si besoin. Enfin, nous utiliserons le modèle final pour générer les métadonnées et les indexer dans des portails de données. Nous envisageons aussi de travailler sur la recherche sémantique des jeux de données.

Remerciements

Ce travail est financée par le projet ANR Flash Semantics4FAIR, contrat ANR-19-DATA-0014-01.

Références

[1] O. Benjelloun, S. Chen, and N. F. Noy. Google dataset search by the numbers. In *19th International Semantic Web Conf.*, pages 667–682, 2020.

[2] F. Beretta. A challenge for historical research : making data FAIR using a collaborative ontology management environment (OntoME). *Semantic Web – Interoperability, Usability, Applicability*, 2020.

[3] C. Brewster, B. Nouwt, S. Raaijmakers, and J. Verhoosel. Ontology-based access control for fair data. *Data Int.*, 2 :66–77, 11 2019.

[4] D. Brickley, M. Burgess, and N. F. Noy. Google dataset search : Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference, WWW2019*, pages 1365–1375. ACM, 2019.

[5] P. L. Buttigieg, N. Morrison, B. Smith, C. J. Mungall, and S. E. Lewis. The environment ontology : contextualising biological and biomedical entities. *J. Biomed. Semant.*, 4 :43, 2013.

[6] Drafting Team Metadata and European Commission Joint Research Centre. INSPIRE Metadata Implementing Rules : Technical Guidelines based on EN ISO 19115 and EN ISO 19119 - V.1.3., 2013.

[7] FAIR Data Maturity Model Working Group RDA. FAIR Data Maturity Model. Specification and Guidelines, June 2020.

[8] J. D. Fernández, N. Lasierra, D. Clement, H. Mason, and I. O. Robinson. Enabling fair clinical data standards with linked data. In *ESWC*, 2020.

[9] G. Guizzardi. Ontology, Ontologies and the "I" of FAIR. *Data Int.*, 2(1-2) :181–191, nov 2020.

[10] A. Jacobsen and et al. FAIR principles : Interpretations and implementation considerations. *Data Intelligence*, 2(1-2) :10–29, 2020.

[11] A. Jacobsen, R. Kaliyaperumal, L. O. B. da Silva Santos, B. Mons, E. Schultes, M. Roos, and M. Thompson. A generic workflow for the data fairification process. *Data Intelligence*, 2(1-2) :56–65, 2020.

[12] F. Karim, M. Vidal, and S. Auer. Compact representations for efficient storage of semantic sensor data. *CoRR*, abs/2011.09748, 2020.

[13] T. J. Kim. Metadata for geo-spatial data sharing : A comparative analysis. *The Annals of Regional Science*, pages 33 :171–181, 1999.

[14] L. Koesten, E. Simperl, T. Blount, E. Kacprzak, and J. Tennison. Everything you always wanted to know about a dataset : Studies in data summarisation. *Int. J. Hum. Comput. Stud.*, 135, 2020.

[15] P. Kremen and M. Necaský. Improving discoverability of open government data with rich metadata descriptions using semantic government vocabulary. *J. Web Semant.*, 55 :1–20, 2019.

[16] L. Lefort, J. Bobruk, A. Haller, K. Taylor, and A. Woolf. A linked sensor data cube for a 100 year homogenised daily temperature dataset. In *5th Inter. Works. on Semantic Sensor Networks*, volume 904, pages 1–16, 2012.

[17] B. Mons, C. Neylon, J. Velterop, M. Dumontier, L. da Silva Santos, and M. D. Wilkinson. Cloudy, increasingly fair ; revisiting the FAIR data guiding principles for the european open science cloud. *Inf. Serv. Use*, 37(1) :49–56, 2017.

[18] R. Raskin. Development of ontologies for earth system science. In *Geoinformatics : Data to Knowledge*. Geological Society of America, 01 2006.

[19] C. Roussey, S. Bernard, G. André, and D. Boffety. Weather data publication on the LOD using SOSA /SSN ontology. *Semantic Web*, 11(4) :581–591, 2020.

[20] L. van den Brink, P. Barnaghi, J. Tandy, G. Atemezing, R. Atkinson, B. Cochrane, Y. Fathy, R. Castro, A. Haller, A. Harth, et al. Best practices for publishing, retrieving, and using spatial data on the web. *Semantic Web*, 10(1) :95–114, 2019.

[21] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1) :1–9, 2016.

[22] M. D. Wilkinson and et al. Interoperability and fairness through a novel combination of web technologies. *PeerJ Comput. Sci.*, 3 :e110, 2017.

[23] J. Wise, A. G. de Barron, A. Splendiani, B. Balali-Mood, D. Vasant, E. Little, G. Mellino, I. Harrow, I. Smith, J. Taubert, et al. Implementation and relevance of fair data principles in biopharmaceutical R&D. *Drug Discovery Today*, 24(4) :933–938, 2019.

Description sémantique des stades de développement phénologique des plantes, cas d'étude de la vigne

Catherine Roussey[1], Xavier Delpuech[2], Marc Raynal[3],
Florence Amardeilh[4], Stephan Bernard[1], Clement Jonquet[5, 6],
Camille Noûs[7]
[1] Université Clermont Auvergne, INRAE, UR TSCF, Aubière
[2] Institut français de la vigne et du vin, Pôle Rhône-Méditerranée, Montpellier
[3] Institut français de la vigne et du vin, Pôle Nouvelle Aquitaine / UMT SEVEN, Bordeaux
[4] Elzeard R&D, Pessac
[5] MISTEA, Université de Montpellier, INRAE, Institut Agro, Montpellier
[6] LIRMM, Université de Montpellier, CNRS, Montpellier
[7] Laboratoire Cogitamus

catherine.roussey@inrae.fr, xavier.delpuech@vignevin.com, marc.raynal@vignevin.com,
florence.amardeilh@elzeard.co, stephan.bernard@inrae.fr, jonquet@lirmm.fr,
camille.nous@cogitamus.fr

Résumé

Le projet ANR "Des Données aux Connaissances en Agronomie et Biodiversité" (D2KAB) met à disposition une archive de bulletins agricoles publiée sur le Web. Pour annoter les bulletins à l'aide des stades de développement des plantes, nous avons besoin d'une nouvelle ressource sémantique. Plusieurs échelles phénologiques existent, chacune de ces échelles définit un ensemble spécifique de stades. L'Institut Français de la Vigne et du vin (IFV) a aligné plusieurs échelles existantes associées à la vigne. Nous présentons ici la création d'un cadre méthodologique pour la description sémantique des stades de développement des plantes fondé sur l'ontologie PPDO (BBCH-based Plant Phenological Description Ontology). [1]

Mots-clés

Ontologie, Web de données, Données liées, Stades phénologiques /de développement, Agriculture, Vigne, BBCH.

Abstract

The French ANR project "Data to Knowledge in Agronomy and Biodiversity" (D2KAB) builds an archive of French agricultural alert newsletters. In order to annotate plant development stages, we need a new semantic resource. Several phenological scales already exist to describe plant development stages. The French Wine and Vine Institute (IFV) has aligned several existing scales related to grapevine. In this paper, we present our work of creating an ontological framework for semantic description of plant development stages ; we introduce the BBCH-based Plant Phenological Description Ontology. [2]

Keywords

Ontology, Wef of Data, Linked Open Data, Phenological / Development stages, agriculture, Grapevine, BBCH

1 Introduction

L'agronomie et l'agriculture sont confrontées à plusieurs défis sociétaux, économiques et environnementaux majeurs, qu'une approche sémantique de la science des données aidera à relever. Le projet ANR *Des Données aux Connaissances en Agronomie et Biodiversité* (D2KAB) [3] illustre comment la science des données sémantiques contribue au développement d'applications agricoles innovantes. L'objectif de D2KAB est de créer un cadre pour transformer les données d'agronomie et de biodiversité en connaissances décrites sémantiquement, interopérables, exploitables et ouvertes. Pour construire un tel cadre, nous nous appuyons sur des ressources sémantiques (par exemple, des terminologies, des vocabulaires ou des ontologies) pour décrire nos données et les publier en tant que données ouvertes liées [6]. Nous utilisons le portail web AgroPortal [4] [13] pour trouver, publier et partager des ressources sémantiques puis nous exploitons ces ressources sémantiques dans des applications dédiées à l'agriculture ou l'environnement.

L'un des scénarios agricoles de D2KAB consiste à construire un navigateur web augmenté pour les bulletins officiels d'alertes agricoles français, appelés *Bulletins*

1. Cet article est une traduction et extension d'un article publié en anglais en Décembre 2020 à la 14ème conférence internationale *Metadata and Semantics Research (MTSR).*

2. This article is a translation and an extension of an article published in English, in December 2020 at the 14th International Conference on Metadata and Semantics Research (MTSR).

3. www.d2kab.org

4. http://agroportal.lirmm.fr

de Santé du Végétal (BSV). Le prototype interrogera une archive de BSV disponible sous forme de fichiers PDF. Chaque bulletin sera annoté sémantiquement, et les annotations seront publiées sur le Web de données liées. Les annotations seront produites, entre autre, à partir de techniques de traitement automatique de la langue appliquées sur les contenus textuels des BSV. Les annotations seront décrites en utilisant les ontologies Semantic Sensor Network combinées au Web Annotation Data Model comme présenté dans de précédents travaux [3].

Des mentions de stades phénologiques sont toujours présentes dans les BSV, afin d'identifier les périodes de sensibilité d'une culture aux facteurs abiotiques (par exemple, le gel printanier) ou à des facteurs biotiques (par exemple, la maladie du mildiou). Afin d'annoter les stades phénologiques des cultures, nous avons besoin d'une ressource sémantique spécifique pour représenter les stades, l'échelle phénologique à laquelle ils appartiennent et les cultures auxquelles ils s'appliquent. Comme il existe plusieurs échelles phénologiques par culture, nous avons aussi besoin d'aligner les stades des échelles différentes concernant la même culture. A noter que la plupart des échelles phénologiques sont dédiées à une seule culture. Une des échelles les plus connues et qui s'applique à plusieurs cultures est l'échelle BBCH [17]. Malheureusement, ces échelles ne sont pas publiées sur le Web de données liées, ni disponibles dans un format du Web sémantique (e.g., SKOS, RDF-S ou OWL), empêchant leur utilisation dans différentes applications agricoles. Notre corpus de BSV montre que différentes échelles ont été utilisées pour observer les stades phénologiques. Par exemple, dans un BSV lié à la vigne, on trouve la phrase : *bourgeon dans le coton (stade 03 ou B ou BBCH 05) dans les secteurs tardifs*, qui est à la fois une référence à un 'label' ("bourgeon dans le coton ") et la codification de plusieurs échelles : (i) 03 est un code de l'échelle Eichhorn-Lorenz [14] ; (ii) B est un code de l'échelle de Baggiolini [5] ; (iii) et BBCH 05 est un code de l'échelle BBCH [17]. Ces codes devront être identifiés automatiquement dans un texte pour être associés à un identifiant pérenne représentant le stade d'une échelle phénologique. De plus, il semble que les labels les plus fréquemment utilisés soient ceux d'une échelle produite par l'Institut Français de la Vigne et du vin (IFV). Malheureusement cette échelle propose uniquement des labels. D'après notre expérience sur l'usage des BSV, il est nécessaire de construire un ensemble de jeux de données alignés pour décrire sémantiquement les stades des différentes échelles.

Cet article est une traduction et extension d'un article précédent [22] ; nous présentons la finalisation de notre travail de transformation en RDF de six échelles (une de plus que l'article précédent) : l'échelle générale BBCH qui s'applique à toutes les cultures [17], l'échelle individuelle BBCH de la vigne [17], l'échelle Eichhorn-Lorenz [14] qui ne concerne que la vigne, l'échelle Baggiolini de la vigne [5], l'échelle produite par IFV qui est une sélection de l'échelle individuelle BBCH de la vigne (intitulée IFV-labels), l'échelle produite par IFV qui est un enrichissement de l'échelle Eichhorn-Lorenz (intitulée IFV-Epicure).

Nous montrerons plus particulièrement notre travail d'alignement entre ces échelles. Nous présenterons la première version de l'ontologie intitulée *BBCH-based Plant Phenological Description Ontology (PPDO)*. Cette ontologie est le support d'un cadre méthodologique pour encoder sémantiquement toute échelle phénologique. Cette ontologie décrit l'échelle générale BBCH sous forme de classes OWL. Puis, chaque stade d'une échelle BBCH spécifique à une culture est représentée par une instance des classes précédentes.

Le reste de l'article est organisé comme suit : La section 2 introduit la phénologie et présente les échelles que nous avons transformées ; La section 3 passe en revue les thésaurus/ontologies existants pour décrire les stades phénologiques ; La section 4 illustre la méthodologie utilisée pour construire l'ontologie PPDO et la peupler avec les stades des échelles de la vigne ; La section 5 discute des avantages et des inconvénients de la publication des échelles phénologiques sous forme de données ouvertes liées.

2 Echelles phénologiques

La phénologie des plantes est l'étude du développement saisonnier des végétaux déterminé par l'influence des variations du climat. Les événements périodiques sont par exemple l'apparition des fleurs, le changement de couleurs des feuilles, etc. Un stade phénologique ou stade de développement des plantes caractérise une phase de développement de la plante pendant son cycle de vie. Par exemple, le stade floraison est observé quand au moins 50% des fleurs sont épanouies. L'étude phénologique d'une plante consiste à observer à quelle date les stades apparaissent. Les variations temporelles sur l'apparition d'un stade peuvent dépendre du climat ou d'autres facteurs comme la variété de la plante.

L'échelle phénologique BBCH améliorée (*Biologische Bundesanstalt, Bundessortenamt und CHemische Industrie*) propose une codification homogène des stades de développement communs à différentes espèces végétales cultivées. BBCH décrit plusieurs ensembles de stades de développement : l'échelle générale et des échelles spécifiques par culture (dites « échelles individuelles ») [17]. L'échelle générale forme le cadre dans lequel les échelles individuelles sont développées. Elle peut également être utilisée pour les cultures n'ayant pas fait l'objet de description d'échelle individuelle : « *Les différentes phases du développement d'une plante sont divisées en dix stades principaux numérotés de 0 à 9. (...) Pour définir précisément les stades d'application ou d'évaluation, il n'est pas conseillé d'utiliser seulement les stades principaux, car ceux-ci recouvrent une durée importante dans le développement de la plante. On utilise les stades de développement secondaires pour déterminer un moment exact du développement. Par rapport aux stades principaux, les stades secondaires définissent des stades limités dans le temps. Ils sont donnés par des valeurs de 0 à 9, qui se suivent à l'intérieur d'un stade principal. On obtient ainsi un code à deux chiffres composé par le stade principal et le stade secondaire.* » [17]. BBCH est

considéré comme une référence pour décrire les stades et la monographie est disponible en anglais, français, allemand et espagnol [15, 17, 16, 18].

Pour la vigne, la première échelle phénologique a été proposée par Baggiolini [4] en 1952. Elle comportait initialement 10 stades qui ont été complétés en 1993 par Baillod et Baggiolini [5] pour atteindre 16 stades. En 1977, Eichhorn et Lorenz (EL) ont proposé une échelle plus détaillée de 24 stades pour la vigne. En 1992, une échelle universelle connue sous le nom de BBCH pour les plantes cultivées mono et dicotylédones a été proposée par Hack et al., puis adaptée à la vigne en 1995 par Lorenz et al. [14]. L'échelle individuelle BBCH de la vigne permet la comparaison avec d'autres espèces, aussi bien annuelles que pérennes, et s'est progressivement imposée comme une échelle de référence dans la communauté scientifique.

En 1995, Coombe [7] a proposé un alignement entre les différentes échelles existantes de la vigne tout en proposant des modifications de l'échelle EL. Cependant, dans la communauté technique agricole française, différentes échelles phénologiques sont parfois utilisées selon les habitudes des techniciens. L'IFV a donc produit une fiche technique [11] pour synthétiser les stades de développement de la vigne les plus utiles et les plus pertinents. Les stades sélectionnés ont été définis à l'aide d'expressions françaises spécifiques et illustrés par des images représentatives. Leurs correspondances avec les trois échelles phénologiques (échelle individuelle BBCH, EL et Baggiolini) ont été indiquées. Cette échelle, qui peut être vu comme une sélection des stades BBCH, porte ici le nom de IFV-Labels.

L'échelle IFV-Epicure complète celle d'EL et propose d'objectiver, par des éléments de mesure (pourcentage ou métrique), certaines définitions de stades trop imprécises : Le développement de la base de données Epicure au début des années 2000, a conduit l'IFV de Bordeaux à standardiser l'observation de stades phénologiques recueillis sur le terrain par les viticulteurs et techniciens. L'échelle IFV-Epicure reprend donc la numérotation proposée par EL, car c'est la plus utilisée en pratique sur le vignoble. Elle propose une numérotation continue du stade repos d'hiver (1) à la fin de la chute des feuilles (47) en occupant tous les numéros de stades laissés vacants par EL. Ce faisant, IFV-Epicure propose un échelonnage plus régulier et affine l'observation de certains stades trop imprécis par adjonction de critères mesurables : par exemple, le stade EL 31 (ou K de Baggiolini) défini par "grains de pois" reste le stade 31 d'IFV-Epicure mais est complété par une évaluation du diamètre des baies (4 à 5 mm).

Même s'il existe d'autres échelles pour d'autres cultures, comme Zadoks (1974), il n'y a manifestement pas de liste de stades phénologiques universellement reconnue ; cependant, en raison de sa large adoption, l'échelle BBCH peut être vue comme un langage pivot pour décrire les stades de développement spécifiques aux cultures. En d'autres termes, les jeux de données RDF décrivant les stades spécifiques d'une culture devront être alignés avec l'échelle générale BBCH (soit via une instanciation, soit via des correspondances explicites entre les échelles).

3 Ressources sémantiques décrivant les stades phénologiques

Aucune des échelles mentionnées précédemment n'est publiée sur le Web de données liées par leurs producteurs. Cependant, certains vocabulaires RDF ou ontologies OWL existants mentionnent des stades de développement. La recherche du mot clé «BBCH» sur AgroPortal nous retourne 3 ressources sémantiques : la Crop Ontology (CO), la Plant Ontology (PO) et SOY (une ontologie décrivant les traits spécifiques du soja). Nous avons également identifié d'autres ressources sémantiques pouvant contenir des description de stades phénologiques, notamment le thésaurus AGROVOC et la Plant Phenology Ontology (PPO) [25].

3.1 AGROVOC

Le thésaurus AGROVOC est publié par l'Organisation des Nations Unies pour l'Alimentation et l'Agriculture (FAO) [23]. Il est édité par une communauté mondiale d'experts et couvre tous les domaines d'intérêt de la FAO, y compris l'agriculture, la sylviculture, la pêche, l'alimentation et les domaines connexes. Il est disponible en 29 langues, avec une moyenne de 35 000 termes par langue et développé en SKOS-XL. La force de ce thésaurus est sa couverture lexicale multilingue. Il est donc souvent utilisé pour annoter ou indexer des documents ou des images relatifs au domaine de l'agriculture.

AGROVOC contient certains stades phénologiques. L'instance de `skos:Concept` intitulé «stades de développement végétal» est la racine d'une hiérarchie de 38 stades. Par exemple, l'URI `http://aims.fao.org/aos/agrovoc/c_2992` identifie le stade «floraison». Notons que la référence à une échelle existante n'est pas indiquée – c'est-à-dire qu'aucun texte associé à l'instance ne fait référence à une échelle en particulier comme BBCH et qu'aucun lien externe ne pointe vers une information de ce type. Idem pour la culture, aucune information ne permet de dire à quelle culture le stade peut s'appliquer. Pour ces raisons, AGROVOC ne correspond pas à notre objectif. Il pourra en revanche être une source externe à laquelle s'aligner pour plus d'interoperabilité.

3.2 La Plant Ontology et la Crop Ontology

La *Plant Ontology* (PO) [8] et la *Crop Ontology* (CO) [24], pour les plantes cultivés, sont deux résultats du projet Planteome [12]. L'objectif était de développer des ontologies de référence en génomique et en phénomique végétale. CO est le regroupement de plusieurs dictionnaires de traits observables spécifiques à une culture, tous liés à PO ainsi qu'à la *Plant Trait Ontology* (TO). L'un des dictionnaires de traits spécifiques à la vigne est l'ontologie *Vitis*,[5] mais elle ne contient aucune mention de stade phénologique. PO couvre cependant toutes les espèces cultivées ; elle contient des termes normalisés, des définitions et des relations décrivant l'anatomie, la morphologie et les stades [26]. Dans PO, un stade phénologique est représentée par une classe. Un stade correspond à un intervalle de

5. `http://agroportal.lirmm.fr/ontologies/CO_356`

temps pendant le cycle de croissance d'un élément végétal. Ainsi, un stade bref est défini comme une sous-classe d'un stade plus long englobant ce stade bref. Un stade est lié à une entité anatomique végétale à l'aide de la propriété objet `has_participant`. Certaines références à des échelles phénologiques ont été trouvées dans les valeurs des propriétés d'annotation `has_related_synonym` de PO : 49 pour l'échelle BBCH, 38 pour l'échelle Zadoks. Par exemple, l'URI `http://purl.obolibrary.org/obo/PO_0007086` correspond au stade "cinq nœuds visibles". Le code de ce stade dans l'échelle BBCH est BBCH 35. Comme la plupart des ontologies de la communauté OBO, PO a pour base l'ontologie "fondatrice" *Basic Formal Ontology* (BFO) [2] et réutilise des propriétés définies dans *Relation Ontology* (RO). Nous avons trouvé dans PO des propriétés (object properties) intéressantes. Les propriétés $precedes$ et $preceded\_by$ indiquent que la fin d'un stade se produit avant le début d'un autre. Ces propriétés définissent des restrictions de domaine et de co-domaine liées aux classes de BFO. Ainsi, réutiliser ces propriétés signifie s'appuyer sur les classes de BFO. PO a été conçu pour annoter les données génomiques et phénomiques. Cette ontologie est plus détaillée que l'échelle BBCH qui elle a été conçue pour les pratiques agricoles. Par conséquent, nous avons jugé PO non pertinente pour notre cas d'usage.

3.3 Plant Phenology Ontology (PPO)

La *Plant Phenology Ontology* (PPO) pour la phénologie végétale fournit un vocabulaire standardisé utile pour l'intégration de données phénologiques hétérogènes du monde entier [25]. Elle s'appuie sur PO ainsi que sur *Phenotype and Trait Ontology* (PATO) pour décrire les phénotypes et les traits. La phénologie n'est pas l'observation des stades phénologiques à proprement parler ; mais l'observation des traits physiques des plantes, comme la présence de feuilles ou de fruits. Ainsi, PPO définit les traits observables et ne se concentre pas sur l'observation de début ou de la fin d'un stade. PPO permet de décrire précisément la présence ou l'absence de certaines entités anatomiques végétales modélisées comme des traits.

Pour conclure, parmi les ressources sémantiques identifiées, nous n'avons trouvé aucune solution à adopter directement pour notre cas d'usage. Bien que BBCH soit souvent référencé, ce qui confirme son rôle de langage pivot, nous n'avons pas trouvé de ressource sémantique qui s'appuierait sur BBCH et offrirait un moyen d'encoder sémantiquement les stades phénologique de toutes les cultures. Par conséquent, nous avons décidé de proposer un nouveau cadre méthodologique pour décrire les échelles phénologiques : ce cadre contiendra une ontologie OWL déclarant un ensemble de classes spécialisant le modèle SKOS ainsi que plusieurs jeux de données SKOS. Un jeu de données SKOS déclare l'ensemble des stades d'une échelle spécifique à une culture – dans le cas de la vigne nous aurons donc encodé et aligné les échelles suivantes : échelle générique et individuelle BBCH, les échelles EL et Baggiolini, les échelles produites par IFV intitulée IFV-labels et IFV-Epicure.

4 Un cadre méthodologique pour décrire les stades phénologiques

Nous avons suivi la méthode *Linked Open Terms*, une méthode d'ingénierie ontologique inspirée du développement logiciel agile [20]. Cette méthode se concentre sur la réutilisation d'éléments (classes, propriétés objet et de type de données) existants dans des ontologies précédemment publiées sur le Web de données liées. La méthode définit des itérations sur les quatre activités suivantes : (1) spécification d'exigences ontologiques, (2) implémentation d'ontologie, (3) publication d'ontologie et (4) maintenance d'ontologie.

4.1 Specification

Nos exigences ontologiques ont été spécifiées par plusieurs questions de compétence. Ces questions expriment des besoins relatifs aux échelles phénologiques de la vigne mais elles peuvent être étendues à toute échelle d'une autre culture.

- Quels sont les labels français / anglais préférés et alternatifs pour un stade donné ?
- Quelle est la définition en français / anglais d'un stade donné ?
- Quels sont les stades principaux et secondaires d'une échelle donnée de la vigne (échelle individuelle BBCH, EL, Baggiolini, IFV-labels, IFV-Epicure) ?
- Quel stade principal de l'échelle individuelle BBCH est lié à un stade secondaire S de l'échelle R (EL, Baggiolini, IFV-labels, IFV-Epicure) ?
- Quel stade secondaire de l'échelle individuelle BBCH est équivalent à un stade secondaire S de l'échelle R (EL, Baggiolini, IFV-labels, IFV-Epicure) ?
- Quel stade secondaire de l'échelle individuelle BBCH est aligné avec un stade secondaire S de l'échelle R (EL, Baggiolini, IFV-labels, IFV-Epicure) ?
- Comment sont ordonnés (suivant / précédent) les stades principaux et secondaires d'une échelle donnée ?

Les alignements entre les échelles ont été spécifiés à l'aide d'un schéma organisant les stades par échelles phénologiques et par ordre chronologique. Ce schéma a d'abord été enrichi par un agronome : son but était de représenter les alignements trouvés dans la littérature. Par exemple dans la Figure 1 les lignes bleues claires représentent les alignements proposés par Coombe [7]. Les lignes rouges représentent les alignements proposés par Lorenz [14]. Les lignes vertes représentent les alignements proposés par Bloesh et Viret en 2008. Ensuite les deux agronomes ont proposé des alignements entre les 6 échelles. L'ontologue intervenait lors de leur discussion (1) pour clarifier le type d'alignements (exact, proche, etc...) entre deux stades, (2) pour proposer de nouveaux alignements déduits à partir des alignements précédents. Dans la figure 1, les traits noirs pleins représentent les alignements de type exact. Les traits noirs hachurés représentent les alignements de type proche.

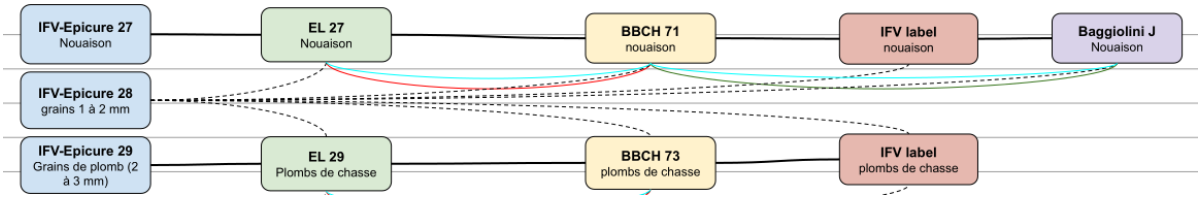

FIGURE 1 – Extrait du schéma de spécification des alignements

4.2 Implémentation

Nous avons choisi de spécialiser le modèle SKOS [1] pour représenter les stades phénologiques ; SKOS est une recommandation du W3C. Ce modèle répond à la plupart de nos questions de compétence relatives à l'usage des labels et des alignements. L'ensemble des stades phénologiques sera une sous-classe de `skos:Concept`. Ainsi, un stade sera décrit avec des labels multilingues préférés et alternatifs. Un stade appartiendra à une échelle phénologique qui sera représentée comme une instance de la classe `skos:ConceptScheme`. Un stade secondaire sera lié à son stade principal par la propriété `skos:broader`. L'ontologie sera stockée dans un fichier OWL. L'ensemble des stades d'une échelle phénologique sera représenté par un ensemble d'instances de `skos:Concept` typées par les classes de l'ontologie.

4.2.1 Modèle de l'échelle générale BBCH

Chaque stade de l'échelle générale BBCH est représenté à la fois par une classe et par une instance ; Une classe regroupe l'ensemble des stades similaires des échelles BBCH relative à une culture (échelle individuelle BBCH) plus le stade de l'échelle BBCH générale correspondant. Ainsi chaque stade d'une échelle BBCH sera donc une instance d'une de ses classes. Ce découpage en classe devrait permettre de retrouver facilement l'instance représentant le stade d'une culture donnée. Par exemple dans la figure 2, la classe intitulée `ppdo:SecondaryStage01` représente l'ensemble des stades secondaires BBCH 01 de toutes les échelles BBCH (générale ou individuelle). L'instance de cette classe intitulée `ppdo:bbch_secondaryStage_BBCH01` représente le stade de l'échelle BBCH générale. Cet individu est relié à l'instance de `skos:ConceptScheme` intitulée `ppdo:bbch_genericScale` par la propriété `skos:inScheme` Comme le montre la figure 2, nous avons créé une classe `ppdo:GrowthStage`, sous-classe de `skos:Concept`. Cette classe est ensuite spécialisée en deux sous-classes, une classe pour les stades principaux et une autre pour les stades secondaires. Ces deux classes sont des classes définies pour expliciter le fait que chaque instance doit être liée à une instance de la classe `ppdo:StageDivision` par la propriété objet `ppdo:hasRank`. Ainsi toute instance de la classe `ppdo:GrowthStage` liée à l'individu `ppdo:principal`, sera automatiquement classée comme instance de la classe `ppdo:PrincipalGrowthStage` et toute instance de la classe `ppdo:GrowthStage` liée à l'individu

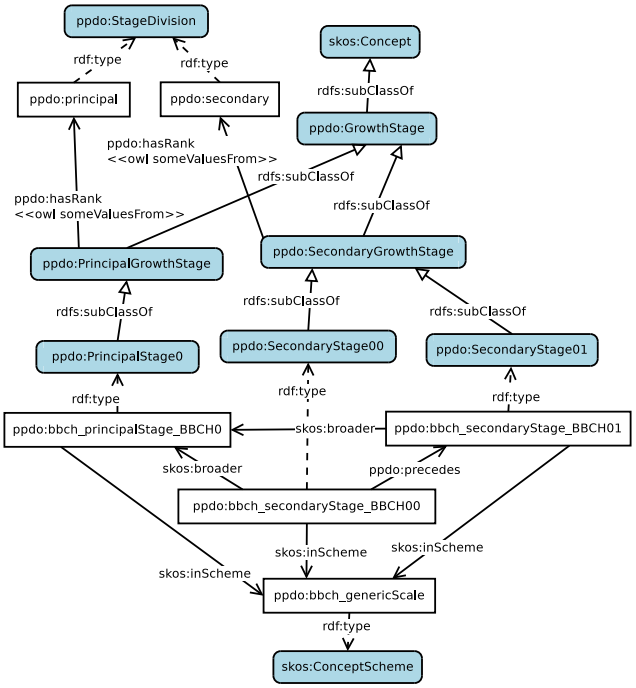

FIGURE 2 – Extrait de l'échelle générale BBCH – le prefixe `ppdo` est : `http://ontology.inrae.fr/ppdo/ontology/`.

`ppdo:secondary` sera automatiquement classée comme instance de la classe `ppdo:SecondaryGrowthStage`. La figure 2 présente des exemples de classes et d'instances représentant des stades principaux et secondaires de l'échelle générale BBCH.

4.2.2 Modèle de l'échelle individuelle BBCH

Un stade de la vigne est représenté par une instance d'une des sous-classes de `ppdo:GrowthStage`. Par exemple, la Figure 3 présente une instance de la classe `ppdo:SecondaryStage01`.
Plusieurs propriétés SKOS sont utilisées pour décrire un stade :
Les labels sont déclarés à l'aide des propriétés d'annotation `skos:prefLabel` et `skos:altLabel`. Les labels préférés sont extraits de la documentation (par exemple, la monographie BBCH). Les labels alternatifs sont extraits de la définition textuelle lorsque cela est possible ;
Les définitions sont déclarées à l'aide de la propriété d'annotation `skos:definition`. La monographie BBCH fournit une définition par langue ;
La propriété objet `skos:inScheme` indique l'échelle à la-

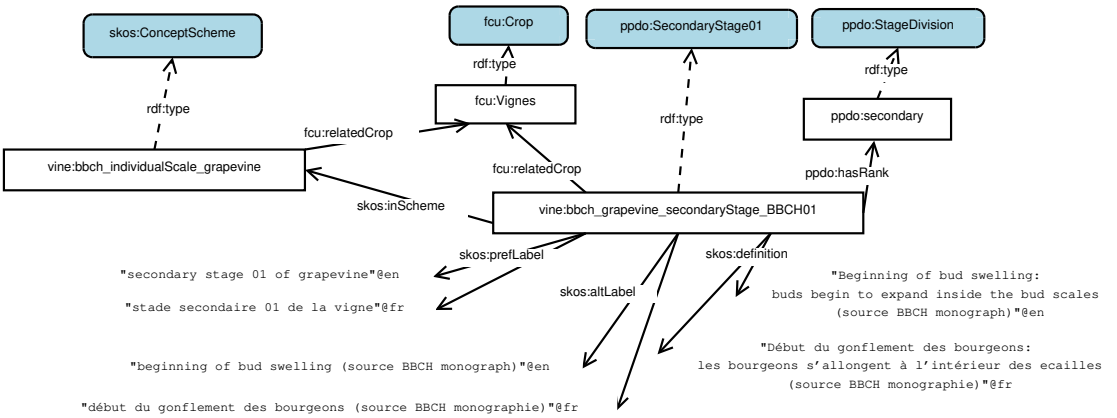

FIGURE 3 – Représentation du stade BBCH 01 de l'échelle individuelle BBCH –le préfixe `vine` signifie `http://ontology.inrae.fr/ppdo/resource/grapevine/`.

quelle appartient le stade. Les échelles individuelles BBCH ou toutes autres échelles sont représentées comme des instances de la classe `skos:ConceptScheme`.

La culture associée à un stade –ou à une échelle-– est décrite à l'aide de la propriété objet `fcu:relatedCrop` qui provient du thésaurus *FrenchCropUsage*,[6] ce thésaurus a été créé pour annoter les BSV [21] ;

Les liens entre les stades principaux et secondaires sont décrits à l'aide des propriétés objet `skos:broader/narrower`. La figure 4 illustre l'utilisation de cette propriété entre le stade principal BBCH 0 et le stade secondaire BBCH 00 de l'échelle individuelle de la vigne ;

L'ordre entre les stades, tel que défini dans l'échelle, est représenté avec les propriétés objet `ppdo:precedes` et `ppdo:follows`. La figure 4 présente un exemple de ce lien entre le stade secondaire BBCH 00 et le stade secondaire BBCH 01 de l'échelle individuelle de la vigne.

4.2.3 Modèle des échelles produites par IFV

Nous modélisons les deux ensembles de stades produits par IFV comme deux nouvelles échelles avec la même méthodologie que celle utilisée pour représenter l'échelle individuelle BBCH de la vigne. De plus, le point d'intérêt est l'alignement des échelles entre elles. Pour ce faire, nous avons utilisé les propriétés d'alignements fournies par SKOS :

Etant donné que les stades des échelles de l'IFV sont uniquement des stades secondaires, chaque stade des échelles de l'IFV est aligné à un stade principal de l'échelle individuelle BBCH de la vigne à l'aide de la propriété objet `skos:broadMatch` ;

De même, chaque stade des échelles de l'IFV est aligné sur un stade secondaire de l'échelle individuelle BBCH à l'aide des propriétés `skos:exactMatch` ou `skos:closeMatch` ou `skos:broadMatch` ou `skos:narrowMatch`.

La figure 4 présente un exemple d'alignement entre un stade de l'échelle IFV-labels et deux stades de l'échelle individuelle BBCH. La même approche est utilisée pour aligner les autres échelles (EL, Baggiolini).

4.2.4 Encodage RDF

Une fois les modèles conçus, ils ont été encodés sous forme de classes et d'instances. Nous avons utilisé l'éditeur d'ontologie Protégé (v5.1.0) [19] avec le plugin Cellfie.[7] Nous avons d'abord créé le modèle général comme une `owl:Ontology` définissant les classes (Figure 2) à partir de l'échelle générale BBCH. Nous avons également ajouté des descriptions de métadonnées (par exemple, auteurs, dates, licences) comme recommandé par [10] ou [9]. Deuxièmement, nous avons peuplé l'ontologie avec des instances représentant tous les stades de l'échelle individuelle BBCH de la vigne, de l'échelle Baggiolini, de l'échelle EL et des deux échelles de IFV pour produire une base de connaissances.

Nous avons utilisé différentes sources pour peupler l'ontologie. Les monographies BBCH ont fourni les informations (labels et descriptions) en quatre langues (français, anglais, espagnol et allemand) pour l'échelle générale et l'échelle individuelle de la vigne. La publication scientifique de l'échelle de Baggiolini et de l'échelle Eichhorn-Lorenz a fourni les informations en anglais. Enfin, l'IFV nous a fourni deux fichiers CSV regroupant les informations en français et en anglais. Ensuite, nous avons extrait les données de manière semi-automatique dans différentes feuilles de calcul avant de les charger dans la base de connaissances à l'aide des règles de transformation Cellfie. Cellfie a également été utilisé pour générer les instances des classes ; les règles de transformation ont créé les liens des propriétés objet : `ppdo:precedes`, `ppdo:follow`, `skos:broader`, `skos:broadMatch`,..., `skos:exactMatch`, `rdf:type`. Pour améliorer la cohérence de la base de connaissance finale, nous avons utilisé des règles SWRL pour déduire les propriétés inverses. Une

6. `http://agroportal.lirmm.fr/ontologies/CROPUSAGE_FR` (prefixe `fcu` dans les figures)

7. Cellfie (`https://github.com/protegeproject/cellfie-plugin`) permet de transformer le contenu des feuilles de calcul en axiomes pour peupler une ontologie.

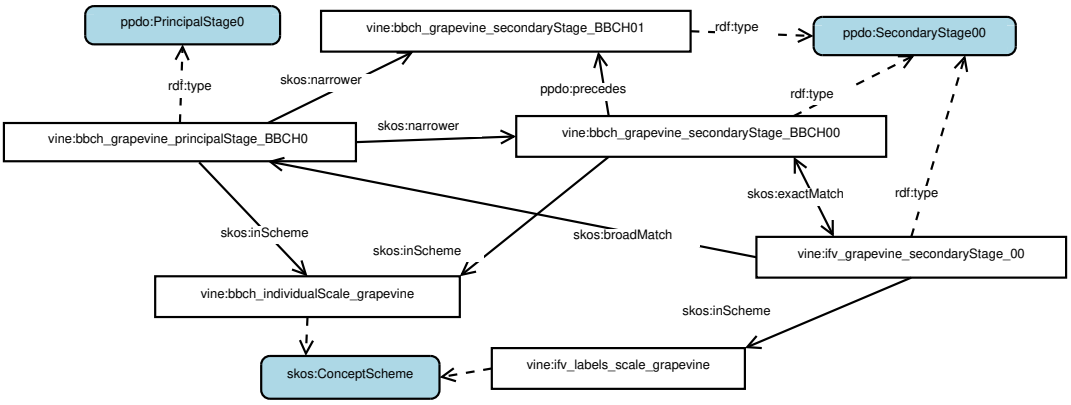

FIGURE 4 – Alignement entre le stade BBCH 01 de la vigne et le stade correspondant dans l'échelle IFV-Labels

dernière vérification a été effectuée à l'aide de l'outil SKOS Play ! [8] : il a permis de visualiser et de contrôler le modèle SKOS et de détecter quelques erreurs.

Le fichier OWL (en syntaxe Turtle) a été utilisé pour peupler un SPARQL endpoint. La conversion a été effectuée à l'aide des outils de la librairie *raptor* [9]. Le serveur SPARQL est jena-fuseki, de la fondation Apache [10].

Nous parlons d'un cadre méthodologique pour décrire le résultat obtenu ; c'est à dire l'ontologie *BBCH based Plant Phenological Development Ontology (PPDO)* ainsi qu'un ensemble de base de connaissances (ou jeux de données) fondées sur cette ontologie, une par culture. [11] La version actuelle de l'ontologie PPDO est étiquetée v1.0. L'ontologie importe SKOS et PROV. Elle se compose de 129 classes et 124 propriétés. L'échelle générale de BBCH est représentée par une centaine d'individus. Les 5 échelles de la vigne sont représentées par 125 individus au total et 1208 assertions de propriété objet et 1696 assertions de propriété d'annotation.

4.3 Publication et maintenance de l'ontologie

L'ontologie PPDO est accessible au public sous différentes formes. Tout d'abord les fichiers OWL et les fichiers qui ont permis sa création sont disponibles dans un dépôt git `https://gitlab.irstea.fr/copain/phenologicalstages`. L'ontologie est également chargée dans un triple store et peut être interrogé sur : `http://ontology.inrae.fr/ppdo/sparql/`. L'ontologie est disponible sur AgroPortal [13] à l'adresse `http://agroportal.lirmm.fr/ontologies/PPDO`. [12] Le suivi des problèmes fourni par Gitlab est l'outil utilisé pour recevoir des commentaires et contrôler les modifications. Dans un proche avenir, nous prévoyons d'ajouter de nouvelles bases de connaissances dédiées à de nouvelles cultures dans le SPARQL endpoint et le dépôt git.

5 Discussion et Conclusion

Cette transformation (RDFisation) des échelles phénologiques de la vigne est un travail collaboratif entre des experts de l'IFV et des ontologues. Le résultat améliorera l'usage et l'interopérabilité des échelles phénologiques actuellement existantes. Par exemple, les labels des échelles de l'IFV ont été améliorés car les labels français et les définitions ont été traduits en anglais grâce à la monographie BBCH. De plus, certaines incohérences et erreurs ont été détectées. Par exemple, l'échelle IFV-Labels mentionnait une référence au stade BBCH 88 de vigne qui n'existe pas dans la monographie BBCH. Au cours de la phase de spécification des alignements les experts de IFV ont décidé d'enrichir l'échelle IFV-Epicure en ajoutant de nouveaux stades. Les alignements entre les stades des différentes échelles ont été clarifiés par l'emploi des propriétés SKOS. Les 6 échelles ont été alignées entre elles alors qu'il n'existait jusqu'à présent que des alignements partiels entre les échelles prises deux à deux.

Nous avons peuplé l'ontologie PPDO avec toutes les échelles de la vigne préconisées par l'IFV. À l'avenir, nous prévoyons de publier des échelles relatives au blé (par exemple, Zadoks) en utilisant le même cadre méthodologique. Nous allons aussi publier les échelles individuelles BBCH de plusieurs légumes.

Le mélange de SKOS et d'OWL permet de regrouper les instances similaires dans une classe pour faciliter leur recherche ultérieure. L'ensemble des stades d'une échelle phénologique donnée est représenté dans un modèle SKOS. En effet, la notion de Scheme au sens de SKOS permet d'identifier une échelle phénologique et de lui associer des métadonnées spécifiques (auteurs, dates, documents sources, etc.). Le mélange des hiérarchies `owl:subClassOf` et `skos:broader/skos:narrower` permet de représenter respectivement des groupes de stades principaux et secondaires et des liens hiérarchiques entre les stades d'une même échelle.

8. `http://labs.sparna.fr/skos-play`
9. `http://librdf.org/raptor/`
10. `https://jena.apache.org/documentation/fuseki2/`
11. Produire une seule base de connaissances avec les instances de toutes les échelles de toutes les cultures n'aurait pas de sens pratique, les usages se faisant culture par culture.
12. A terme, AgroPortal stockera également les bases de connaissances par variété sous forme de vues ; mais à ce jour le portail ne permet pas de visualiser les instances ni de les regrouper dans différents `skos:ConceptScheme`.

La base de connaissance produite pour la vigne est actuellement disponible via un SPARQL endpoint à l'INRAE. À la fin du projet D2KAB, nous prévoyons de publier sur le Web de données liées notre archive BSV et les annotations associées. Ensuite, afin d'améliorer la découverte et l'interopérabilité de l'ontologie PPDO, nous alignerons formellement ses classes avec des concepts du thésaurus Agrovoc représentant ces stades (pour le moment nous avons déclaré simplement des liens `rdf:seeAlso`). Les alignements pourraient être complétés en alignant PPDO avec PPO. Le type d'alignement devra être étudié avec soin car les points de vue de ces deux ontologies divergent : PPO et PO représentent des processus de développement précis de chaque organe végétal. En revanche, les stades des échelles phénologiques ne sont qu'un instantané de certaines phases typiques de développement des plantes. Ces stades sont utiles pour planifier les pratiques agricoles.

Dans le cadre de l'évolution vers l'agriculture numérique, une autre perspective pour ce travail serait de rendre plus interopérables les données produites, par exemple en viticulture, par l'arrivée prochaine sur le marché de piquets connectés capables d'observations fines et répétées transmises en temps réel. Des prises de vue quotidiennes permettront assez prochainement de caractériser très finement la croissance et le développement des plantes voisines de ces postes d'observations par des algorithmes de reconnaissances de forme à partir d'images appropriées.

Remerciements

Ce travail a été réalisé avec le soutien du projet "Des Données aux Connaissances en Agronomie et Biodiversité (D2KAB– www.d2kab.org) financé par l'Agence Nationale de la Recherche (ANR-18-CE23-0017) ; ainsi qu'avec l'aide de l'Institut Français de la Vigne et du vin `www.vignevin.com`. Nous remercions également Thibaut Scholasch (Fruition Sciences) pour son aide à la traduction des labels.

Références

[1] Miles Alistair and Bechhofer Sean. SKOS Simple Knowledge Organization System. W3C Recommendation, W3C, August 2009.

[2] Robert Arp, Barry Smith, and Andrew D Spear. *Building ontologies with basic formal ontology*. MIT Press, 2015.

[3] Sophie Aubin, Pierre Bisquert, Patrice Buche, Juliette Dibie, Liliana Ibanescu, Clement Jonquet, and Catherine Roussey. Recent progresses in data and knowledge integration for decision support in agri-food chains. In *30èmes Journées Francophones d'Ingénierie des Connaissances, IC'19*, pages 43–59, Toulouse, France, July 2019.

[4] M Baggiolini. Les stades repères dans le développement annuel de la vigne et leur utilisation pratique. *Revue Romande d'Agriculture, de Viticulture et d'Arboriculture*, 1 :4–6, 1952.

[5] M Baillod and M Baggiolini. Les stades repères de la vigne. *Rev. Suisse Vitic. Arboric. Hortic*, 25(1) :7–9, 1993.

[6] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *Semantic Web and Information Systems*, 5(3) :1–22, 2009.

[7] Bryan G Coombe. Growth stages of the grapevine : adoption of a system for identifying grapevine growth stages. *Australian journal of grape and wine research*, 1(2) :104–110, 1995.

[8] Laurel Cooper, Ramona L. Walls, Justin Elser, Maria A. Gandolfo, Dennis W. Stevenson, Barry Smith, Justin Preece, Balaji Athreya, Christopher J. Mungall, Stefan Rensing, Manuel Hiss, Daniel Lang, Ralf Reski, Tanya Z. Berardini, Donghui Li, Eva Huala, Mary Schaeffer, Naama Menda, Elizabeth Arnaud, Rosemary Shrestha, Yukiko Yamazaki, and Pankaj Jaiswal. The Plant Ontology as a Tool for Comparative Plant Anatomy and Genomic Analyses. *Plant and Cell Physiology*, 54(2) :e1, Dec. 2012.

[9] Biswanath Dutta, Anne Toulet, Vincent Emonet, and Clement Jonquet. New Generation Metadata vocabulary for Ontology Description and Publication. In *11th Metadata and Semantics Research Conference, MTSR'17*, volume 755 of *CCIS*, pages 173–185, Tallinn, Estonia, Nov. 2017. Springer.

[10] Daniel Garijo and M. Poveda Villalon. A checklist for complete vocabulary metadata. Technical report, WIDOCO, April 2017.

[11] IFV. Les stades phénologique de la vigne, 2017.

[12] P. Jaiswal, L. Cooper, J. L. Elser, A. Meier, M-A. Laporte, C. Mungall, B. Smith, E. KS. Johnson, M. Seymour, J. Preece, X. Xu, R. S. Kitchen, B. Qu, E. Zhang, E. Arnaud, S. Carbon, S. Todorovic, and D. Wm. Stevenson. Planteome : A resource for Common Reference Ontologies and Applications for Plant Biology. In *24th Plant and Animal Genome Conference, PAG'16*, San Diego, USA, January 2016.

[13] Clement Jonquet, Anne Toulet, Elizabeth Arnaud, Sophie Aubin, Esther Dzalé Yeumo, Vincent Emonet, John Graybeal, Marie-Angélique Laporte, Mark A. Musen, Valeria Pesce, and Pierre Larmande. AgroPortal : a vocabulary and ontology repository for agronomy. *Computers and Electronics in Agriculture*, 144 :126–143, January 2018.

[14] D.H. Lorenz, K.W. Eichhorn, H. Bleiholder, R. Klose, U. Meier, and E. Weber. Growth Stages of the Grapevine : Phenological growth stages of the grapevine (Vitis vinifera L. ssp. vinifera)—Codes and descriptions according to the extended BBCH scale. *Australian Journal of Grape and Wine Research*, 1(2) :100–103, July 1995.

[15] Uwe Meier. *Entwicklungsstadien mono- und dikotyler Pflanzen : BBCH Monografie*. Open Agrar Repositorium, 2018.

[16] Uwe Meier. *Etapas de desarrollo de las plantas monocotiledóneas y dicotiledóneas : BBCH Monografia*. Open Agrar Repositorium, 2018.

[17] Uwe Meier. *Growth stages of mono- and dicotyledonous plants : BBCH Monograph*. Open Agrar Repositorium, 2018.

[18] Uwe Meier. *Stades phénologiques des mono-et dicotylédones cultivées : BBCH Monographie*. Open Agrar Repositorium, 2018.

[19] Mark A. Musen. The protégé project : a look back and a look forward. *AI Matters*, 1(4) :4–12, June 2015.

[20] María Poveda-Villalón. A reuse-based lightweight method for developing linked data ontologies and vocabularies. In *9th Extended Semantic Web Conference, ESWC'12*, volume 7295 of *LNCS*, pages 833–837. Springer, 2012.

[21] Catherine Roussey, Stephan Bernard, Francois Pinet, Xavier Reboud, Vincent Cellier, Ivan Sivadon, Danièle Simonneau, and A. Bourigault. A methodology for the publication of agricultural alert bulletins as LOD. *Computers and Electronics in Agriculture*, 142 :632–650, Nov. 2017.

[22] Catherine Roussey, Xavier Delpuech, Florence Amardeilh, Stephan Bernard, and Clement Jonquet. Semantic Description of Plant Phenological Development Stages, starting with Grapevine. In *14th International Conference on Metadata and Semantics Research (MTSR2020)*, Madrid, Spain, November 2020.

[23] Johannes Keizer Sachit Rajbhandari. The AGROVOC Concept Scheme ; A Walkthrough. *Integrative Agriculture*, 11(5) :694–699, May 2012.

[24] Rosemary Shrestha, Elizabeth Arnaud, Ramil Mauleon, Martin Senger, Guy F. Davenport, David Hancock, Norman Morrison, Richard Bruskiewich, and Graham McLaren. Multifunctional crop trait ontology for breeders' data : field book, annotation, data discovery and semantic enrichment of the literature. *AoB Plants*, 2010, May 2010.

[25] Brian J. Stucky, Rob Guralnick, John Deck, Ellen G. Denny, Kjell Bolmgren, and Ramona Walls. The Plant Phenology Ontology : A New Informatics Resource for Large-Scale Integration of Plant Phenology Data. *Frontiers in Plant Science*, page 517, May 2018.

[26] Ramona L. Walls, Laurel Cooper, Justin Elser, Maria Alejandra Gandolfo, Christopher J. Mungall, Barry Smith, Dennis W. Stevenson, and Pankaj Jaiswal. The Plant Ontology Facilitates Comparisons of Plant Development Stages Across Species. *Frontiers in Plant Science*, 10 :631, June 2019.

Formalisation du concept d'affordance dans l'ontologie Thing Description

V. Charpenay
Laboratoire d'informatique, de modélisation et d'optimisation des systèmes (LIMOS)

victor.charpenay@emse.fr

Résumé

Cet article présent l'ontologie Thing Description (TD), une ontologie pour décrire objets connectés et systèmes cyber-physiques sur le web. L'ontologie TD formalise le concept d'« affordance » comme la relation qui existe entre une requête HTTP envoyée par un agent, la réponse qui sera donnée par le serveur et les effets de cet échange agent/serveur sur le monde physique.
Les axiomes de l'ontologie TD, qui font intervenir des termes des ontologies Semantic Sensor Networks et Smart Applications Reference, sont évalués vis-à-vis de la « commandabilité » de l'objet ou du système décrit. Une formalisation logique de cette notion issue de la théorie du contrôle est proposée dans l'article.

Mots-clés

Web des objets, Thing Description, commandabilité, système cyber-physique, planification automatique.

Abstract

This paper presents the Thing Description (TD) ontology, an ontology to describe connected devices and cyber-physical systems on the Web. The TD ontology formalizes the concept of 'affordance' as the relationship between an HTTP request sent by an agent, the response provided by the server and the effects of the agent/server communication in the physical world.
The axioms of the TD ontology, involving terms from the Semantic Sensor Networks and Smart Applications Reference ontologies, are evaluated against the 'controllability' of the device or system being described. The paper introduces a logical formalization of this notion inherited from control theory.

Keywords

Web of Things, Thing Description, Controllability, Cyber-Physical System, AI Planning.

1 Introduction

Le web des objets a pour principe fondamental d'exposer des objets connectés à travers une interface web, dite « REST » (pour *Representational State Transfer*) [11]. Cette approche apporte principalement des avantages en termes d'ingénierie logicielle mais à l'heure actuelle, il n'existe pas de formalisme théorique établi pour REST. Notamment, l'approche REST invoque la notion d'« affordance », qui n'a pourtant pas de fondement théorique dans le contexte du web. Ce néologisme, qui émane du champ de la psychologie et du design [10], désigne la capacité d'un objet à suggérer son usage. Sur le web, les objets en questions sont des documents, qu'un client manipule à travers des hyperliens (pour y accéder) et des formulaires web (pour modifier leur état). Hyperliens et formulaires web seraient donc les affordances des documents qui suggèrent au client comment manipuler un document en particulier [5].
Cette notion d'affordance est prépondérante pour permettre à des clients de contrôler les objets exposés sur le web de manière autonome. Les affordances offertes par hyperliens et formulaires web doivent être non seulement comprises des humains mais aussi d'agents logiciels, qui agissent en navigant d'une ressource à l'autre et en remplissant des formulaires.
Dans cet article, nous proposons un cadre formel pour les affordances exposées sur le web par des objets connectés. Ce cadre formel se présente comme une ontologie, dont les termes sont déjà utilisés dans le standard *Thing Description* (TD) du *World Wide Web Consortium* (W3C) [8]. L'objectif principal de l'ontologie TD est de permettre à des agents autonomes d'appliquer les techniques d'intelligence artificielle (IA) classiques, telles que du raisonnement sur des situations ou de la planification, à des descriptions d'objets connectés et, plus largement, à des systèmes cyber-physiques.
Comme tout ontologie du web, l'ontologie TD cherche à réutiliser les ontologies existantes, notamment celles standardisées par le W3C ou d'autres instituts de normalisation. L'ontologie TD est donc définie par rapport aux ontologies *Semantic Sensor Network* (SSN), *OWL Time*, *HTTP in RDF* et l'ontologie de provenance PROV-O, toutes du W3C, ainsi que *Smart Applications Reference* (SAREF), un standard de l'institut européen des normes de télécommunication. Voir table 1 pour les *namespaces* associés à chacune de ces ontologies.
Cet article prolonge un article de 2020 qui présentait l'ontologie TD vis-à-vis de questions de compétences précises, posées par le groupe de travail W3C autour du web des objets [4]. Alors que l'article de 2020 basait son évaluation (entre autres) sur une tâche unique de « sélection d'affordance », le travail suivant présente une généralisation des tâches possibles que des agents autonomes peuvent effec-

TABLE 1 – Ontologies référencées dans l'article

Ontologie	*Namespace*
SSN	`http://www.w3.org/ns/ssn/`
OWL Time	`http://www.w3.org/2006/time#`
HTTP in RDF	`ttp://www.w3.org/2011/http#`
PROV-O	`http://www.w3.org/ns/prov#`
SAREF	`https://saref.etsi.org/core/`
TD	`https://www.w3.org/2019/wot/td#`

tuer sur des systèmes cyber-physiques. Cette généralisation est basée sur la « commandabilité » des systèmes, une notion issue de la théorie du contrôle.
La notion de commandabilité, ainsi que d'autres préliminaires sont définis dans la suite de l'article (partie 2), qui donne ensuite les axiomes de l'ontologie, après généralisation vis-à-vis l'article de 2020 (partie 3) et un exemple complet d'instantiation de l'ontologie (partie 4). L'article se termine sur une synthèse de l'approche (partie 5).

2 Préliminaires

2.1 Actions et temporalité

Les techniques d'IA classiques sont toutes basées sur une représentation formelle du temps et/ou des actions effectuées par des agents autonomes. De nombreuses théories du temps ont été proposées, comme la logique temporelle linéaire (LTL) [9], beaucoup utilisée pour de la vérification formelle, ou l'algèbre d'intervalles d'Allen [2]. Ces deux approches utilisent des modèles différents ; la première représente le temps comme un ensemble discret de points, la seconde comme un ensemble d'intervalles. Il existe des équivalentes, cependant, selon les extensions syntaxiques considérées [6].
Dans le contexte du web sémantique, l'outil logique de référence est OWL, le langage des ontologies web. OWL est en revanche très rarement utilisé pour faire du raisonnement temporel ou de la planification. La raison, bien qu'il n'en existe pas de preuve formelle à notre connaissance, tiendrait au fait que le formalisme sous-jacent à OWL (la logique de description $\mathcal{SROIQ}$) ne permet d'encoder ni les logiques temporelles par points comme LTL, ni l'algèbre d'Allen [1].
Les travaux de recherches se rapprochant le plus de OWL pour représenter temporalité et actions ont été effectués par Artale et Franconi [3] et sont basés sur $\mathcal{HS}$ [7], une logique plus expressive mais indécidable (pour le problème de satisfaction). Artale et Franconi reprennent l'algèbre d'Allen comme modèle temporel de base. Les termes de cette algèbre sont définis dans l'ontologie du W3C OWL Time [2].
Dans la suite de l'article, nous utiliserons OWL Time comme modèle temporel de base pour les axiomes de l'ontologie TD. Par souci de lisibilité, les axiomes seront exprimés dans un forme plus classique, en tant que formules

1. des opérateurs non-standards de logique de description, incompatibles avec $\mathcal{SROIQ}$ d'un point de de la décidabilité du raisonnement, permettent cependant de le faire : un opérateur de point fixe pour LTL, et la disjonction de roles pour l'algèbre d'Allen.

2. mais la sémantique de l'algèbre, elle, n'est pas complètement axiomatisée dans l'ontologie.

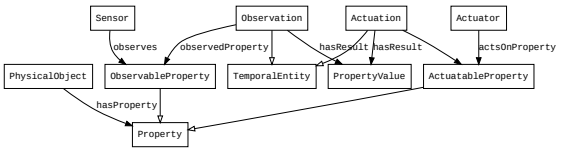

FIGURE 1 – Vocabulaire et axiomes RDFS (hiérarchies de classes, domaines/images de propriétés) d'une ontologie de base pour le web des objets

de logique de premier ordre dans la syntaxe *Common Language Interchange Format* (CLIF) [1]. L'équivalence avec la logique proposée par Artale et Franconi ne sera pas traité ici.
L'algèbre d'Allen (et donc OWL Time) définit treize relations possibles entre intervalles de temps. Si `t1` et `t2` sont des intervalles de temps, alors on peut par exemple définir `(before t1 t2)`, `(meets t1 t2)`, `(overlaps t1 t2)` ou `(equals t1 t2)`. La relation `equals`, ici, n'est pas une égalité stricte. `t1` et `t2` coïncident dans le temps mais ces deux intervalles peuvent représenter des événements distincts.
Les observations faites par des capteurs ou les activations effectuées par des actionneurs sont typiquement les intervalles sur lesquels raisonner dans le web des objets. Ces termes sont présentés plus en détail dans la partie suivante.

2.2 *Semantic Sensor Networks*

L'ontologie TD se base sur un ensemble de termes préexistants, pour la plupart tirés du standard W3C *Semantic Sensor Network* (SSN). Les termes de cette ontologie de base sont synthétisée en un diagramme (figure 1), dont le langage graphique utilise les notions courantes du web sémantique.
Les classes (prédicats unaires) apparaissent comme des rectangles et les relations [3] (prédicats binaires) apparaissent à travers des connexions entre classes. Une connexion non-labelisée implique une hiérarchie de classe. C'est par exemple le cas entre `Observation` et `TemporalEntity` (la classe de tous les intervalles de temps). Une connexion labélisée implique, elle, des axiomes de domaine(s) et d'image(s). La relation `observedProperty` est par exemple définie entre les classes `Observation` et `ObservableProperty`.
Ces types d'axiomes, à la base de RDF Schema (RDFS), permettent principalement de définir des structures de données en graphe. À travers cet article, nous ajoutons d'autres axiomes aux termes de SSN dans le cadre du web des objets. Ces axiomes sont exprimés en tant que formules en logique de premier ordre, comme discuté précédemment. Une distinction sera faite entre les formules ouvertes (à voir comme des motifs syntaxiques récurrents), contenant des variables libres non-quantifiées, et les formules fermées (les

3. plus généralement appelées « propriétés » dans la terminologie RDF. Le terme est cependant ambigü ici car il désigne aussi les propriétés d'objets physiques.

axiomes eux-mêmes).
Par exemple, l'ontologie SSN définit les classes `Observation` et `Actuation`, qui associent une propriété physique à une valeur observée ou fixée par un agent. Les formules suivantes matérialisent ces associations, ce sont des formules ouvertes :

(OBS)

```
(and
  (Observation o)
  (observedProperty o p)
  (hasResult o v))
```

(ACT)

```
(and
  (Actuation a)
  (actsOnProperty a p)
  (hasResult a v))
```

Les variables libres apparaissant dans ces formules ouvertes sont ensuite quantifiées dans les deux axiomes suivants (par souci de concision, les classes `TemporalEntity`, `Property` et `PropertyValue`) ont été raccourci en `T`, `P` et `V`) :

```
(forall ((T o) (P p) (V v) (V v'))
  (if OBS_{o,p,v}
    (not (exists (T o')
      (and
        OBS_{o',p,v'}
        (finishes o' o) (not (= v v')))))))

(forall ((T a) (T o) (P p) (V v) (V v'))
  (if
    (and
      ACT_{a,p,v}
      OBS_{o,p,v'}
      (or (before a o) (meets a o))
      (not (exists (T a')
        (and
          ACT_{a',p,v''}
          (or (before a a') (meets a a'))
          (or (before a' o) (meets a' o))
          (not (= v v'')))))))
    (= v v')))
```

Les axiomes ci-dessus expriment respectivement que deux observations faites pour une même propriété au même instant doivent avoir le même résultat et que toute observation doit avoir le même résultat que la dernière activation la précédent.
L'axiome suivant donne la contrainte qu'une propriété doit être associée à une valeur en tout instant :

```
(forall (T o) (P p)
  (if OBS_{o,p,v}
    (exists (T o') (V v')
      (and (meets o o') OBS_{o',p,v'}))))
```

Ensemble, ces axiomes permettent de réduire les systèmes décrits avec SSN à un ensemble de lignes de temps (une pour chaque propriété physique du système) sur lesquelles chaque point, correspondant à un instant donné, est la valeur de propriété mesurable par un capteur à cet instant. Ces axiomes ne font pas partie des ontologies dont nous empruntons le vocabulaire, ils serviront cependant à valider formellement l'intérêt de l'ontologie TD pour diverses techniques d'IA.
Dans l'article original de 2020 sur l'ontologie TD, plus de la moitié des objets décrit dans le jeu de données utilisée pour l'évaluation [4] servait à contrôler un système d'éclairage. L'exemple suivant en est inspiré.

Exemple 1. *Les formules suivantes décrivent un interrupteur on/off et un capteur d'illuminance, à trois niveaux de luminosité (obscur, moyen, clair).*

```
(and
  (PhysicalObject switch)
  (hasProperty switch state)
  (ObservableProperty state)
  (ActuatableProperty state)
  (forall (T t)
    (if (or OBS_{t,state,v} ACT_{t,state,v})
      (or
        (= v on)
        (= v off)))))

(and
  (Sensor lightSensor)
  (hasProperty lightSensor level)
  (ObservableProperty level)
  (not (ActuatableProperty level))
  (forall (T t)
    (if (or OBS_{t,level,v} ACT_{t,level,v})
      (or
        (= v dark)
        (= v average)
        (= v bright)))))
```

À partir de l'ontologie de base présentée ici, il serait possible de définir des lois physiques. Un exemple en sera donné par la suite. La caractérisation complète des lois qui régissent observations et activations vont cependant au-delà du périmètre de cet article. Le fondement de l'article est ailleurs, dans la commandabilité des systèmes cyber-physiques.

2.3 Commandabilité

La notion de commandabilité est une notion importante de la théorie du contrôle [12] car elle caractérise simplement en termes matriciels l'efficacité d'une procédure de contrôle dans des systèmes linéaires. Nous la redéfinissons ici dans un cadre logique comme la satisfiabilité d'une formule de précédence entre un état initial donné et un état final souhaité (état cible).

Définition 1. *Soit S une formule spécifiant le comportement d'un système à base d'objets physiques à partir d'un intervalle de temps `now`. Soit ϕ_t une formule temporelle avec la variable libre `t` telle que `(T t)`. ϕ_t est un « état cible » pour lequel est définie la formule suivante :*

4. https://w3c.github.io/wot-thing-description/testing/report.html

(K)

```
(forall (T t')
  (if (before now t')
    (exists (T t)
      (and ϕ_t (before t' t)))))
```

Le système spécifié par S est contrôlable si et seulement si la formule $S \wedge K$ est satisfiable, c'est-à-dire qu'il existe un modèle de premier ordre M tel que :

$$M \models S \wedge K$$

Cette caractérisation logique de la notion de commandabilité peut intuitivement se formuler ainsi : un système est contrôlable s'il est possible d'atteindre l'état cible depuis l'état initial. L'état cible n'est cependant pas atteint en toute circonstances (les propriétés activables ont en effet plusieurs activations possibles). C'est le rôle d'agents autonomes d'effectuer les actions nécessaires pour atteindre cet état.
Par ailleurs, la formule K nécessite que l'intervalle `t` soit atteignable depuis n'importe quel état depuis l'état initial (cf `forall (T t')`). Cette condition permet de considérer le cas ou des changements spontanés s'opèrent sur les objets une fois l'état cible atteint, du fait de lois physiques. Cette notion généralisée de commandabilité est aussi appelée « stabilisabilité » dans la littérature sur la théorie du contrôle.
Les états cibles qui nous intéressent dans le cadre du web des objets sont ceux définis avec l'ontologie de base présentés dans la partie précédente, par exemple à travers l'observation d'une valeur particulière. En reprenant l'exemple 1, on peut par exemple définir comme état cible une observation où l'éclairage est au plus haut : $\text{OBS}_{t,\texttt{level},\texttt{bright}}$.

3 Axiomes

La finalité de l'ontologie TD est de décrire l'interface web permettant à un agent logiciel autonome d'interagir avec des objets physiques, c'est-à-dire d'en observer les propriétés et les contrôler. Dans le web des objets, l'action des agents sur les objets prend la forme d'opérations effectuées sur des ressources web à travers des formulaires.
Dans le formalisme présenté dans cet article, tous les atomes apparaissant dans les formules (par exemple, `switch`, `state` ou `off`) sont considérés comme des ressources web, dont une représentation concrète peut être déréférencée par les agents. L'ontologie TD définit des axiomes permettant aux agents de connaître les implications sur un objet physique d'une opération web.
Le diagramme donnant le vocabulaire et les axiomes RDFS de l'ontologie TD (figure 3) inclut des termes importés d'autres ontologies (*HTTP in RDF* et PROV-O, pour la relation `wasGeneratedBy`). Il est à noter que la classe `Thing`, elle, appartient au *namespace* de l'ontologie TD. Elle est distincte de son homonyme définie par OWL, la classe mère du langage.
Les termes importés permettent de représenter par des formules les messages échangés entre agents et serveurs web. Ces messages sont à la base des opérations web.

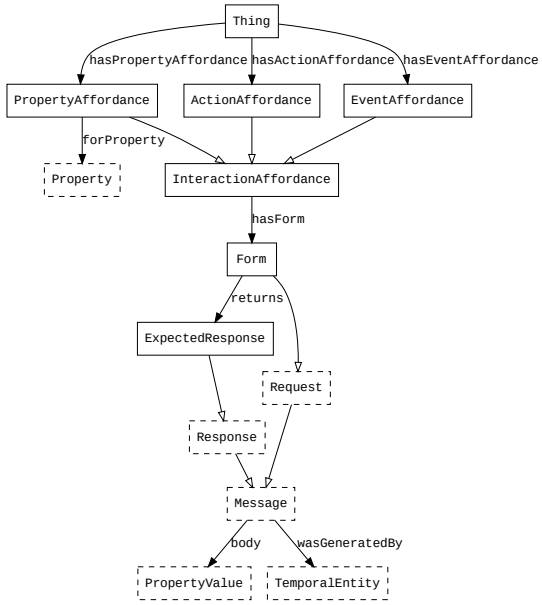

FIGURE 2 – Vocabulaire et axiomes RDFS de l'ontologie TD (vocabulaire extérieur en pointillé)

3.1 Opérations et affordances

Une opération est l'intervalle inscrit entre l'envoi d'une requête et celui d'une réponse. Dans le cadre du web des objets, les opérations sont rarement matérialisées, l'ontologie TD n'inclut donc pas de vocabulaire spécifique pour les opérations. On peut cependant caractériser une opération en n'utilisant que le vocabulaire d'*OWL Time*, comme ci-dessous.

(OP)

```
(and
  (Request req) (Response resp)
  (meets t1 op) (isMetBy t2 op)
  (wasGeneratedBy req t1)
  (wasGeneratedBy resp t2)
  (not (exists (T t2')
    (and
      (before t2' t2)
      (wasGeneratedBy resp t2')))))
```

Une représentation schématique des intervalles `t1`, `op` et `t2` est donnée en figure 3. Dans la formule définissant `op`, la non-existence d'un intervalle `t2'` garantit l'unicité de `op` : seule la première réponse succédant à la requête est prise en compte.
À partir de la notion d'opération, on peut maintenant définir formellement ce qu'est une affordance.

(AF)

```
(and
  (InteractionAffordance alpha)
  (hasForm alpha req)
  (returns req resp))
```

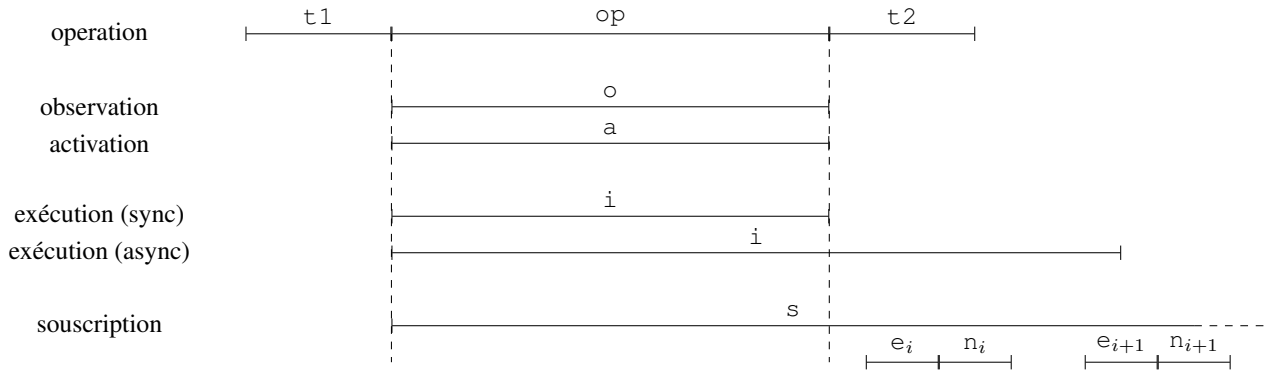

FIGURE 3 – Representation schématique des axiomes liés aux affordances sur des opérations REST

Dans la structure ci-dessus, l'affordance est associée à une requête (un formulaire web, *form* en anglais) et une réponse particulière. La notion de formulaire web est ici à prendre au sens large. Il ne s'agit pas que d'un simple formulaire HTML mais d'un patron de requête qu'un agent web peut envoyer à un serveur.
Dans sa version la plus simple, une `InteractionAffordance` spécifie quelle réponse un agent obtiendra en soumettant un formulaire.

Définition 2. *Une affordance est la relation qui existe entre une requête REST potentielle (un formulaire web) et la réponse attendue à cette requête :*

```
(forall (alpha req resp)
  (if AF_{alpha,req,resp}
    (forall (T t1)
      (if (wasGeneratedBy req t1)
        (exists ((T op) (T t2))
          OP_{op,t1,t2,req,resp})))))
```

Cependant, cette définition n'est pas suffisante pour permettre à des agents autonomes d'agir de manière informée. Ce type de connaissance peut déjà être formalisé à travers des descriptions de services web (qui décrivent les structures de données en entrée et en sortie d'un service). OpenAPI [5] est par exemple une spécification normalisée qui uniformise la description d'interfaces REST.
La notion d'affordance provient de recherches en psychologie et en design. Dans sa définition originelle, proposée par James Gibsons, une affordance est rattachée à un objet, elle est fournie par lui à un agent comme action potentielle. Sur le web, on a comparé le concept d'hyperlien à une affordance, en particulier dans son rendu visuel (curseur engageant à cliquer). Sur le web des objets, les objets physiques fournissent des affordances aux agents à travers la classe d'`InteractionAffordance`.
Les sous-classes d'`InteractionAffordance` permettent d'axiomatiser plus en détail l'effet d'une opération sur le monde physique et, ainsi, d'inférer des plans d'actions sur la base de la commandabilité des objets décrits.

5. https://www.openapis.org/

Il est à noter que la classe `Thing`, qui regroupe toutes les entités physiques fournissant des affordances, est distincte de celle de `PhysicalObject`. Dans l'architecture du web des objets, l'objet qui fournit l'affordance n'est en effet pas nécessairement l'objet d'une action. Le type d'affordance présenté dans la partie suivante permet de faire la distinction.

3.2 Observations et activation simples

La principale sous-classe d'`InteractionAffordance` définit des affordances associées à une propriété physique.

(PAF)

```
(and
  (PropertyAffordance alpha)
  (forProperty alpha p))
```

Définition 3. *Une affordance de propriété est l'affordance de lire ou écrire la valeur d'une propriété physique :*

```
(forall (alpha req resp p)
  (iff
    (and
      AF_{alpha,req,resp} PAF_{alpha,p}
      (ObservableProperty p))
    (forall ((T op) (T t1) (T t2))
      (if OP_{op,t1,t2,req,resp}
        (exists (T o)
          (and
            OBS_{o,p,v}
            (equals op o)
            (body resp v)))))))

(forall (alpha req resp p)
  (iff
    (and
      AF_{alpha,req,resp} PAF_{alpha,p}
      (ActuatableProperty p))
    (forall ((T op) (T t1) (T t2) (V v))
      (if (and OP_{op,t1,t2,req,resp} (body req v))
        (exists (T a)
          (and
            ACT_{a,p,v}
            (equals op a)))))))
```

Comme mentionné précédemment, on distingue `Thing` et `PhysicalObject` et, de la même manière, on distingue `PropertyAffordance` et `Property`. Par exemple, si l'appareil `lightSensor` fournit une affordance pour la propriété `level`, le niveau de luminosité observé n'est pas la propriété du capteur mais celle de la pièce dans laquelle il se trouve. Un autre capteur situé dans la même pièce pourrait fournir une affordance distincte pour la même propriété. Par ailleurs, la pièce elle-même pourrait fournir une affordance sur sa propre luminosité, masquant ainsi l'infrastructure technique d'acquisition de données.
Les questions de compétences de notre article de 2020 élicitaient le besoin formulé par le W3C d'identifier différentes combinaisons possibles entre ces classes. Outre le cas où deux affordances s'appliquent à la même propriété, il est aussi possible qu'une même affordance s'applique à plusieurs propriété (par exemple lorsqu'une station météo fournit une mesure de température et d'humidité dans le même message).

3.3 Invocations et souscriptions

Les affordances pour observer ou agir sur les propriétés physiques sont les plus utilisées en pratique (84% des 371 affordances incluses dans notre jeu de données de 2020). Cependant, elles ne suffisent pas complètement à permettre à tout système d'être contrôlable à travers le web. D'autres formes plus complexes d'affordances sont nécessaires : `ActionAffordance` et `EventAffordance`.
Intuitivement, lorsque une propriété change de manière continue et rapide, le délai introduit par le fait d'envoyer des requêtes REST (discrètes) pour l'observer ne permet pas toujours à des agents de réagir à temps. Une requête peut cependant exiger de l'objet d'effectuer lui-même une action (s'il s'agit d'un système cyber-physique) ou de notifier l'agent lors d'événements particuliers. C'est l'intérêt des deux sous-classes d'affordances présentées dans cette partie.
Une affordance d'action est associée à une action, qui accepte des paramètres en entrée et qui retourne une valeur en sortie.

(AAF)

```
(and
  (ActionAffordance alpha)
  (hasInputSchema alpha schi)
  (hasOutputSchema alpha scho))
```

Définition 4. *Une affordance d'action est l'affordance d'invoquer l'exécution d'une action :*

```
(forall (alpha req resp schi scho)
  (iff
    (and AF_{alpha,req,resp} AAF_{alpha,schi,scho})
    (forall ((T op) (T t1) (T t2))
      (if (and OP_{op,t1,t2,req,resp} (body req schi))
        (exists (T i)
          (and
            (forInvocation alpha i)
            (or
              (and
                (equals op i)
                (body resp scho))
              (and
                (starts op i)
                (body resp i)))))))))
```

Une `ActionAffordance` peut modéliser l'un des deux motifs récurrents dans les interfaces REST, selon le modèle d'éxecution de l'action invoquée : synchrone ou asynchrone. Dans le premier cas, l'invocation dure le temps de l'opération. Elle commence à la réception de la requête et se termine à l'envoi de la réponse. Dans le second cas, l'invocation peut se prolonger au-delà de l'opération. L'agent peut différencier ces deux motifs grâce à la nature de la réponse obtenue (une valeur de sortie `scho` ou l'invocation `i` elle-même, en tant qu'entité temporelle). En pratique, lorsque l'invocation est asynchrone, déréfencer l'entité temporelle retournée permet de connaître le statut de l'invocation (« en cours », « terminée », « annulée », ou autres).
Il est important ici de distinguer deux types d'actions en jeu dans des affordances. Les actions invoquées par un agent sont effectuées par l'objet ou système sous contrôle. Cependant, les invocations elles-mêmes sont des actions (de délégation) qui sont, elles, effectuées par l'agent. Dans un problème de planification, par exemple, les actions à planifier sont celles de l'agent, donc toutes les opérations qui lui sont offertes à travers des affordances, et pas uniquement les actions invocables à travers une affordance d'action.
Une affordance d'événement joue un rôle analogue à l'affordance d'action. Si l'affordance d'action permet de paramétrer des activations, l'affordance d'événement permet de paramétrer les observation faites par les agents.

(EAF)

```
(and
  (EventAffordance alpha)
  (hasSubscriptionSchema alpha schs)
  (hasNotificationSchema alpha schn))
```

Définition 5. *Une affordance d'événement est l'affordance de souscrire à un événement afin de recevoir une notification à chaque occurrence de l'événement :*

```
(forall (alpha req resp schs schn)
  (iff
    (and AF_{alpha,req,resp} AAF_{alpha,schs,schn})
    (forall ((T op) (T t1) (T t2))
      (if (and OP_{op,t1,t2,req,resp} (body req schs))
        (exists (T s)
          (and
            (forSubscription alpha s)
            (starts op s)
            (body resp s)))))))
```

On peut noter que l'axiome ci-dessus n'utilise pas `schn` (le schéma de notification). En effet, pour pouvoir axiomatiser qu'une notification est envoyée à chaque événement, il est d'abord nécessaire de caractériser l'événement, en

termes d'observations et/ou d'activations. Même sans caractériser l'action ou de l'événement en jeu, pourtant, l'intérêt d'axiomatiser des affordances d'action ou d'événement est d'établir l'existence d'entités temporelles (l'invocation `i` et la souscription `s`, respectivement) qui peuvent ensuite servir à formuler des contraintes plus spécifiques sur des sous-classes d'affordance. Afin de relier une affordance à `i` ou `s`, on utilise les relations `forInvocation` et `forSubscription`. Nous en donnerons un exemple dans la partie suivante. Les axiomes complets pour ces relations ne sont cependant pas trancrites ici.
Dans le cas d'une affordance d'événement, toute caractérisation de l'événement devrait utiliser la formule suivante pour décrire une notification (avec la variable libre `e`) :

(N)

```
(exists ((T n) m)
  (and
    (meets e n)
    (Message m)
    (wasGeneratedBy m n)
    (body m schn)))
```

L'ontologie TD telle que publiée par le W3C inclut aussi un schéma pour annuler une souscription. Cette partie n'est pas détaillée ici.

3.4 Boucle de commande

Le théorème suivant garantit que l'ajout des classes `ActionAffordance` et `EventAffordance` à `PropertyAffordance` suffit à décrire tout type de système sur le web (à la condition que le système décrit est commandable, pour un état cible défini à travers l'ontologie de base du web des objets).

Théorème 1. *On suppose que chaque activation est précédée d'une observation (boucle de commande) et que, par ailleurs, il ne peut y avoir qu'une seule opération à un instant donné (éxecution linéaire).*
Dans ces conditions, les trois types d'affordances de propriété, d'action et d'événement sont suffisants pour représenter tout système contrôlable sur le web.

Démonstration. (esquisse) Considérons le modèle M qui, par hypothèse, satisfait $S \wedge K$.
On suppose l'existence d'une activation `a1`, nécessaire à la commandabilité. On construit M' tel qu'une nouvelle opération `op1` englobe l'activation (`(contains op1 a1)`). On définit ensuite une affordance `alpha`, telle que `(PropertyAffordance alpha)`, qui garantisse l'existance d'`op1`.
S'il existe une autre activavtion `a2` telle que `(meets a1 a2)`, on définit `alpha` telle que `(ActionAffordance alpha)`, de telle sorte que l'action invoquée englobe `a1` et `a2`.
On fait de même avec les observations qui précèdent `a1` et `a2`. Si une observation `o1` n'est pas collée à `a1`, on définit une affordance de propriété. Sinon (`(meets o1 a1)`), on définit une affordance d'événement. □

Les trois types d'affordances ne sont en revanche pas nécessaires à garantir la commandabilité du système décrit. Certains systèmes définissent uniquement des affordances d'action pour modifier la valeur d'une propriété activable. La possibilité d'écrire une propriété à travers une `PropertyAffordance` est donc redondant avec la définition d'`ActionAffordance`. C'est pour permettre de modéliser des systèmes particuliers, comme les systèmes de gestion de bâtiments BACnet, que ce choix de la redondance a été fait par le W3C. Les objets BACnet ne définissent que des affordances de propriété ; une modélisation par affordances d'actions serait fastidieux.
Si les axiomes liés aux affordances servent de base à un raisonnement sur la temporalité des actions et des événements, décrire des objets et systèmes physiques avec l'ontologie TD se fait avec les seules formules PAF, AAF, et EAF.

Exemple 2. *Les formules suivantes donne la description TD de l'interrupteur et du capteur d'illuminance de l'exemple 1.*

```
(and
  (Thing switch)
  (hasPropertyAffordance switch paf1)
  (forProperty paf1 state)
  (hasActionAffordance switch aaf))


(and
  (Thing lightSensor)
  (hasPropertyAffordance switch paf2)
  (forProperty paf2 level)
  (hasEventAffordance lightSensor eaf))
```

4 Instantiation

L'objectif de l'ontologie TD est de simplifier la description d'objets ou systèmes physiques tout en garantissant l'application de techniques d'IA pour des agents autonomes sur le web. Les techniques particulières à appliquer ne rentrent cependant pas dans le cadre de l'article, uniquement axé sur la représentation de la connaissance liée aux objets connectés.
Comme mentionné dans la présentation de `ActionAffordance` et `EventAffordance`, une axiomatisation complète sans restriction sur les actions ou événements associés n'est pas possible. Les exemples présentés dans la partie suivante précisent comment une affordance d'action ou d'événement peut inclure les définitions nécessaires à du raisonnement ou de la planification.
Pour ce faire, on définit de nouveaux axiomes à partir des classes définies dans l'ontologie SAREF. SAREF inclut notamment les sous-classes de `Property` suivantes : `Light` et `OnOffState`. Avec ces deux classes, on peut par exemple définir une loi physique simplifiée qui stipule que le niveau de luminosité d'une pièce dépend de la position de l'interrupteur dans la pièce.

```
(forall
  ((P st) (P l) (T o) (T o') (V v) (V v'))
```

```
(if
  (and
    (OnOffState st) (Light l)
    OBS_{o,st,v} OBS_{o',l,v'}
    (contains o o'))
  (and
    (if (= v off) (= v' dark))
    (if (= v on) (= v' average)))))
```

L'exemple ci-dessus est volontairement simplifié. Le niveau de luminosité réel d'une pièce dépend par exemple du nombre d'éclairages dans la pièce et de leur puissance individuelle.
L'ontologie SAREF définit aussi des classes de « commandes » qui permettent de rendre plus spécifique une activation sur une propriété. L'axiome suivant assigne par exemple l'une des deux classes `OnCommand` et `OffCommand` a une activation selon la valeur associée à l'activation. Par ailleurs, l'interrupteur étant un appareil passif, l'axiome définit que l'état observé de l'interrupteur résulte toujours d'une activation antérieure.

```
(forall ((P st) (T o) (V v))
  (if (and (OnOffState st) OBS_{o,st,v})
    (exists (T a)
      (and
        ACT_{a,st,v}
        (meets a o)
        (if (= v on) (OnCommand a))
        (if (= v off) (OffCommand a))))))
```

Dans notre article de 2020, on supposait qu'une commande s'applique à l'état de l'appareil qui exécute la commande. Ici, on introduit un lien explicite via `forProperty` et l'on caractérise plus précisément l'action exécutée par l'appareil (à travers la relation `forInvocation`) : il s'agit d'une activation simple. Les classes `OnCommand` et `OffCommand` sont des sous-classes de `OnOffCommand`.

```
(forall (alpha schi scho st i)
  (if
    (and
      AAF_{alpha,schi,scho} (OnOffState st)
      (forProperty alpha st)
      (forInvocation alpha i))
    (and
      (OnOffCommand i)
      ACT_{a,st,schi}))
```

Pour finir, SAREF permet aussi de décrire des événements simples, à travers la classe `EventFunction`, qui est le domaine de la relation `hasThresholdMeasurement`. De la même manière que l'on a spécifié la nature de l'action invoquée par un interrupteur, on peut spécifier le type d'événement associé à une souscription sur un capteur de luminosité, en donnant une mesure seuil.

```
(forall (alpha schs schn f o p v)
  (if
    (and
      EAF_{alpha,schs,schn}
      (hasThresholdMeasurement alpha st)
      OBS_{o,p,v}
      (forProperty alpha p)
      (forSubscription alpha s))
    (forall (T o')
      (if (and (contains s o') OBS_{o',p,v})
        N_{o',schn}))))
```

La mesure seuil est ici considérée comme une observation simple qui, lorsqu'elle se produit, déclenche une notification. L'observateur est ici le capteur. L'agent n'est qu'un observateur « par délégation » de la propriété physique.

Exemple 3. *Les formules suivantes reprennent les descriptions de l'exemple 2 et y ajoutent des « annotations » SAREF*

```
(and
  (LightSwitch switch)
  (OnOffState state)
  (forProperty aaf state))

(and
  (hasFunction lightSensor f)
  (EventFunction f)
  (hasThresholdMeasurement f t)
  (observedProperty t level)
  (hasResult t dark)
  (Light level)
  (forProperty eaf level))
```

Du fait de l'utilisation d'annotations SAREF dans les descriptions TD de l'interrupteur et du capteur d'illuminance, un agent est en mesure d'inférer les faits suivants :

- pour atteindre l'état cible $OBS_{t,level,dark}$, il est nécessaire qu'il y ait une activation `a` telle que $ACT_{a,state,off}$ (idem pour `average` et `on`);
- les affordances `paf1` et `aaf` peuvent être utilisées de manière équivalente pour activer l'interrupteur ;
- le système composé de `switch` et `levelSensor` n'est pas commandable pour l'état cible $OBS_{t,level,bright}$.

Dans le cas de l'affordance d'action caractérisée par une `OnOffCommand` comme dans le cas de l'affordance d'événement caractérisée par une `EventFunction`, l'action ou l'événement est en fait une activation ou observation simple. Les affordances sont donc redondantes avec les affordances de propriétés définies par les objets. Cependant, le cadre formel donné ici au travers de l'ontologie TD permet de caractériser des actions ou événements de complexité arbitraire, à travers un langage générique fait de relations temporelles entre observations et activations.

5 Conclusion

L'ontologie TD formalise la notion d'affordance sur le web, ce qui, à notre connaissance, n'avait encore jamais été proposé. La conception des axiomes de cette ontologie a été motivée par la vision selon laquelle des agents autonomes peuvent naviguer sur le web (sémantique) et agir de manière informée sur les ressources auxquels il ont accès. Si l'ontologie TD est une première étape vers le développement de tels agents, d'autres barrières sont à surmonter pour réaliser cette vision.

En particulier, du fait d'une représentation en logique du premier ordre des axiomes, il reste à étudier la décidabilité et la complexité de différentes tâches de raisonnement basées sur l'ontologie TD, comme par exemple la satisfaction de la spécification d'un système à partir de ses affordances afin d'en établir la commandabilité. Dans le cas où le problème général est indécidable, des pistes de validation formelles par modèles finis pourraient être explorées.

Références

[1] Information technology — common logic (cl) — a framework for a family of logic-based languages, 2018.

[2] James F. Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5) :531–579, 1994.

[3] A. Artale and E. Franconi. A temporal description logic for reasoning about actions and plans. 9 :463–506, 1998.

[4] Victor Charpenay and Sebastian Käbisch. On modeling the physical world as a collection of things : The w3c thing description ontology. In *The Semantic Web*, volume 12123, pages 599–615. Springer International Publishing, 2020.

[5] Roy Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. phdthesis, 2000.

[6] Michael Fisher, Dov M. Gabbay, and L. Vila. *Handbook of temporal reasoning in artificial intelligence*. Number v. 1 in Foundations of artificial intelligence. Elsevier, 1st ed edition, 2005.

[7] Joseph Y. Halpern and Yoav Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4) :935–962, 1991.

[8] Sebastian Kaebisch, Takuki Kamiya, Michael McCool, Victor Charpenay, and Matthias Kovatsch. Web of things (WoT) thing description, 2020.

[9] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems : Specification*. Springer New York, 1992.

[10] Cesare Pautasso, Erik Wilde, and Rosa Alarcon. *REST : Advanced Research Topics and Practical Applications*. Springer New York, 2014.

[11] Erik Wilde. Putting things to REST, 2007.

[12] Jerzy Zabczyk. *Mathematical Control Theory*. Birkhäuser Boston, 2008.

Raisonnement embarqué et distribué pour le Web des Objets : un état de l'art

Alexandre Bento[1], Lionel Médini[1], Kamal Singh[2], Frédérique Laforest[1]

[1] Université de Lyon, INSA Lyon, UCBL, CNRS, LIRIS UMR 5205, Villeurbanne, France
[2]Université de Lyon, UJM, CNRS, LaHC UMR 5516, Saint-Etienne, France

{alexandre.bento,lionel.medini,frederique.laforest}@liris.cnrs.fr, kamal.singh@univ-st-etienne.fr

Résumé

Le projet Constrained Semantic Web of Things (CoSWoT) a pour objectif de définir une plateforme pour le développement d'applications distribuées et intelligentes pour le Web des Objets utilisant les technologies du Web sémantique. Les graphes de connaissances et le raisonnement à base de règles constituent les éléments clés du projet. Cet article propose un état de l'art des travaux intéressants pour le raisonnement embarqué et distribué dans le cadre du Web des Objets, et trace des lignes directrices pour la mise en place d'un tel raisonnement dans le projet CoSWoT.

Mots-clés

Web des objets, raisonnement, architecture distribuée, optimisation

Abstract

The Constrained Semantic Web of Things (CoSWoT) project aims to define a platform for the developement of smart and distributed applications for the Web of Things that uses the technologies of the semantic Web. Knowledge graphs and rule-based reasoning are the key elements of the project. This paper proposes a state of the art of interesting works on embedded and distributed reasoning in the frame of the Web of Things. It also gives directions to set up such reasoning in the CoSWoT project.

Keywords

Web of things, reasoning, distributed architecture, optimisation

1 Introduction

Le Web des Objets (WoT) [1] désigne les techniques permettant d'utiliser des objets de l'Internet des Objets (IoT) avec les technologies du Web. Différents organismes de standardisation (IETF [2], W3C [3], ETSI [4]) proposent des solutions pour l'interopérabilité sémantique dans l'IoT.
Les objectifs du projet CoSWoT [5] sont de proposer une architecture logicielle distribuée compatible avec le WoT et embarquée dans des dispositifs contraints en ressources (capacité de calcul, mémoire, énergie...). Cette architecture ajoutera des fonctionnalités de raisonnement à des dispositifs de capacités diverses, en répartissant les tâches de traitement entre les dispositifs.
L'analyse des cas d'usage du projet dans les domaines du batiment intelligent et de l'e-agriculture ont permis d'identifier que la plateforme doit fournir un raisonnement à base de règles sur des données dynamiques représentées sous la forme de multiples graphes de connaissances. Plus précisément, les besoins métier sont : (i) un raisonnement incrémental et distribué sur des règles métier incluant la détection de dépassements de seuils issus de l'agrégation temporelle et spatiale de données, (ii) l'appel à des fonctions externes (par exemple, des calculs arithmétiques) dans la tête des règles, (iii) le traitement de flux de données, et (iv) l'intégration du processus de raisonnement dans d'autres outils de la plateforme CoSWoT (stockage, traçabilité...).
Le raisonnement distribué a fait l'objet d'une attention particulière au cours des dernières décennies [1, 2] pour traiter de très grands ensembles de données RDF sur des machines de même architecture dans des clusters ou sur Internet [3, 4]. La distribution dans les architectures edge [6] ou fog [7] porte sur des configurations très différentes [5]. Tout d'abord, les dispositifs impliqués ont des capacités de calcul hétérogènes et des architectures matérielles variées. Deuxièmement, les dispositifs ont une connectivité plus ou moins limitée. Ces caractéristiques invalident les résultats de recherches antérieures, et soulèvent le défi de développer des stratégies de raisonnement efficaces en termes de distribution des données et des tâches de raisonnement dans un contexte d'hétérogénéité, et aussi près des sources de données que possible.
Dans cet article, nous présentons l'état de l'art de la recherche en raisonnement dans les domaines qui concernent les contraintes du projet CoSWoT : objets contraints, raisonnement local, et raisonnement distribué. Nous concluons ensuite avec quelques pistes de recherche.
Pour des raisons de place, nous partons du principe que le lecteur connait le domaine du Web sémantique [6] et les

1. https://www.w3.org/WoT/
2. https://www.ietf.org/
3. https://www.w3.org/
4. https://www.etsi.org/
5. https://coswot.gitlab.io/
6. https://fr.wikipedia.org/wiki/Edge_computing
7. https://fr.wikipedia.org/wiki/Fog_computing

principes du raisonnement à base de règles [7].

2 Raisonnement et objets contraints

2.1 Contraintes liées à l'IoT

Les objets contraints ou autonomes ont différentes caractéristiques qui rendent difficiles la communication et l'exécution de tâches complexes [8, 9]. Pour instancier un raisonneur, choisir les règles à y déployer et exécuter un algorithme de raisonnement sur de tels objets, il faut prendre en compte les différentes dimensions identifiées ci-dessous.

Architecture matérielle. Certains dispositifs disposent d'unités de traitement aux architectures non standards ou aux capacités de calcul limitées. [10] cite le cas de dispositifs de type microcontrôleurs au bus de données de 8 bits. De telles unités peuvent ralentir, voire empêcher le traitement de règles "à forte expressivité", c'est-à-dire plus susceptibles de produire des faits qui eux-mêmes déclencheront l'exécution de règles. Il existe donc un lien entre la complexité maximale atteignable par un algorithme de raisonnement et le type de processeur ou microcontrôleur sur lequel ce raisonnement peut être déployé. Cela peut se produire en particulier lorsque les données arrivent en flux et qu'il faut les traiter à la volée.
Par ailleurs, comme tous les algorithmes, la chaîne de déploiement doit être adaptée au portage du code source du raisonneur sur l'architecture matérielle de chaque dispositif cible. La diversité des dispositifs utilisés dans une architecture IoT complexifie cette chaîne. Enfin, il faut bien entendu que le code puisse être déployé sur le support de persistance du dispositif (ROM, EEPROM, Flash), c'est-à-dire que sa taille soit inférieure à celle de ce support [8].

Mémoire de travail. Certains dispositifs ne peuvent ni stocker ni travailler sur des graphes de connaissances volumineux. [8] propose également une catégorisation des objets en fonction de la taille de la mémoire de travail disponible [8]. Cette contrainte s'applique à la fois à la taille du graphe d'entrée (faits explicites) mais aussi au graphe déduit (faits explicites + implicites). Certains cas d'utilisation imposent de raisonner sur des flux de données correspondant par exemple à une fenêtre temporelle ; la taille de celle-ci peut aussi être limitée par la mémoire disponible.

Énergie. Le raisonnement sur des dispositifs autonomes en énergie peut être limité par deux aspects : la communication (en particulier sans fil), qui représente la principale source de consommation énergétique d'un dispositif, ainsi que l'intensivité des calculs, dont le coût est moindre et qui peut s'avérer rentable si elle permet de diminuer la charge utile des communications réseau [11]. Dans la perspective d'un algorithme de raisonnement distribué, il convient donc de minimiser les communications entre les dispositifs en évaluant le coût des différentes tâches de raisonnement et leur faisabilité en local.

Connectivité. De nombreux travaux ont été menés autour des problématiques de transmission de données entre nœuds d'un réseau dans l'IoT [8, 9, 10]. Un nœud à la connectivité limitée, notamment pour des raisons d'économie d'énergie, ne prendra probablement pas part à un protocole de raisonnement distribué pour traiter des données provenant d'autres sources. Mais pour les raisons exposées plus haut, il peut aussi avoir des difficultés à raisonner sur ses propres données. C'est pourquoi, pour certains capteurs isolés et connectés périodiquement, il faut savoir choisir entre délivrer des données brutes, des observations sémantisées ou des faits implicites issus de déductions à partir de ces observations.

2.2 Déploiement sur des objets contraints

Les technologies du Web sémantique, et en particulier les raisonneurs classiques (voir section 3) sont trop gourmands en ressources pour être portés directement sur des dispositifs contraints. Il n'existe par exemple que quelques travaux qui intègrent du raisonnement dans des dispositifs contraints, et plusieurs d'entre eux sont conçus pour les téléphones mobiles [12, 13, 14, 15] et non pour des dispositifs plus contraints comme des capteurs [16, 17] ou des microcontrôleurs. Dans cette partie, nous présentons les travaux de la littérature qui implémentent les différentes parties de la "stack" Web sémantique et raisonnement sur des objets contraints.

Échange de données RDF. Plusieurs implémentations proposent la gestion de données RDF dans des objets contraints. Wiselib [16] permet à un microcontrôleur de type iSense [9] d'héberger et de servir directement à ses clients des triplets RDF à l'aide des protocoles CoAP et 6LowPan. [18] propose un serveur RDF utilisant CoAP et JSON-LD pour interagir avec les capteurs et les actionneurs déployés sur un microcontrôleur encore plus contraint : Arduino Uno [10]. Plus généralement, [19] propose un format de sérialisation de données RDF en binaire qui s'appuie sur EXI4JSON et CBOR pour réduire considérablement la taille des faits échangés entre un raisonneur et un client ou entre plusieurs raisonneurs.

Outils de traitement pour RDF. Pour pouvoir déployer un moteur d'inférence sur un objet, il faut que les outils sous-jacents soient disponibles dans l'environnement d'exécution de l'objet. C'est pourquoi nous recensons les outils de stockage et traitement de graphes RDF en fonction des langages de programmation dans lesquels ils sont implémentés et de leur disponibilité sur les différentes plateformes matérielles.

Les outils de base des technologies du Linked Data et du Web sémantique ont des implémentations bien connues telles que Jena [11], librdf [12], N3.js [13] ou RDFlib [14]. En termes de performances, [20] a démontré que la bibliothèque RDF

8. Cette classification date de 2014, les seuils définis ne sont plus à jour.

9. https://www.quarbz.com/Wireless%20Sensor%20Network/2.%20iSense%20Devices%20and%20Modules.pdf
10. https://store.arduino.cc/arduino-uno-rev3
11. Implémenté en Java, http://jena.apache.org/
12. Implémenté en C, http://librdf.org/
13. Implémenté en JavaScript, https://github.com/rdfjs/N3.js
14. Implémenté en Python, https://rdflib.readthedocs.io

Sophia [15] développée en Rust [16], est plus rapide pour charger un graphe RDF en mémoire et répondre à des requêtes simples sur ce graphe. À l'heure de la rédaction de ce document, les chaînes de compilation et de déploiement de code Rust sur les plateformes matérielles d'objets contraints sont de plus en plus répandues, stables et disponibles pour une part de plus en plus importante de ces plateformes.

Dans le même esprit, WASMTree [21] est une implémentation en Rust et Web Assembly qui permet de construire des triplestores RDF en JavaScript. Elle permet de stocker et interroger des datasets RDF en utilisant une stratégie d'indexation intelligente, et surpasse les bibliothèques de référence. Cependant, elle est limitée au stockage et à la récupération de quads RDF au niveau syntaxique, et des travaux supplémentaires sont nécessaires pour travailler au niveau sémantique requis par le raisonnement à base de règles.

Raisonnement sur mobile et Web. Même si les smartphones ont des capacités de calcul beaucoup plus élevées que les appareils généralement considérés comme contraints, nous mentionnons ici le déploiement de tâches de raisonnement sur des appareils mobiles. [12] et [15] ont adapté des raisonneurs de référence sur des téléphones Android, et ont montré que ces machines n'avaient pas les capacités de calcul nécessaires pour exécuter correctement des tâches de raisonnement. Bien que les spécifications des appareils mobiles aient évolué depuis [17], ces travaux montrent que les raisonneurs existants ne sont pas adaptés aux dispositifs contraints. [13] présente un raisonneur développé sur iOS pour iPhone qui offre des performances décentes, mais la principale optimisation présentée provient de l'utilisation du langage de programmation Swift natif d'iOS, au lieu de langages standards. La généricité de cette approche est donc discutable. Dans le même ordre d'idée, le raisonneur HyLAR [22] décrit dans les sections suivantes peut s'exécuter dans un client Web, éventuellement mobile. Programmé en JavaScript pour être portable, il peut s'exécuter indifféremment côté serveur et côté client, et a été porté dans un Web worker [18] pour améliorer ses performances. Mais il nécessite une machine virtuelle JS pour s'exécuter, qui n'est en général pas disponible pour les objets contraints.

Raisonnement et architectures matérielles. Sans avoir été spécifiquement développé pour les objets contraints, Inferray [23] accélère également le raisonnement en l'optimisant en fonction de la taille des mots-mémoire de l'achitecture sur laquelle il est déployé. Il utilise une structure de données basée sur des tableaux pour l'inférence in-memory en utilisant des algorithmes de tri-jointure et de tri-comptage personnalisés. Il gère l'inférence des ensembles de règles RDFS, ρdf, et RDFS-Plus et a montré de très bonnes performances sur certains ensembles de données. Cependant, il ne supporte pas la suppression des faits.

15. https://github.com/pchampin/sophia_rs
16. https://www.rust-lang.org/
17. Nous n'avons pas trouvé de travaux plus récents sur cette question dans la litérature.
18. https://html.spec.whatwg.org/multipage/workers.html#workers

3 Raisonnement local

En termes de performances, l'algorithme de raisonnement naïf bouclant sur les règles est loin d'être optimal lorsque les règles sont interdépendantes. Plusieurs optimisations sont présentées ci-dessous.

3.1 Optimisations du raisonnement local

RETE [24] structure les règles sous la forme d'un arbre de préfixes nommé "trie" [19]. Le trie RETE est composé de deux types de nœuds [20] (Figure 1). Les nœuds alpha représentent les conditions atomiques dans le corps des règles. Les nœuds bêta effectuent des opérations de jointure entre les sorties de deux autres nœuds. Les faits compatibles activent les nœuds du trie et le résultat est conservé en mémoire. Ainsi, lorsque de nouveaux faits arrivent, il n'est pas nécessaire de réévaluer les faits précédents. Le raisonnement est finalisé quand les nœuds feuilles sont atteints, c'est-à-dire quand tous les nœuds alpha d'une règle ont été activés et toutes les opérations de jointure des nœuds bêta ont été effectuées. L'algorithme RETE n'explore qu'une partie de l'arbre de règles, au lieu de parcourir chaque règle pour chaque fait explicite ; cela permet d'augmenter la vitesse d'exécution, au prix de l'empreinte mémoire. Diverses optimisations ont été proposées pour l'algorithme RETE, dont certaines sont présentées ci-après.

$\text{RETE}_{\text{pool}}$ [14] réduit l'empreinte mémoire de RETE en limitant la duplication des données pendant le raisonnement. Il utilise une mémoire partagée pour tous les nœuds alpha du réseau. De cette façon, les doublons sont éliminés à l'insertion. Dans les cas où un triplestore RDF est utilisé avec le raisonneur, un autre niveau de duplication est éliminé, chaque nœud alpha ayant des références aux triplets contenus dans le triplestore plutôt qu'une copie locale. Cela permet d'économiser de la mémoire, au détriment de la vitesse d'exécution.

COROR [17] réduit la consommation mémoire de RETE. Il crée un graphe de dépendances entre les règles, puis charge en mémoire uniquement les règles nécessaires : pour une ontologie et un jeu de règles donnés, les règles ne pouvant pas aboutir à la création de nouveaux faits ne sont pas considérées. Ensuite, il décompose l'algorithme RETE en deux phases. La première phase effectue un appariement entre les faits et les conditions des nœuds alpha, comme le fait l'algorithme RETE classique. La seconde restructure l'arbre RETE en utilisant les statistiques de la première phase : les règles et les conditions sont réorganisées pour que les plus sélectives soient analysées en premier. COROR permet de réduire l'empreinte mémoire de 74 % en moyenne, par rapport à l'implémentation de référence.

3.2 Raisonnement incrémental

Au fil du temps, les données d'une application peuvent évoluer, modifiant le graphe de connaissances sur lequel s'effectue le raisonnement. Les faits implicites qui en découlent sont donc impactés par ces modifications. Parmi

19. https://en.wikipedia.org/wiki/Trie
20. https://www.sparklinglogic.com/rete-algorithm-demystified-part-2/

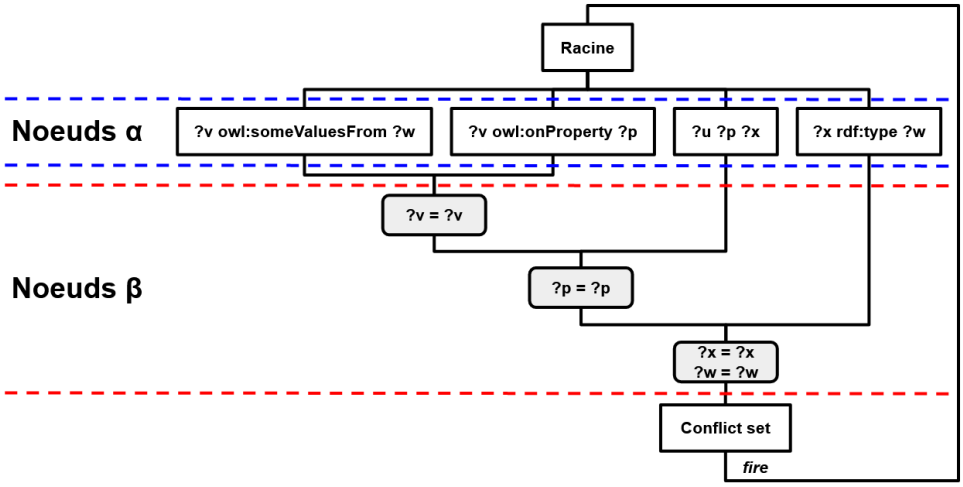

FIGURE 1 – Un exemple d'arbre RETE pour la règle
(?v owl:someValuesFrom ?w) ∧ (?v owl:onProperty ?p) ∧ (?u ?p ?x) ∧ (?x rdf:type ?w) → (?u rdf:type ?v).

les algorithmes de matérialisation présentés ci-avant, tous ne supportent pas l'insertion de faits, et aucun d'entre eux n'en supporte la suppression. Pour résoudre ce problème, des raisonneurs incrémentaux ont été définis ; ils permettent à la fois l'insertion et la suppression (aussi appelée maintenance incrémentale) des faits explicites.

Lorsqu'un fait explicite est supprimé, l'algorithme Delete / Rederive (DRed) [25] supprime d'abord tous les faits implicites qui en dépendent. Ensuite, il relance le raisonnement sur les faits non supprimés, ce qui permet de redériver certains faits supprimés. D'autres travaux utilisent une variante de l'algorithme DRed, comme [26].

Pour permettre le raisonnement incrémental, RDFox [27] utilise l'algorithme Backward/Forward : lorsqu'un fait explicite est supprimé, il cherche immédiatement des dérivations alternatives pour les faits qui en découlent. En comparaison avec DRed, le gain en performance est particulièrement visible avec les faits implicites qui découlent de nombreuses déductions en chaîne (par exemple, rdfs:subClassOf). GraphDB [21] utilise la même approche. La Figure 2 illustre les approches DRed et Backward/Forward. Dans cet exemple, lorsque le fait E_1 est supprimé, DRed supprime I_1 et I_2 pour les redériver ensuite via E_2, tandis que Backward/Forward identifie la dérivation de I_1 via E_2 et ne le supprime donc pas.

HyLAR+ [28] propose une approche de raisonnement incrémental dite "à base de tags". Lorsqu'un fait explicite est retiré, il n'est pas supprimé mais marqué comme invalide ; aucun autre calcul n'est nécessaire pour traiter la suppression des faits implicites. Lors d'une requête, les faits explicites invalides et les faits implicites dont les conditions ne sont plus remplies sont filtrés. Lors de multiples cycles d'insertions / suppressions, HyLAR+ permet un gain en temps jusqu'à plus de 80%. Cette approche est efficace lorsque des faits apparaissent et disparaissent régulièrement, comme dans le cas de dépassements de seuils. Elle est moins adaptée pour des mesures brutes. Comme RETE, elle a tendance à privilégier la performance au détriment de l'espace mémoire.

21. https://graphdb.ontotext.com/documentation/free/reasoning.html#retraction-of-assertions

3.3 Raisonnement sur flux

Dans le cas d'applications utilisant des données arrivant en flux, celles-ci doivent être traitées en temps réel ou quasi-réel. Des compromis doivent être trouvés entre la complexité du raisonnement demandé et la vitesse de traitement des données. Les ingrédients du raisonnement sur flux proviennent des systèmes de gestion des flux de données et des systèmes de traitement des événements complexes [29]. La plupart des approches utilisent un opérateur de fenêtrage temporel pour effectuer le raisonnement sur des sous-ensembles de données.

Slider [30] est un raisonneur sur flux à chaînage avant multithread optimisé en mémoire. Chaque thread correspond à une règle et une règle peut instancier plusieurs threads. Chaque règle a une mémoire tampon dédiée qui stocke les faits du flux entrant avant que chacun soit traité par un de ses threads ; les faits implicites et explicites sont stockés dans un triplestore partagé par tous les threads, et avec une gestion fine de la concurrence d'accès. Cependant, Slider ne gère pas de fenêtres temporelles et ne permet que l'insertion de faits.

IMaRS (Incremental Materialization for RDF Streams) [31] suppose que les flux RDF sont constitués de triplets horodatés. Comme il utilise une fenêtre glissante fixe, il associe un cachet d'expiration à chaque fait. IMaRS est d'un ordre de grandeur plus rapide que DRed jusqu'à 0,1 % de changements dans les données, et deux ordres plus rapides jusqu'à 2,5 %. Cependant, dans le cas où les données changent de plus de 13 %, il obtient des performances pires qu'une approche naïve où la matérialisation est effectuée à chaque fois que le contenu de la fenêtre change.

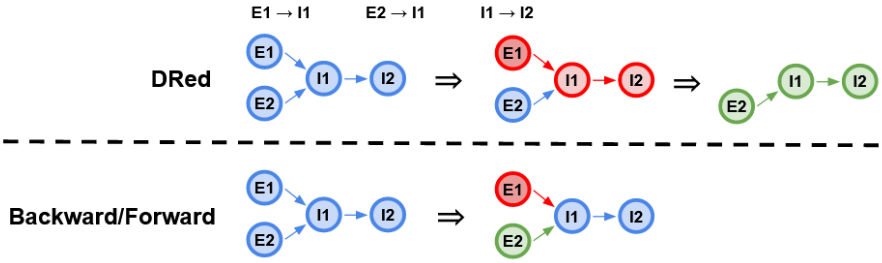

FIGURE 2 – Un exemple de raisonnement incrémental avec DRed et Backward/Forward. Les suppressions sont présentées en rouge, et les redérivations en vert.

4 Raisonnement distribué

4.1 Parallélisation

RDFox [27] est un algorithme rapide de raisonnement parallélisé qui distribue uniformément la charge de travail sur plusieurs cœurs d'un processeur. Plus précisément, les faits explicites et implicites sont ordonnés dans l'ABox. Un cœur extrait de l'ABox un fait qui n'a pas encore été traité. S'il satisfait une condition d'une règle, RDFox essaie d'appliquer les autres conditions sur les faits qui le précèdent dans l'ABox. Il utilise pour l'ABox un schéma d'indexation rapide basé sur des tables de hachage : les triplets RDF sont encodés dans un tableau à six colonnes, contenant les trois ressources du triplet et trois pointeurs vers des listes chaînées de triplets contenant respectivement le même sujet, prédicat ou objet. Cette structure permet un accès en lecture très rapide. Cependant, tous les cœurs partagent le même espace mémoire, ce qui n'est pas transposable à une architecture distribuée. Par ailleurs, l'optimisation de l'inférence de RDFox ne concerne pas l'ordre des faits et des règles, mais implémente un réordonnancement glouton des conditions atomiques dans le corps de chaque règle.

[32, 33] proposent des méthodes pour le traitement parallèle de règles par partitionnement et distribution de la charge de travail. Leur méthode est basée sur RETE et consiste à exécuter indépendamment et simultanément les nœuds alpha et bêta de l'algorithme RETE. Par exemple, les jointures des nœuds bêta d'une même profondeur peuvent être exécutées indépendamment et donc simultanément.

4.2 Distribution

Distribution des données. Lorsque les données sont distribuées, les différents raisonneurs interagissent les uns avec les autres en échangeant des messages contenant des faits implicites et explicites. Il faut veiller à ce que le coût d'échange des messages ne soit pas supérieur au coût d'envoi des données à un emplacement central, ou que ce coût soit contre-balancé par d'autres avantages.

Le sharding [34] fait référence au partitionnement horizontal des données. Lorsque les données sont partitionnées de cette manière, chaque partition de données est traitée ou exploitée séparément. De cette façon, il est possible de répartir la charge sur différentes machines et d'augmenter la fiabilité en évitant les points uniques de défaillance. Cette technique est valable dans les cas où il est possible d'identifier des partitions indépendantes.

Le raisonnement contextuel [35] permet de raisonner avec différents points de vue comme dans le cas d'une fédération de données, ou de raisonner avec différentes croyances (systèmes multi-agents). On considère ici des ontologies distribuées qui décrivent le monde selon différents points de vue ; chaque ontologie décrit le contexte du nœud où elle se trouve. Ces ontologies peuvent être construites et évoluer indépendamment les unes des autres. Il est alors parfois nécessaire d'établir des alignements entre les différentes ontologies locales. [36] propose un algorithme pour effectuer un raisonnement contextuel sur un réseau d'ontologies alignées. Leur architecture est distribuée et un raisonneur global traite les alignements. Le raisonneur global communique avec les autres raisonneurs en utilisant OWL-Link [37]. Chaque nœud utilise HermiT [38] comme raisonneur OWL.

Distribution des règles. EDR [39] est une approche de raisonnement distribué basée sur le paradigme REST. Sa topologie est hiérarchique. Les nœuds déclarent les données qu'ils peuvent fournir ainsi que les données qu'ils requièrent. EDR décompose les règles en différentes parties. Chaque nœud utilise un moteur d'inférences basé sur SHACL, qui lui permet soit d'appliquer lui-même la règle reçue, soit de transférer la règle à un nœud fils. Au final, les règles sont installées aussi profondément que possible dans la topologie du réseau. On notera que les nœuds avec EDR n'ont connaissance que de leurs voisins immédiats. De plus, EDR ne prend pas en compte les changements de topologie ni les défaillances de nœuds. EDR est donc difficilement adaptable. Le placement des règles ne tient pas compte non plus des contraintes de mémoire et d'énergie.

Distribution des tâches de raisonnement. MORe [40] est un méta-raisonneur qui répartit les tâches de classification d'ontologies entre un raisonneur OWL2-EL et un raisonneur OWL2-Full. Ce dernier étant beaucoup plus lourd, MORe réserve son usage aux parties qui ne peuvent pas être traitées par OWL2-EL. Le raisonnement est ainsi plus efficace. Le résultat est l'union logique des résultats des deux raisonneurs.

HyLAR-Framework [22] [22] permet aux applications Web

22. https://github.com/ucbl/HyLAR-Framework/

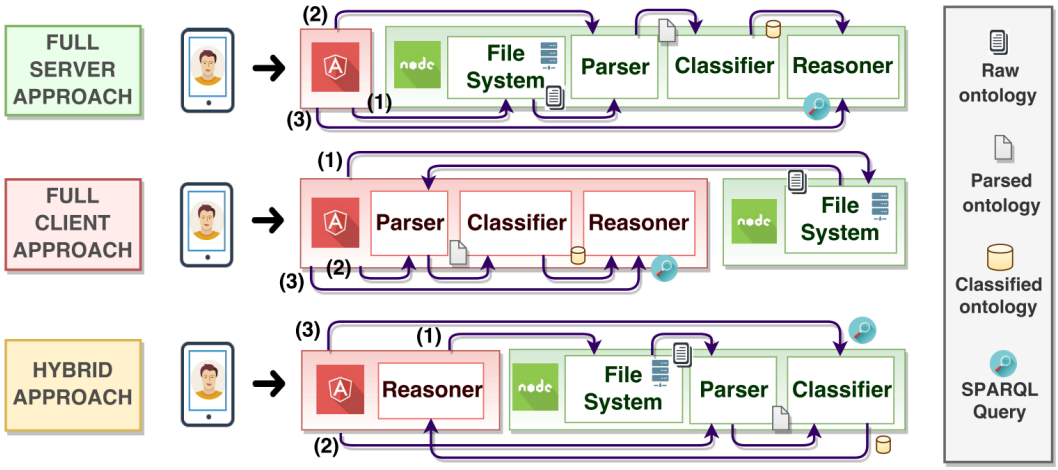

FIGURE 3 – Répartitions possibles des tâches de raisonnement avec le framework HyLAR [22]

d'effectuer des tâches de raisonnement indifféremment côté serveur ou côté client, en déployant le même raisonneur [23] aux deux endroits. Une utilisation classique est de réaliser les opérations lourdes et identiques pour tous les clients côté serveur, puis de déléguer le reste des inférences aux clients, en fonction des mises à jour incrémentales de leurs bases de connaissances locales (Figure 3). Ce framework peut aussi déterminer automatiquement s'il sera plus efficace de traiter les règles côté serveur et d'envoyer ensuite la fermeture déductive du graphe au client, ou de n'envoyer que sa réduction et de déléguer le raisonnement au client. Pour cela, il évalue les capacités de calcul du client et la qualité de sa connexion réseau, en lançant un "benchmark" correspondant à un calcul simple. Les information obtenues à l'aide de ce benchmark sont toutefois d'une qualité assez basique, compte tenu des mécanismes de protection des données en vigueur dans les API JavaScript côté client.

5 Conclusion et questions ouvertes

Le raisonnement réparti dans les architectures fog ou edge pour le Web des Objets pose de nouveaux défis, qui n'ont pour l'instant pas trouvé de solution dans la littérature. Dans cet article, nous avons identifié des problématiques et les travaux qui peuvent aider à résoudre ces défis, sur les thématiques du raisonnement embarqué, de l'optimisation du raisonnement local, du raisonnement incrémental, de la gestion de flux de données, et du raisonnement distribué. Mais de nombreux défis restent à relever, nous en citons quelques uns ci-après.

Il existe quelques approches de raisonnement embarqué dans des dispositifs contraints. Un des défis est de concevoir un raisonneur efficace en termes de mémoire, de bande passante et de consommation d'énergie. Il doit avoir une faible empreinte et définir des structures de données et algorithmes de raisonnement les plus efficaces. Celle-ci devra également être efficace en termes de consommation d'énergie et éviter une trop grande quantité de données échangées.

23. HyLAR [22], implémenté en JavaScript.

Une façon de réduire les échanges de données pourrait également consister à échanger des connaissances dans de nouveaux formats compressés. Une majorité des contributions étudiées ont été évaluées sur des smartphones, dont la puissance de calcul est supérieure à ce que nous considérons comme des appareils contraints. Ainsi, les raisonneurs développés dans ce contexte ne correspondent pas à notre projet dédié au Web des Objets. Néanmoins, les smartphones fonctionnent sur batterie et l'énergie reste une ressource critique à optimiser. Ainsi, certaines propositions faites pour les smartphones pourront être reprises pour points de départ. Par ailleurs, dans le contexte du Web des Objets et du projet CoSWoT en particulier, il est nécessaire de concevoir des raisonneurs distribués, capables de travailler sur des données distribuées.

Au sujet du raisonnement distribué, un partitionnement devra être défini pour répartir le travail entre les nœuds. Ce partitionnement pourra concerner tant les données que les règles. Plusieurs options seront étudiées. Par exemple, on peut définir des graphes de connaissances correspondant chacun à une "couche d'information" (matériel, plateforme, domaine d'application, configuration, préférences de l'utilisateur, etc.).

Une autre question ouverte est liée à l'agrégation des données. Certains cas d'utilisation de CoSWoT nécessitent de calculer des valeurs moyennes sur les données en continu, dans une fenêtre temporelle et géographique donnée. Comment ces fonctions d'agrégation seront-elles intégrées au processus de raisonnement ? Si l'agrégation est séparée du raisonnement, comment les deux processus seront-ils articulés ?

La littérature scientifique doit également encore être explorée dans d'autres domaines, notamment l'indexation efficace de triplets, l'optimisation de l'ordonnancement des règles, l'encodage des données en mémoire, l'optimisation logicielle orientée matériel ou encore l'optimisation multi-objectifs.

Remerciements

Ce travail est soutenu par la subvention ANR-19-CE23-0012 de l'Agence Nationale de la Recherche pour le projet CoSWoT.

Références

[1] Philippe Adjiman, Philippe Chatalic, François Goasdoué, Marie-Christine Rousset, and Laurent Simon. Distributed reasoning in a peer-to-peer setting: Application to the semantic web. *Journal of Artificial Intelligence Research*, 25:269–314, 2006.

[2] L Serafini and A Tamilin. Distributed reasoning architecture for the semantic web. In *Proceedings of the Second European Semantic Web Conference, ESWC*, 2005.

[3] Eyal Oren, Spyros Kotoulas, George Anadiotis, Ronny Siebes, Annette ten Teije, and Frank van Harmelen. Marvin: Distributed reasoning over large-scale semantic web data. *Journal of Web Semantics*, 7(4):305–316, 2009.

[4] Aidan Hogan, Jeff Z Pan, Axel Polleres, and Stefan Decker. Saor: template rule optimisations for distributed reasoning over 1 billion linked data triples. In *International Semantic Web Conference*, pages 337–353. Springer, 2010.

[5] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P Jue. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, 98:289–330, 2019.

[6] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.

[7] Pedro Barahona, François Bry, Enrico Franconi, Nicola Henze, and Ulrike Sattler. *Reasoning Web: Second International Summer School 2006, Lisbon, Portugal, September 4-8, 2006, Tutorial Lectures*, volume 4126. Springer, 2006.

[8] Carsten Bormann, Mehmet Ersue, and Ari Keranen. Terminology for constrained-node networks. *Internet Engineering Task Force (IETF): Fremont, CA, USA*, pages 2070–1721, 2014.

[9] Asma Haroon, Munam Ali Shah, Yousra Asim, Wajeeha Naeem, Muhammad Kamran, and Qaisar Javaid. Constraints in the iot: the world in 2020 and beyond. *Constraints*, 7(11):252–271, 2016.

[10] Zach Shelby, Klaus Hartke, and Carsten Bormann. The constrained application protocol (coap). 2014. *URL https://tools. ietf. org/html/rfc7252*, 2014.

[11] Christopher M Sadler and Margaret Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 265–278, 2006.

[12] Carlos Bobed, Roberto Yus, Fernando Bobillo, and Eduardo Mena. Semantic reasoning on mobile devices: Do androids dream of efficient reasoners ? *Journal of Web Semantics*, 35:167–183, 2015.

[13] Michele Ruta, Floriano Scioscia, Filippo Gramegna, Ivano Bilenchi, and Eugenio Di Sciascio. Mini-me swift: the first mobile owl reasoner for ios. In *European Semantic Web Conference*, pages 298–313. Springer, 2019.

[14] William Van Woensel and Syed Sibte Raza Abidi. Optimizing semantic reasoning on memory-constrained platforms using the rete algorithm. In *European Semantic Web Conference*, pages 682–696. Springer, 2018.

[15] Roberto Yus, Carlos Bobed, Guillermo Esteban, Fernando Bobillo, and Eduardo Mena. Android goes semantic: Dl reasoners on smartphones. In *Ore*, pages 46–52, 2013.

[16] Henning Hasemann, Alexander Kröller, and Max Pagel. Rdf provisioning for the internet of things. In *2012 3rd IEEE International Conference on the Internet of Things*, pages 143–150. IEEE, 2012.

[17] Wei Tai, John Keeney, and Declan O'Sullivan. Resource-constrained reasoning using a reasoner composition approach. *Semantic Web*, 6(1):35–59, 2015.

[18] Remy Rojas, Lionel Médini, and Amélie Cordier. Toward Constrained Semantic WoT. In *Seventh International Workshop on the Web of Things (WoT 2016)*, pages 31 – 37, Stuttgart, Germany, November 2016. W3C, ACM New York, NY, USA.

[19] Victor Charpenay, Sebastian Käbisch, and Harald Kosch. Towards a binary object notation for rdf. In *European Semantic Web Conference*, pages 97–111. Springer, 2018.

[20] Pierre-Antoine Champin. Sophia: a Linked Data and Semantic Web toolkit for Rust. In Erik Wilde and Mike Amundsen, editors, *The Web Conference 2020: Developers Track*, Taipei, Taiwan, April 2020.

[21] Julian Bruyat. Web assembly pour le web sémantique. Master's thesis, Université Claude Bernard Lyon 1, 2020. http://bruy.at/BRUYAT_Rapport_WasmPourWebSem.pdf.

[22] Mehdi Terdjimi, Lionel Médini, and Michael Mrissa. Hylar: Hybrid location-agnostic reasoning. In *ESWC Developers Workshop 2015*, page 1, 2015.

[23] Julien Subercaze, Christophe Gravier, Jules Chevalier, and Frederique Laforest. Inferray: fast in-memory rdf inference. 2016.

[24] Charles L Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. In *Readings in Artificial Intelligence and Databases*, pages 547–559. Elsevier, 1989.

[25] Ashish Gupta, Inderpal Singh Mumick, and Venkatramanan Siva Subrahmanian. Maintaining views incrementally. *ACM SIGMOD Record*, 22(2):157–166, 1993.

[26] Jacopo Urbani, Alessandro Margara, Ceriel Jacobs, Frank Van Harmelen, and Henri Bal. Dynamite: Parallel materialization of dynamic rdf data. In *International Semantic Web Conference*, pages 657–672. Springer, 2013.

[27] Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. Rdfox: A highly-scalable rdf store. In *International Semantic Web Conference*, pages 3–20. Springer, 2015.

[28] Mehdi Terdjimi, Lionel Médini, and Michael Mrissa. Web reasoning using fact tagging. In *Companion Proceedings of the The Web Conference 2018*, pages 1587–1594, 2018.

[29] Daniele Dell'Aglio, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. Stream reasoning: A survey and outlook. *Data Science*, 1(1-2):59–83, 2017.

[30] Jules Chevalier, Julien Subercaze, Christophe Gravier, and Frédérique Laforest. Incremental and directed rule-based inference on rdfs. In *International Conference on Database and Expert Systems Applications*, pages 287–294. Springer, 2016.

[31] Daniele Dell'Aglio and Emanuele Della Valle. Incremental reasoning on rdf streams., 2014.

[32] Martin Peters, Christopher Brink, Sabine Sachweh, and Albert Zündorf. Rule-based reasoning on massively parallel hardware. In *SSWS@ ISWC*, pages 33–49, 2013.

[33] Martin Peters, Christopher Brink, Sabine Sachweh, and Albert Zündorf. Scaling parallel rule-based reasoning. In *European Semantic Web Conference*, pages 270–285. Springer, 2014.

[34] Pramod J Sadalage and Martin Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2013.

[35] Fausto Giunchiglia and Chiara Ghidini. Local models semantics, or contextual reasoning= locality+ compatibility. *KR*, 98:282–289, 1998.

[36] Jérémy Lhez, Chan Le Duc, Thinh Dong, and Myriam Lamolle. Decentralized reasoning on a network of aligned ontologies with link keys. In *International Semantic Web Conference*, pages 418–434. Springer, 2019.

[37] Thorsten Liebig, Marko Luther, Olaf Noppens, and Michael Wessel. Owllink. *Semantic Web – Interoperability, Usability, Applicability*, 2(1):23–32, 2011.

[38] Rob Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient owl reasoner. In *Owled*, volume 432, page 91, 2008.

[39] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Edr: A generic approach for the dynamic distribution of rule-based reasoning in a cloud-fog continuum. *Semantic Web Journal*, 2019.

[40] A.A. Romero, B.C. Grau, I. Horrocks, and Ernesto Jiménez-Ruiz. More: A modular owl reasoner for ontology classification. *CEUR Workshop Proceedings*, 1015:68–74, 01 2013.

Proposition d'un modèle de trajectoires multi-aspects et multi-niveaux appliqué au tourisme

C. Cayèré[1], C. Sallaberry[2], C. Faucher[1], M.-N. Bessagnet[2], P. Roose[3]

[1] La Rochelle Université, L3i, La Rochelle, France
[2] Université de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Pau, France
[3] Université de Pau et des Pays de l'Adour, E2S UPPA, LIUPPA, Anglet, France

21 mai 2021

Résumé

Dans un contexte d'analyse de traces de mobilité touristique, nous avons conçu un modèle de trajectoire sémantique répondant à des besoins spécifiques exprimés par des experts du tourisme. Ainsi, ce modèle prend en compte : (i) la description de séquences d'épisodes imbriqués/hiérarchisés, (ii) la définition d'aspects sémantiques intégrant les dimensions spatiale, temporelle et thématique et (iii) l'association d'aspects sémantiques à des positions ou encore à des épisodes de trajectoire. Chacune de ces caractéristiques est nécessaire au traitement et à l'analyse de données de mobilité touristique que nous détaillerons. À des fins de validation, nous expérimentons notre modèle sur deux cas d'usage de traces de mobilité en extérieur que nous avons analysées dans une chaîne de traitement dédiée. Nous montrons également que notre modèle est générique et extensible.

Mots-clés

Modèle de trajectoire sémantique, trajectoire multi-aspect, trace de mobilité, pratiques touristiques

Abstract

We designed a semantic trajectory model responding to specific needs expressed by tourism analyst experts. Thus, this model takes into account : (i) the description of sequences of nested/hierarchical episodes, (ii) the definition of semantic aspects integrating spatial, temporal and thematic dimensions, and (iii) the association of such semantic aspects to positions or to trajectory episodes. Each of these features is necessary for the processing and analysis of tourist mobility data, which we will detail. For validation purposes, we experiment our model on two use cases of outdoor mobility traces that we computed in a processing chain. We also show that our model is generic and extensible.

Keywords

Semantic trajectory model, multi-aspect trajectory, mobility track, tourist practices

1 Introduction

Le projet Région Nouvelle Aquitaine DA3T (Dispositif d'Analyse des Traces numériques pour la valorisation des Territoires Touristiques) a pour objectif d'améliorer la gestion et la valorisation des territoires touristiques littoraux de la Nouvelle-Aquitaine en utilisant des traces de mobilité touristiques à la fois en intérieur et en extérieur et des données de contexte. Nous disposons de jeux de données provenant de sources variées décrivant des visites touristiques (p. ex. des traces de déplacement de touristes dans la ville de La Rochelle ou encore dans les ateliers de découverte de la Cité du vin de Bordeaux). Pour répondre à l'objectif du projet, géographes et informaticiens souhaitent mettre en place des méthodes et des outils permettant de réaliser des traitements manuels et automatiques dont le résultat permettra d'extraire des connaissances profitables aux acteurs de l'aménagement des territoires.
Dans le cadre de cet article, nous utiliserons des traces de mobilité en extérieur appartenant à des touristes volontaires. Ils ont accepté d'utiliser notre application mobile de capture de déplacements durant leur visite de la ville mais également de participer à un entretien mené à la fin de leur séjour afin de compléter les données brutes de déplacement. Ces traces de mobilité touristique, les entretiens, les traitements dédiés à l'aménagement du territoire mettent en exergue des besoins de représentation de mobilité de granularités diverses (p. ex. des suites de points, des suites de segments de trajectoires, etc.). La représentation d'informations contextuelles calculées (p. ex. la vitesse) ou issues de ressources externes (p. ex. la météo) est également importante. Qualifiées d'aspects sémantiques, ces informations contextuelles seront associées à des positions ou encore à des segments de trajectoires. Ainsi, nous décrivons une mobilité à travers différentes représentations sémantiques : segments annotés par des aspects sémantiques. Ces segments décrivent des caractéristiques de déplacement : (i) segments sémantiques disjoints (p. ex. il marche tout en mangeant une glace) ; (ii) segments sémantiques composés/imbriqués (p. ex. à la fin de sa visite, le touriste achète un livre dans le magasin du musée). Le modèle DA3T que nous proposons relève deux principaux défis : décrire une

trajectoire comme une suite de segments ou épisodes disjoints ou imbriqués ; décrire chaque épisode par des caractéristiques ou aspects sémantiques de dimension spatiale, temporelle et thématique.
L'article est organisé comme suit. La partie 2 décrit les travaux relatifs aux données de mobilité en énonçant, dans un premier temps, quelques définitions essentielles puis en abordant, dans un second temps, la modélisation des trajectoires sémantiques dans la littérature. La partie 3 expose le scénario de motivation du projet DA3T en décrivant les données manipulées (c.-à-d. les traces de mobilité de touristes ainsi que des données d'enrichissement potentielles) puis en spécifiant les besoins des géographes et des aménageurs quant à la modélisation des trajectoires sémantiques. Les verrous de recherche visés sont ensuite exposés. La partie 4 présente notre contribution : le modèle DA3T dédié à la représentation de trajectoires sémantiques. La partie 5 met à l'épreuve le modèle DA3T à travers deux cas d'usage. Enfin, la partie 6 conclut et présente les perspectives de travaux futurs.

2 Travaux connexes

2.1 Quelques définitions

Le déplacement est un phénomène de nature continue observable partout dans notre environnement physique. Pour faciliter la capture et le stockage d'un déplacement, il est discrétisé ; c.-à-d. qu'il est simplifié en une suite de **positions** géolocalisées et horodatées, appelée **trace de mobilité**. Ainsi, la position p d'un objet mobile o à un instant t est un tuple $p = (o, x, y, t, D)$ avec x et y les coordonnées spatiales (qu'elles soient géographiques ou planaires), t le temps de la capture et D un ensemble de données complémentaires capturées en même temps que la position (p. ex. la vitesse, la précision, etc.).
Selon [8], une **trajectoire brute** d'un objet mobile est un segment de la trace de mobilité qui a de l'intérêt pour une application donnée. Dans notre contexte de visiteurs à La Rochelle, les trajectoires peuvent être construites sur un critère temporel (p. ex. une trajectoire par jour) en faisant abstraction des parties de trace de mobilité en dehors du département de la Charente-Maritime.
Souvent la trajectoire brute est limitée pour comprendre tous les enjeux d'un déplacement. Cette constatation a donné lieu à la création du concept de **trajectoire sémantique** [8] qui propose de lier des données sémantiques soit à la trajectoire, soit à un segment de la trajectoire, soit à une position de la trajectoire. Une donnée sémantique est simple ou complexe. Une donnée sémantique simple décrit un objet du monde réel sous la forme d'une **annotation** textuelle (p. ex. la Tour Saint-Nicolas peut être décrite par son *nom*). Une donnée sémantique complexe, quant à elle, décrit un objet du monde réel sous la forme d'un agrégat de caractéristiques (ensemble de données de différents types). Dans ce dernier cas, nous parlons d'**aspect** sémantique [6] (p. ex. un *point d'intérêt* peut être décrit par son *nom*, sa *localisation*, ses *heures d'ouverture*, etc.). Un segment de la trajectoire associé à des données sémantiques s'appelle communément un **épisode** [12].
Pour passer d'une trajectoire brute à une trajectoire sémantique, plusieurs traitements dits d'enrichissement peuvent être mis en œuvre. Par exemple, la **segmentation** est le fait de diviser la trajectoire en segments de trajectoire selon un certain critère. L'**annotation** (manuelle ou automatique) est le fait d'attacher des données sémantiques à la trajectoire, un segment de la trajectoire ou une position de la trajectoire. Ces deux traitements peuvent être confondus, il est possible de segmenter et d'annoter avec le même critère. Une **interprétation** de la trajectoire est une séquence d'épisodes obtenue après segmentation ou annotation de la trajectoire [10].
Nous présentons ci-après des modèles de trajectoires sémantiques exploitables dans le domaine du tourisme.

2.2 Modèles de trajectoires sémantiques

Nous avons identifié trois catégories de modèles de trajectoires sémantiques. (i) La première est dédiée à la segmentation de la trajectoire en une suite d'arrêts et de déplacements. (ii) La seconde vise la segmentation de la trajectoire en une suite d'épisodes. (iii) La troisième modélise des phénomènes du monde réel à des fins d'enrichissement mais indépendamment de toute trajectoire et segmentation de trajectoire.
En ce qui concerne la première catégorie (i), les travaux de [9], datant de 2008, introduisent un modèle basé sur les arrêts et déplacements (en anglais, *stops and moves*) qui segmente la trajectoire en temps d'arrêt et temps de déplacement. Ce modèle enrichit chaque segment avec des annotations textuelles (données sémantiques simples) qui en font des épisodes. Il a été repris dans de nombreux autres travaux [1][11][5]. Notons que les épisodes ainsi manipulés se limitent toujours à des arrêts et déplacements annotés.
La seconde catégorie (ii) vise la segmentation d'une trajectoire en une suite d'épisodes. Ici, les épisodes sont également enrichis par des données sémantiques (p. ex. il peut-être pertinent de réaliser une segmentation et un enrichissement de la trajectoire basée sur les activités touristiques). Ainsi, l'ontologie Baquara 2 [4] et l'ontologie STEP [7] s'appuient sur la notion d'épisode sémantique pour enrichir les trajectoires. Indépendamment de la catégorie, [4] et [7] intègrent également une hiérarchie entre les épisodes dans laquelle ils peuvent être décomposés en sous-épisodes.
Enfin, la troisième catégorie (iii) modélise des phénomènes du monde réel à des fins d'enrichissement mais indépendamment de toute trajectoire et segmentation de trajectoire. L'annotation d'une ou de plusieurs trajectoires ou positions se fait a posteriori. Le modèle MASTER [6] présente, en 2019, une nouvelle approche qui ne repose pas sur la segmentation de la trajectoire. Elle vise l'association de données sémantiques complexes avec des trajectoires, appelée trajectoires multi-aspects. Ainsi, une trajectoire est enrichie avec des objets du monde réel pertinents pour l'analyse, appelés aspects. Un aspect peut être attaché à une ou plusieurs trajectoires ou à une ou plusieurs positions de trajectoire. Il faut noter qu'un aspect peut avoir une existence sans être pour autant rattaché à des trajectoires ou des positions (p.

ex. l'aspect Tour de la Lanterne est décrit d'abord indépendamment de toute trajectoire mais peut ensuite être lié à des trajectoires ou des positions). Dans ce modèle les aspects possèdent des attributs qui ont des valeurs uniquement textuelles.

3 Projet DA3T : scénario de motivation

Le scénario de motivation, issu du projet DA3T, se découpe en trois parties. La première décrit les spécificités des données manipulées dans le cadre du projet. La deuxième exprime les besoins des géographes en termes de représentation et de traitement de ces données. Enfin, la troisième décrit les verrous scientifiques correspondants.

3.1 Traces de mobilité touristiques et données d'enrichissement

Nous collectons trois types de données : (i) des traces de mobilité des visiteurs de la ville de La Rochelle, (ii) des données issues des entretiens de tout ou partie de ces mêmes visiteurs et (iii) des données contextuelles permettant l'enrichissement de ces traces (météo, points d'intérêt, etc.).
Notre démarche de collecte des données comporte plusieurs phases : (1) promotion de notre projet auprès de visiteurs dans les offices de tourisme de La Rochelle, (2) collecte des traces de mobilité des touristes volontaires grâce à l'application mobile Geoluciole qui, à intervalles de temps réguliers, capture la position du téléphone (c-à-d. de son porteur), (3) réalisation, a posteriori, d'entretiens semi-directifs avec ces mêmes touristes pour demander des précisions et des explications sur leurs déplacements et, enfin, (4) recherche, dans l'Open Data, de données contextuelles en lien avec la ville. À l'issue de ces étapes, nous disposons d'un ensemble de traces de mobilité brutes, collectées grâce à Geoluciole, d'entretiens et de données contextuelles diverses.

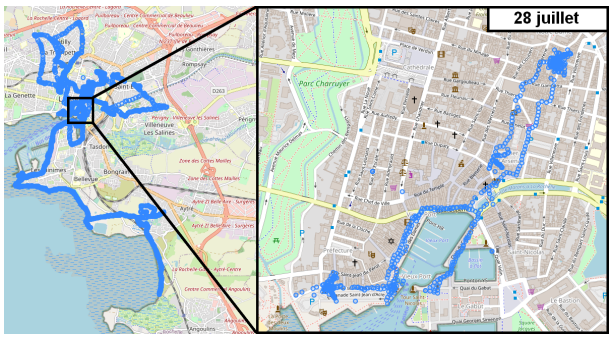

FIGURE 1 – Trace de mobilité issue de l'application Geoluciole appartenant à un visiteur volontaire en vacances à La Rochelle pendant un mois. À gauche, les données collectées et affichées ici concernent uniquement la deuxième semaine de la visite. À droite, seul le jour 4 de la deuxième semaine (28/07) du séjour est isolé.

La figure 1 montre, à gauche, un exemple de traces issues de Geoluciole. L'extrait d'entretien qui correspond à la journée du 28 juillet, isolée à droite sur la figure, est : *"Le 28/07, visite des tours : d'abord de la Chaîne, puis la Lanterne puis on a voulu aller tour St. Nicolas mais c'était trop tard donc on a fait demi-tour. Puis, on est allé visiter le musée du protestantisme vers 17h30 je crois, ça fermait à 18h."*. Les différents points d'intérêt mentionnés dans l'entretien sont en relation avec la trace mais, nous comprenons, grâce au discours, que la tour Saint-Nicolas n'a pas été visitée à cause de l'heure tardive. L'entretien peut ainsi compléter la trace. Les activités d'un visiteur peuvent être menées en séquence ou en parallèle (p. ex. *[...] j'ai voulu trouver un autre endroit où aller lire et je me suis souvenu qu'il y avait de l'herbe par-là donc ouais, j'y suis allé*). En somme, l'entretien peut compléter les silences de la trace et vice-versa. Par ailleurs, nous intégrons des données dites d'enrichissement des traces. Nous disposons de nombreuses données de contexte, disponibles sur l'*Open Data* telles que la météo, les points d'intérêt, les événements sociaux, etc. Elles sont de différents types et sont décrites, généralement, par trois dimensions : spatiale, temporelle et thématique. Chaque dimension n'est pas forcément renseignée.

3.2 Expression des besoins des géographes

Le projet DA3T est un projet pluridisciplinaire et les géographes et aménageurs exploitent les données de mobilité touristique et d'enrichissement selon des représentations que nous allons préciser dans cette partie. Une trajectoire est décrite par une suite de positions (géolocalisées et horodatées) tout comme elle peut-être décrite, à un niveau d'abstraction supérieur, par une séquence d'*épisodes* (chacun étant également composé d'une suite de positions). Par exemple, la trajectoire de la Figure 1, peut-être décomposée en épisodes, chacun correspondant à une suite de positions journalières. Nous l'avons déjà précisé, un aspect sémantique est une valeur ajoutée, issue notamment de sources contextuelles, indispensables aux géographes et aménageurs. Le triptyque spatial, temporel et thématique est mobilisé pour la description d'un aspect sémantique : par exemple, un événement des Francofolies peut-être décrit par une *localisation*, une *durée* et un *libellé*. Ainsi, nous comprenons bien qu'il sera intéressant d'associer un aspect sémantique à une position ou à un épisode. Par exemple, une trajectoire pourrait correspondre à une séquence d'épisodes décrivant les différents moyens de locomotion (*bus*, *marche*, *vélo*) mis en œuvre successivement par le touriste. Enfin, de telles séquences d'épisodes peuvent se décomposer récursivement en sous niveaux hiérarchiques. Par exemple, un épisode correspondant à la pratique touristique *découverte* pourrait se décomposer en une séquence de deux sous-épisodes *visite* et *restauration*.

3.3 De la trace touristique à la trajectoire sémantique : verrous et hypothèses de travail

Les caractéristiques des données de mobilité touristique et les attentes spécifiques des géographes et aménageurs qui les exploitent mettent en exergue des besoins de modélisa-

tion bien particuliers. Nous devons, dans un même modèle, être en mesure de décrire des positions, des trajectoires brutes et des trajectoires sémantiques, intégrant les caractéristiques décrites dans la partie 3.2. Ainsi, à l'issue d'un traitement d'une trace de mobilité (c.-à-d. pré-traitement, enrichissement, filtrage, etc.), nous obtenons une ou plusieurs trajectoires brutes (c.-à-d. des segments de la trace de mobilité qui a de l'intérêt pour une application donnée) ou encore des trajectoires sémantiques (c.-à-d. des segments de la trace enrichie avec une ou plusieurs interprétations spécifiques basées sur des données contextuelles de sources diverses).

Les verrous résident dans (i) la modélisation de séquences d'épisodes imbriqués/hiérarchisés pour enrichir les trajectoires ; (ii) la modélisation générique des données d'enrichissement intégrant les dimensions spatiale, temporelle et thématique ; (iii) l'association de données d'enrichissement à une position ainsi qu'à des épisodes de trajectoire.

Nous faisons l'hypothèse de ré-utiliser et d'étendre les notions de granularité [4][7] et de multi-aspects [6] afin de résoudre les 3 verrous (i), (ii) et (iii) précédents. Il n'existe pas, à notre connaissance, de modèle de trajectoire sémantique intégrant ces différentes notions simultanément.

4 Modèle de trajectoire sémantique DA3T

La figure 2 montre notre modèle de trajectoires sémantiques qui est décomposé en trois parties distinctes :

La partie *Raw data level* (c.f. figure 2, bloc 3) rassemble les classes représentant les données brutes collectées. Les données générales relatives aux objets mobiles sont dans la classe *MobileObject* et les données plus spécifiques relatives à une catégorie d'objets mobiles en particulier (p. ex. les touristes volontaires de notre projet) sont dans les classes correspondantes qui en héritent (p. ex. la classe *GeolucioleVisitor*). La trace de mobilité d'un objet mobile est décrite grâce aux classes *Position* et *Trajectory*. Il peut y avoir plusieurs types de positions (p. ex. dans notre projet, *IndoorPosition* pour les positions de visiteurs dans les musées et *OutdoorPosition* pour les positions collectées avec Geoluciole) qui ont différents attributs, mais qui héritent toutes de la classe générique *Position*. Un objet mobile possède une suite de positions qui décrit le déplacement capturé au complet, c.-à-d. sa trace de mobilité. Les trajectoires sont des sous-parties de cette trace de mobilité qui présentent un intérêt pour une application donnée. Ce modèle est **générique** et **extensible** en fonction du contexte applicatif. Les parties qui peuvent être étendues sont la classe *MobileObject*, à laquelle il est possible d'ajouter des classes enfants représentant de nouveaux types d'objets mobiles (p. ex. une classe d'objets mobiles *Vehicle* héritant de *MobileObject*) et la classe *Position* à laquelle il est possible d'ajouter des classes enfants représentant de nouveaux types de positions (p. ex. une classe de positions *GPSPosition* héritant de *Position*)

La partie *Semantic data level* (c.f. figure 2, bloc 1) regroupe les données sémantiques. Comme dans le modèle MASTER [6], nous souhaitons représenter les données sémantiques sous la forme d'aspects sémantiques. Notre modèle est donc qualifié de modèle **multi-aspects**. Quatre classes principales représentent ces aspects (*Aspect*, *AspectType*, *Attribute* et *Value*). L'aspect représente un phénomène du monde réel identifié comme ayant de l'intérêt pour une application en question. Le type d'aspect représente la catégorie de ce phénomène (p. ex. *pratique touristique*, *point d'intérêt*, *moyen de déplacement*, *météo*, etc.) et possède des attributs spécifiques (p. ex. les *points d'intérêt* sont chacun caractérisés par un *nom*, une *localisation*, un *type*, etc.). Un type d'aspect peut avoir des sous-types (p. ex. un type *mode de transport* peut avoir comme sous-type *voiture*, *vélo*, *bus*, etc.). Lors de la création d'un aspect, au minimum un type d'aspects lui est associé et chaque attribut associé à ce type est instancié grâce à la classe d'association *Value* (p. ex. l'aspect *tour de la lanterne* est un *point d'intérêt* qui a pour *nom* : *Tour de la Lanterne*, pour *localisation* : *[46.1558333,-1.1569444]*, pour *type* : *tour*, etc.). Dans le modèle original, les attributs sont uniquement instanciés sous la forme de chaînes de caractères. Dans notre modèle, nous avons choisi d'ajouter des classes pour distinguer les attributs temporels, spatiaux et thématiques. De plus, certaines classes du modèle peuvent être reliées à des concepts provenant d'ontologies grâce à un attribut *uri* (p. ex. pour décrire des aspects de type *point d'intérêt*, on peut s'appuyer sur des ontologies externes comme celle de DataTourisme [1]). Une URI peut soit décrire un type d'aspect, soit un attribut d'aspect, soit une valeur d'un aspect particulier.

La partie *Interpretation level* (c.f. figure 2, bloc 2) sert de lien entre les données brutes et les données sémantiques. La classe *semanticMeaning* est une classe reprise du modèle MASTER qui sert précisément à faire ce lien. Un sens sémantique peut être attaché au visiteur à une position de son déplacement, mais également à un épisode de la classe *Episode*. Il est possible d'exprimer la granularité des épisodes grâce au lien de composition récursif signifiant qu'un épisode peut être précisé par d'autres épisodes. Ainsi, notre modèle est qualifié de modèle **multi-niveaux** [4]. Une trajectoire spécifique peut être liée à une ou plusieurs interprétations de la classe *Interpretation*. Une interprétation est une séquence d'épisodes particulière (p. ex. la trajectoire d'un touriste peut avoir une interprétation pour décrire la *météo* au cours du déplacement, une autre interprétation pour décrire les *pratiques touristiques*, etc.) [10]. La classe *Pattern* permet de faire des regroupements de trajectoires qui présentent des caractéristiques spatiales, temporelles ou sémantiques communes (p. ex. les trajectoires où les visiteurs ont fait les mêmes activités, les trajectoires qui ont traversé les mêmes quartiers, etc.). Enfin la classe *MobileObjectCategory* permet de classer les visiteurs dans des catégories particulières selon leurs comportements (p. ex. les touristes, les habitants, etc.).

Nous avons présenté un modèle multi-aspects et multi-

1. Lien vers l'ontologie DataTourisme : https ://framagit.org/datatourisme/ontology

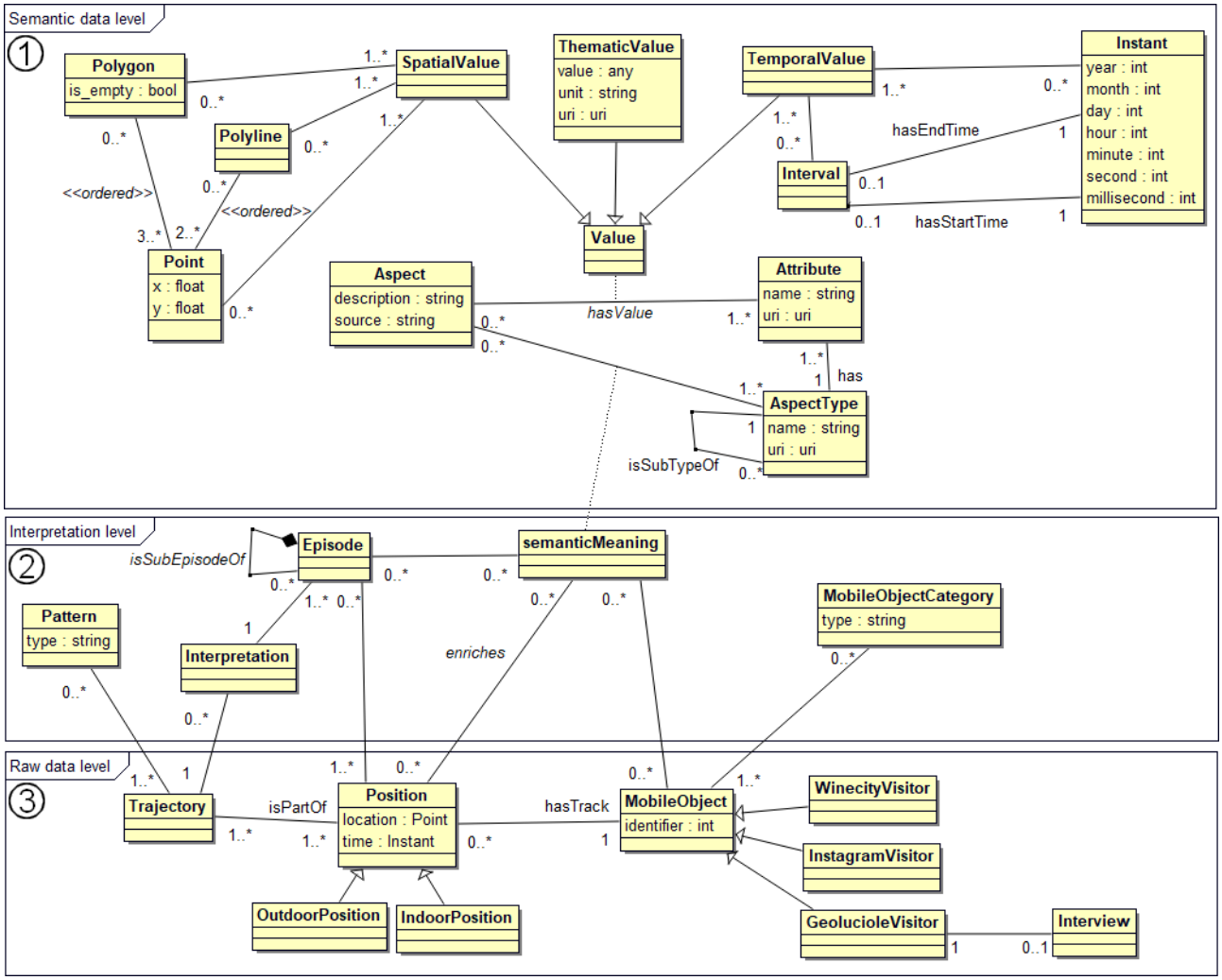

FIGURE 2 – Modèle de trajectoire sémantique

niveaux qui est générique et extensible. Utilisons maintenant de vrais cas d'usage pour le tester.

5 Expérimentation du modèle DA3T

L'objectif de cette partie est de présenter l'instanciation de notre modèle de trajectoire sémantique dans le contexte particulier du projet DA3T. Les traitements des données sont réalisés grâce à une plateforme modulaire où toutes les données en entrée et en sortie des services sont des instanciations du modèle. Dans un premier temps, nous présentons brièvement la plateforme modulaire [3] à travers un premier cas d'usage. Dans un second temps, le modèle est mis à l'épreuve selon deux cas d'usage dont le premier.

5.1 Plateforme de traitement modulaire

La plateforme de traitement modulaire créée et utilisée dans le projet a fait l'objet de deux articles [3] [2]. Il s'agit d'une plateforme permettant de construire des chaînes de traitement personnalisées à l'aide de services. Chaque service est un composant logiciel qui effectue un traitement spécifique. Ils peuvent être classés dans différentes catégories qui regroupent les services avec des objectifs similaires (c.-à-d. *Pré-traitement*, *Filtrage*, *Enrichissement*, *Modification*, *Agrégation* et *Visualisation*). Une chaîne de traitement est une suite de services qui répond à une question spécifique sur un jeu de données.

Ainsi, la figure 3 montre un exemple de chaîne de traitement qui permet de répondre à la question (1) : Quelles sont les traces de mobilité qui sont passées par le quartier *Les Minimes* et par le quartier *Saint-Nicolas* dans la même journée ?

Nous utilisons ici des données issues de Geoluciole. Cette chaîne de traitement prend en entrée des traces de mobilité brutes, affiche en sortie une carte avec toutes les trajectoires répondant à la problématique et enchaîne des services de quatre catégories différentes.

Pour commencer, la catégorie *Pré-traitement* (c.f. figure 3, catégorie 1) regroupe les services qui pré-traitent les données. Les traces de mobilité des touristes sont nettoyées avec le service *Nettoyage basé sur la précision de la cap-*

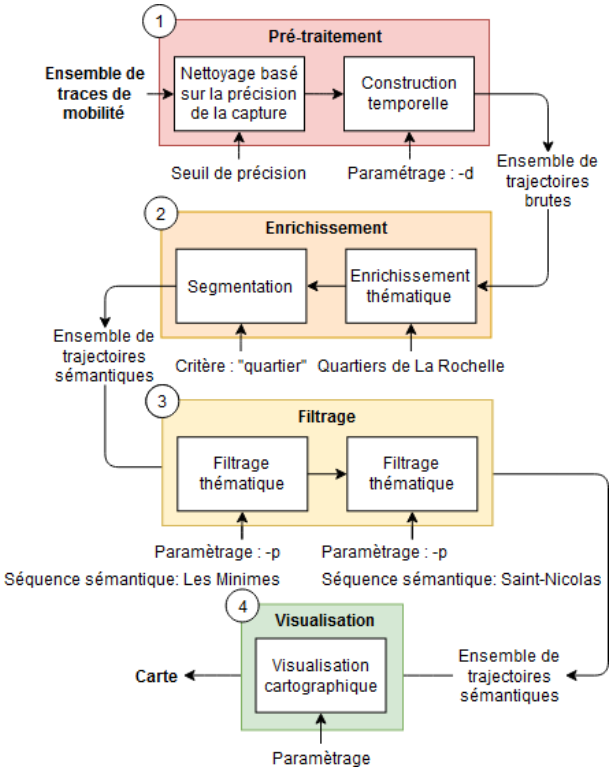

FIGURE 3 – Chaîne de traitement personnalisée permettant de répondre à la question (1)

ture. Ce service s'appuie sur la valeur de précision enregistrée au moment de la capture de la position et indique, en mètres, le rayon d'incertitude autour de la position. Il est à noter que certains services ont été spécifiquement développés pour accepter certains types de données (ici, p. ex. il faut que les positions des traces en entrée disposent de cette valeur de précision). Le service suivant *Construction temporelle* construit des trajectoires brutes à partir des traces de mobilité nettoyées. La question pousse à nous intéresser aux trajectoires des visiteurs à l'échelle temporelle de la journée. Ainsi, le paramétrage *-d* (pour *day*) indique que le service construit une trajectoire par jour et par personne.
La catégorie *Enrichissement* (c.f. figure 3, catégorie 2) regroupe les services qui transforment une trajectoire brute en une trajectoire sémantique. Le service *Enrichissement sémantique* lit des données sémantiques externes aux positions de la trajectoire. Ici, chaque position des trajectoires est liée à l'aspect représentant le quartier de La Rochelle dans lequel elle se trouve. Le service suivant *Segmentation* construit, pour chaque trajectoire, une interprétation basée sur l'enrichissement réalisé précédemment, c.-à-d. une séquence d'épisodes composée des différents quartiers de La Rochelle traversés. Le résultat de ce service est un ensemble de trajectoires sémantiques.
La catégorie *Filtrage* (c.f. figure 3, catégorie 3) regroupe les services qui filtrent les trajectoires selon des critères spécifiques. Ici, le service *Filtrage thématique* filtre les trajectoires et donne comme résultat celles qui présentent une certaine sous-séquence sémantique dans leur séquence d'épisodes correspondante. Ici, le service est appelé deux fois : la première fois, il filtre les trajectoires qui sont passées par le quartier *Les Minimes* et la seconde fois, il filtre les trajectoires qui sont passées par le quartier *Saint-Nicolas*.
Enfin, la catégorie *Visualisation* (c.f. figure 3, catégorie 4) regroupe les services qui permettent de visualiser des résultats. Le service *Visualisation cartographique* permet de représenter les données en entrée sur une carte selon un certain paramétrage. Ici, seules les trajectoires qui passent par les quartiers *Les Minimes* et *Saint-Nicolas* apparaîtront sur la carte (c.f. figure 4). Comme chaque trajectoire représente le déplacement d'une personne pour un jour spécifique, nous avons répondu à la question (1).

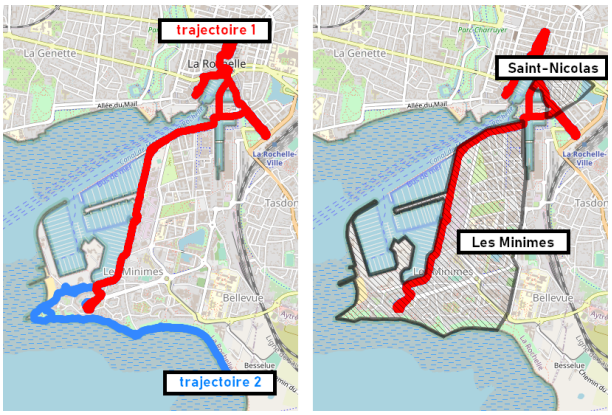

FIGURE 4 – À gauche, résultat après le service *Construction temporelle*, à droite, résultat après le service *Filtrage thématique*

5.2 Mise en œuvre avec deux cas d'usage

Nous allons maintenant instancier notre modèle de trajectoire sémantique selon deux cas d'usage.
Dans un premier temps, nous allons reprendre le cas d'usage de la partie précédente avec la question (1) et nous appuyer sur la chaîne de traitement présenté en figure 3. Rappelons que les entrées et sorties de chaque service sont des instanciations du modèle. Afin de tester son intégration à la plateforme, nous observerons l'instanciation du modèle à une étape de la chaîne de traitement. La figure 6 montre l'instanciation du modèle juste avant le filtrage. Après le filtrage, la trajectoire 1 apparaîtra en sortie car elle passe par les deux quartiers spécifiés. Par souci de lisibilité, la figure représente la trace de mobilité d'un seul visiteur Geoluciole.
Nous souhaitons maintenant répondre à la question (2) : Quelles sont les pratiques touristiques des visiteurs lorsqu'ils sont dans le quartier *Centre-ville* de La Rochelle ?
Nous allons expliquer la chaîne de traitement à travers la figure 5 qui schématise un exemple d'interprétation d'une trajectoire. Dans un premier temps, nous partons des traces de mobilité des visiteurs. Nous nous intéressons aux pratiques touristiques en centre-ville, par conséquent, le dé-

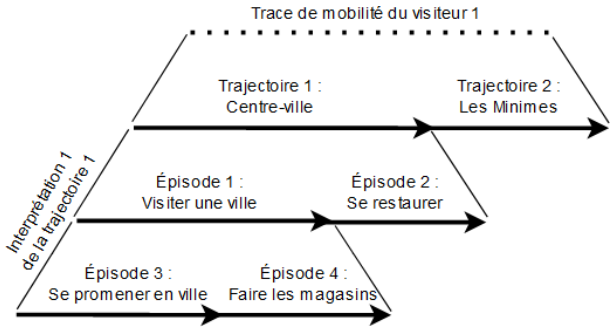

FIGURE 5 – Exemple d'interprétation d'une trajectoire

coupage des traces en trajectoire se fonde sur des critères spatiaux et utilise les zones décrivant les quartiers de La Rochelle. Nous aboutissons, dans l'exemple, à deux trajectoires pour le visiteur 1, une, se déroulant dans le quartier *Centre-ville*, l'autre, dans le quartier *Les Minimes*. Chaque position est enrichie avec la pratique touristique qui lui est associée. Nous construisons ensuite les interprétations des trajectoires basées sur cet enrichissement. Dans l'exemple, l'interprétation 1 est composée de quatre épisodes décrivant les pratiques touristiques réalisées par le visiteur à ces moments-là. Il est à noter que les épisodes 3 et 4 sont imbriqués dans l'épisode 1. Il suffit, ensuite, d'utiliser un service de filtrage afin de ne sélectionner que les trajectoires se situant dans le quartier *Centre-ville*, puis, grâce à un service de visualisation, d'afficher les pratiques touristiques qui leur sont associées. Nous obtenons, en résultat, une liste de pratiques touristiques réalisées au centre-ville de La Rochelle. Dans l'exemple la liste est composée des aspects *Visiter une ville*, *Se promener en ville* et *Faire les magasins* et *Se restaurer*.

La figure 7 montre l'instanciation du modèle juste avant le filtrage. Dans un premier temps, chaque position est enrichie avec les aspects sémantiques représentant les pratiques touristiques qui leur correspondent. Chaque pratique est liée à un concept d'une ontologie du domaine que nous utilisons dans le projet, cependant sa description sort du cadre de cet article. Par souci de lisibilité, les liens entre les classes *Position* et *semanticMeaning* ne sont pas affichés mais les positions 1, 2 et 3 sont associées à la pratique de *Visiter la ville*, les positions 1 et 2 sont aussi associées à la pratique *se_promener_en_ville*, la position 3 est aussi associée à la pratique de *faire_les_magasins*, enfin, la position 4 est associée à la pratique *se_restaurer*. La segmentation s'appuie ensuite sur cet enrichissement pour créer une interprétation relative aux pratiques touristiques. Nous pouvons voir que quatre épisodes ont été créés. L'épisode 1 correspond à *visiter_la_ville* et l'épisode 2 correspond à *se_restaurer*. Les épisodes 3 et 4 composent l'épisode 1 et correspondent respectivement à *se_promener_en_ville* et *faire_les_magasins*.

Cette partie a permis de démontrer l'intérêt du modèle à travers deux cas d'usage réels. La combinaison entre le modèle et la plateforme de chaîne de traitement a permis de répondre à des questions que se posent les géographes de notre projet. Nous avons pu voir que toutes les données manipulées s'intègrent dans le modèle et qu'il permet de représenter tous les cas particuliers spécifiques à notre contexte.

6 Conclusion et perspectives

Le projet DA3T a pour objectif de proposer des modèles, méthodes et outils dédiés au traitement de données de mobilité touristique au service de l'aménagement et de la valorisation des territoires de la côte atlantique. Dans cet article, nous nous sommes focalisés sur le modèle DA3T. Ce modèle intègre la notion de séquence d'épisodes multi-niveaux. Chaque épisode, disjoint ou imbriqué, est enrichi par des aspects sémantiques qui permettent de représenter les objets du monde réel à l'aide d'attributs de dimensions spatiale, temporelle et thématique. Le modèle permet ainsi d'associer plusieurs interprétations (c.-à-d. plusieurs séries d'épisodes sémantiques) à une trajectoire. Il est déjà implémenté dans la plateforme de traitement personnalisée DA3T où les données en entrée et en sortie de chaque service sont des instances du modèle. Nous l'avons testé et validé sur des données de mobilité en extérieur. Nous souhaitons maintenant le mettre en œuvre sur des données en intérieur correspondant à des visites de musées. De plus, nous expérimentons des services d'agrégation de trajectoires et de visiteurs comportant des similarités sémantiques (classes *Pattern* et *MobileObjectCategory*). Enfin, nous travaillons sur des services exploitant les entretiens des visiteurs pour enrichir leurs trajectoires.

Références

[1] M. Baglioni, J. Macedo, C. Renso, and M. Wachowicz. An Ontology-Based Approach for the Semantic Modelling and Reasoning on Trajectories. In *Advances in Conceptual Modeling – Challenges and Opportunities*, Lecture Notes in Computer Science, pages 344–353, Berlin, Heidelberg, 2008.

[2] C. Cayèré. Plateforme etl dédiée à l'analyse de lamobilité touristique dans une ville. In *Actes du Forum jeunes chercheuses jeunes chercheurs d'INFORSID*, pages 13–15, 2020.

[3] C. Cayèré, C. Faucher, C. Sallaberry, M.-N. Bessagnet, and P. Roose. Tools for processing digital trajectories of tourists. In *Mobile Data Management*, pages 232–233, June 2020.

[4] R. Fileto, C. May, C. Renso, N. Pelekis, D. Klein, and Y. Theodoridis. The Baquara2 knowledge-based framework for semantic enrichment and analysis of movement data. *Data & Knowledge Engineering*, 98 :104–122, July 2015.

[5] A. Frihida, D. Zheni, H. B. Ghezala, and C. Claramunt. Modeling Trajectories : A Spatio-Temporal Data Type Approach. In *2009 20th International Workshop on Database and Expert Systems Application*, pages 447–451, August 2009.

[6] R. D. Mello, V. Bogorny, L. O. Alvares, L. H. Z. Santana, C. A. Ferrero, A. A. Frozza, G. A. Schreiner, and

C. Renso. MASTER : A multiple aspect view on trajectories. *Transactions in GIS*, page tgis.12526, May 2019.

[7] T. P. Nogueira, R. B. Braga, C. T. de Oliveira, and H. Martin. FrameSTEP : A framework for annotating semantic trajectories based on episodes. *Expert Systems with Applications*, 92 :533–545, February 2018.

[8] C. Parent, S. Spaccapietra, C. Renso, G. L. Andrienko, N. V. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. A. F. de Macêdo, N. Pelekis, Y. Theodoridis, and Z. Yan. Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4) :42 :1–42 :32, 2013.

[9] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot. A conceptual view on trajectories. *Data & Knowledge Engineering*, 65(1) :126–146, April 2008.

[10] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. SeMiTri : A Framework for Semantic Annotation of Heterogeneous Trajectories. *EDBT 2011*, 2011.

[11] Z. Yan, J. Macêdo, C. Parent, and S. Spaccapietra. Trajectory Ontologies and Queries, 2008.

[12] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty. A Hybrid Model and Computing Platform for Spatio-semantic Trajectories. In *The Semantic Web : Research and Applications*, Lecture Notes in Computer Science, pages 60–75, Berlin, Heidelberg, 2010. Springer.

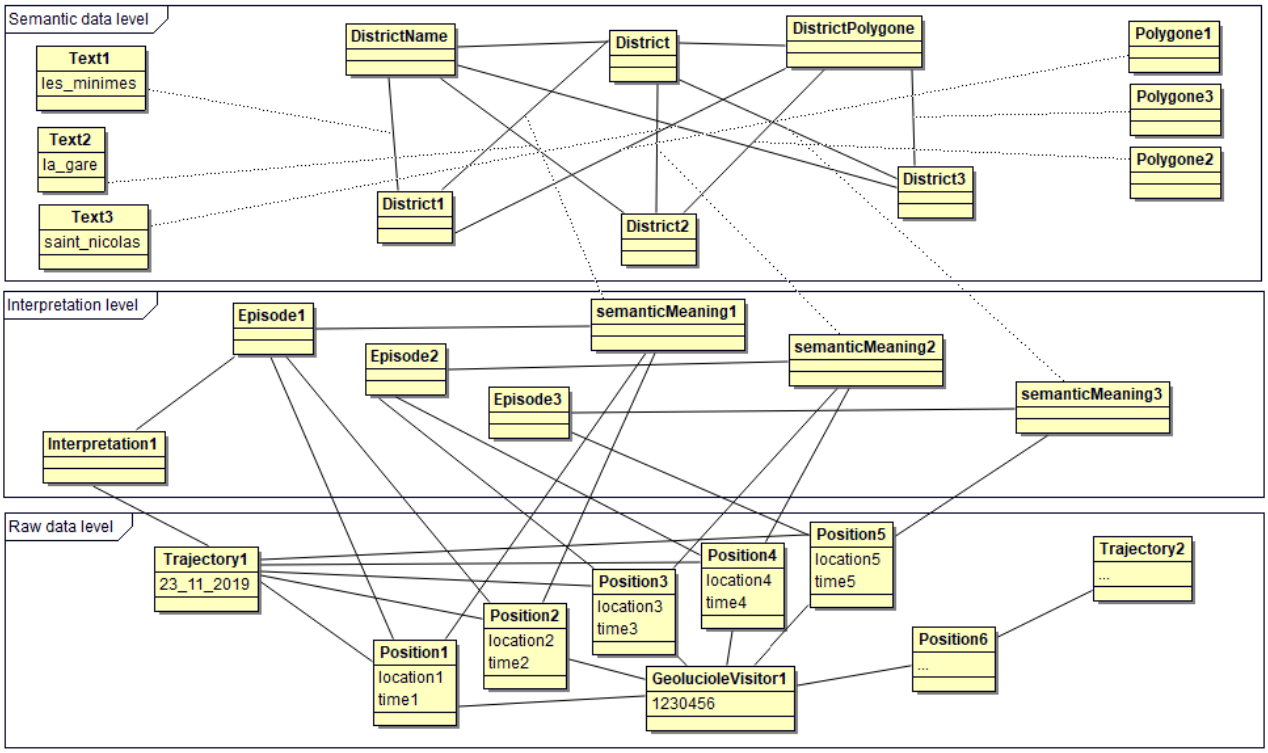

FIGURE 6 – Instanciation du modèle durant le processus de traitement pour répondre à la question (1)

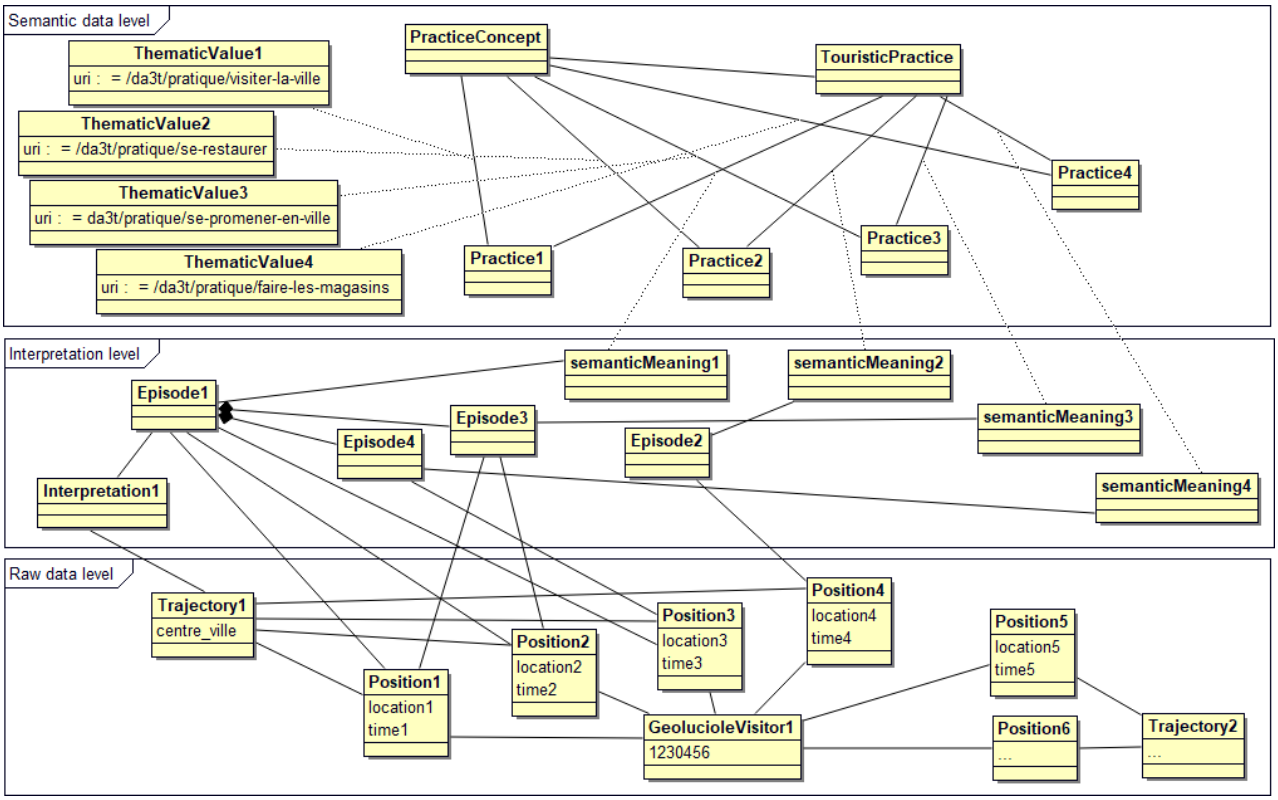

FIGURE 7 – Instanciation du modèle durant le processus de traitement pour répondre à la question (2) sans les liens entre les classes *Position* et *semanticMeaning* (par soucis de lisibilité)

Quelle place accorder aux objets abstraits dans les ontologies fondatrices ?

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 visons à éclaircir les fondements métaphysiques des ontologies en proposant d'assimiler les catégories ontologiques à des types d'objets abstraits de pensée. Pour préciser notre notion d'objet abstrait, nous nous dotons d'une théorie ontologique de l'intentionnalité en revenant aux sources de l'école brentanienne et, à cette occasion, nous réhabilitons l'état d'affaires abstrait. Nous reprenons alors nos travaux sur l'ontologie des événements pour identifier ces derniers à des états d'affaires abstraits. Sur la base de ces engagements, nous esquissons une nouvelle ontologie fondatrice.

Mots-clés

Ontologie fondatrice, réalisme conceptuel, intentionnalité, objet abstrait, objet concret, événement abstrait

Abstract

In this article, we aim to shed light on the metaphysical foundations of ontologies by proposing to equate ontological categories with types of abstract objects of thought. To clarify our notion of abstract object, we equip ourselves with an ontological theory of intentionality by returning to the sources of the Brentanian school and, on this occasion, we rehabilitate the abstract state of affairs. We then resume our work on the ontology of events to identify them as abstract states of affairs. On the basis of these commitments, we outline a new foundational ontology.

Keywords

Foundational ontology, conceptual realism, intentionality, abstract object, concrete object, abstract event

1 Introduction

Les ontologies développées en Ingénierie des Connaissances comme en Ontologie Appliquée se présentent communément sous la forme d'un catalogue de catégories structuré au moyen de liens de généralité (*subsomption*). Une approche courante, et fortement encouragée, pour établir ce catalogue consiste à se fonder sur des principes de l'Ontologie Formelle pour organiser le haut niveau de l'ontologie au moyen d'un ensemble de quelques catégories abstraites, cette collection étant nommée ontologie « fondatrice » (en anglais : *foundational*).

Les principes guidant la définition de ces ontologies fondatrices consistent à décider de la *portée* de l'ontologie (en termes de *domaines* couverts), du *grain* ou du *niveau de description* des entités considérées, mais aussi – et cette question doit être distinguée des précédentes – de *modes d'être* (ou d'*existence*) des entités. Traditionnellement les métaphysiciens distinguent trois modes d'*être* – physique, mental et social – cette tri-partition se déclinant en une bi-partition dès lors qu'on considère la pensée humaine comme pivot [23] : les entités physiques existent indépendamment de toute pensée humaine[1] ; les entités mentales et sociales dépendent au contraire de l'humanité, resp. d'un sujet unique et d'un collectif de sujets. Les termes « concret » et « abstrait » sont communément utilisés pour dénommer ces deux principales catégories d'entités. On notera du reste que, lorsqu'un métaphysicien dévoile une ontologie fondatrice, ce qui est le cas en Fig. 1 de Frédéric Nef [21, p. 78], le premier principe structurant consiste justement à distinguer les concrets des abstraits.

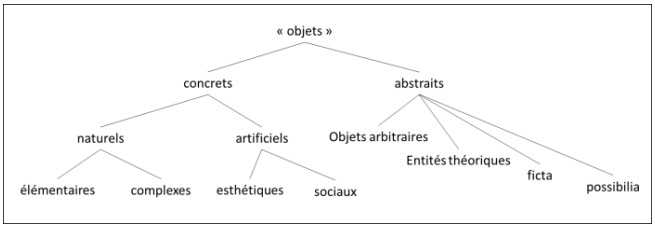

Fig. 1 : ontologie fondatrice (tiré de [21])

Comme on le voit donc en Fig. 1, les modes d'être sont priorisés dans l'organisation des catégories. En Ontologie Appliquée, en revanche, la pratique courante est inverse. Nous illustrons ce fait à l'aide des exemples d'ontologies fondatrices que sont DOLCE [16] et BFO [11].

Dans DOLCE, les entités abstraites, au sens où nous venons de les définir, semblent ressortir de deux catégories principales (*cf.* Fig. 2). D'une part, la catégorie `Non-Physical Endurant` (ou, plus spécifiquement, `Non-Physical Object`), subsumant justement les catégories `Mental Object` et `Social Object`. D'autre part, la catégorie `Abstract` couvrant des ensembles (`Set`), des faits (`Fact`) et des valeurs de qualités comme '250 grammes' (`Region`)[2]. De fait, le premier principe de structuration de l'ontologie revient à distinguer entre des continuants (`Endurant`) et des occurrents (`Perdurant`), ces entités ayant des qualités (`Quality`),

[1] Le terme « physique » est à entendre dans un sens large pour couvrir la strate des organismes.

[2] Cette catégorie `Region` couvre à son tour des régions temporelles, spatiales et des espaces conceptuels de valeurs de qualités (`Qualia`).

lesquelles ont des valeurs (`Qualia`).

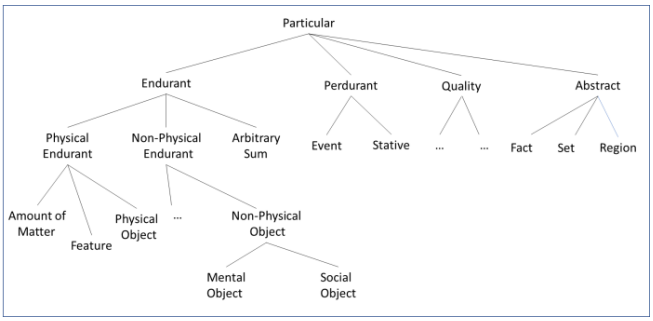

Fig. 2 : extrait de l'ontologie fondatrice DOLCE [16]

L'ontologie BFO, pour sa part, adopte, à l'instar de DOLCE, une distinction principale entre des continuants (`SNAP Entity`) et des occurrents (`SPAN Entity`) (*cf.* Fig. 3). Une différence importante par rapport à DOLCE, toutefois, est que les entités abstraites ne sont pas prises en compte, ce qui relève d'un choix délibéré des auteurs.

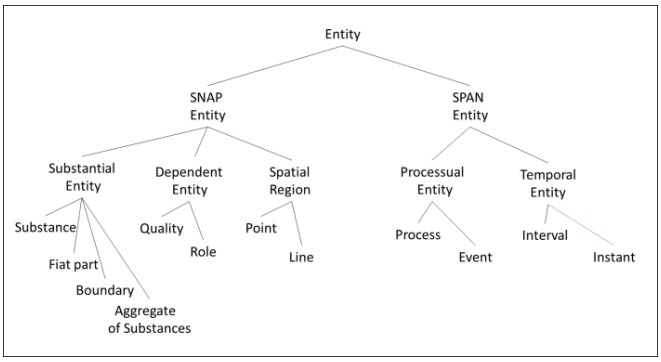

Fig. 3 : extrait de l'ontologie fondatrice BFO [11]

Ce choix, comme l'a très bien analysé Gary Merrill [19], repose sur plusieurs hypothèses, à savoir une thèse sémantique (TS), une thèse métaphysique (TM) et une doctrine méthodologique pour le développement d'ontologies qualifiées de « scientifiques », comme celles développées dans le domaine biomédical (DM) :

(TS) – Les termes scientifiques comme « lésion de la peau », « infection bactérienne », *etc.*, réfèrent *directement* à des entités réelles (concrètes).

(TM) – Les *universaux* existent réellement (concrètement).

(DM) – Les catégories ontologiques sont des universaux ; ces universaux sont identifiés par les experts des domaines concernés ; le rôle des ontologues est de recenser ces universaux et de les organiser en un catalogue structuré au moyen de liens de subsomption.

À l'instar de Merrill, et contre l'opinion des auteurs de BFO qui ont une nouvelle fois défendu ces thèses [28], nous choisissons d'endosser d'autres thèses. Ce choix est motivé par le fait d'attribuer aux catégories ontologiques une nature différente.
En premier lieu, nous constatons que les ontologies couramment développées, notamment dans le domaine biomédical[3], couvrent des domaines d'entités non physiques comme des comptes rendus d'hospitalisation, des protocoles de soin, des organisations de santé et des professions de santé. À ce propos, comme nous l'avons défendu dans des travaux récents, nous considérons que les *événements*, que ce soit des *états* (ex : 'le patient est fiévreux') ou des *changements d'états* (ex : 'la température du patient augmente'), sont des entités abstraites et non concrètes [12][14]. De fait, nous en concluons que les ontologies fondatrices doivent avoir pour portée à la fois des entités abstraites et concrètes, à l'instar de DOLCE. Mais, plus fondamentalement, nous posons la question de ce que représente la catégorie `Physical Object` de DOLCE et les sous-catégories lui étant habituellement rattachées, notamment des catégories d'artefacts comme `Table`, `Chaise`, *etc*. La réponse que nous apportons est que ces entités ne sont pas des objets physiques *simpliciter* mais des *représentations* d'objets physiques auxquels des propriétés sociales comme une *fonction* leur sont attribuées. Il convient donc de les identifier à des objets abstraits *représentant* des objets physiques. De fait, nous formulons une proposition prenant le contre-pied des thèses des auteurs de BFO :

(TS)' – les termes (notamment scientifiques) réfèrent *indirectement* à des entités réelles concrètes ; la référence est médiatisée par des objets abstraits de pensée.

(TM)' – Les objets abstraits de pensée *représentent* des particuliers concrets

(DM)' – Les catégories ontologiques sont des types abstraits d'objets de pensée ; ces objets abstraits sont construits par les experts des domaines concernés ; le rôle des ontologues est de les recenser et de les organiser au moyen de liens de subsomption.

Dans cet article, nous posons les bases d'une théorie des objets abstraits supportant les thèses venant d'être énoncées. À cette fin, nous nous référons à des travaux menés au tournant du 20ème siècle en psychologie et philosophie portant sur l'*intentionnalité*. Nous revenons aux sources de l'école brentanienne pour établir une conception de l'objet *inexistent* comme *être représenté* et nous réhabilitons l'*état d'affaires abstrait* comme entité infra-propositionnelle (Section 2). Par la suite, en prolongation de nos travaux dans le domaine des entités occurrentes, distinguant entre processus physiques concrets et événements abstraits, nous identifions états d'affaires et événements abstraits (Section 3). En Conclusion, nous esquissons une nouvelle ontologie fondatrice

2 Abstraits *vs* concrets

Dans cette section, nous nous dotons d'un cadre ontologique général pour rendre compte de phénomènes intentionnels, autrement dit d'actes ou d'états de pensée dirigés vers un objet. Ce cadre, nous le posons d'emblée, est un modèle à 4 termes : *acte* / *contenu mental* / *objet mental* / *chose(s) externe(s)*. Pour en préciser la signification, nous remontons aux théories avancées au tournant du 20ème siècle par Franz Brentano et ses

[3] Un grand nombre de telles ontologies peuvent être consultées sur le Bioportal : https://bioportal.bioontology.org/

élèves, tout particulièrement Kasimir Twardowski et Alexius Meinong[4]. Ces théories comportent des thèses psychologique et ontologique. Pour notre propos, nous privilégions la dimension ontologique.

2.1. La doctrine ontologique de l'intentionnalité chez Brentano

La théorie de l'intentionnalité de Brentano, celle du jeune Brentano de la *Psychologie du point de vue empirique* [5], a fait l'objet de nombreuses interprétations. Si l'on s'en tient aux faits couramment admis[5], le jeune Brentano (mais également le Brentano réiste d'après 1911) a défendu un modèle à 3 termes : *acte* / *« contenu-objet » mental* / *chose(s) externe(s)*[6]. Rappelons que la motivation de Brentano était avant tout de rendre compte de phénomènes (ou actes) de pensée visant des choses effectives existantes (la question des référents non-existants viendra ultérieurement). Prenons justement comme exemple d'acte mental la *présentation* (*Vorstellung*) par un sujet d'un objet matériel physique – le sujet *se représente* un objet physique. Selon Brentano, l'essence de ce phénomène (en tant qu'acte de pensée) est d'être dirigé vers un objet immanent, à la fois interne et inséparable de l'acte. Dans le même temps, toutefois, l'acte est également dirigé vers l'entité effective – l'objet matériel physique. Pour accéder plus complètement à l'ontologie de l'intentionnalité de Brentano, il est primordial de se rappeler ce passage fameux [5, p. 33] :

> Les phénomènes qu'il [le physicien] étudie et qui concernent la lumière, le son, la chaleur, le lieu, le mouvement local n'ont pas d'existence véritable (…). Ils constituent les signes d'une réalité effective dont l'action produit leur représentation. Mais l'image qu'ils en donnent ne correspond aucunement à cette réalité, et la connaissance qu'on en peut tirer demeure bien imparfaite. Nous pouvons dire qu'il existe quelque chose qui, dans telles ou telles conditions, devient la cause de telle ou telle sensation ; nous pouvons également démontrer qu'il doit s'y rencontrer des relations analogues à celles que représentent les manifestations spatiales, les grandeurs et les formes. Mais il faut s'en tenir là. La vérité des phénomènes physiques n'est, suivant l'expression consacrée, qu'une vérité relative.

Ce paragraphe est important pour comprendre le réalisme et la notion brentanienne de « donné » (pour emprunter un terme contemporain). Le réel physique extérieur à la conscience d'un sujet ne lui est pas donné directement. Seules lui sont données, dans des expériences de perception externe, des manifestations ou signes du réel sous la forme d'impressions sensibles. Ces impressions résultent causalement d'une activité du réel physique. C'est alors l'interprétation de ces signes qui conduit le sujet à se construire une représentation du monde réel physique. Deux thèses caractérisent ainsi le réalisme de Brentano. D'une part, l'objet immanent à l'acte intentionnel jouit d'un véritable statut ontologique[7]. D'autre part, l'objet immanent possède, à titre de référence de l'acte, un corrélat effectif mais dont le sujet n'a qu'une connaissance imparfaite.

Examinons plus en détail le lien existant entre l'objet immanent et le corrélat effectif de l'acte intentionnel. Sur ce point, il semble que Brentano ait oscillé entre deux conceptions. Selon Chrudzimski [*ibid.*], Brentano, dans sa *Habilitationsschrift* (1867) dédiée à la philosophie de l'esprit chez Aristote, décrit une doctrine *représentationnelle* des objets immanents consistant à les doter de propriétés distinctes de (et représentant) celles de l'objet concret de référence. Ultérieurement, à l'occasion de cours donnés à Vienne entre 1880 et 1890, Brentano testera auprès de ses étudiants une doctrine (plus connue) *présentationnelle* consistant à doter l'objet immanent de la même propriété que celle de l'objet de référence, dans une sorte de détachement de la forme aristotélicienne[8].

Pour notre part, nous optons pour une conception *représentationnelle* de l'objet immanent, en phase avec le passage cité *supra*. Brentano rompt avec le réalisme naïf d'Aristote, mais tout en gardant la figure de la substance aristotélicienne comme référence. Simplement, celle-ci devient le modèle abstrait de la chose réelle, plutôt que la chose elle-même qui verrait sa forme détachée. L'objet mental étant construit, celui-ci porte ses propriétés différemment de son corrélat effectif. Ses propriétés sont « encodées » (pour utiliser un terme contemporain), ce qui souligne que le fait d'« avoir » des propriétés relève d'une construction mentale.

2.2. L'*être représenté* chez Twardowski

Venons-en à Twardowski, dont la théorie de l'intentionnalité nous est connue principalement par son essai *Sur la théorie du contenu et de l'objet des représentations* [30]. La motivation première de Twardowski est de rendre compte des

[4] L'école de Brentano a joué un rôle important dans l'élaboration des théories philosophiques contemporaines du sens et de la référence. Notre démarche dans cet article est motivée par la conviction que les choix théoriques effectués par les chercheurs susmentionnés conservent toute leur pertinence et que ce leg peut être repris, au moins en partie. La taille de l'article ne permettant pas d'évoquer des théories compétitives, nous renvoyons le lecteur intéressé aux contributions rassemblées par Jocelyn Benoist dans [3].

[5] *Cf.* notamment Barry Smith [27], Jocelyn Benoist [2] et Alexius Chrudzimski [6], tous tenants de la théorie ontologique « standard » de l'intentionnalité de Brentano.

[6] La mise entre guillemets du terme 'contenu-objet' souligne les questionnements que Brentano gardera sa vie durant sur la nature de cette entité. Nous la désignerons par le terme « objet immanent » dans la suite de cette section.

[7] On peut rapprocher l'objet mental de pensée de Brentano de l'*être simpliciter* de Bertrand Russell [24, §427] : « There is only one kind of being, namely *being simpliciter*, and only one kind of existence, namely, *existence simpliciter*. Being is that which belongs to every conceivable term, to every possible object of thought… Numbers, the Homeric gods, relations, chimeras, and four-dimensional spaces all have being, for if they were not entities of a kind, we could make no propositions about them… For what does not exist must be something, or it would be meaningless to deny its existence; and hence we need the concept of being, as that which belongs even to the non-existent ». Ce parallèle est d'autant plus intéressant à dresser que, comme nous le rappellerons infra, les deux philosophes reviendront sur les largesses supposées de leurs ontologies pour en réduire, chacun à sa manière, la portée.

[8] Toujours selon Chrudzimski [*ibid.*], Brentano n'ayant pas publié cette seconde doctrine, visiblement considérée comme une hypothèse de travail à laquelle il finira par renoncer, pourra s'en défaire dans ses écrits postérieurs à 1904.

représentations *anobjectuelles* (ne possédant pas de référence effective) bolzaniennes, comme ces représentations comportant des déterminations contradictoires [le carré rond] ou ne se référant à aucune entité rencontrée jusqu'à présent [la montagne d'or]. La stratégie twardowskienne est alors d'admettre l'existence d'objets « donnés » par ces représentations, au motif qu'ils soient les supports de déterminations, pouvant être éventuellement contradictoires. Dans le cas des représentations anobjectuelles, une détermination de ces objets est justement de *ne pas exister*, au sens où il ne leur correspond pas de chose concrète existante. En élevant au rang d'objet pensé (l'équivalent de l'être *simpliciter* russellien) ces objets impossibles, ou non encore rencontrés, Twardowski les fait participer de phénomènes d'intentionnalité et propose dès lors que chaque représentation comporte à la fois un contenu et un objet[9]. L'acte de pensée-représentation est nécessairement dirigé vers un objet mental immanent (nous dirons qu'il 'a pour sujet' cet objet), et peut être également dirigé vers une chose effective (nous dirons qu'il 'a pour référence' cette chose). La distinction entre contenu et objet se fonde sur une ontologie de la représentation : un objet est « donné » ou « représenté » par un contenu ; *être représenté* est un mode d'existence qui se distingue d'*être effectivement*.

La nomination, avec son exigence de référentialité, a servi de modèle à Twardowski. En utilisant un nom, par exemple « le vainqueur de Iéna », un allocutaire indique se représenter un contenu [le vainqueur de Iéna] et désigner une chose, en l'occurrence une personne. En distinguant contenu et objet de référence, Twardowski retrouve le phénomène sémantique de la synonymie avec le fait qu'un contenu différent, par exemple [le vaincu de Waterloo], puisse co-référer à un même objet externe. En distinguant plus finement contenu et sujet de la représentation, le modèle permet de rendre compte du fait que l'allocutaire soit ignorant de cette identité de référence – les objets sujets sont pour lui sans lien.

Une métaphore intéressante que Twardowski utilise pour caractériser son ontologie de la représentation est celle du tableau. En peignant un paysage, l'artiste réalise un acte dont le résultat est un objet matériel physique, un tableau constitué d'une toile et de peinture. À la question d'identifier l'« objet peint », au sens de l'objet représenté par la peinture, deux objets distincts sont candidats. D'une part, le paysage réel servant de référence. Mais celui-ci peut très bien ne pas exister, auquel cas, pour poursuivre la métaphore, nous avons affaire à un tableau *anobjectuel*. D'autre part, le paysage tel que présenté par le tableau, autrement dit tel que donné par les formes de peinture inscrites sur la toile, celles-ci jouant le rôle de contenu. On notera que ce dernier paysage – sujet du tableau - peut ne correspondre à aucun paysage rencontré jusqu'à présent (par exemple s'il comporte une montagne d'or) ou peut révéler des objets impossibles (comme dans le cas de lithographies de Maurits Escher). Cette métaphore du tableau illustre bien la distinction du paysage *sujet* et du paysage de *référence*.

Chez Twardowski, l'objet immanent brentanien est ainsi défini comme un *être représenté* et, concomitamment, son domaine s'élargit pour couvrir, outre les réels, les *irrealia* (*ficta* ou chimères) mais aussi, comme nous venons de le rappeler, les objets impossibles. À côté de ces *simples*, une extension requiert tout particulièrement notre attention, celle de *complexes* ou états d'affaires (*Sachverhalt*)[10]. Leur reconnaissance tient à une analyse que Twardowski en vient progressivement à promouvoir concernant les jugements relationnels exprimés par des phrases comme « la pomme est mûre » ou « Paul salue Marie ». Selon Twardowski, ces jugements (leurs contenus) portent sur un objet principal, resp. 'l'être mûr de la pomme' et 'la salutation de Paul à Marie'. Ces états d'affaires s'avèrent distincts de relations, mais également de propositions car ils ne portent pas de valeur de vérité (et, de fait, les propositions [la pomme est mûre] et [la pomme n'est pas mûre] ont le même état d'affaires comme objet principal.

Meinong, dans sa théorie de l'*objet général* suivra Twardowski dans cet élargissement de l'objet, y compris dans la reconnaissance de l'état d'affaires et selon la même caractérisation[11]. Dans sa publication *Sur les objets d'ordre supérieur et leur rapport à la perception interne* [17], Meinong, faisant état de réflexions analogues menées par Twardowski, développe ainsi une théorie de complexes correspondant à des objets *superiora* fondés unilatéralement sur des objets *inferiora*, une théorie (Meinong insiste) permise grâce à la distinction entre contenus et objets. Par la suite, dans son *On assumptions* [18], Meinong attribuera à de tels complexes le fait de jouer le rôle d'objet principal de jugements en les distinguant, comme l'avait fait Twardowski, de la proposition

[9] Cet octroi du statut d'objet pensé aux objets impossibles, repris par Meinong, sera fortement critiqué par Russell dans son *On denoting* [25], au motif de violer la sacro-sainte loi logique de contradiction et, *in fine*, de saper toute forme de pensée. Dans la stratégie russellienne pour rétablir le primat de la logique vis-à-vis de l'ontologie, par contre, le prix à payer est une analyse non-intuitive de la structure logique de phrases. *Cf.* Frédéric Nef, [21, pp. 99-101 et pp. 149-151] ; *cf.*, également, Benoist, [2, chap. V, § *Russell : le détour par la syntaxe*].

[10] Arianna Betti, dans son *Propositions et états de choses chez Twardowski* [4], nous indique que Twardowski a commencé à élaborer une théorie de l'état de choses dans son [30], théorie qu'il a complétée à l'occasion d'un cours de logique qu'il a donné à Vienne à l'hiver 1894-1895 et dont les notes ont été préservées. Betti, toutefois, prend le parti d'identifier le *Sachverhalt* twardowskien à un objet idéal réticulant des choses réelles. Au contraire, Smith [*ibid.*, chap. 6 *Kasimir Twardowski On content and object*, §4 *Sachverhalt vs Judgment-Content: Immanence and Idealism*] prend soin de rappeler que le *Sachverhalt* introduit par Twardowski est une entité mentale. Ce n'est que plus tard, sous l'influence de Husserl, que le philosophe polonais rompra avec le psychologisme. Pour notre part, nous conservons l'interprétation mentaliste originelle. Pour éviter toute confusion, nous préférons le terme « état d'affaires » à « état de choses », qui a une connotation concrète.

[11] Comme le rappelle Nef [*ibid.*, pp. 156-159], Russell dénoncera de son côté l'existence de ces états d'affaires abstraits dans son *On the Nature of Truth* [27], au motif de ne pouvoir établir une conception de la vérité fondée sur une correspondance entre ces complexes abstraits et des faits concrets. Selon Russell, si l'exemple du complexe 'la mort de Charles 1er sur l'échafaud' peut laisser imaginer un lien avec un fait concret, en revanche le complexe 'la mort de Charles 1er dans son lit' ne laisse guère d'espoir d'établir une correspondance avec des faits concrets. De fait, pour Russell, une proposition ne peut être fausse et l'argument demeure toujours de ne pouvoir accorder d'existence à des pensées sans référence concrète. Dans le présent article, nous nous apprêtons à élargir la notion de *correspondance*, ou de *représentation*.

(l'*Objectiv* pour Meinong)[12].
Dans cet article, nous n'irons pas plus loin dans l'analyse de ces travaux et nous prévenons le lecteur que, bien nous ayons mentionné Meinong, nous ne nous engageons pas vis-à-vis des positions ontologiques que Meinong développera ultérieurement. Nous résumons donc les choix théoriques que nous souhaitons retenir.

En résumé, nous distinguons dans notre cadre ontologique deux types d'être (d'existence), *être pensé* et *être effectif*, et nous qualifions d'objets abstraits et d'objets concrets les deux catégories d'entités relevant de ces modes d'existence respectifs. Nous concevons *être pensé* comme *être représenté* et, concernant le phénomène de représentation, nous distinguons l'acte ou l'état *occurrent* (l'épisode de pensée) de son contenu et de son objet *continuants*. La distinction entre objets abstraits et concrets s'accompagne d'une distinction entre deux catégories de propriétés/relations, que nous nommons momentanément propriétés *conceptuelles* et *ordinaires*.
Dans la suite de l'article, nous complétons ce cadre ontologique. Les directions ne manquent pas ! Rappelons que notre motivation est d'évaluer la place à accorder aux objets abstraits dans les ontologies fondatrices et, de ce fait, d'étudier la complémentarité entre objets abstraits et concrets. Dans cette optique, nous allons prioriser l'analyse du changement d'objets concrets et ceci nous va donner l'occasion de mettre en scène les événements abstraits.

3 Processus concrets et événements abstraits

— Pour affiner le cadre ontologique général que nous venons d'adopter, et poursuivre notre enquête sur la place à accorder aux objets abstraits, nous nous tournons maintenant vers le domaine des entités qualifiées de « survenantes » ou « occurrentes ». À ce propos, nous avons rappelé en Introduction que l'opposition *continuant* vs *occurrent* constitue un principe structurant premier pour les ontologies BFO et DOLCE. Récemment, nous avons proposé que les *événements*, habituellement considérés comme des entités concrètes, soient au contraire identifiés à des entités abstraites [12][14]. Cette proposition s'accompagne de la mise en scène d'autres primitives ontologiques, notamment des *processus physiques* et des *liens de connexion physiques*. Nous avons ainsi postulé un monde structuré comme suit :

(i) Le monde concret est notamment peuplé d'*objets* et de *processus* ;
(ii) Ces entités concrètes endurantes ont une vie consistant en des *connexions* se créant et se défaisant dans le temps ;
(iii) Des sujets pensant, immergés dans le monde, se construisent des représentations pour interagir avec lui ; parmi ces représentations figurent des *événements* rendant compte, pour ces sujets, de l'histoire du monde.

Dans cette section, nous comptons reprendre ces thèses. Comme elles ont déjà fait l'objet de présentations lors de précédentes journées IC, nous nous contenterons d'en rappeler les principales idées, sans chercher à les défendre[13]. À cette fin, nous considérons à titre d'illustration l'exemple du changement temporel d'objets physiques. L'élément nouveau que nous comptons ajouter (en §3.3), est de montrer que les *états d'affaires abstraits* définis en §2 constituent un cadre naturel pour accueillir les événements abstraits.
Considérons donc le changement temporel, communément défini comme le fait qu'une substance 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 instant T_2. Selon Peter Geach [10], cette simple caractérisation soulève d'emblée un problème métaphysique complexe. Intuitivement, selon le choix de *F*, on peut distinguer de *vrais* et de *faux* changements : le fait qu'un morceau de beurre fonde paraît correspondre à un vrai changement, au contraire du fait que le prix d'une plaquette de beurre augmente. Dans le second cas, la raison que l'on peut avancer est que les propriétés physiques de l'objet ne sont pas impliquées causalement dans le changement. Selon Geach, seul le vrai changement existe vraiment (est une entité concrète)[14].
Pour tâcher de caractériser les vrais changements, tournons-nous vers le mouvement d'un corps physique. Sur ce terrain, un autre problème nous attend, identifié antérieurement au précédent. Cette espèce de changement a été définie au début du 20ème siècle par Russell ainsi [24, §442] : « *Motion is the occupation, by one entity, of a continuous series of places at a continuous series of times* ». Le problème bien connu de cette caractérisation est de traiter le mouvement comme s'il n'était fait que d'immobilités : la dynamique du mouvement n'est pas prise en compte. Cette fois, des données récentes de la métaphysique des processus et des événements permettent d'apporter une réponse.

3.1. Processus physiques

L'enjeu, pour rendre compte de la dynamique du mouvement,

[12] Dans [18, §8 *Judged Objectives*], Meinong affirme : « If someone says, e.g., in regard to a parliamentary election that was preceded by intense public excitement, that no disturbance of the peace took place, then in the first place no one will deny that "something" is known by means of the judgment in question – assuming that it is correct. Yet one might at the outset suppose that this "something" is nothing but the object thought of by the one who meaningfully expresses that judgment – or in other words, the object "disturbance of the peace" ». L'objet 'Disturbance of the peace' est ainsi l'objet principal de l'objectif 'that no disturbance of the peace has occurred'. L'interprétation que nous donnerons plus loin est que les expressions « took place » et « has occurred » tiennent pour des propriétés de l'état d'affaires.

[13] Le lecteur souhaitant en avoir une présentation détaillée peut se référer à [13] et [15].

[14] Faute d'être capable d'exhiber un critère métaphysique permettant de distinguer entre vrais et faux changements, Geach se contentera de nommer « propriétés de Cambridge » (en référence aux positions des philosophes de Cambridge McTaggart et Russell) les propriétés mobilisées dans de faux changements, eux-mêmes qualifiés à leur tour de « changements de Cambridge ». Qu'il s'agisse de vrais changements ou de changements de Cambridge, contrairement à Geach, comme nous allons la rappeler, nous leur conférons une existence abstraite.

est d'expliquer comment il est possible à un objet d'entrer et de sortir d'une position. Nous suivons ici la proposition de la philosophe Carol Cleland de faire appel à la notion d'*effort*, au cœur de la physique newtonienne [7]. Selon Cleland, convoquant à la suite de Newton l'expérience consistant à faire tourner autour de soi un objet maintenu par une ficelle, 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 et lui conférer une place dans notre inventaire ontologique [*ibid.*, p. 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.

Ce processus de mouvement, selon Rowland Stout [29], endure dans le temps à la manière de l'objet en étant pleinement présent à différents moments [*ibid.*, p. 26] :

> The phrase, 'What is happening now', is naturally taken to denote a whole process; and we do want to claim that what is happening now is literally identical with what is happening at some other time – the very same process.

Qui plus est, selon Anthony Galton [8], le processus peut lui-même changer en portant des propriétés différentes à des temps différents [*ibid.*, p. 6] :

> Like objects, processes can change: the walking can get faster, or change direction, or become limping. All around us processes undergo changes: the rattling in the car becomes louder, or changes rhythm, or may stop, only to start again later. The flow of the river becomes turbulent; the wind veers to the north-west.

Suivant ces auteurs, la caractérisation du vrai changement passe ainsi par l'introduction d'un nouvel endurant, à côté de l'objet physique, le *processus physique*. Ajoutons que, lorsqu'ils se manifestent, les processus nous apparaissent ancrés dans des objets : il s'agit du processus de mouvement d'*une balle*, du mûrissement d'*un fruit*, de la fonte d'*un glacier*, de l'oxydation d'*une pièce de métal*, *etc*[15]. Galton et Riichiro Mizoguchi [9] rendent compte de ce lien fort entre objets et processus au moyen de la relation d'*énaction* définie ainsi : un objet *énacte* contingentement un processus quand il est le siège du processus et lorsque ce dernier, par son activité, induit un réel changement de l'objet.

3.2. Faits de connexion concrets

Suivant la conception du processus que nous venons d'arrêter, celui-ci est causalement responsable d'effets manifestés par l'objet l'énactant. Reprenons notre exemple de mouvement. Lorsqu'un objet se meut, un processus de mouvement est responsable du fait que l'objet change de localisation et se trouve occuper des positions différentes à différents temps. Nous avons ainsi affaire à une série de faits de localisations :

> <Loc, O, Pos_1, I_1>, <Loc, O, Pos_2, I_2>, … ; chaque fait, représenté entre crochets <,,,>, correspond à l'occupation *Loc* par l'objet *O* d'une position Pos_i à un instant I_i.

Par ailleurs, ayant admis qu'un processus peut lui-même changer, par exemple qu'un processus de marche peut s'accélérer, ceci nous conduit à considérer une série de faits de vitesses :

> <Inhère, $Marcher_{Paul}$, $Vitesse_1$, I_1>, <Inhère, $Marcher_{Paul}$, $Vitesse_2$, I_2>, … ; chaque fait correspond à l'inhérence *Inhère* au processus $Marcher_{Paul}$ d'une vitesse $Vitesse_i$ à un instant I_i.

La conception des faits que nous venons d'adopter repose sur de nouveaux engagements ontologiques, le premier d'entre eux étant, bien sûr, celui de l'existence concrète attribuée à de tels faits. Cette thèse de l'existence de faits concrets est à rapprocher de celle de David Armstrong [1]. Vis-à-vis des faits armstrongiens, nous soulignons plusieurs différences importantes.

En premier lieu, nous limitons le lien entre les constituants (représentés dans notre notation en deuxième et troisième position) à un lien de « connexion ». Par lien de connexion, précédemment nommé propriété/relation ordinaire, nous entendons l'opération d'un principe physique liant des entités concrètes dont l'existence concomitante est nécessaire (dans notre exemple de faits de localisation, lesdites entités sont un objet physique et une région spatiale). Le lien de connexion est ainsi de nature différente de celle de la propriété conceptuelle dont les *relata* sont des objets abstraits – ce qui permet que les corrélats concrets représentés ne soient pas présents lorsque la relation tient[16]. Le fait de restreindre nos faits concrets à des liens de connexion physiques nous rapproche d'une proposition faite par Kevin Mulligan, Peter Simons et Barry Smith [20], reposant sur une interprétation nouvelle de l'*état de choses* défini par Ludwig Wittgenstein dans son *Tractatus* [31].

> 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.

La proposition de Mulligan *et coll.* est celle d'états de choses structurés au moyen de relations de *fondation* (des relations de dépendance existentielle). À l'instar de ces auteurs, nous considérons cette proposition comme un chalenge à relever.

Par ailleurs, nous considérons des faits temporisés (*tensed*), en indexant les liens de connexion sur le temps et en optant pour des *instants* de temps[17]. La conséquence de ce choix est que de

[15] Il convient de noter que cet ancrage que nous évoquons ne signifie pas que le processus existe *dans* ces objets. Des exemples de processus sont des interactions (électromagnétiques ou gravitationnelles) impliquant plusieurs objets. L'ancrage correspond au fait qu'un effet d'un processus se manifeste par un changement d'un objet. À ce propos, signalons que des processus peuvent exister mais voir leurs effets se contrarier, comme lorsqu'en poussant une porte celle-ci me résiste en demeurant immobile.

[16] La distinction entre *relations* et *connexions* est un travail figurant à l'agenda de la recherche contemporaine en métaphysique. Le lecteur intéressé peut se référer à l'ouvrage de Frédéric Nef [22], dans lequel est esquissée une métaphysique des connexions.

[17] Comme rappelé dans [14], une telle option est courante dans les théories dénommées 'at at'. Allant au-delà, nous avons défendu dans [14] une théorie *présentiste* du temps consistant à considérer que seuls des *instants* d'une durée non nulle existent réellement. Cet engagement

tels faits n'endurent qu'un instant. On notera que cette existence fugace de faits est en parfaite cohérence avec l'idée intuitive que l'on peut se faire du changement : les changements (notamment continus) dans le monde sont à mettre au crédit des faits disparaissant et apparaissant instantanément.

3.3. Evénements abstraits

Revenons au mouvement pour en déterminer la nature. En identifiant le mouvement à une série de faits de localisations distinctes successives, nous en avons fait une entité étendue dans le temps. Les événements en général, contrairement aux processus, sont communément considérés comme existant en accumulant des parties temporelles (selon la théorie du *perdurantisme*). L'engagement que nous avons pris vis-à-vis du temps – seuls existent réellement des instants – nous empêche d'accueillir le mouvement dans notre inventaire des entités concrètes. Par contre, rien ne nous empêche, si ce n'est la transgression d'une tradition bien ancrée en métaphysique contemporaine, de leur accorder une existence abstraite. Ce geste nous conduit à distinguer, pour ce qui les concerne, les propriétés d'*existence* et d'*occurrence*, malheureusement communément confondues.

Considérons l'exemple d'une personne – Paul – marchant. Comme défini en §3.1 et §3.2, Paul énacte un processus de marche – nommons le 'Marcher$_{\text{Proc}}$' – l'amenant à se déplacer en occupant successivement des positions distinctes. Envisageons maintenant de décrire ce que fait (ou a fait) Paul, en lien avec sa marche, sur une certaine période de temps. Ceci peut nous conduire à penser à 'la marche de Paul jusqu'à la gare ce matin'. Ce faisant, nous conférons une existence à une nouvelle entité – nommons la 'Marche$_{\text{Évén}}$'. L'engagement ontologique que nous prenons consiste à considérer que cette narration, histoire ou encore – événement – est un construit psychologique, existant mentalement dans la tête du sujet le pensant (et n'existant que mentalement !).

Entre processus et événements s'instaure une double relation : d'un côté, plusieurs narrations à partir de faits impliquant un même processus peuvent être produites (ex : 'la marche vivifiante de Paul ce matin', 'les dix premières minutes de la marche de Paul ce matin') ; d'un autre côté, un événement tel 'Marche$_{\text{Évén}}$' a pu donner lieu à plusieurs processus comme 'Marche$_{\text{Proc}}$', notamment si Paul a flâné sur le trajet en s'interrompant et en reprenant sa marche. Dans les exemples que nous venons de considérer sont évoqués un (ou des) événement(s) se référant à un objet énactant un (ou plusieurs) processus, ce qui correspond à de vrais changements au sens de Geach. Mais il est des événements pour lesquels l'objet de référence n'énacte pas de processus, comme lorsque l'objet est transporté (l'objet est alors un composant d'un système énactant globalement un processus de mouvement, sans que l'objet exerce lui-même d'activité). Enfin, nous pouvons citer des mouvements (apparents, justement !) ne reflétant pas l'activité de l'objet, comme pour le lever ou le coucher du soleil.

L'événement ainsi défini peut être rapproché de l'état d'affaires abstrait – objet complexe de représentation tel qu'envisagé notamment par Twardowski et Meinong (§2). On se convaincra sans hésitation que 'Marche$_{\text{Évén}}$' joue un rôle de représentation du monde. Toutefois, la notion de *représentation* ici mobilisée doit être entendue dans un sens plus large qu'une relation entre deux *relata*, puisque, selon nos engagements ontologiques, aucun événement concret n'existe. Les jugements exprimés par des phrases comme « Paul marche » ou « Paul salue Marie » – que nous analysons logiquement comme [*la marche de Paul* occurre] et [*la salutation de Paul à Marie* occurre] – nécessitent d'évaluer, non l'existence d'une entité concrète, mais celle de plusieurs faits à différents instants. Nous évaluons ainsi si l'histoire (l'état d'affaires) a effectivement pris place sur une période de temps excédant l'instant.

Nous définissons dès lors la propriété d'*occurrer*, ou de *survenir*, ainsi : un événement *occurre* à un temps *t* ssi les faits correspondant aux conditions de satisfaction de l'événement tiennent au temps *t*.

4 Conclusion

Dans cet article, nous avons pris plusieurs engagements ontologiques nous conduisant à esquisser une nouvelle ontologie fondatrice (*cf.* Fig. 4). Par « nouvelle », nous signifions que le réalisme sous-tendant cette ontologie est différent de celui qui a présidé à la définition d'ontologies fondatrices comme BFO, et même DOLCE.

La différence essentielle tient à la nature revendiquée des catégories ontologiques. Nous les identifions à des *types d'objets abstraits* de pensée, et non à des *universaux* concrets (comme défendu par Barry Smith). On peut dès lors parler d'ontologie « conceptuelle » ou « épistémique ». Ces objets abstraits sont des représentations du monde, des moyens que nous utilisons pour viser *indirectement* le monde.

Suivant cette fois la nature des objets, concrets ou abstraits, nous avons identifié deux modalités de *représentation*. Les objets concrets représentent des corrélats-choses dans une relation 1:1. Les objets abstraits, en tout cas les événements, représentent des collections de corrélats-états de choses (des liens de connexion) lesquels tiennent à différents instants, dans une relation 1:n. Les événements représentent ainsi l'histoire du monde en nous renseignant sur les stabilités (états) et changements du monde. Ils ne représentent pas des événements concrets : le monde physique et ses narrations par des sujets ne participent pas de la même strate réelle.

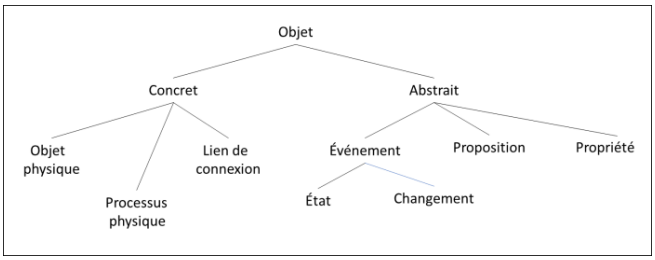

Fig. 4 : esquisse d'une nouvelle ontologie fondatrice

Le lecteur notera que nous n'avons pas retenu la distinction *continuant* vs *occurrent* comme principe structurant. Suivant en effet nos engagements ontologiques, toutes les entités sont des *continuants*, c'est-à-dire des entités existant pleinement dans le

n'étant pas pertinent pour le présent article, nous ne le développons pas.

temps (au sens d'une pleine identité), ces entités venant à exister et pouvant cesser d'exister.
Le lecteur aura par ailleurs noté que notre esquisse d'ontologie fondatrice repose sur des travaux contemporains en métaphysique, comme la théorie des propriétés et celle des faits. Ce sont autant de chantiers sur lesquels nous comptons à l'avenir avancer pour évaluer et affiner notre proposition.

5 Références

[1] D.M. Armstrong, *A World of States of Affairs*, Cambridge University Press, 1997.

[2] J. Benoist, *Représentations sans objet : aux origines de la phénoménologie et de la philosophie analytique*, Paris, PUF, collection « Epiméthée », 2001.

[3] J. Benoist (éd.), *Propositions et états de choses : entre être et sens*, Librairie Philosophique J. Vrin, Paris, 2006.

[4] A. Betti, Propositions et états de choses chez Twardowski, *Dialogue*, vol. 14, pp. 469-92, 2005.

[5] F. Brentano, *Psychologie du point de vue empirique*, Librairie Philosophique J. Vrin, Paris, 2008 ; trad., par M. de Gandillac, de *Psychologie vom empirischen Standpunkt*, vol. I, O. Kraus (ed.), Leipzig: Meiner, 1874.

[6] A. Chrudzimski, Brentano and Aristotle on the Ontology of Intentionality, dans D. Fisette & G. Fréchette (eds.), *Themes from Brentano*, Amsterdam:Rodopi, pp. 121-137, 2013.

[7] C.E. Cleland, The Difference Between Real Change and Mere Cambridge Change, *Philosophical Studies*, vol. 60, pp. 257-280, 1990.

[8] A. Galton, On What Goes On: The ontology of processes and events, dans R. Ferrario & W. Kuhn (eds.), *Proc. of the Fourth International Conference on Formal Ontology in Information Systems (FOIS2006)*, pp. 4-11, 2006.

[9] A. Galton & R. Mizoguchi, The water falls but the waterfall does not fall: New perspectives on objects, processes and events, *Applied Ontology*, vol. 4, pp. 71-107, 2009.

[10] P. Geach, What actually Exists? In *Proc. of the Aristotelian Society*, Supplementary Volumes, vol. 42, pp. 7-16, 1968.

[11] P. Grenon & B. Smith, SNAP and SPAN: Towards dynamic spatial ontology, *Spatial Cognition and Computation*, vol. 87, pp. 69-103, 2004.

[12] G. Kassel, Processes Endure, Whereas Events Occur. In S. Borgo, R. Ferrario, C. Masolo & L. Vieu (eds.), *Ontology Makes Sense*: *Essays in honor of Nicola Guarino*, Frontiers in Artificial Intelligence and Applications, vol. 136, IOS Press, pp. 177-193, 2019.

[13] G. Kassel, Trois conceptions du processus : les raisons d'un choix, dans N. Hernandez (éd.), *Actes des 30èmes Journées Francophones d'Ingénierie des Connaissances (IC 2019)*, pp. 199-214, 2019.

[14] G. Kassel, Physical processes, their life and their history, *Applied Ontology*. vol. 15, n° 2, pp. 109-133, 2020.

[15] G. Kassel, Événements abstraits et états d'affaires « occurrent-facteurs », In S. Ferré (éd.), Actes des Journées Francophones d'Ingénierie des Connaissances (IC-PFIA 2020), pp. 40-55, 2020.

[16] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, & L. Schneider. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology, WonderWeb Deliverable D18, Final Report, vr. 1.0, 2003.

[17] A. Meinong, Sur les objets d'ordre supérieur et leur rapport à la perception interne, dans D. Fisette & G. Fréchette (eds.), *À l'école de Brentano, de Würzbourg à Vienne*, Paris, J. Vrin, pp. 261-341, 2007 ; trad., par G. Fréchette, de Über Gegenstände höherer Ordnung und deren Verhältmis zur inneren Wahrnehmung, *Zeitschrift für Psychologie und Physiologie der Sinnesorgane*, vol. 21, pp. 182-272, 1899.

[18] A. Meinong, *On Assumptions*, Berkeley and Los Angeles, University of California Press, 1983 ; trad. anglaise, par J. Heanue, de *Über Annahmen*, Seconde édition, Leipzig: J.A. Barth, 1910.

[19] G.H. Merrill, Ontological realism: Methodology or misdirection?, *Applied Ontology*, vol. 5, pp. 79-108, 2010.

[20] K. Mulligan, P. Simons & B. Smith, Truth-Makers, *Philosophy and Phenomenological Research*, vol. 44, pp. 287-321, 1984.

[21] F. Nef, *L'objet quelconque. Recherches sur l'ontologie de l'objet*. Librairie Philosophique J. Vrin, Paris, 1998.

[22] F. Nef, *L'Anti-Hume. De la logique des relations à la métaphysique des connexions*, Librairie Philosophique J. Vrin Paris, 2017.

[23] R. Poli, Levels of Reality and the Psychological Stratum. *Revue internationale de philosophie*, vol. 2, n° 236, pp. 163-180, 2006.

[24] B. Russell, *Principles of Mathematics*, Cambridge, UK: Cambridge University Press, 1903.

[25] B. Russell, On Denoting, *Mind*, New Series, vol. 14, n° 56, pp. 479-493, 1905.

[26] B. Russell, On the Nature of Truth, dans *Proc. of the Aristotelian Society*, New Series, vol. 7, pp. 28-49, 1906-1907.

[27] B. Smith, *Austrian Philosophy*, Brentano's Legacy. Chicago, Open Court, 1995.

[28] B. Smith & W. Ceusters, Ontological realism: A methodology for coordinated evolution of scientific ontologies. *Applied Ontology*, vol. 5, pp. 139-188, 2010.

[29] R. Stout, Processes, *Philosophy*, vol. 72, n° 279, pp. 19-27, 1997.

[30] K. Twardowski, Sur la théorie du contenu et de l'objet des représentations, dans J. English (éd.), *Husserl – Twardowski, sur les objets intentionnels (1893-1901)*, Paris, J. Vrin, pp. 85-200, 1993 ; trad., introduction et notes par J. English de *Zur Lehre vom Inhalt und Gegenstand der Vorstellungen*. *Eine psychologogische Untersuchung*, Vienne, Hölder, 1894.

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

Découvrabilité et réutilisation de données produites par des workflows : un cas d'usage en génomique

Alban Gaignard[1], Hala Skaf-Molli[2], Khalid Belhajjame[3]

[1] l'institut du thorax, INSERM, CNRS, University of Nantes, Nantes, France
[2] LS2N, University of Nantes, Nantes, France
[3] LAMSADE, PSL, Université Paris-Dauphine, Paris, France

alban.gaignard@univ-nantes.fr
hala.skaf@univ-nantes.fr
kbelhajj@googlemail.com

Résumé

Les systèmes de workflows ont largement contribué à améliorer la reproductibilité des expériences scientifiques. Cependant, relativement peu de travaux ont porté sur la réutilisation des données produites au cours de l'exécution. Dans cet article, nous faisons l'hypothèse que ces données intermédiaires doivent être considérées comme des objets de premier ordre, qui doivent être conservés et publiés. Non seulement cela permettra d'économiser des ressources de calcul et de stockage, mais surtout cela facilitera et accélérera l'évaluation de nouvelles hypothèses. Pour aider les scientifiques à annoter ces données, nous exploitons plusieurs sources d'information : i) les informations de provenance capturées lors de l'exécution des workflows, et ii) les annotations de domaine qui sont fournies par des catalogues sémantiques d'outils, tels que Bio.Tools. Finalement, nous montrons, sur un scénario réel de bioinformatique, comment des graphes de provenance peuvent être transformés et résumés, à destination des utilisateurs et des machines.

Mots-clés

FAIR, reproductibilité, workflows scientifiques.

Abstract

Workflow systems have played an important role in facilitating the reproducibility of scientific experiments, yet, little work has been devoted to enhance the reuse of produced data. We argue that these intermediate data should be considered as first-order objects, which are worthy of preservation and publication. Not only will this save computational resources, but more importantly it will ease and accelerate the evaluation of new hypotheses. To help scientists annotate such produced data, we exploit multiple sources of information : i) provenance information captured during the execution of workflows, and ii) domain annotations provided by semantic catalogs of tools, such as Bio.Tools. Finally, we show, on a real bioinformatics scenario, how provenance graphs can be transformed and synthesized, for human and machine use.

Keywords

FAIR, reproducibility, scientific workflows.

1 Introduction

Les sciences dirigées par les données ont amené ces dernières années un changement de paradigme. L'évaluation d'hypothèses scientifiques repose de plus en plus sur des codes informatiques d'analyse, organisés sous la forme de *pipelines* (ou *workflows*), et exécutés sur des masses de données[1, 21, 2, 11]).

Pour répondre aux enjeux de reproductibilité [25], les scientifiques ont été encouragés à ne pas seulement rendre compte de leurs résultats, mais aussi à documenter leurs méthodes et expériences, ainsi que l'ensemble des données analysées et produites. Un certain nombre de méthodes et d'outils ont été proposés pour aider les scientifiques dans cette tâche [9, 16, 3].

Malgré l'intérêt de ces propositions visant à faciliter la répétabilité des expériences, elles n'apportent pas encore de réponse quant à la réutilisation des données produites.

Nous faisons l'hypothèse que toutes les données produites par les *workflows* associés aux expériences doivent être considérées comme des objets de premier ordre, afin d'être plus facilement découvrables, accessibles et finalement réutilisables par les membres de la communauté scientifique.

Dans cet article, nous montrons comment nous pouvons combiner les métadonnées de provenance avec des connaissances externes associées aux *workflows* et aux outils bioinformatiques (Bio.Tools [19]) pour promouvoir le partage et la réutilisation des données traitées. **Notre objectif principal est de promouvoir la réutilisation des données traitées afin de limiter la duplication des efforts de calcul et de stockage associés à la ré-exécution de *workflows***.

Les contributions de cet article sont les suivantes :

— un scénario concret de réutilisation de données produites par des *workflows* dans le domaine de la bioinformatique,
— une approche basée sur des graphes de connais-

sances pour l'annotation sémantique des données brutes,
— une évaluation expérimentale de l'approche à l'aide d'un *workflow* réel en bioinformatique.

Cet article est organisé comme suit. La section 2 présente les motivations et définit le problème scientifique. La section 3 détaille l'approche FRESH proposée. La section 4 présente nos résultats expérimentaux. La section 5 résume les travaux de l'état de l'art. Enfin, les conclusions et les travaux futurs sont exposés dans la section 6. Les lecteurs intéressés peuvent accéder à la version étendue de ce travail qui a été publiée dans un journal de langue anglaise [15].

2 Motivations et problématique

Nous motivons notre proposition à partir d'un *workflow* de séquençage d'exomes. Il consiste (1) à aligner les séquences d'ADN codantes d'un échantillon sur un génome de référence et (2) à détecter leurs mutations génétiques. La figure 1 résume les tâches d'analyse bioinformatique. Pour des raisons de clarté, nous masquons dans ce scénario certaines des étapes de traitement mineures telles que le tri des bases d'ADN.

Dans des conditions réelles, de telles analyses nécessitent beaucoup de temps de calcul et de capacité de stockage. A titre d'exemple, des *workflows* similaires sont exécutés en production au centre national de séquençage (CNRGH). Pour un échantillon typique en séquençage d'exome (9,7 Go compressé), 18,6 Go sont nécessaires pour stocker les données compressées d'entrée et de sortie. Sur une infrastructure de calcul *HPC* (7 nœuds de calculs avec 28 cœurs Intel Broadwell), 2 heures et 27 minutes sont nécessaires pour produire un fichier de variants VCF annoté, ce qui correspondrait sur un seul CPU à 158 heures cumulées pour un seul échantillon, soit 6 jours de calcul sur un seul CPU.

Nous faisons donc l'hypothèse que la réutilisation des données déjà analysées, est essentielle pour accélérer la recherche sur des sujets similaires ou connexes. Dans le *workflow* de la figure 1, `GRCh37` est considéré comme hautement réutilisable car il constitue un atlas de référence pour les séquences génomiques humaines, et il résulte de l'état des connaissances scientifiques à un moment donné. Par ailleurs, les fichiers `BAM` peuvent également être considérés comme plus réutilisables que les données brutes car ils ont été alignés sur cet atlas et bénéficient donc des connaissances associées à cet atlas génomique. Par exemple, ils fournissent la relation entre les séquences et les gènes connus, ils peuvent être visualisés à l'échelle du génome, ou bien réutilisé pour générer des séquences brutes non alignées.

Pour les scientifiques il est très difficile de répondre à des questions telles que "puis-je réutiliser ces fichiers dans le contexte de mes travaux ?". Dans cet exemple, si l'on souhaite réutiliser le fichier de variants final, il faut absolument connaître la version du génome de référence ainsi que le contexte scientifique de l'étude, les phénotypes associés aux échantillons, ainsi que les relations possibles entre échantillons. Enfin, il est également essentiel de disposer d'informations précises sur l'algorithme de détection de variants en raison des seuils de détection internes [22]. Plus généralement, il faut non seulement des informations détaillées de provenance concernant l'historique des traitements de données, mais également des annotations de domaine basées sur des vocabulaires contrôlés (problème 1). Ces vocabulaires existent mais l'annotation des données traitées avec des concepts spécifiques à un domaine demande beaucoup de temps et d'expertise (problème 2).

Dans ce travail, nous montrons comment améliorer la réutilisation des données (intermédiaires) de *workflows* en tirant parti (1) des efforts de la communauté visant à cataloguer sémantiquement les outils en bioinformatique, et (2) des capacités d'automatisation et de capture d'information de provenance des systèmes de gestion de *workflows* pour automatiser l'annotation des données traitées.

3 Approche

FRESH est une approche visant à améliorer la découvrabilité (*Findability*) et la réutilisation (*Reusability*) des données produite et analysées par des *workflows* dans le domaine de la génomique. Les principes FAIR [26, 27] et les approches Linked data [6, 5] constituent les piliers conceptuels et technologiques de cette démarche.

Nous abordons la question de la découvrabilité en nous appuyant sur les données liées sur le web (*Linked Data*), à savoir l'association d'un URI à chaque entité, la mise en relation de ces entités sous la forme de graphes de connaissances RDF, et l'utilisation de vocabulaires contrôlés qui définissent la nature de ces entités et leurs relations.

Très liée au contexte scientifique, la réutilisation des données est plus difficile. Des éléments de réponse ont été proposés pour le partage FAIR des données génomiques [12], cependant, proposer et évaluer la réutilisabilité des données est toujours un défi et un travail en cours [28]. Dans ce travail, nous nous concentrons sur les données réutilisables comme annotées avec des informations suffisamment complètes permettant une meilleure traçabilité, interprétabilité à la fois par des utilisateurs ou des machines.

Pour une meilleure traçabilité, les graphes de provenance sont nécessaires afin de suivre le processus de génération des données.

Pour une meilleure interprétabilité, des informations contextuelles [26] sont nécessaires, par exemple : (i) les hypothèses de recherche, les laboratoires de recherche, les conditions expérimentales, les résultats antérieurs (publications scientifiques), et (ii) le contexte technique en termes de matériel, de méthodes, de sources de données, de logiciels utilisés (algorithmes, requêtes). Ces données doivent être annotées avec des vocabulaires spécifiques d'un domaine. Pour expliciter les connaissances associées aux étapes d'analyse de données, nous pouvons nous appuyer sur l'ontologie EDAM [1] qui est activement développée et utilisée dans le cadre du registre d'outils bioinformatiques

1. `http://edamontology.org`

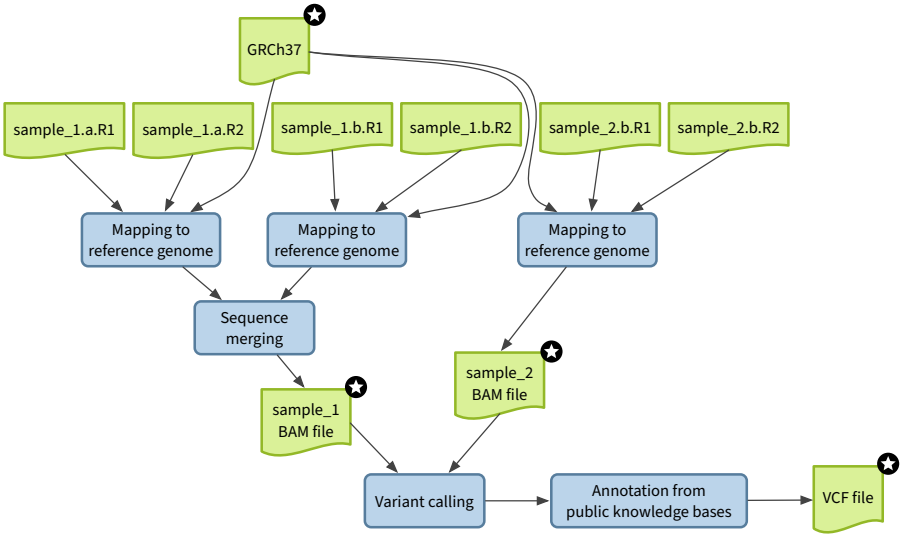

FIGURE 1 – Un *workflow* de bioinformatique typique visant à détecter et annoter des mutations génétiques. Les données sont en vert et les étapes de traitement sont en bleu.

Bio.Tools [2], et qui organise les concepts et relations sémantiques dans le domaine de la bioinformatique. Cependant, cette ontologie ne permet pas de décrire le contexte scientifique associé à un *workflow*. Pour résoudre ce problème, nous nous appuyons sur l'ontologie *Micropublications* [10] qui a été proposée pour représenter formellement les approches scientifiques, les hypothèses, ou les éléments de preuve, dans la perspective de faciliter l'exploitation des articles scientifiques par des algorithmes.

La figure 2 illustre notre approche pour améliorer la réutilisabilité des données. La première étape consiste à capturer la provenance pour toutes les exécutions d'un *workflow*. PROV-O [3] est le standard *de facto* pour décrire et échanger des graphes de provenance. Bien que la capture de la provenance puisse être facilement gérée dans les moteurs de *workflows*, il n'existe pas de moyen systématique de relier une activité PROV-O (l'exécution réelle d'un outil) à l'agent logiciel correspondant *i.e.* Le logiciel responsable du traitement des données). Pour résoudre ce problème, nous proposons de fournir, au moment de la conception du workflow, l'identifiant de l'outil dans le catalogue des outils. Cela permet de générer une trace de provenance qui associe chaque exécution (`prov:wasAssociatedWith`), et donc chaque donnée consommée et produite, à l'identifiant du logiciel.

Ensuite, nous assemblons un graphe de connaissances bioinformatiques qui relie (1) les annotations des outils, recueillies dans le registre Bio.Tools, fournissant des informations sur les fonctions des outils (opérations bioinformatiques EDAM) et le type de données qu'ils consomment et produisent, (2) l'ontologie EDAM complète, pour accéder par exemple aux définitions et synonymes, (3) le graphe PROV-O résultant de l'exécution d'un *workflow* qui fournit des métadonnées de provenance techniques et génériques, et (4) le contexte expérimental en utilisant les micropublications pour décrire les questions et hypothèses scientifiques associées à l'expérience.

Enfin, sur la base de requêtes de provenance spécifiques au domaine, la dernière étape consiste à extraire quelques données significatives du graphe de connaissances, afin de fournir aux scientifiques des résultats intermédiaires ou finaux plus réutilisables, et de fournir des historiques de données découvrables et interrogeables par les machines.

Dans le reste de cette section, nous nous appuyons sur le langage de requête SPARQL pour interagir avec le graphe de connaissances en termes d'extraction et d'enrichissement des connaissances.

```
SELECT ?d_label ?title ?f_def ?st WHERE {
    ?d rdf:type prov:Entity ;
       prov:wasGeneratedBy ?exec ;
       rdfs:label ?d_label .

    ?exec prov:wasAssociatedWith ?tool ;
          prov:wasStartedBy ?wf .

    ?tool dc:title ?title ;
       biotools:has_function ?f .

    ?f rdfs:label ?f_label ;
       oboInOwl:hasDefinition ?f_def .

    ?wf mp:supports ?c .
    ?c rdf:type mp:Claim ;
       mp:statement ?st .
}
```

Requête 1 – Requête SPARQL permettant de lier les données produite aux outils/algorithmes.

La requête 1 vise à lier des données avec la définition de l'opération bioinformatique dont elles résultent. Dans cette requête SPARQL, nous identifions d'abord les données (`prov:Entity`), l'exécution de l'outil dont elles résultent (`prov:wasGeneratedBy`), et le logiciel utilisé (`prov:wasAssociatedWith`). Ensuite, nous obtenons du

2. http://bio.tools
3. https://www.w3.org/TR/prov-o/

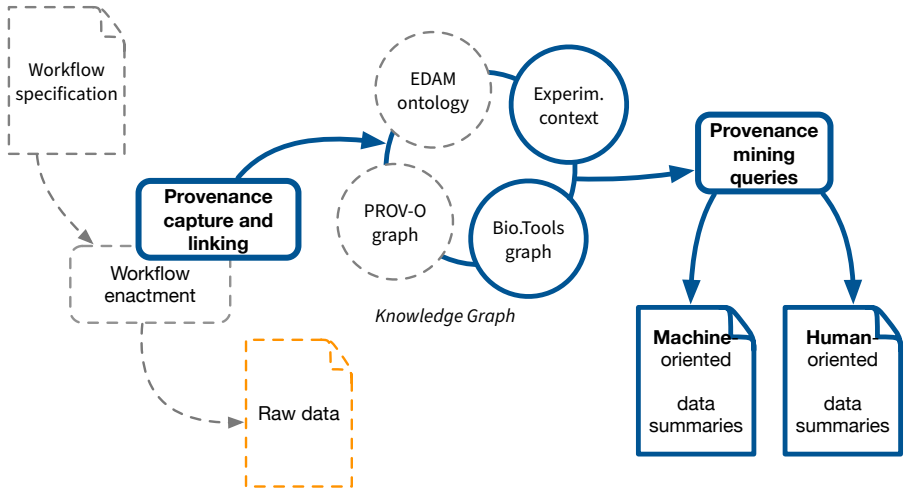

FIGURE 2 – Base de connaissances produite à partir des informations de provenance et des annotations des outils afin d'automatiser la production de résumés destinés aux utilisateurs et aux machines.

sous-graphe Bio.Tools l'annotation EDAM qui spécifie la fonction de l'outil (`biotools:has_function`). La définition de la fonction de l'outil est obtenue dans l'ontologie EDAM (`oboInOwl:hasDefinition`). Enfin, nous identifions le contexte scientifique de l'expérience en faisant correspondre les déclarations exprimées en langage naturel (`mp:Claim`, `mp:statement`).

La requête 2 montre comment un motif de provenance peut-être transformé pour fournir un résumé des principales étapes de traitement, en s'appuyant sur une ontologie de domaine. L'idée consiste d'abord à identifier tous les liens de dérivation de données (`prov:wasDerivedFrom`). Ensuite les exécutions d'outils sont identifiées ainsi que les agents logiciels correspondants, et la fonctionnalité des outils. Nous exploitons la propriété `biotools:has_function`. Une fois ce patron identifié, un nouveau graphe est créé à l'aide d'une clause CONSTRUCT. Il représente une chaîne ordonnée d'étapes de traitement (`p-plan:wasPreceededBy`).

```
CONSTRUCT {
    ?x2 p-plan:wasPreceededBy ?x1 .
    ?x2 prov:wasAssociatedWith ?t2 .
    ?x1 prov:wasAssociatedWith ?t1 .
    ?t1 biotools:has_function ?f1 .
    ?f1 rdfs:label ?f1_label .
    ?t2 biotools:has_function ?f2 .
    ?f2 rdfs:label ?f2_label .
} WHERE {
    ?d2 prov:wasDerivedFrom ?d1 .

    ?d2 prov:wasGeneratedBy ?x2 ;
        prov:wasAssociatedWith ?t2 ;
        rdfs:label ?d2_label .

    ?d1 prov:wasGeneratedBy ?x1 ;
        prov:wasAssociatedWith ?t1 ;
        rdfs:label ?d1_label .

    ?t1 biotools:has_function ?f1 .
    ?f1 rdfs:label ?f1_label .

    ?t2 biotools:has_function ?f2 .
    ?f2 rdfs:label ?f2_label .
}
```

Requête 2 – Requête SPARQL permettant de produire un *workflow* abstrait.

4 Résultats expérimentaux

4.1 Graphes de provenance

Nous avons expérimenté notre approche sur un *workflow* de séquençage d'exome [4], conçu et exploité par la plateforme de génomique et de bioinformatique GenoBird. Il met en œuvre le scénario de motivation que nous avons présenté dans la section 2. Nous supposons que, sur la base de l'approche présentée précédemment, le *workflow* a été exécuté, la provenance associée a été capturée et le graphe de connaissances a été assemblé.

Le graphe de provenance consiste en un graphe RDF avec 555 triplets exploitant l'ontologie PROV-O.

L'interprétation de ce graphe de provenance est difficile d'un point de vue humain en raison du nombre de noeuds et d'arêtes et, surtout, de l'absence de termes spécifiques à un domaine.

4.2 Résumés de données destinés aux utilisateurs

Sur la base de la requête 1 et d'un modèle textuel, nous montrons dans la figure 3 des phrases qui ont été générées automatiquement à partir du graphe de connaissances. Elles ont pour but de fournir aux scientifiques des informations explicites sur la façon dont les données ont été produites, et sur leur contexte scientifique, en utilisant des termes spécifiques au domaine.

Les procédures complexes d'analyse des données nécessiteraient un long texte et de nombreuses articulations logiques pour être compréhensibles. Les diagrammes visuels fournissent une représentation compacte pour le traitement de données complexes et constituent donc un moyen intéressant d'assembler des résumés de données pour les scientifiques.

4. https://gitlab.univ-nantes.fr/bird_pipeline_registry/exome-pipeline

```
The file <VCF/hapcaller.recal.combined.annot.
gnomad.vcf.gz> results from tool
<gatk2_variant_annotator-IP> which <Predict the
effect or function of an individual single
nucleotide polymorphism (SNP).>
It was produced in the context of <Rare Coding
Variants in ANGPTL6 Are Associated with Familial
Forms of Intracranial Aneurysm>
```

FIGURE 3 – Résumé textuel basé sur l'ontologie EDAM.

TABLE 1 – Temps de calcul pour la génération des résumés de données

Graphe RDF	chargement	résumés text.	NanoPub.	Visu.
218 906 triplets	22.7s	1.2s	61ms	1.5s

Un exemple de diagramme récapitulatif est fourni par la Figure 4. Les flèches noires représentent le flux logique du traitement de données, et les ellipses noires représentent la nature du traitement de données, en termes d'opérations EDAM. Le diagramme montre que les fichiers en bleu résultent d'une étape de traitement effectuant une opération de type *"SNP annotation"*, telle que définie dans l'ontologie EDAM.

Ces visualisations fournissent aux scientifiques les moyens de positionner un résultat intermédiaire, par exemple des séquences génomiques alignées sur un génome de référence (fichier BAM), ou des variants génomiques (fichier VCF) dans le contexte d'un processus complexe d'analyse. Alors qu'un bioinformaticien expert n'aura pas besoin de ces résumés, nous considérons que visualiser et rendre explicites ces résumés est d'un intérêt majeur pour mieux réutiliser les données scientifiques, voire fournir un premier niveau d'explication en termes de concepts spécifiques au domaine d'application.

4.3 Résumés de données destinés aux machines

Les principes *Linked Data* préconisent l'utilisation de vocabulaires contrôlés et d'ontologies pour fournir des connaissances lisibles par l'homme et par la machine. Nous montrons dans la figure 5 comment des annotations spécifiques au domaine (EDAM), peuvent être agrégées et partagées entre des machines en exploitant le vocabulaire NanoPublication. Ces résumés peuvent être indexés et découverts sémantiquement, conformément aux principes de *Findability* de FAIR.

4.4 Implémentation

Nous avons légèrement étendu le moteur de workflow Snakemake [20] avec un module de capture de provenance [5]. Nous avons également développé un *crawler* [6] qui construit un jeu de données RDF à partir du registre Bio.Tools. Les résultats présentés dans la section 4 ont été obtenus en exécutant un Notebook Jupyter [7].

5. https://github.com/albangaignard/snakemake/tree/research-objects

6. https://github.com/bio-tools/biotoolsRdf

7. https://github.com/albangaignard/fresh-toolbox

Nous avons simulé l'exécution du *workflow* d'analyse de données d'exome pour évaluer le temps de calcul des résumés de données, à partir d'un graphe de connaissances RDF. Cette simulation a permis de ne pas être impacté par les temps de calcul réels de l'analyse des données génomiques. Le tableau 1 décrit les temps de calcul en utilisant un ordinateur portabe MacBook Pro Core i5 de 16 Go et 2,9 GHz. Nous avons mesuré 22,7 secondes pour charger en mémoire le graphe de connaissances complet (218 906 triplets) décrivant l'exécution du *workflow* via son graphe de provenance, le registre d'outils Bio.Tools et l'ontologie EDAM. Les résumés de données textuels ont été obtenus en 1.2s, la *NanoPublication* a été générée en 61 ms, et enfin il a fallu 1.5 s pour générer une visualisation sous la forme de graphe du résumé. Ce sur-coût peut être considérée comme négligeable par rapport aux ressources informatiques nécessaires pour analyser des données de séquençage d'exome, comme indiqué dans la section 2.

4.5 Discussion

Les résultats expérimentaux montrent qu'il est possible de générer des résumés de données qui fournissent des informations précieuses sur les données du *workflow*. Nous nous concentrons sur les annotations spécifiques au domaine afin de promouvoir la découvrabilité et la réutilisation des données, en lien avec les principes FAIR [8] avec une attention particulière pour les *workflows* en génomique.

Pour ce qui est de la découvrabilité, FRESH répond en partie aux exigences F1 ((Les (méta)données ont un identifiant unique et persistant), F2 (Les données sont décrites avec des métadonnées riches) et F3 (Les métadonnées incluent clairement et explicitement l'identifiant des données qu'elles décrivent) car (i) nous attribuons des identifiants uniques universels (UUID) aux entités de provenance et (ii) nous réutilisons les ontologies NanoPublication et EDAM pour partager et réutiliser les données produites. Les nanopublications générées pourraient être publiées soit via un serveur SPARQL, soit par le réseau de serveurs NanoPublication.

Pour la réutilisabilité, FRESH addresse R1.2 (les (méta)données sont associées à des informations de provenance détaillées) et R1.3 (les (méta)données répondent aux standards adoptés par les communautés). Comme illustré dans les sections précédentes, FRESH peut être utilisé pour générer des résumés de données destinés aux scientifiques ou aux machines.

Toujours dans le contexte de l'analyse des données génomiques, un scénario de réutilisation typique consisterait à exploiter les variantes génomiques annotées, pour effectuer une analyse statistique des variantes rares. Si l'on considère qu'aucune sémantique n'est attachée aux noms de fichiers ou d'outils, les informations de provenance générique ne permettraient pas de fournir des informations sur la nature du traitement de données. En regardant le diagramme destinés aux utilisateurs, ou en laissant un algorithme interroger la nanopublication, destinée aux machines, produite par FRESH, les scientifiques seraient plus facilement en mesure de comprendre que le fichier résulte d'une annotation

8. https://www.go-fair.org/fair-principles

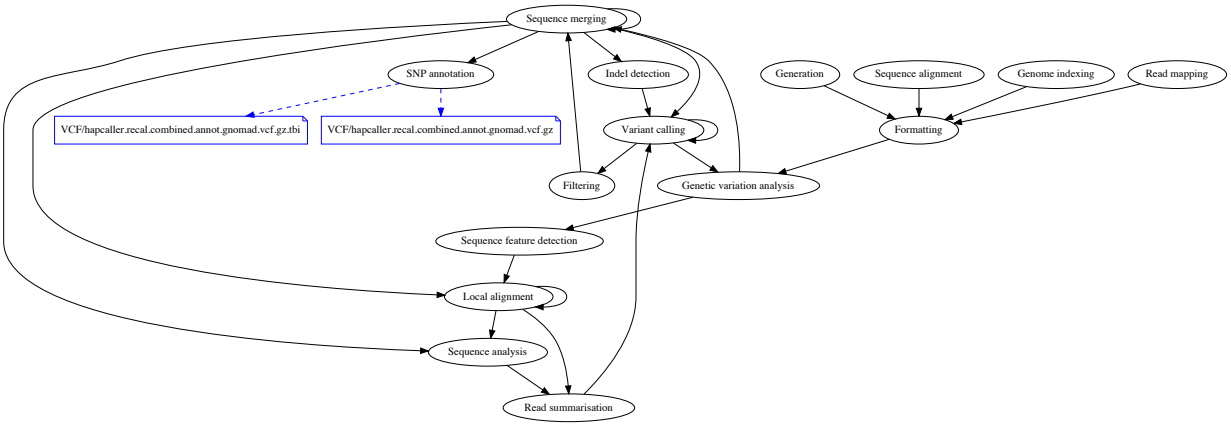

FIGURE 4 – Un diagramme compilé automatiquement à partir du graphe de provenance et de connaissances spécifiques du domaine et destiné aux scientifiques.

```
:head {
    _:np1 a np:Nanopublication .
    _:np1 np:hasAssertion :assertion .
    _:np1 np:hasProvenance :provenance .
    _:np1 np:hasPublicationInfo :pubInfo .
}

:assertion {
  <http://snakemake-provenance/Samples/Sample1/
  BAM/Sample1.merged.bai> rdfs:seeAlso
  <http://edamontology.org/operation_3197> .

  <http://snakemake-provenance/VCF/hapcaller.
  indel.recal.filter.vcf.gz> rdfs:seeAlso
  <http://edamontology.org/operation_3695> .
}
```

FIGURE 5 – Extrait d'une *NanoPublication* permettant d'aggréger des assertions spécifiques du domaine, des informations de provenance et de publication.

de polymorphismes génétiques (SNP) qui a été précédée d'une étape de détection de variants elle-même précédée d'une étape de détection d'insertion/délétion (Indel).

Nous nous sommes concentrés dans ce travail sur le domaine de la bioinformatique et avons exploité Bio.Tools, un effort communautaire à grande échelle visant à cataloguer sémantiquement les algorithmes/outils disponibles. Dès que des catalogues d'outils sémantiques seront disponibles pour d'autres domaines, FRESH pourra être appliqué afin d'améliorer la découvrabilité et la réutilisation des données traitées. Même s'ils sont plus récents, des efforts similaires s'adressent à la communauté de la bioimagerie grâce à la mise en place du registre de bioimagerie BISE (projet européen COST Neubias), et l'extension de l'ontologie EDAM pour la bioimagerie.

Dans ce travail, nous avons validé notre solution manuellement dans le cadre d'un *workflow* réel de génomique. Un ensemble de données et de workflows de référence permettrait de plus facilement évaluer les approches visant à améliorer la réutilisation de données scientifiques. Ces données et workflows de référence pourraient résulter des activités de différentes communautés scientifiques afin de mieux répondre aux enjeux des sciences ouvertes et reproductibles.

5 Etat de l'art

Notre approche est liée aux travaux visant à faciliter la préservation, la reproductibilité et la réutilisation des ressources scientifiques. OBI (Ontology for Biomedical Investigations) [7] et le modèle ISA (Investigation, Study, Assay) [23] sont deux modèles largement utilisés dans le domaine des sciences de la vie pour décrire des travaux scientifiques. Research Objects [4] propose une suite d'ontologies visant à agréger des spécifications de *workflows*, leurs exécutions et leur contexte scientifique. ReproZip [9] est une autre solution qui permet de créer des archives comprenant les dépendances nécessaires pour reproduire un *workflows*.

Les solutions ci-dessus aident les scientifiques à agréger leurs informations dans un seul conteneur. Cependant, elles n'aident pas pour l'annotation des résultats d'expériences. Pour ce problème, Alper *et al.* [3] et Gaignard *et al.* [16] ont développé des solutions qui permettent de dériver des annotations à partir des *workflows* et de les résumer.

L'ontologie PROV-O, recommandation du W3C et ses extensions, ProvONE [9], OPMW [10], Wfprov [11] ou P-Plan [17], présentent un intérêt particulier pour notre travail. Elles permettent de poser des questions sur le "pourquoi" et le "comment" de la production des données. Cependant, répondre à des questions telles que "ces données sont-elles utiles pour mon expérience ?" ou bien "sont-elles de qualité suffisante ?" est difficile et nécessite des annotations spécifiques d'un domaine, non couvertes par les modèles de provenance génériques.

Dans nos travaux précédents [16], nous avons proposé *PoeM* une approche pour générer des rapports d'expérience basés sur la provenance et les annotations des utilisateurs.

9. https://purl.dataone.org/provone-v1-dev
10. https://www.opmw.org
11. http://purl.org/wf4ever/wfprov#

SHARP [13, 14] étend *PoeM* pour les *workflows* s'exécutant sur différents systèmes et produisant des traces de provenance hétérogènes. Dans ce travail, nous nous appuyons sur ces travaux pour annoter et résumer les informations de provenance, en nous concentrant sur les données plutôt que sur le *workflow* lui-même.

La proposition de Garijo et Gil [18] est peut-être la plus proche de la nôtre dans le sens où elle se concentre sur les données, et génère des textes à partir des informations de provenance. Dans ce travail, nous nous concentrons plutôt sur l'annotation des données intermédiaires du *workflow*.

[24] vise à identifier les similitudes entre les *workflows* à partir de leur structure, des noms de leurs modules, et des auteurs associés. Notre objectif est différent dans la mesure où nous voulons promouvoir la réutilisation non seulement des *workflows*, mais aussi des données produites, en nous appuyant sur des techniques de résumé.

Cerezo et al. [8] ont proposé un modèle de *workflow* conceptuel, proche du domaine d'expertise de l'utilisateur final, visant à améliorer le partage et la réutilisation des *workflows* scientifiques. Dans notre approche, nous nous concentrons sur la réutilisation des données intermédiaires produites alors que Cerezo *et al.* se concentrent sur la réutilisation du processus de transformation de données lui-même. De plus, notre approche est basée sur l'exécution des *workflows*, et tend à limiter la sollicitation d'experts du domaine, en exploitant des catalogues d'outils déjà annotés sémantiquement.

Notre travail est également lié aux efforts de la communauté scientifique pour créer des entrepôts ouverts pour la publication de données scientifiques. Par exemple, Figshare [12] et Dataverse [13], qui aident les institutions universitaires à stocker, partager et gérer tous les résultats de leurs recherches. Les résumés de données que nous produisons peuvent être publiés dans ces entrepôts. Cependant, nous pensons que les résumés que nous produisons sont mieux adaptés aux entrepôts qui publient des graphes de connaissances, par exemple celui créé par le projet whyis (http ://tetherless-world.github.io/whyis/).

6 Conclusion et perspectives

Dans cet article, nous proposons une approche visant à rendre les données des *workflows* scientifiques plus facilement découvrables et réutilisables, à partir d'un exemple dans le domaine de la génomique. Pour cela, nous générons des résumés de données, à partir de métadonnées de provenance et d'une ontologie en bioinformatique. FRESH permet de produire des résumés de données concis pour les scientifiques et pour les machine. Les résultats expérimentaux montrent l'efficacité de FRESH en termes de temps de calcul, négligeable par rapport aux ressources informatiques requises pour analyser les données de génomique. Afin d'évaluer notre approche, nous souhaiterions mener une étude auprès des plate-formes de bioinformatique fédérées dans le cadre de l'infrastructure nationale de recherche IFB. Ces plate-formes développent et exécutent à grande échelle des *workflows* d'analyse de données dans le domaine de la génomique. Cette communauté de bioinformaticiens fait face aux enjeux de reproductibilité des analyses, de partage et de réutilisation des données, et pourra permettre d'évaluer la pertinence des résumés de données introduits dans FRESH. Cependant, mettre en place une telle étude nécessite des développements logiciels *open source* pour intégrer la capture de méta-données de provenance dans des moteurs de workflows utilisés en routine tels que SnakeMake, NextFlow ou encore Galaxy.

Dans un contexte de sciences ouvertes et reproductibles, il nous parait critique d'inciter les scientifiques (1) à produire des annotations sémantiques de haute qualité pour décrire les workflows et algorithmes (2) à produire des résumés de données sémantiques et inter-opérables afin de promouvoir la découvrabilité et la réutilisation des données scientifiques.

12. https://figshare.com/
13. https://dataverse.org/

Remerciements

Nous remercions la plateforme BiRD (Biogenouest, IFB) pour son soutien technique et l'utilisation de son infrastructure.

Références

[1] G. R. Abecasis, A. Auton, B., et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422) :56–65, 2012.

[2] P. Alper. *Towards harnessing computational workflow provenance for experiment reporting*. PhD thesis, University of Manchester, UK, 2016.

[3] P. Alper, K. Belhajjame, et al. Automatic versus manual provenance abstractions : Mind the gap. In *8th USENIX Workshop on the Theory and Practice of Provenance, TaPP 2016, Washington, D.C., USA, June 8-9, 2016.* USENIX, 2016.

[4] K. Belhajjame, J. Zhao, et al. Using a suite of ontologies for preserving workflow-centric research objects. *J. Web Semant.*, 32 :16–42, 2015.

[5] C. Bizer, T. Heath, et al. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3) :1–22, 2009.

[6] C. Bizer, Vidal M.-E., and H. Skaf-Molli. Linked open data. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*. Springer, 2017.

[7] R. R. Brinkman, M. Courtot, et al. Modeling biomedical experimental processes with obi. In *Journal of biomedical semantics*, volume 1, page S7. BioMed Central, 2010.

[8] N. Cerezo and J. Montagnat. Scientific workflow reuse through conceptual workflows on the virtual imaging platform. In *Proceedings of the 6th Workshop on Workflows in Support of Large-scale Science*, WORKS '11, pages 1–10, New York, NY, USA, 2011. ACM.

[9] F. Chirigati, R. Rampin, et al. Reprozip : Computational reproducibility with ease. In *Proceedings of the 2016 International Conference on Management of Data*, pages 2085–2088. ACM, 2016.

[10] T. Clark, P. N. Ciccarese, et al. Micropublications : A semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of Biomedical Semantics*, 2014.

[11] S. Cohen Boulakia, K. Belhajjame, et al. Scientific workflows for computational reproducibility in the life sciences : Status, challenges and opportunities. *Future Generation Comp. Syst.*, 75 :284–298, 2017.

[12] M. Corpas, N. V. Kovalevskaya, et al. A fair guide for data providers to maximise sharing of human genomic data. *PLoS Computational Biology*, 14(3), 2018.

[13] A. Gaignard, K. Belhajjame, et al. SHARP : harmonizing and bridging cross-workflow provenance. In *The Semantic Web : ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portorož, Slovenia,Revised Selected Papers*, pages 219–234, 2017.

[14] A. Gaignard, K. Belhajjame, et al. SHARP : harmonizing cross-workflow provenance. In *Proceedings of the Workshop on Semantic Web Solutions for Large-scale Biomedical Data Analytics co-located with 14th Extended Semantic Web Conference, SeWeBMeDA@ESWC 2017, Portoroz, Slovenia.*, pages 50–64, 2017.

[15] A. Gaignard, H. Skaf-Molli, and K. Belhajjame. Findable and reusable workflow data products : A genomic workflow case study. *Semantic Web*, 11(5) :751–763, 2020.

[16] A Gaignard, H. Skaf-Molli, et al. From scientific workflow patterns to 5-star linked open data. In *8th USENIX Workshop on the Theory and Practice of Provenance*, 2016.

[17] D. Garijo and Y Gil. Augmenting PROV with plans in PPLAN : scientific processes as linked data. In *Proceedings of the Second International Workshop on Linked Science 2012 - Tackling Big Data, Boston, MA, USA, November 12, 2012*. CEUR-WS.org, 2012.

[18] Y. Gil and D. Garijo. Towards automating data narratives. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces*, IUI '17, pages 565–576, New York, NY, USA, 2017. ACM.

[19] J. Ison, K. Rapacki, et al. Tools and data services registry : A community effort to document bioinformatics resources. *Nucleic Acids Research*, 2016.

[20] J. Köster and S. Rahmann. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19) :2520–2522, 2012.

[21] J. Liu, E. Pacitti, et al. A survey of data-intensive scientific workflow management. *Journal of Grid Computing*, 13(4) :457–493, 2015.

[22] N. D. Olson, S. P. Lund, et al. Best practices for evaluating single nucleotide variant calling methods for microbial genomics, 2015.

[23] P. Rocca-Serra, M Brandizi, et al. Isa software suite : supporting standards-compliant experimental annotation and enabling curation at the community level. *Bioinformatics*, 26(18) :2354–2356, 2010.

[24] J. Starlinger, S. Cohen-Boulakia, and U. Leser. (Re)use in public scientific workflow repositories. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.

[25] V. Stodden. The scientific method in practice : Reproducibility in the computational sciences. *SSRN Electronic Journal*, 2010.

[26] M. D. Wilkinson et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3, 2016.

[27] M. D. Wilkinson et al. A design framework and exemplar metrics for fairness. *Scientific Data*, 5, 2018.

[28] M. D. Wilkinson et al. Fairmetrics/metrics : Proposed fair metrics and results of the metrics evaluation questionnaire. 2018.

Découverte de règles contextuelles pour prédire la présence d'amiante dans les bâtiments

Thamer Mecharnia[1,2], Lydia Chibout Khelifa[2], Fayçal Hamdi[3], Nathalie Pernelle[4], Celine Rouveirol[4]

[1] LISN, Université Paris Saclay, Orsay, thamer@lri.fr
[2] Centre Scientifique et Technique du bâtiment (CSTB), Champs sur Marne, prenom.nom@cstb.fr
[3] CEDRIC, CNAM - Conservatoire National des Arts et Métiers, Paris, faycal.hamdi@cnam.fr
[4] LIPN, Université Sorbonne Paris-Nord, CNRS UMR 7030, Villetaneuse, prenom.nom@lipn.univ-paris13.fr

18 mai 2021

Résumé

Le Centre Scientifique et Technique du Bâtiment (CSTB) a été sollicité pour développer un outil d'aide à l'identification des matériaux contenant de l'amiante dans les bâtiments. Dans ce contexte, nous avons développé une approche, nommée CRA-Miner, qui utilise des techniques de programmation logique inductive (PLI) pour découvrir des règles à partir d'un graphe de données décrivant des bâtiments et des diagnostics d'amiante. La référence des produits spécifiques utilisés lors de la construction n'étant jamais spécifiée, CRA-Miner considère les données temporelles, les types de produits et les informations contextuelles pour rechercher l'ensemble de règles candidates qui pourront être utilisées pour prédire la présence d'amiante dans les éléments de construction. Les expériences menées sur le graphe de connaissances fourni par le CSTB montrent qu'une F-Mesure prometteuse peut être obtenue.

Mots-clés

Découverte de règles, graphe de connaissances, données temporelles, amiante.

Abstract

The Scientific and Technical Center for Building (CSTB) was asked to develop a tool to help identify materials containing asbestos in buildings. In this context, we have developed an approach, named CRA-Miner, which uses inductive logic programming (ILP) techniques to discover logic rules from a data graph describing buildings and asbestos diagnostics. Since the reference of the specific products used during construction is never specified, CRA-Miner considers the temporal data, the types of products and the contextual information to find the set of candidate rules which can then be used to deduce the presence of asbestos in construction elements. The experiments carried out on the knowledge graph provided by CSTB show that a promising F-Measure can be obtained.

Keywords

Rule mining, knowledge graph, temporal data, asbestos.

1 Introduction

La nocivité de l'amiante est identifiée depuis le début du 20ème siècle. L'inhalation d'air contenant des fibres d'amiante peut entraîner des maladies telles que le cancer des poumons et de la muqueuse thoracique. Cependant, en raison de ses qualités ignifuges, de nombreux pays ont largement utilisé l'amiante dans les bâtiments, en particulier de 1950 à 1970. Même si il est maintenant illégal d'utiliser cette dangereuse fibre minérale, celle-ci est toujours présente dans de nombreux bâtiments. Aussi, l'identification des parties de construction contenant de l'amiante est une tâche importante afin de procéder à un désamiantage. Les professionnels inspectent régulièrement les bâtiments et prélèvent des échantillons pour détecter la présence d'amiante dans les composants des bâtiments mais il est nécessaire de hiérarchiser le trop grand nombre de tests possibles.

Dans le cadre du PRDA [1], le CSTB (Centre Scientifique et Technique du Bâtiment) a été sollicité pour développer un outil en ligne d'aide à l'identification de matériaux contenant potentiellement de l'amiante dans les bâtiments afin de guider l'opérateur dans la préparation de son programme de suivi (Projet ORIGAMI). La difficulté réside dans le fait que les descriptions de bâtiments disponibles ne précisent que les classes de produits utilisés, sans donner leurs références exactes ou toute autre information à leur sujet (e.g. fournisseurs, etc.). Dans [6], une approche basée sur l'ontologie ASBESTOS a été définie, qui permet d'estimer la probabilité de présence de produits amiantés dans un bâtiment. Cette première approche hybride combine des méthodes statistiques et des méthodes basées sur des règles pour générer cette probabilité en se basant sur l'année de construction du bâtiment et des ressources externes fiables mais incomplètes décrivant les produits amiantés existant sur le marché à la même période. Cependant, les

1. Plan de recherche et de développement amiante lancé par la Direction de l'Habitat, de l'Urbanisme et des Paysages (DHUP), rattachée à la Direction Générale de l'Amiante, de l'Habitat et de la Nature (Ministre du Logement et de l'Habitat Durable)

experts supposent que si le type de produit et l'année de construction peuvent être exploités pour prédire la présence d'amiante, le contexte dans lequel le produit est utilisé peut également être pertinent (i.e. la région dans laquelle se situe le bâtiment, les éléments de construction dans lesquels apparaît le produit, ainsi que les autres produits utilisés). Récemment, le CSTB a mis à disposition un ensemble de diagnostics réalisés sur un grand nombre de bâtiments. Ces données ont été représentées à l'aide de l'ontologie ASBESTOS proposée dans [6]. L'objectif est de définir une approche qui vise à apprendre des règles contextuelles à partir de ces données sémantiques en utilisant des prédicats permettant de représenter l'appartenance d'une date de construction à un intervalle temporel. De nombreuses approches de fouille de règles ont été proposées qui peuvent apprendre à classer les données en fonction de leur description RDF [5, 4, 10]. Cependant, aucune d'entre elles n'utilise le type de prédicat numérique dont nous devons disposer pour prendre en compte cet aspect temporel.

Dans cet article, nous proposons une approche basée sur l'ontologie qui découvre des règles qui peuvent être utilisées pour estimer la probabilité de l'existence de produits amiantés dans un bâtiment. L'approche proposée s'inspire des techniques de Programmation Logique Inductive (PLI) de type *générer et tester*, mais se concentre sur la découverte de règles qui décrivent le produit et son contexte par un ensemble de prédicats déclarés comme potentiellement pertinents par l'expert. Sur la base des relations de subsomption et des connaissances générales sur l'évolution de l'utilisation de l'amiante au fil des années, l'algorithme découvre un ensemble de règles qui prédisent la présence d'amiante dans les produits d'un composant de bâtiment. L'originalité de l'approche CRA-Miner est de se baser sur un contexte sémantique, des heuristiques dédiées aux propriétés de type part-of omniprésentes dans les descriptions des bâtiments et des contraintes temporelles utilisant des seuils calculés.

Dans la section 2, nous présentons des travaux connexes. En section 3, nous décrivons l'ontologie Asbestos, puis en section 4, l'approche de classification. La section 5 présente les résultats obtenus sur des données réelles du CSTB en les comparant à une *baseline* et aux résultats obtenus avec AMIE3. Enfin, la section 6 tire un ensemble de conclusions et propose des orientations de recherches futures.

2 Travaux connexes

Dans le contexte des graphes de connaissances (KG), l'exploration de règles peut être utilisée pour enrichir les graphes (prédiction de liens ou de types, ajout de nouveaux axiomes, liaison d'entités), ou pour détecter les triples RDF erronés. Motivées par le besoin de passage à l'échelle, la plupart des approches récentes de prédiction de liens/types sont basées sur des méthodes d'apprentissage profond et de plongement de graphes qui permettent de traduire des vecteurs de grande dimension en espaces de dimension relativement faible [9]. Néanmoins, d'autres applications pour lesquelles des règles interprétables sont nécessaires pour comprendre et maintenir une certaine connaissance du domaine sont toujours intéressées par la découverte de règles logiques. C'est le cas de l'approche que nous proposons dans cet article.

De nombreuses approches se sont intéressées à l'apprentissage de règles et de concepts dans les graphes de connaissance. Les approches d'apprentissage de concepts telles que DL-Foil [3] ou DL-FOCL [10] permettent d'apprendre des définitions de concepts représentées en logique de description. Ces approches s'appuient sur des stratégies de type *separate-and-conquer* qui permettent de construire une disjonction de solutions partielles, pouvant être spécialisées à l'aide d'opérateurs de raffinement basés sur la subsomption, de façon à couvrir autant d'exemples positifs que possible tout en excluant (presque) tous les exemples négatifs. Cependant, ces approches, si elles permettent de générer des définitions de concepts dans des logiques de description expressives, ne recherchent pas toutes les solutions partielles, et donc toutes les définitions. De plus, elles ne permettent pas d'utiliser des prédicats instanciés par des constantes ou de rechercher des valeurs seuils (e.g. $X \leq 17$ pour définir un mineur). Les approches de classification FOLDT (first-order logical decision tree) telles que TILDE (Top-down induction of logical decision trees) ([1]) sont basées sur des arbres de décision dans lesquels les noeuds peuvent partager des variables et impliquer des prédicats numériques comportant des valeurs seuils. Cependant, ces dernières approches n'utilisent pas la sémantique de l'ontologie dans l'exploration de l'espace de recherche.

Les graphes de connaissances sont de plus en plus nombreux et volumineux mais les données ne sont pas nécessairement complètes. Les approches telles que AMIE3 [5] et RUDIK [8] s'intéressent à la découverte d'ensembles de règles exprimées en logique du premier ordre (clauses de Horn) dans des données RDF volumineuses. Pour disposer de contre-exemples, ces approches se basent sur l'hypothèse de complétude partielle (PCA) qui suppose que lorsqu'un objet est représenté pour une entité et une propriété spécifique, tous les objets sont représentés (i.e. les autres étant considérés comme contre-exemples). Pour mieux contrôler l'espace de recherche, AMIE3 se base les mesures de qualité et limite le nombre d'atomes qui apparaissent dans la règle. RUDIK permet de découvrir des règles qui utilisent la négation pour identifier des contradictions et qui comportent des prédicats permettant de comparer des valeurs numériques ou littérales. Cependant, ces constantes doivent être définies dans le graphe de connaissance et associées à deux variables de la règle, l'approche ne permettant pas de découvrir une constante de référence comme "$\hat{a}ge(X, a), a \geq 18 \rightarrow adulte(X)$", ce qui est l'un des objectifs dans notre application. D'autres approches telles que [2] peuvent être guidées par la sémantique de l'ontologie pour éviter de construire des règles sémantiquement redondantes. Cependant, l'auteur a montré que l'exploitation des capacités de raisonnement pendant le processus d'apprentissage ne permet pas d'exploiter les règles sur

les grands graphes. L'approche [6] est une première approche permettant d'exploiter l'ontologie et des données temporelles pour estimer la probabilité qu'un produit soit amianté mais ces connaissances temporelles sont issues de ressources externes incomplètes (ANDEVA et INRS).
Dans ce travail, nous cherchons à découvrir des règles de classification à partir d'exemples positifs et négatifs décrits dans le graphe de connaissances (GC) fourni par le CSTB, afin d'estimer la probabilité de présence d'amiante (faible ou élevée) d'un produit utilisé dans un bâtiment. Les produits commercialisés utilisés étant inconnus, nous nous focalisons sur des règles contextuelles qui suivent des modèles spécifiques définis par des experts du domaine. Comme l'année de construction a un impact important sur la possible présence d'amiante, des opérateurs de comparaison SWRL [2] sont exploités pour comparer une année de construction à une année de référence qui maximise la confiance de la règle pour un type de produit (ex. *SWRL :lowerThanOrEqual(YEAR, ref_year)*). Aucune des approches sémantiques mentionnées précédemment ne permet d'exploiter un tel prédicat numérique dans les règles générées.

3 Ontologie Asbestos

Dans cette section, nous présentons la partie haute de l'ontologie Asbestos (c.f. Figure 1) qui a été construite en exploitant les documents du CSTB, les connaissances experts, et les besoins de prédiction dans le projet ORIGAMI ([6]).

- Building : construction caractérisée par un code CSTB qui correspond à un type de bâtiment donné, le type de bâtiment (e.g. école, maison, etc.), l'année de construction et l'adresse du bâtiment.
- Structure : espace faisant partie du bâtiment (e.g. balcon, toit, escalier, etc.).
- Location : localisation composant une structure (e.g. porte, fenêtre, mur, etc.).
- Product : produit utilisé dans une localisation (e.g. colle, enduit, etc.).
- Diagnostic Characteristic : représente le résultat des diagnostics amiante. La valeur de $has\_diagnostic$ est "positive" lorsque le produit contient de l'amiante ou "negative" sinon.

L'ontologie Asbestos décrit 8 sous-classes de structures, 19 sous-classes de localisation et 38 sous-classes de produit.

4 L'approche CRA-Miner

Dans cette section, nous décrivons tout d'abord les règles logiques contextuelles que nous voulons fournir aux experts pour les aider à détecter les matériaux contenant de l'amiante dans le bâtiment. Nous présentons ensuite l'algorithme CRA-Miner qui permet de générer ces règles à partir de l'ontologie ASBESTOS peuplée.

4.1 Règles contextuelles pour la prédiction de l'amiante

Une règle contextuelle pour la prédiction de l'amiante (CRA) est une conjonction de prédicats qui conclut sur la présence ou l'absence d'amiante dans un produit P. Nous considérons la borne supérieure hors-contexte de l'espace de recherche $\top$ suivante : $product(P),\ has\_diagnostic\_characteristic(P, D) \rightarrow has\_diagnostic(D, Value)$. L'ensemble des règles contextuelles qui peuvent être construites à partir de cette borne supérieure est défini en utilisant un contexte conceptuel. Ce contexte est utilisé par les experts pour sélectionner les éléments de l'ontologie décrivant le produit P pouvant avoir un impact sur la présence d'amiante. Ces prédicats utilisés pour spécialiser la règle représentent un biais de langage tel que défini en Programmation Logique Inductive (PLI) [7].

Définition 1 (Contexte conceptuel) Un contexte conceptuel CO est défini par un sous-graphe de l'ontologie, c'est-à-dire un ensemble de classes et de propriétés, qui déterminera les prédicats utilisables dans le corps de la règle.

Exemple 1 $CO = \{product, location, structure, contain, has\_location, has\_region, has\_year, has\_structure, has\_diagnostic\_characteristic\}$ est un exemple de contexte conceptuel.

Une règle contextuelle est basée sur le vocabulaire de l'ontologie sélectionné dans le contexte conceptuel et les spécialisations du prédicat $SWRL : CompareTo$ qui peut être ajouté pour introduire des contraintes sur l'année de construction du bâtiment (i.e. intervalles ouverts) :

Définition 2 (Règle contextuelle) Soit CO un contexte conceptuel, une règle contextuelle $\overrightarrow{B} \rightarrow h$, où $\overrightarrow{B} = \{B_1, B_2, ..., B_n\}$, est telle que $\forall B_i \in \overrightarrow{B}, \exists B_j \in CO \cup \{SWRL : CompareTo\}$ s.t. $B_i \sqsubseteq B_j$ et h est le prédicat $has\_diagnostic$ qui est instancié par la valeur "positive" ou "negative".

Une règle contextuelle doit également respecter les propriétés de fermeture et de connectivité définies dans les approches de fouille de règles telles que [5].

Exemple 2 La règle suivante est une règle contextuelle connexe et fermée qui peut être formée avec le contexte CO défini dans l'exemple 1 :

glue(P), contain(L, P), has_location(S, L), painting(P2), contain(L, P2), has_structure(B, S), has_year(B, Y), has_region(B, "Paris"), lessThanOrEqual(Y, "1950"), has_diagnostic_characteristic(P, D) → has_diagnostic(D, "positive")

Cette règle exprime qu'une colle présente dans un bâtiment parisien construit avant 1950, qui est utilisée dans la même localisation qu'une peinture, est potentiellement amiantée.

Des contraintes supplémentaires sont définies pour réduire la complexité du contexte et limiter la taille de l'espace

2. https://www.w3.org/Submission/SWRL/

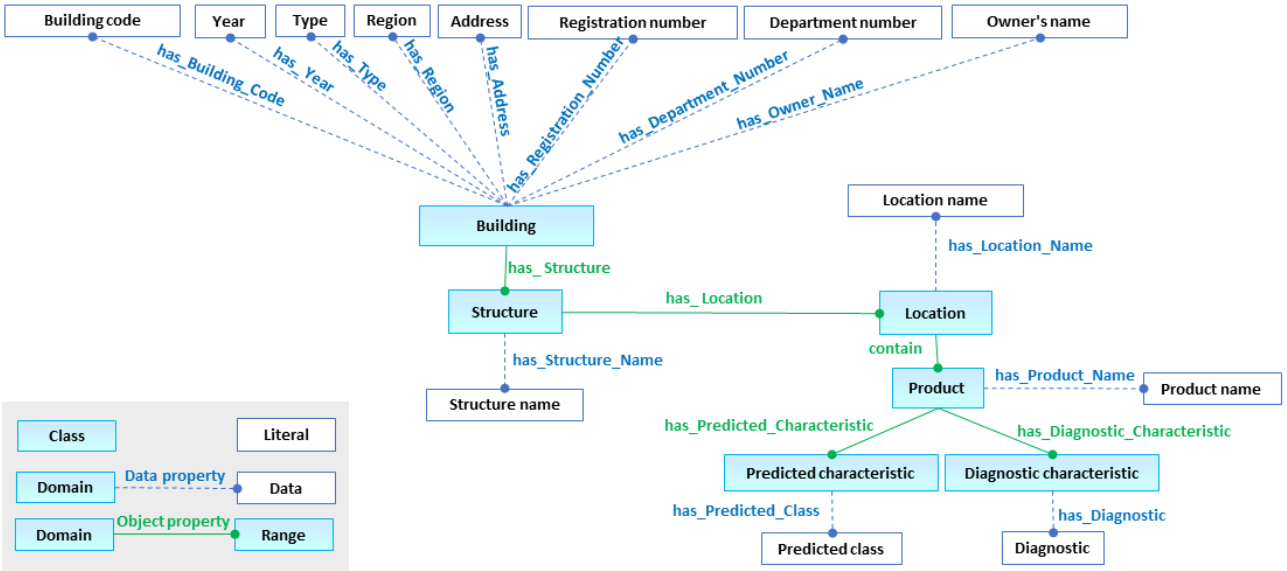

FIGURE *1 – Concepts principaux de l'ontologie Asbestos*

de recherche pour les propriétés multi-valuées décrivant les parties de bâtiments (i.e. $contain$, $has\_location$, et $has\_structure$).
L'expert peut tout d'abord définir le nombre maximum d'occurences des autres composants du bâtiment qui peuvent apparaître dans le corps de la règle : $maxSibS$ est utilisé pour définir le nombre de structures frères de la structure qui contient le produit P, $maxSibL$ est le nombre maximum de localisation frères, et $maxSibP$ représente le nombre maximum de produits frères.

Example 3 Si l'expert considère que le type des autres structures présentes dans le bâtiment ne peut influencer la présence d'amiante dans P, alors $maxSibS = 0$ et l'approche ne pourra construire la règle suivante :

Coating(P), contain(L, P), Location(L), has_location(S1, L), Vertical_Separator(S1), has_structure(B, S1), has_structure(B, S2), Floor(S2) , has_year(B, Y), has_region(B, "Lyon"), SWRL :lessThanOrEqual(Y, "1963"), has_diagnostic_characteristic(P, D) → has_diagnostic(D, "positive")

En effet, la structure $S2$ ne devrait pas être considérée (frère de $S1$ par la propriété $has\_structure$, $S1$ contenant le produit cible).

Enfin, les experts du CSTB considèrent que seule la prise en compte des types de produits les plus spécifiques peut impacter le choix du produit cible commercialisé utilisé et donc la présence d'amiante. Par exemple, la présence d'un revêtement (i.e. $coating$) dans la même localisation qu'un produit cible de type colle n'est pas significative tandis que la présence d'un revêtement de sol peut impacter le choix de la colle commercialisée utilisée. Une hypothèse similaire est réalisée pour les localisations et les structures. Aussi, seules les classes les plus spécifiques sont ajoutées dans les relations de type part-of considérées.

Pour mesurer la qualité des règles, nous utilisons les mesures de qualité classiques de *head coverage* (hc) [5] et de confiance ($conf$) qui ont été définies pour les règles relationnelles.
Le *head coverage* (hc) représente la ratio entre le support, i.e. le nombre de prédictions correctes de $has\_diagnostic(D, \text{"positive"})$ (resp. $has\_diagnostic(D, \text{"negative"})$) généré par la règle, et le nombre de diagnostics $has\_diagnostic(D, \text{"positive"})$ (resp. $has\_diagnostic(D, \text{"negative"})$) qui sont présents dans le graphe de connaissance :

$$hc(\overrightarrow{B} \rightarrow has\_diagnostic(D, val)) = \frac{supp(\overrightarrow{B} \rightarrow has\_diagnostic(D, val))}{\#(D, val) : has\_diagnostic(D, val)}$$

La confiance ($conf$) est définie par le ratio entre le support de la règle et le nombre de diagnostics différents qui participent à une instanciation du corps de la règle.

$$conf(\overrightarrow{B} \rightarrow has\_diagnostic(D, val)) = \frac{supp(\overrightarrow{B} \rightarrow has\_diagnostic(D, val))}{\#D : \exists X_1, \ldots, X_n : \overrightarrow{B}}$$

L'objectif est de découvrir toutes les règles les plus générales qui sont conformes aux contraintes du biais de langage et qui sont telles que $hc \geq minHc$ et $conf \geq minConf$.

4.2 Evolution de la présence d'amiante au fil du du temps

Il a été montré dans [6] que le nombre de produits commercialisés amiantés sur le marché reste stable jusqu'en 1972, puis diminue pour atteindre 0 en 1997, lorsque l'usage de

l'amiante est interdit en France. En effet, soit les produits ont été désamiantés, soit ils ont été abandonnés. Ainsi, même si la probabilité d'amiante diffère d'une classe de produit à une autre (e.g. les adhésifs ont perdu leur amiante plus tôt et plus rapidement que d'autres classes de produits), nous savons que cette probabilité diminue avec le temps. Aussi, si une règle contextuelle conclut sur l'absence d'amiante pour les produits utilisés dans les bâtiments construits après une année donnée Y_1, la confiance ne pourra qu'être égale ou augmenter pour $Y_2 \geq Y_1$. Cette caractéristique est exploitée pour élaguer l'espace de recherche lorsque le prédicat $greaterThanOrEqual$ ou $lessThanOrEqual$ est généralisé.

4.3 Algorithme CRA-Miner

Le but de l'algorithme CRA-Miner est de générer toutes les règles contextuelles permettant de prédire la présence d'amiante dans les produits à partir des exemples positifs et négatifs décrits dans le graphe de connaissances (GC) et telles que $hc \geq minHC$ et $conf \geq minConf$. L'algorithme de type *descendant générer et tester*, spécialise la borne supérieure de l'espace de recherche T en considérant la hiérarchie des classes de produit, en ajoutant des contraintes sur la localisation et la structure de ce produit, sur la présence de produits, localisations ou structures apparaissant dans le même composant, ainsi que des contraintes temporelles sur l'année de construction.

L'algorithme a comme entrées le graphe de connaissances, le biais de langage, un seuil $minConf$ sur la confiance, un seuil $minHC$ sur le *head coverage* de la règle, ainsi que les valeurs de $maxSibP$, $maxSibL$ et $maxSibS$ qui limitent le nombre de frères de produits, de localisations et de structures à ajouter à la règle.Le résultat est un ensemble $\mathcal{CR}$ de règles contextuelles.
L'exploration de l'espace de recherche est guidée par les relations de subsomption de l'ontologie (exploration top-down des produits cibles, de leurs localisations et de leurs structures) et exploite le fait que le nombre de produits amiantés est décroissant au fur et à mesure des années. A chaque étape de spécialisation, les règles construites qui possèdent dans une valeur de confiance et une valeur de *head coverage* plus grande que les seuils sont stockées dans l'ensemble $\mathcal{CR}$. Pour toutes les règles telles que $conf = 1$ ou $hc < minHc$, la spécialisation s'arrête.

Nous décrivons les étapes de l'algorithme pour le contexte le plus général qui a été défini par les experts du CSTB, i.e. le contexte CO défini dans l'exemple 1. L'algorithme comporte les 5 étapes suivantes :

1- **Spécialisation de $\top$ en utilisant des sous-classes de produit** :
Dans cette phase, nous remplaçons dans le $\top$ la classe $product$ par toutes les classes plus spécifiques (e.g. enduit, peinture, etc.) tant que $hc \geq minHC$ et générons donc toutes les règles "*hors-contexte*" qui peuvent être trouvées pour chaque classe de produit sans tenir compte des autres composants ou de la date de construction.

2- **Spécialisation par ajout d'une contrainte temporelle**. Pour chaque règle hors-contexte générée par l'étape précédente, nous ajoutons le chemin de propriété nécessaire pour atteindre l'année de construction. à partir du produit cible P : $has\_location(S, L), contain(L, P), has\_structure(B, S)$, $has\_year(B, Y)$. Le prédicat *SWRL :lowerThanOrEqual(Y, y)* (pour une règle qui conclut sur "positive") ou *SWRL :greaterThanOrEqual(Y, y)* (pour "negative"), est également ajouté pour comparer l'année de construction Y à une année de référence y qui maximise la confiance et préserve $hc \geq minHc$.
Par exemple, si la règle R1 suivante est générée par la première étape :
R1 : *coating(P), has_diagnostic_characteristic(P, D)* $\rightarrow$ *has_diagnostic(D, "positive")*
Cette règle peut être spécialisée de la façon suivante :
R2 : *coating(P), has_location(S, L), contain(L, P), has_structure(B, S), has_year(B, Y), SWRL :lowerThanOrEqual(Y, 1980), has_diagnostic_characteristic(P, D)* $\rightarrow$ *has_diagnostic(D, "positive")*

Pour découvrir la meilleure année de référence, CRA-Miner explore les valeurs d'année possibles de la plus récente à la plus ancienne, et considère différemment les règles qui concluent sur "$negative$" et "$positive$".
La Figure 2 montre comment la confiance évolue de 1946 à 1997 pour une règle qui conclut sur "$positive$" et pour une une classe de produit. Quand l'année de référence diminue, le *head coverage* hc diminue et la confiance $conf$ augmente. Pour couvrir le nombre maximum de diagnostics en maximisant la confiance, l'exploration s'arrête quand $hc < minHc$ (i.e. 1966 sur la figure 2). La dernière année explorée telle que $hc \geq minHc$ et telle que la confiance reste maximum (i.e. 1970 sur la figure 2) est choisie. Un processus similaire mais symétrique est appliqué pour choisir y pour les règles concluant sur "$negative$".

3- **Spécialisation par localisation et/ou par structure** (prédicat 'Location' et prédicat 'Structure').
Les hiérarchies de localisations et structures sont explorées pour spécialiser les règles générées en étape 1 et 2 avec des composants de bâtiment spécifiques qui contiennent le produit cible P.
Par exemple, la règle R1 peut être spécialisée en spécifiant que la localisation est un mur et que la structure est un balcon.
R3 : *coating(P), wall(L), balcony(S), has_location(S, L), contain(L, P), has_structure(B, S), has_year(B, Y), SWRL :lowerThanOrEqual(Y, 1980), has_diagnostic_characteristic(P, D)* $\rightarrow$ *has_diagnostic(D, "positive")*

4- **Enrichissement par la région** Toutes les règles générées peuvent être enrichies par la propriété 'has_region' qui représente la région dans lequel le bâtiment est situé.

5- **Spécialisation en ajoutant d'autres composants**. Dans cette étape, de nouvelles propriétés sont ajoutées qui représentent des produits spécifiques frères, des localisations spécifiques frères ou des structures spécifiques frères : $contain(L, P_i)$ et $C_p(P_i)$ où i varie de 0 à $maxSiblingP$ et C_p est une feuille de la hiérarchie de produits, puis $has\_location(S, L_j), C_l(L_j)$ où j varie de 0 à $maxSiblingL$ et C_L est une feuille de la hiérarchie des localisations), et $has\_structure(S, L_j), C_l(L_j)$ où j varie de 0 à $maxSiblingS$ et C_S et une feuille de la hiérarchie des structures.

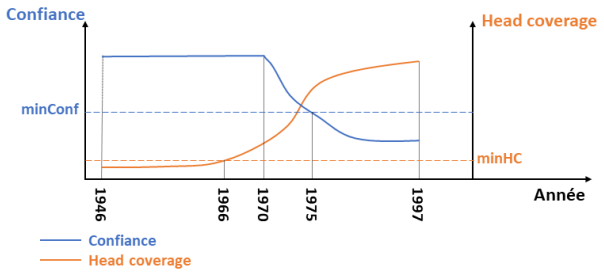

FIGURE 2 – *Évolution de la confiance et du head coverage d'une règle concluant sur "positive"*

5 Expérimentations

Nous avons évalué notre approche sur un GC peuplé à partir d'un ensemble de diagnostics fournis par le CSTB. Il comporte 51970 triplets qui décrivent 2998 instances de produit, 341 localisations, 214 structures et 94 bâtiments. L'année de construction de ces bâtiments varie entre 1948 et 1997. Nous avons 1525 produits contenant de l'amiante et 1473 produits sont sans amiante (les données sont disponibles dans le GitHub [3]).
Le but de l'expérimentation est (1) d'apprendre des règles sur un sous-ensemble de diagnostics et d'étudier la qualité de la prédiction qui peut être faite sur les produits restants (2) de comparer ces résultats à une approche naïve qui n'utilise que les classes de produits (3) de comparer les résultats de notre approche avec AMIE3 [5]. Pour évaluer notre approche, nous avons divisé les données du GC en 3 tiers, et nous avons réalisé une validation croisée. Comme nous disposons de nombreuses classes de produits de tailles différentes, nous avons fixé un seuil de *head-coverage* à $minHC = 0,001$ pour observer le plus de règles possible, puis nous avons évalué les résultats lorsque $minConf$ varie de $0,6$ à 1 en utilisant les mesures classiques de précision, rappel, F-Mesure et exactitude (accuracy). Le nombre maximum de frères a été fixé à 0 pour les structures et à 3 pour les localisations et les produits.

La table 1 montre que CRA-miner découvre 75 règles en moyenne. Les résultats montrent que des composants frères sont effectivement exploités pour prédire la présence d'amiante : 29 règles comportent au moins un produit frère (au maximum 2) et 16 règles comportent une localisation frère. Parmi les 75 règles, 14 d'entre elles exploitent une contrainte temporelle.

Nous avons utilisé une approche pessimiste qui consiste à choisir de classer un produit comme positif si au moins une règle découverte conclut à la présence d'amiante.

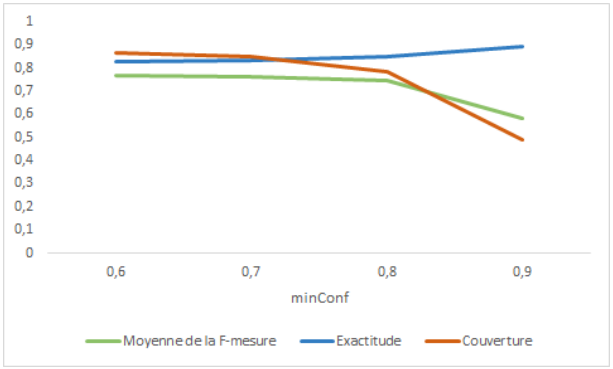

FIGURE 3 – *Résultats de CRA-Miner selon différents seuils de minConf*

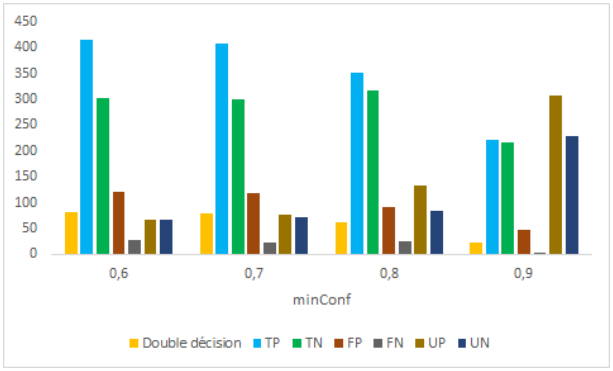

FIGURE 4 – *Résultats détaillés de CRA-Miner selon différents seuils de minConf*

La Figure 3 présente la moyenne de la F-mesure, l'exactitude et la couverture des données (i.e. ratio des produits qui peuvent être classés dans l'ensemble des produits de test), lorsque le seuil $minConf$ varie. Quand le seuil $minConf$ augmente, la couverture des données diminue mais l'exactitude augmente. La meilleure moyenne de F-mesure, $0,77$, (moyenne entre la F-mesure positive et négative) est obtenue pour un $minConf$ fixé à $0,6$. Avec ce seuil, nous pouvons décider pour 87% de l'échantillon de test. La Figure 4 présente les résultats détaillés (TN : vrais négatifs, TP : vrais positifs, FN : faux négatifs, FP : faux positifs, UN : négatifs non classifiés, UP : positifs non classifiés) et le nombre de produits qui ont été classés à la fois comme positifs et négatifs par différentes règles (double décision). Ce chiffre montre que le seuil de $0,6$ entraîne seulement 82 décisions contradictoires parmi les mille produits.
Plus précisément, les vrais positifs TP (resp. Vrais négatifs TN) sont les produits contenant de l'amiante (resp. ne

3. https://github.com/ThamerMECHARNIA/DATA-IC2021

Statistiques	CRA-Miner	$l = 4$	$l = 6$	Baseline
# règles	75	45	91	24
Double décision	82	50	277	0
TP	415	381	473	146
TN	303	288	264	257
FP	121	146	226	30
FN	28	74	32	24
UP	66	54	3	338
UN	67	58	0	204
Pos. précision	**77%**	72%	68%	83%
Pos. rappel	82%	75%	**93%**	29%
Pos. F-mesure	**0,79**	0,73	**0,79**	0,43
Neg. précision	**92%**	80%	89%	91%
Neg. rappel	**62%**	59%	54%	52%
Neg. F-mesure	**0,74**	0,68	0,67	0,66
Moy. F-mesure	**0,77**	0,71	0,73	0,55
Exactitude	**0,83**	0,75	0,74	0,88
Couverture	87%	89%	**100%**	46%

TABLE *1 – Comparaison de CRA-Miner avec une approche non contextuelle (baseline) et AMIE3 avec* $l = 4$ *et* $l = 6$ *(minHC=0.001, minConf=0.6)*

contiennent pas de l'amiante) classés par les règles découvertes comme positifs (resp. négatifs). Les faux positifs FP (resp. Faux négatifs FN) sont les produits sans amiante (resp. avec amiante) classés par les règles comme positifs (resp. négatifs), tandis que les produits non classés sont soit positifs (UP), soit sans amiante (UN) dans le KG.
Nous avons comparé l'approche contextuelle CRA-Miner avec une baseline exploitant uniquement la classe de produit. Cela nous permet d'estimer l'intérêt de considérer la hiérarchie des produits et le contexte dans lequel ils ont été utilisés. La Figure 5 montre que la F-mesure et la couverture sont beaucoup plus faibles quel que soit le seuil $minConf$. Ainsi, pour $minConf = 0.6$, le tableau 1 montre que la baseline ne permet de classer que 46% des échantillons de test et obtient une moyenne de F-mesure de $0,55$. En effet, CRA-miner permet de découvrir des règles contextuelles complexes telles que :

"*plaster-based_plaster_or_smooth_sprayed_cement_under_floats(?P), has_location(?S, ?L), contain(?L, ?P), underlays_of_wall_fabrics(?P2), contain(?L, ?P2), has_structure(?B, ?S), has_year(?B, ?Y), has_diagnostic_characteristc(?P, ?D), lessThanOrEqual(?Y, "1997-01-01T00 :00 :00") → has_diagnostic(?D, "positive")*"

ou

plaster_or_cement_based_coating(?P), has_location(?S, ?L), contain(?L, ?P), smoothing_bubbling_leveling_plasters(?P2), contain(?L, ?P2), has_structure(?B, ?S), has_year(?B, ?Y), has_diagnostic_characteristic(?P, ?D), lessThanOrEqual(?Y, "1991-01-01T00 :00 :00") → has_Diagnosis(?D, "positive")

Nous avons également comparé ces résultats avec ceux pouvant être obtenus avec AMIE3 [5] en utilisant les mêmes seuils de $minConf$ et $minHC$, et en fixant le nombre de prédicats des règles recherchées à $l = 4$ et $l = 6$ (cf. table 1)[4]. Notre approche permet d'atteindre une meilleure F-mesure que celle obtenue avec [5] (0,77 contre 0,73 pour $l = 6$, $l = 6$ étant le nombre de prédicats permettant à AMIE3 d'obtenir les meilleurs résultats en termes de F-Mesure et d'exactitude). AMIE3 a pu découvrir 91 règles (75 avec notre approche) ce qui lui permet de couvrir 100% des données test (87% avec notre approche). En revanche, AMIE3 obtient une exactitude plus faible (0,74 contre 0,83 avec CRA-Miner). Cette importante couverture est accompagnée de nombreuses doubles décisions (277). L'approche suivie étant pessimiste (i.e. si un produit est associé à deux décisions différentes, on le considère comme amianté), AMIE3 trouve plus de TP (473 contre 415 avec CRA-Miner) mais également deux fois plus de FP (226 contre 121 avec CRA-Miner), et les TN sont aussi moins nombreux (seulement 264 contre 303 avec CRA-Miner). De manière générale, le fait de disposer d'un contexte sémantique et de pouvoir représenter des intervalles de temps permet de découvrir des règles comportant plus de prédicats tout en améliorant la lisibilité des contraintes temporelles pour un expert du domaine (i.e. une règle sera définie pour un intervalle de temps, ce qui n'est pas possible avec AMIE3 où une année ne pourra apparaître que sous forme d'une constante).

Ces expérimentations ont tout d'abord montré que tous les prédicats du contexte qui ont été sélectionnés par l'expert sont pertinents pour classer le produit. En effet, la baseline

4. Même si AMIE3 est utilisée pour chercher uniquement des règles concluant sur $has\_Diagnosis$, une longueur > 6 ne permet pas d'obtenir de résultats en moins d'une semaine

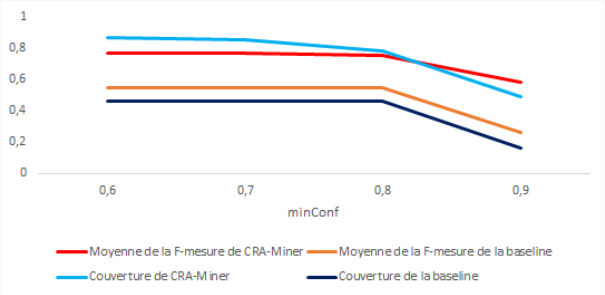

FIGURE 5 – *Comparaison entre CRA-Miner et la baseline non contextuelle selon différents seuils de minConf*

obtient un rappel très faible, et les résultats montrent que tous les prédicats ont été utilisés dans au moins une règle. La comparaison avec un autre système d'exploration de règles [5] montre que CRA-Miner obtient les meilleures valeurs de précision, F-mesure et exactitude, avec une valeur de couverture plus faible mais élevée (87%). Puisqu'il est plus important de détecter les exemples positifs que négatifs, nous avons choisi d'appliquer une stratégie pessimiste, et les résultats montrent que nous obtenons un meilleur rappel pour les exemples positifs que pour les exemples négatifs. Cependant, ce choix affecte la précision des positifs et d'autres stratégies pourraient être envisagées (ex : stratégies de vote, règles ordonnées en fonction de leur sémantique et/ou de leur confiance).

6 Conclusion

Dans ce papier, nous avons présenté l'approche de découverte de règles CRA-Miner qui prédit la présence d'amiante dans les produits en se basant sur un contexte sémantique, des heuristiques dédiées aux propriétés de type part-of et des contraintes temporelles utilisant des seuils calculés par le système. Les expérimentations montrent qu'une bonne F-mesure et une bonne couverture peuvent être obtenus et que ces résultats sont meilleurs que ceux pouvant être obtenus par un autre système de type générer et tester tel que AMIE3.

Dans des travaux futurs, nous envisageons d'explorer la possibilité de combiner cette approche avec une approche hybride telle que définie dans [6] afin d'en améliorer la couverture. Comme les résultats de cette approche doivent être utilisés par les experts amiante du CSTB pour prioriser les produits à diagnostiquer, nous devons également ordonner les produits positifs, non classifiés et négatifs en fonction des règles appliquées, et définir une interface qui permet de présenter et expliquer cet ordre aux experts.

Références

[1] Hendrik Blockeel and Luc De Raedt. Top-down induction of first-order logical decision trees. *Artificial intelligence*, 101(1-2) :285–297, 1998.

[2] Claudia d'Amato, Andrea G. B. Tettamanzi, and Duc Minh Tran. Evolutionary discovery of multi-relational association rules from ontological knowledge bases. In Eva Blomqvist, Paolo Ciancarini, Francesco Poggi, and Fabio Vitali, editors, *EKAW 2016, Italy, November 19-23, 2016, Proceedings*, volume 10024 of *Lecture Notes in Computer Science*, pages 113–128, 2016.

[3] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. DL-FOIL concept learning in description logics. In *Inductive Logic Programming, ILP 2008, Czech Republic, September 10-12, 2008, Proceedings*, volume 5194 of *Lecture Notes in Computer Science*, pages 107–121, 2008.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs : Methods and applications. *IEEE Data Eng. Bull.*, 40(3) :52–74, 2017.

[5] Jonathan Lajus, Luis Galárraga, and Fabian Suchanek. Fast and exact rule mining with amie 3. In *Extended Semantic Web Conference (ESWC)*, volume 12123 of *Lecture Notes in Computer Science*, pages 36–52. Springer, 2020.

[6] Thamer Mecharnia, Lydia Chibout Khelifa, Nathalie Pernelle, and Fayçal Hamdi. An approach toward a prediction of the presence of asbestos in buildings based on incomplete temporal descriptions of marketed products. In Mayank Kejriwal, Pedro A. Szekely, and Raphaël Troncy, editors, *K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*, pages 239–242. ACM, 2019.

[7] Stephen Muggleton and Luc De Raedt. Inductive logic programming : Theory and methods. *J. Log. Program.*, 19/20 :629–679, 1994.

[8] S. Ortona, V. V. Meduri, and P. Papotti. Robust discovery of positive and negative rules in knowledge bases. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1168–1179, 2018.

[9] Heiko Paulheim, Volker Tresp, and Zhiyuan Liu. Representation learning for the semantic web. *J. Web Semant.*, 61-62 :100570, 2020.

[10] Giuseppe Rizzo, Nicola Fanizzi, and Claudia d'Amato. Class expression induction as concept space exploration : From dl-foil to dl-focl. *Future Gener. Comput. Syst.*, 108 :256–272, 2020.

Vers des Classifieurs Ontologiquement Explicables

G. Bourguin, A. Lewandowski, M. Bouneffa, A. Ahmad

Université du Littoral Côte d'Opale, LISIC

{gregory.bourguin, arnaud.lewandowski, mourad.bouneffa, adeel.ahmad}@univ-littoral.fr

Résumé

Répondant au besoin d'explicabilité des IA qui utilisent l'Apprentissage Profond (AP), ce papier explore les apports et la faisabilité d'un processus de création de classifieurs explicables basés sur des ontologies. La démarche est illustrée par l'utilisation de l'ontologie des Pizzas pour créer un classifieur d'images qui fournit des explications visuelles impliquant une sélection de features ontologiques. Nous proposons une implémentation en complétant un modèle d'AP avec des tenseurs ontologiques générés à partir de l'ontologie exprimée avec la Logique de Description.

Mots-clés

Apprentissage automatique, ontologie, explicabilité, classifieur.

Abstract

In order to meet the explainability requirement of AI using Deep Learning (DL), this paper explores the contributions and feasibility of a process designed to create ontologically explainable classifiers while using domain ontologies. The approach is illustrated with the help of the Pizzas ontology that is used to create an image classifier that is able to provide visual explanations concerning a selection of ontological features. The approach is implemented by completing a DL model with ontological tensors that are generated from the ontology expressed in Description Logic.

Keywords

Machine learning, ontology, explainability, classifier

1 Introduction

Ces dernières années ont été marquées par une large démocratisation de solutions basées sur l'Apprentissage Automatique (AA), en particulier l'Apprentissage Profond (AP). Si la prolifération de ces nouveaux outils destinés à supporter des utilisateurs dans des tâches très diverses a démontré leur grande utilité, elle s'est aussi accompagnée de questionnements concernant la confiance que l'on peut leur accorder. De nombreux papiers de recherche ont ainsi souligné le problème de l'opacité des algorithmes d'AA, et le besoin prégnant envers de nouvelles solutions permettant d'entrouvrir ces boites noires pour faciliter leur acceptation. Cette problématique est au cœur du mouvement XAI [2] (eXplainable AI) et d'un grand nombre de travaux de recherches tournés vers l'explicabilité des IA (cf. partie 2).

Comme le souligne [12], expliquer clairement à un utilisateur final le rationnel ayant mené à une décision peut être aussi important que la décision elle-même. Pour ce faire, il est nécessaire que la solution proposée soit non seulement capable de fournir des explications quant à ses décisions, mais aussi que ces explications soient compréhensibles par l'utilisateur, c'est- à-dire qu'elles soient en adéquation avec son niveau d'abstraction. Le niveau d'abstraction auquel les explications doivent être proposées dépend donc des connaissances de l'utilisateur, de son expertise, voire de son point de vue. Le défi pour les concepteurs est alors de créer des IA explicables capables de combler l'écart sémantique entre les entités manipulées par les algorithmes, et celles permettant d'expliquer leurs décisions.

Issues du domaine de l'Ingénierie des Connaissances, les ontologies ont pour but de capturer les connaissances liées aux domaines d'expertises des utilisateurs, et de permettre aux algorithmes de les manipuler. Les ontologies sont d'ores et déjà utilisées par les chercheurs en AA dans le but d'augmenter les jeux de données utilisés pour entrainer les modèles : sélection de données basées sur leurs propriétés ontologiques, ajouts d'étiquettes déduites de l'ontologie. Cependant, l'utilisation des ontologies qui nous intéresse dans ce papier est celle qui implique des moteurs d'inférence dans le calcul du résultat : les chercheurs utilisent les algorithmes de l'AA pour identifier des entités ontologiques de « bas niveau » (ex. la présence d'objets dans une image), puis injectent ces informations dans un moteur d'inférence afin d'effectuer des déductions à haut niveau d'abstraction (ex. classification de l'activité humaine) [3]. Ce type de processus est particulièrement intéressant du point de vue de l'explicabilité du fait que l'inférence ontologique est un processus déductif qui peut être expliqué. Toutefois, la littérature souligne que l'inférence ontologique coûte cher [8], et ce type de processus n'est généralement mis en œuvre que lorsque la tâche de classification est trop complexe pour les algorithmes d'AA classiques.

Prenant acte du besoin d'explicabilité, l'objectif de ce papier est d'explorer un processus de création de classifieurs automatiques capables de fournir des explications fondées sur une ontologie. Nous ne focalisons pas ici sur les moyens permettant d'améliorer la classification, mais sur les apports d'une ontologie pour l'explicabilité. Nous explorons aussi la faisabilité d'une telle approche en utilisant des outils classiques de l'AP, et proposons une solution permettant de compléter un modèle d'AP avec des tenseurs (au sens

de Tensorflow [1]) générés à partir d'assertions exprimées en Logique de Description (DL).
Nous avons choisi d'exemplifier notre démarche en réutilisant la fameuse ontologie des Pizzas proposée par l'Université de Manchester. Notre objectif n'étant pas la performance de classification, le but de notre classifieur est d'étiqueter des images synthétiques représentant des pizzas avec les classes définies dans l'ontologie. Du point de vue de l'explicabilité, il s'agit de générer des heatmaps différenciant les garnitures (toppings) qui correspondent aux définitions ontologiques des classes.
La 2ème partie de ce papier propose un état de l'art du besoin et des solutions pour l'explicabilité des IA, ceci en focalisant sur les approches qui veulent y intégrer des connaissances grâce aux ontologies. La 3ème partie illustre les apports d'une approche fondée sur un raisonnement ontologique pour l'explicabilité, et introduit la démarche générique que nous proposons. La 4ème partie en présente l'application via l'implémentation d'un classifieur d'images ontologiquement explicable. La 5ème partie présente des réflexions qui découlent de cette expérience, avant de conclure dans la partie 6.

2 Explicabilité

Il est un consensus établi sur l'importance que revêt l'utilisation d'IA dotées de capacités d'apprentissage, de raisonnement et d'adaptation pour l'accomplissement de tâches informatiques de plus en plus complexes indispensables au développement des activités humaines [26]. Les systèmes basés sur l'AP sont de plus en plus performants, mais deviennent en corollaire de plus en plus complexes et opaques. Ils apparaissent comme des boites noires [9], rendant très problématique l'intervention humaine pour la compréhension de leurs décisions, ainsi que pour le contrôle de leur exécution, de leur déploiement, et de leur évolution [2][13]. En conséquence, le besoin de transparence et surtout d'explicabilité des IA s'est révélé crucial avec des aspects liés à la confiance qu'un utilisateur peut leur accorder en matière de sureté de fonctionnement de systèmes critiques pilotés par l'IA, mais également en matière d'éthique et du respect de règles légales et sociales telles que la non ségrégation et le respect de la vie privée [11].
Dans ce papier, nous nous intéressons à l'explicabilité de l'IA en tant que justifications des décisions compréhensibles par les utilisateurs. La nécessité de fournir des explications est un besoin ancien étant apparu dès les premières implémentations de systèmes experts [25]. Les systèmes à base de règles, ainsi que les algorithmes d'AA réputés plus transparents, comme la régression linéaire et les arbres de décision, ont également besoin d'outils simplifiant, résumant, et expliquant leurs prédictions. Cela peut se traduire par une restitution de la trace d'exécution des règles appliquées pour aboutir à une décision, ou encore par la simplification d'un arbre de décision en remplaçant des nœuds et arcs de niveaux de granularité fine par des concepts plus généraux issus de la terminologie du domaine de l'utilisateur [2]. Cependant l'application de ces techniques s'avère très difficile, voire impossible, en ce qui concerne l'explicabilité des systèmes à base d'AP.

1. https ://www.tensorflow.org/guide/tensor

2.1 Les méthodes *Post hoc*

Les outils pour l'explicabilité des systèmes d'AP mettent majoritairement en œuvre des approches dites *post hoc* permettant de fournir des explications sur des modèles préexistants. La plupart de ces méthodes sont aussi appelées agnostiques du fait qu'elles sont applicables à tout algorithme d'AP.
Les techniques d'explication utilisent en majorité les facteurs d'importance des features d'entrée et reposent sur l'idée d'associer à chacune une valeur traduisant l'importance de son rôle dans la prédiction. Il est ainsi possible d'obtenir des explications sur une prédiction particulière, ou des explications plus globales exprimées par différents graphiques associant feature, facteur d'importance, et prédictions.
Un des domaines où ce type de travaux est le mieux représenté est celui de la vision par ordinateur (CV, Computer Vision) basée sur les réseaux de neurones convolutionnels (CNN). En CV, les features correspondent aux pixels d'une image fournie en entrée. Les propositions consistent à mettre en correspondance les prédictions et les pixels qui ont conduit à une classification [27]. Pour ce faire, diverses approches ont été adoptées avec en particulier les travaux consistant à explorer l'architecture des réseaux pour déterminer comment les couches intermédiaires perçoivent le monde extérieur [23]. Une des méthodes les plus représentatives de ces travaux est la méthode Grad-CAM (Gradient-Weighted Class Activation Mapping) [28] (et ses dérivées) qui utilise le gradient d'un concept cible pour produire des heatmaps identifiant les régions de l'image qui ont le plus participé à sa reconnaissance.
Les techniques utilisant une identification des features d'entrée pour expliquer un modèle ne sont pas limitées au domaine de la CV. Ainsi, des outils tels que LIME (Local Interpretable Model-Agnostic Explanations) [19] permettent aussi bien d'identifier des pixels dans une image pour un problème de CV, que d'identifier les termes participant à une prédiction dans un modèle de NLP (Natural Language Processing). La technique utilisée est l'explication par simplification qui consiste à construire des modèles linéaires sur des sous-parties du système global : il s'agit de produire des explications à partir de perturbations locales en simulant le fonctionnement du modèle boîte noire par un modèle naturellement transparent. Si dans le cas de LIME, les perturbations sont générées par le système, d'autres systèmes permettent d'explorer les prédictions à la suite de changements de valeurs de certaines features dans un processus interactif d'explication et par une analyse de type What-If [17], fournissant ainsi une sorte d'analyse contrefactuelle. Enfin, on peut aussi citer l'outil SHAP [15] (SHapely Additive exPlanation) qui s'inspire de la théorie des jeux, ou plus particulièrement de celle qui consiste à trouver la manière la plus équitable de distribuer les gains aux joueurs en

se basant sur leur taux de contribution lors de la partie. Les joueurs représentent ici les features du modèle. Une revue systématique de ce type d'approche est effectuée dans [30]. Les techniques et outils évoquées ci-avant se sont avérés très utiles dans de nombreux travaux pour fournir des explications sur le fonctionnement de modèles d'IA. Cependant, comme le souligne [14], et comme nous le verrons dans la partie 3.2, ces approches ne garantissent aucunement que les explications fournies soient compréhensibles par les utilisateurs.

2.2 Explicabilité & Ontologies

Un des objectifs principaux de l'explicabilité est de fournir aux utilisateurs, souvent des spécialistes de domaines, une description compréhensible de la manière dont le système a produit une prédiction, ou des facteurs clés qui ont conduit à cette prédiction.

L'apport des ontologies dans l'explicitation et l'axiomatisation de la sémantique d'un domaine n'est plus à démontrer. Pour de nombreux auteurs, il est apparu évident que les ontologies peuvent servir à fournir des explications adéquates et cohérentes aux conclusions d'un modèle d'AP [5]. Par exemple, dans [22], les auteurs opèrent une méthode *post-hoc* de mise en correspondance entre l'entrée d'un réseau de neurones et les classes d'une ontologie suggérée, et génèrent automatiquement des règles en Logique de Description (DL) à partir des instances classifiées pour obtenir des expressions qui opèrent comme des explications.

Dans [1] les auteurs proposent une architecture de réseaux de neurones dans laquelle des couches dites sémantiques sont introduites pour produire des explications. Ce type d'approche est formalisée dans [14] et développée plus avant dans [16] sous le concept de *semantic bottleneck* : il s'agit de construire un classifieur qui intègre dès sa conception des couches sémantiques spécifiques qui permettent au module d'AP d'extraire des features sémantiques qui sont elles-mêmes utilisées pour calculer la classification finale. La contribution pondérée des features sémantiques permet de fournir des explications quant à la prédiction, et aide à comprendre les erreurs de classification. Il faut toutefois noter que même si ces travaux parlent de sémantique, ils ne mettent pas en œuvre une approche ontologique.

Enfin, nous souhaitons citer les travaux tels [3][6] qui s'intéressent à l'interprétation d'images fondée sur des ontologies. Dans ces travaux, un processus d'AP comme la détection d'objets ou la segmentation sémantique est utilisé pour identifier des features qui correspondent aux concepts de l'ontologie, et qui servent ensuite à inférer des déductions de plus haut niveau d'abstraction. Ces travaux impliquent un raisonnement à base de DL dans le but de rendre possible ou d'améliorer des tâches de classification complexes. Même s'ils ne focalisent pas explicitement sur la problématique d'explicabilité, une classification basée sur la DL est intrinsèquement explicable, et ces solutions peuvent de fait fournir des explications au niveau d'abstraction de l'ontologie impliquée.

2.3 Positionnement

Notre but est de fournir des explications tout en mettant en exergue, dans les données d'entrée, les features qui ont participé aux prédictions. De ce point de vue, nous nous inspirons des outils tels que Grad-CAM ou LIME. Cependant, à la différence de ces outils, notre approche n'est pas agnostique : même si notre démarche se veut générique, les explications que nous voulons fournir sont intimement liées au domaine visé et les features que nous voulons mettre en exergue se doivent d'être au niveau d'abstraction des utilisateurs, c.à.d, de notre point de vue, des features ontologiques.

L'approche que nous développons n'est pas non plus *post-hoc* puisque nous verrons que l'ontologie est ici directement impliquée dans le processus même de création du classifieur. Elle est similaire à celles utilisant des *semantic bottlenecks*, à la différence près que la sémantique est ici fournie par une ontologie qui, de plus, sert directement au calcul de la prédiction.

De ce point de vue, nous sommes fortement inspirés par les travaux impliquant ontologies et DL pour l'interprétation d'images à haut niveau d'abstraction. Notre démarche est cependant aussi quelque peu différente en focalisant résolument sur le problème d'explicabilité, et en proposant des classifieurs explicables qui impliquent des ontologies y compris dans des tâches de classification qui n'en auraient *a priori* pas besoin. Enfin, l'interprétation à haut niveau d'abstraction peut impliquer des moteurs d'inférence externes qui s'intègrent difficilement dans un pipeline classique d'AA : comme le souligne [8], l'inférence ontologique est un mécanisme coûteux. C'est pourquoi nous proposons une mise en œuvre qui n'utilise pas de moteur d'inférence ontologique « classique » comme Jena, Hermit ou Pellet, mais qui repose sur des modules de raisonnement spécifiques que nous générons automatiquement à partir des définitions de l'ontologie, et qui utilisent les mêmes technologies d'implémentation que les modèles d'AP (cf. 4.2).

3 Explicabilité Ontologique

3.1 Domaine d'illustration : les pizzas

Pour illustrer notre démarche, nous avons choisi de réutiliser l'ontologie des Pizzas (Université de Manchester). Les raisons sont multiples, mais la principale est le fait que cette ontologie est accessible et possède une très grande notoriété.

L'ontologie des pizzas définit un ensemble de classes de pizzas (ex. *Napoletana*), sous-classes de la classe *NamedPizza* (elle-même sous-classe de *Pizza*). Les définitions utilisent principalement la propriété d'objet *hasTopping* dont le domaine est la classe *Pizza*, et l'image est la classe *PizzaTopping* qui est la superclasse de garnitures telles que les anchois (*AnchoviesTopping*), etc.

L'ontologie définit 22 sous-classes de *NamedPizza* à partir de 36 sous-classes de *PizzaTopping*. Pour simplifier la construction du jeu de données (les images de pizzas), nous avons choisi de focaliser sur 14 sous-classes de *NamedPizza* en impliquant 16 sous-classes de *PizzaTopping*. On

trouvera ainsi par exemple la *Napoletana* définie par :

$$\begin{aligned} Napoletana \equiv & Pizza \\ & \sqcap (\exists\, hasTopping.AnchoviesTopping) \\ & \sqcap (\exists\, hasTopping.OliveTopping) \\ & \sqcap (\forall\, hasTopping \\ & .(AnchoviesTopping \sqcup OliveTopping)) \end{aligned} \quad (1)$$

Notre objectif étant de mettre en relation les résultats d'un classifieur d'images avec les définitions de l'ontologie, nous avons constitué un jeu de données dont les exemples sont étiquetés avec les sous-classes de *NamedPizza*, et dont les images correspondent à l'ontologie, c'est-à-dire que les toppings apparaissant dans l'image étiquetée correspondent à la définition ontologique de l'étiquette.
D'autres chercheurs ont déjà constitué des jeux de données contenant des images de pizzas [18] : aucun ne correspond aux définitions fournies par l'ontologie. De plus, notre but dans ces expérimentations n'est pas d'optimiser la classification (en termes de précision, etc.), mais d'étudier les apports et la faisabilité d'une démarche impliquant une ontologie pour améliorer l'explicabilité d'un classifieur. Inspirés par les travaux de [18] qui génèrent des images de pizzas synthétiques pour obtenir un jeu de données contrôlé, nous avons mis en œuvre une méthode similaire et créé un module Pizzaïolo qui génère des images synthétiques de pizzas à partir de l'ontologie en combinant des cliparts de toppings (cf. Figure 1). Les images utilisent volontairement la même base (seule la répartition des toppings varie en nombre, position et orientation) de manière à forcer un quelconque classifieur (non ontologique) à focaliser sur les toppings pour différencier les pizzas.

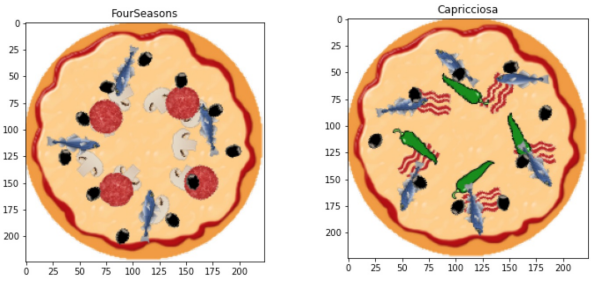

FIGURE 1 – Pizzas synthétiques ontologiques.

La tâche de classification de ces images étant assez simple, nous n'avons généré qu'un «petit» jeu de données totalement équilibré de 200 pizzas par classe.

3.2 Problèmes d'approche non ontologique

Pour illustrer les problèmes liés aux outils pour l'explicabilité, nous avons construit et entrainé un classifieur « classique » basé sur un CNN dont la base est formée par une architecture VGG19 [24] pré-entrainé sur Imagenet [20], à laquelle nous avons simplement ajouté une couche Dense (256) puis un SoftMax (14 classes de pizzas). Les données étant simples, le classifieur a pu être entrainé pour atteindre une précision de 100% sur un ensemble de test constitué de 20% des échantillons.
Nous avons ensuite utilisé les outils dérivés de LIME [19] et Grad-CAM [23] qui expliquent le résultat d'une classification en générant une heatmap mettant en valeur les pixels de l'image qui ont principalement participé à la prédiction. Nos images étant constituées de manière à ce que les seuls éléments qui différencient les pizzas soient les toppings présents sur l'image, on peut espérer que les heatmaps focalisent sur les pixels qui y correspondent.

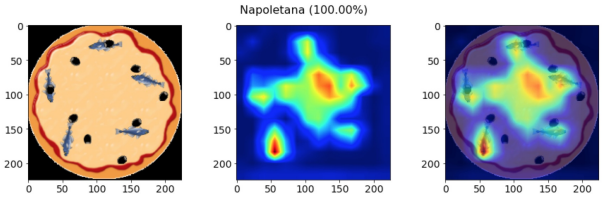

FIGURE 2 – Explication fournie par Grad-CAM.

La Figure 2 montre les résultats obtenus avec Grad-CAM pour la classification d'une *Napoletana* qui est constituée uniquement d'olives et d'anchois (cf. définition partie 3.1). Les résultats issus de LIME et Grad-CAM sont similaires. On peut constater que le CNN focalise bien sur les anchois. Cependant, il ignore les olives, tout en focalisant aussi sur une partie de la base (vide de toppings). On peut alors considérer que pour le CNN, cette pizza est une *Napoletana*, du fait qu'elle a des anchois et du vide : ce qui ne correspond bien entendu pas à la définition que l'on attendait.

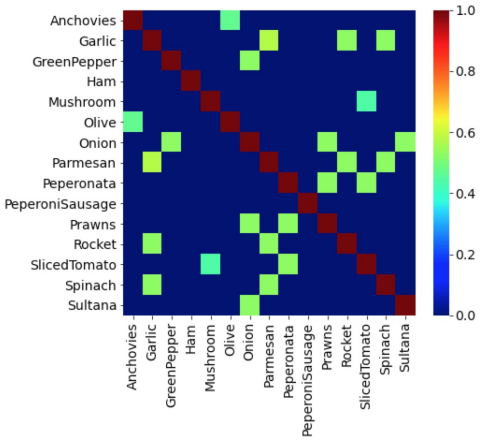

FIGURE 3 – Corrélation ontologique des toppings.

Toutefois, l'ontologie des pizzas permet d'expliquer ce phénomène si on l'utilise pour générer une matrice de corrélation ontologique des toppings (Figure 3). Cette matrice révèle en quelle mesure les toppings sont corrélés dans les définitions ontologiques des pizzas : on peut constater que les anchois (*AnchoviesTopping*) apparaissent systématiquement avec des olives (*OliveTopping*). Par contre, les olives apparaissent fréquemment avec d'autres toppings. De fait, pour le CNN, sur une *Napoletana* constituée uniquement

d'anchois et d'olives, le discriminant est la présence d'anchois. De plus, l'étude des définitions de toutes les classes de l'ontologie (non illustrée ici) révèle que la *Napoletana* est la pizza qui possède le moins de toppings, ce qui peut expliquer pourquoi le CNN a aussi considéré la zone vide comme un discriminant.

Ces remarques n'ont aucunement pour but de discréditer les outils tels que Grad-CAM et LIME. Comme nous venons de le montrer, ils se révèlent très utiles pour expliquer comment fonctionne un classifieur. Cependant, ces explications ne peuvent généralement être interprétées que par des spécialistes en IA et, comme l'ont souligné les travaux tentant d'associer une sémantique aux filtres des CNN [10], cet exemple démontre que le niveau d'abstraction des discriminants qui émergent de l'entraînement d'un CNN n'est pas en adéquation avec celui d'un expert des pizzas. De fait, ces outils ne paraissent pas les plus indiqués pour fournir des explications facilement interprétables par des experts du domaine.

3.3 Approche proposée

La démarche que nous proposons a pour but de créer un classifieur et de fournir des explications au niveau d'abstraction des experts du domaine d'application, c.à.d., de notre point de vue, dans les termes d'une ontologie. Dans notre exemple, il s'agit de classifier des images de pizzas avec les classes de l'ontologie, et de générer des heatmaps qui correspondent aux définitions ontologiques de ces classes.

Les étapes de la réalisation consistent à :

(a) Construire un ensemble C constitué des classes de l'ontologie prédites en sortie du classifieur.

(b) Soit D l'ensemble des axiomes de l'ontologie qui définissent les concepts de C :
D = {d | $\exists$ c $\in$ C, d $\equiv$ c est un axiome de l'ontologie}.
Soit P l'ensemble des propriétés de l'ontologie impliquées dans D.
Soit R l'ensemble des images de P tel que :
R = {r | r=range(p), p $\in$ P}.
Construire l'ensemble F des features ontologiques f $\in$ F, c.à.d. des triplets (c, p, r) impliqués dans D et qui seront utilisés dans l'explication d'une classification.

(c) Mettre en œuvre une technique d'AA permettant de construire l'ensemble FI $\subseteq$ F des features ontologiques identifiées (assertions satisfaites) dans une donnée envoyée au classifieur et qui sont de la forme FI = {fi $\in$ F | fi $\equiv$ $\exists$ p.r}

(d) Mettre en œuvre un raisonnement ontologique qui utilise D et FI pour calculer CI $\subseteq$ C, l'ensemble des classes ci identifiées pour une donnée.

(e) Utiliser l'ensemble des axiomes DI $\subseteq$ D tel que DI = {di $\equiv$ ci} et l'ensemble FI pour expliquer la classification CI.

Dans notre exemple :

(a) C = {c $\sqsubseteq$ Pizza}
ex. Napoletana

(b) D = {d $\equiv$ c}
ex. ($\exists$ hasTopping . AnchoviesTopping) $\sqcap$
($\exists$ hasTopping . OliveTopping) $\sqcap$
($\forall$ hasTopping .
(AnchoviesTopping $\sqcup$ OliveTopping))
P = { hasTopping }
R = {r $\sqsubseteq$ PizzaTopping }
ex. AnchoviesTopping
F = {(c $\sqsubseteq$ Pizza, hasTopping, r $\sqsubseteq$ PizzaTopping)}
ex. (Napoletana, hasTopping, AnchoviesTopping)

(c) Module de segmentation sémantique (cf. 4.1)

(d) Module OntoClassifier (cf. 4.2)

(e) Projection des assertions OWL (cf. 4.3)

Il est à noter que les étapes (b) et (c) sont fortement liées du fait qu'il serait vain de construire F avec des features ontologiques qui ne pourraient pas être extraites des données. Dans notre exemple, nous avons focalisé sur la propriété *hasTopping* du fait qu'elle est en adéquation avec les définitions des pizzas, mais aussi parce que la présence des toppings (éléments de R) peut être déduite de l'image. Le fait qu'il n'y ait ici qu'une seule propriété dans P (*hasTopping*) est lié à l'exemple : il serait tout à fait possible de considérer plusieurs propriétés différentes à extraire des données d'entrée pour inférer une classification.

On peut aussi souligner que le niveau d'abstraction des explications est intimement lié au niveau d'abstraction des features ontologiques. En effet, si dans notre exemple il sera possible d'expliquer qu'une image représente une *Napoletana* du fait qu'elle est constituée d'anchois et d'olives, le classifieur n'aura pas d'explication à fournir sur la manière dont il a décidé qu'une zone de l'image représente un anchois. Pour pouvoir le faire, il faudrait raffiner l'ontologie en donnant une définition des toppings eux-mêmes à partir de features ontologiques de plus bas niveau. Néanmoins, il faut rappeler que toute démarche pour l'explicabilité est confrontée au fait qu'à un certain niveau d'abstraction, on considère ne plus devoir fournir d'explications. On peut par exemple citer [12] qui propose un classifieur d'espèces d'oiseau mêlant CNN et NLP pour proposer des explications : l'outil peut expliquer qu'une image représente un Albatros du fait que l'oiseau possède un bec jaune, etc., mais il ne tente pas de démontrer ce qu'est un bec jaune.

Enfin, on peut remarquer que cette démarche engendre la création d'un pipeline de classification possédant 2 entités principales : un module utilisant une technique d'AP pour extraire les features ontologiques, suivi d'un module de raisonnement ontologique. Comme nous l'avons souligné, cette décomposition est similaire à celle que l'on peut trouver dans les solutions dédiées à identifier des classes de haut niveau d'abstraction comme les activités humaines [3][6]. Cependant, dans ces travaux, l'ontologie est principalement mise en œuvre pour aider la classification. La démarche que nous proposons est de partir de l'ontologie dans le but explicite de fournir des explications, y compris pour des problèmes de classification qui n'auraient *a priori* pas besoin d'ontologie. Nous verrons aussi dans la suite du papier que nous proposons une solution originale nommée

OntoClassifier pour la partie raisonnement de notre pipeline de classification.

4 Classifieur Ontologique

Cette partie présente l'implémentation de notre démarche sur l'exemple des pizzas présenté précédemment. Cette implémentation est constituée de 2 principaux modules : un module d'AP de segmentation sémantique destiné à identifier les features ontologiques (FI) présentes dans une image, et un module ontologique nommé OntoClassifier destiné à calculer les classes CI qui peuvent être déduites de FI, tout en étant capable de fournir des explications. Ces 2 modules sont implémentés et associés en Tensorflow 2. Le pipeline général de notre classifieur est présenté dans la Figure 4. Une image fournie en entrée de ce pipeline est traitée séquentiellement (flèches vers la droite) par le module d'AP (cf. 4.1), puis par le module ontologique (cf. 4.2) pour obtenir en sortie l'ensemble CI des classes identifiées pour cette image. Le mécanisme d'introspection du module ontologique permet ensuite de fournir des explications à propos de chaque classe identifiée ci ∈ CI (flèches vers la gauche) en utilisant sa définition ontologique di ∈ DI et les features fi ∈ FI correspondantes qui peuvent de plus être projetées sur (mises en exergue dans) l'image de départ (cf. 4.3).

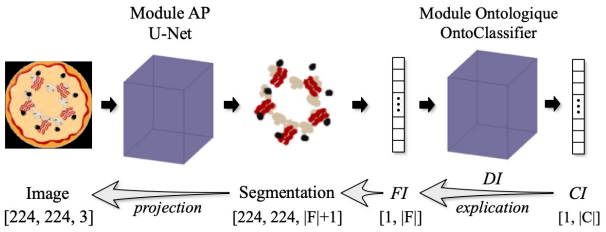

FIGURE 4 – Classifieur ontologiquement explicable.

4.1 Module AP : Segmentation Sémantique

La première partie du pipeline de classification a pour mission d'identifier les features ontologiques fi ∈ FI, c.à.d. la satisfiabilité des assertions correspondantes aux triplets de F sachant qu'ici :

$$F = \{(c \sqsubseteq \text{Pizza}, \text{hasTopping}, r \sqsubseteq \text{PizzaTopping})\}$$

Nous avons choisi d'utiliser une technique de segmentation sémantique dont l'objectif est d'étiqueter chaque pixel d'une image avec les classes de {r ⊑ PizzaTopping}. Notre jeu de données étant simple et totalement contrôlé, nous avons mis en œuvre une architecture de modèle basée sur U-Net [20], et généré les masques de segmentation nécessaires à l'entraînement du modèle en même temps que nos images. Cette implémentation de U-Net (basée sur MobileNetV2 [21] avec les poids d'Imagenet [7]) reçoit en entrée des images de pizzas constituées de 3 canaux (RGB) (224x224x3) pour obtenir en sortie une segmentation de l'image en 17 canaux (224x224x17). Chaque canal correspond à une des 16 sous-classes de *PizzaTopping*, excepté 1 canal destiné à recevoir les pixels qui ne correspondent à aucun topping.

La partie centrale de la Figure 4 montre comment une image fournie en entrée de l'U-Net est segmentée : pour représenter cette segmentation, nous avons ici superposé les différents canaux en leur attribuant chacun une couleur différente. Chaque couleur/canal correspond à une classe ontologique de topping (r ∈ R). Dans la suite du papier, cette représentation sera nommée *masque ontologique* dans le sens où les pixels de ce masque permettent (par projection) d'identifier la classe de topping à laquelle correspondent les pixels de l'image d'entrée.

4.2 Module Ontologique : OntoClassifier

La présence de pixels dans une couche de segmentation peut être interprétée comme la présence d'une feature ontologique (∃ hasTopping . topping), topping ∈ R, ce qui permet de déduire l'ensemble FI des assertions satisfaites pour chaque image traitée par le modèle. Il reste alors à raisonner à partir de FI en utilisant l'ensemble des définitions D pour en déduire les classes CI qui sont applicables à l'image.
Ce processus de raisonnement à partir de propriétés extraites de l'image est en partie similaire à celui qu'on peut trouver dans divers travaux mêlant AP et ontologies pour effectuer des interprétations à haut niveau d'abstraction. L'approche classique est de peupler l'ontologie avec des instances représentant les exemples à classifier, puis d'effectuer des déductions avec un raisonneur ontologique comme Jena, Hermit ou encore Pellet. Cependant, comme souligné dans [8], ce processus est coûteux du fait qu'il faille compléter le modèle d'AP par des outils externes, et que ces outils destinés à raisonner sur la globalité d'une ontologie sont bien plus lents que les pipelines d'AP utilisés pour créer des classifieurs. Dans la démarche que nous proposons, nous n'avons pas besoin de toute la puissance d'un raisonneur ontologique pour déduire CI à partir de FI et D. Nous avons donc créé le module OntoClassifier dont le constructeur génère un ensemble de tenseurs à partir de l'ontologie, en particulier de C, D et F.

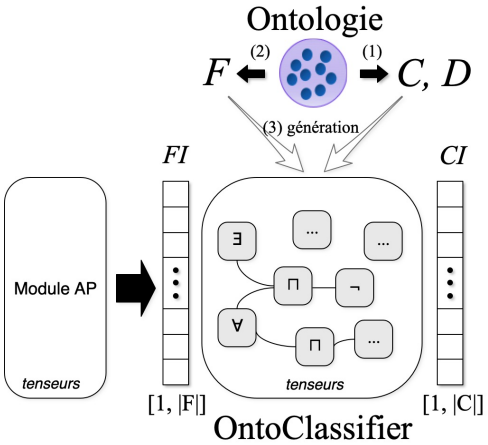

FIGURE 5 – Génération d'un OntoClassifier.

Ce processus est illustré dans la Figure 5 : après la sélection des classes et définitions visées pour construire C et D (1), et des features ontologiques qui constituent F (2),

un ensemble de tenseurs est automatiquement généré (3) : ces tenseurs sont typés et interconnectés grâce à la décomposition des expressions OWL contenues dans D. Les opérateurs de construction considérés sont la conjonction ($\sqcap$), la disjonction ($\sqcup$), la négation ($\neg$), les restrictions existentielles ($\exists r.c$), universelles ($\forall r.c$), et de cardinalité. L'OntoClassifier se base également sur l'hypothèse que les classes de R (range des features ontologiques) sont disjointes. Rappelons enfin que si dans l'exemple nous n'avons utilisé qu'une seule propriété ontologique (*hasTopping*), l'OntoClassifier est tout à fait capable de gérer un ensemble F contenant plusieurs propriétés (cf. 3.3).
L'OntoClassifier résultant est alors prêt à compléter le pipeline de classification en sortie du module d'AP (Figure 4, partie droite). Cet assemblage permet de calculer la satisfiabilité d'assertions ontologiques complexes comme la définition d'une *Napoletana* (cf. 3.1) ou encore des expressions impliquant des super-classes de toppings telles :

$$\text{CheesyPizza} \equiv \exists\ \text{hasTopping}\ .\ \text{CheeseTopping}$$

$$\text{VegetarianPizza} \equiv\ \neg\ (\exists\ \text{hasTopping}\ .\ \text{FishTopping}) \sqcap \neg\ (\exists\ \text{hasTopping}\ .\ \text{MeatTopping})$$

Enfin, l'OntoClassifier étant composé d'un graphe de tenseurs qui représente la décomposition des éléments de D, ce module permet de remonter le graphe des assertions pour identifier les éléments de FI – ou l'absence d'éléments – qui les ont satisfaites dans un exemple donné.

4.3 Résultats

Le pipeline de classification étant en place, il ne reste plus qu'à lui envoyer des images pour obtenir une classification. Pour commencer, nous aimerions souligner que la génération de l'OntoClassifier sous forme de tenseurs permet l'intégration directe de la dimension ontologique dans le pipeline de classification. Le modèle global résultant est alors bien plus rapide que dans le cas où le raisonnement ontologique est délégué à un moteur d'inférence externe. À titre d'exemple, en utilisant notre module AP de segmentation sémantique couplé à une instance de raisonneur Hermit sur nos machines (I9- 10850K à 3.6 GHz, 32 Go DDR4 3200MHz, GPU RTX 3080), il faut en moyenne 130s pour classifier 100 images de pizzas. Avec l'OntoClassifier, sur les mêmes machines, la classification des mêmes données prend en moyenne 1,6s. Il faut bien entendu relativiser cette différence car, comme nous l'avons souligné dans la partie 4.2, un module comme Hermit est destiné à raisonner sur une ontologie dans sa globalité en considérant de manière exhaustive l'ensemble des relations qui peuvent être inférées, alors que le raisonnement réalisé par un OntoClassifier focalise uniquement sur les relations ontologiques qui servent à calculer les classes de C, c'est à dire celles explicitement visées par le classifieur. Comme dans la partie 3.2, et sur le même jeu de données assez simples d'images synthétiques de pizzas, ce pipeline a pu être entraîné pour atteindre une précision de 100% sur un ensemble de test constitué de 20% des échantillons.
La Figure 6 montre l'exemple d'une image de *Fiorentina*. Le classifieur fournit la liste des classes qui ont été détectées (*Fiorentina*, *Cheesy*, *Vegetarian*). Le masque ontologique (résultant de la segmentation) fournit la liste des ingrédients. Ce masque peut être projeté sur l'image « à la Grad-CAM », mais nous avons choisi ici de l'afficher séparément pour plus de lisibilité. On peut noter que contrairement à ce que nous avions montré en 3.2, le niveau d'abstraction est ici en adéquation avec celui de l'ontologie : aucun topping n'est ignoré par le classifieur et dans le masque, les entités mises en valeur correspondent à la définition ontologique des classes, et chaque topping est différencié et identifiable grâce à un code couleur.

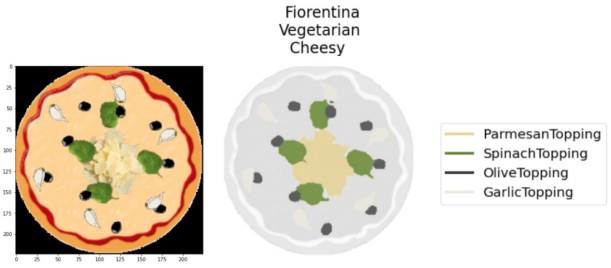

FIGURE 6 – Classification et segmentation ontologique.

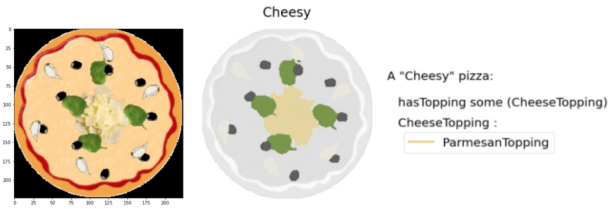

FIGURE 7 – Explications visuelles ontologiques.

Le système permet de plus de focaliser sur une des classes identifiées et d'utiliser l'introspection de l'OntoClassifier pour expliquer cette classification. Ainsi, la Figure 7 illustre une focalisation sur la classe *Cheesy* identifiée pour l'image de la Figure 6. Ce focus reprend le masque ontologique, et y ajoute (en partie droite) une explication qui met en correspondance la définition en OWL de la classe avec les pixels de l'image qui ont participé, en tant que features ontologiques, à la satisfiabilité des assertions qui la composent. Dans la même idée, la Figure 8 illustre le fait que l'OntoClassifier peut aussi expliquer le résultat d'une classification du fait de l'absence de features.

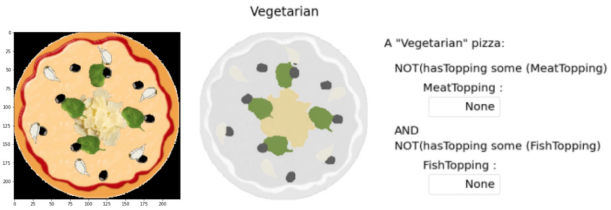

FIGURE 8 – Explications par des features absentes.

5 Discussion

Il est vrai que la démarche présentée ici demande un travail préparatoire supplémentaire autour de la création d'une ontologie pour réifier le niveau d'abstraction des utilisateurs, de la construction des ensembles C, D et F, ainsi que la mise en œuvre d'une technique d'AP plus complexe que pour une « simple » classification. On peut cependant aussi noter que cette approche, en plus de résulter en un classifieur ontologiquement explicable, possède d'autres avantages.
Tant que l'ensemble des features ontologiques (F) ne change pas, il est facilement possible d'ajouter de nouvelles classes dans C, d'en supprimer, ou de les modifier, et d'intégrer ces évolutions au pipeline de classification sans avoir à ré-entraîner le modèle. Il suffit de (re-)générer l'OntoClassifier pour que la nouvelle version intègre les nouvelles étiquettes, et soit capable d'expliquer pourquoi une image y correspond.
Comme le soulignent [29] et [4], la notion de point de vue est importante, y compris dans le domaine des ontologies : le sens des choses est pluriel, elles ont d'ailleurs souvent des définitions différentes selon les points de vue. La démarche et les outils que nous proposons peuvent apporter des éléments de réponse au besoin de multi points de vue puisqu'il est possible de générer plusieurs OntoClassifiers dédiés à des points de vue différents sur le même domaine. Pour ce faire, il est possible de composer différents ensembles C (un par point de vue), et de générer les différents OntoClassifiers qui viendront compléter le même module d'AP, proposant ainsi des pipelines de classification et d'explicabilité dédiés à chaque point de vue.
Enfin, notre démarche permet aussi d'introduire la notion de point de vue au niveau de l'explicabilité elle-même. En effet, pour un même ensemble de classes C, il est possible de créer différents ensembles de définitions D, et donc de générer des classifieurs explicables selon diverses définitions ontologiques. Dans l'ontologie des Pizzas, une *Vegetetarian* peut ainsi être définie de plusieurs manières :

(1) VegetarianPizza $\equiv \neg$ ($\exists$ hasTopping . FishTopping) $\sqcup$ $\neg$ ($\exists$ hasTopping . MeatTopping)

(2) VegetarianPizza $\equiv \forall$ hasTopping . VegetarianTopping

Si ces 2 définitions aboutissent à la même classification, elles permettent de créer des OntoClassifiers qui fourniront des explications visuelles avec des points de vue différents, focalisant sur (mettant en valeur) la présence ou l'absence soit (1) des toppings sous-classes de poissons et viandes (cf. Figure 8), soit (2) des toppings sous-classes d'ingrédients végétariens.

6 Conclusion

Répondant au besoin d'explicabilité des IA, nous avons exploré dans ce papier les apports et la faisabilité d'un processus de création de classifieurs ontologiquement explicables du point de vue des utilisateurs du domaine. Nous avons proposé une démarche générique inspirée de diverses approches de l'état de l'art qui permettent l'explicabilité, et qui résulte en une architecture basée sur 2 modules, l'un dédié à l'extraction de features ontologiques par des techniques d'AP, l'autre dédié au raisonnement ontologique et nommé OntoClassifier.
De manière à intégrer la dimension ontologique au cœur même du classifieur sans trop alourdir le pipeline de classification résultant, nous avons introduit un outil permettant la génération de l'OntoClassifier, ce module étant implémenté automatiquement sous la forme d'un graphe de tenseurs ontologiques construit directement à partir des définitions fournies par l'ontologie.
Nous avons exemplifié notre démarche par la création d'un classifieur d'images dédié au domaine de la fameuse ontologie des Pizzas, et illustré les possibilités offertes par l'OntoClassifier, aussi bien du point de vue de la classification, que de celui de l'explicabilité visuelle des prédictions en utilisant les définitions OWL de l'ontologie. Comme nous l'avons souligné, le jeu de données mis en oeuvre dans cet exemple était simple et totalement contrôlé, ceci dans le but d'aider à la formalisation et à l'illustration de la démarche que nous proposons. Nos travaux en cours impliquent d'ores et déjà cette démarche et ces outils dans un projet d'envergure dédié à un autre domaine, sur des données réelles, et pour une tâche de classification d'images à grain fin.
Les éléments que nous avons présentés restent cependant à être étoffés. Nous avons en particulier pour l'instant principalement focalisé sur les possibilités offertes par un classifieur ontologiquement explicable et n'avons pas encore travaillé sur l'ergonomie des interfaces homme-machine qui permettront aux utilisateurs finaux de manipuler et d'explorer les explications fournies par le système : ce point fera l'objet de futurs travaux. De plus, les explications visuelles proposées ici utilisent directement les expressions OWL pour les relier à l'image : cette représentation demande aussi à être améliorée, puis à être évaluée avec des utilisateurs.
Les éléments exposés dans ce papier fournissent néanmoins des bases solides nous permettant d'entamer de nouveaux projets ayant besoin de classifieurs explicables au niveau d'abstraction de leurs utilisateurs. Il permettent de plus d'imaginer de nouvelles fonctionnalités prometteuses, voire nécessaires dans le cadre de projets impliquant des acteurs aux cultures différentes, comme par exemple le fait d'envisager l'explicabilité des IA tout en considérant le besoin de multi points de vue.

Références

[1] P. Angelov and E. Soares. Towards explainable deep neural networks (xDNN). *Neural networks : the official journal of the International Neural Network Society*, 130 :185–194, 2020.

[2] A. Arrieta, N. Díaz-Rodríguez, J. Ser, Adrien Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI) : Concepts, taxonomies, opportunities and challenges toward responsible AI. *ArXiv*, abs/1910.10045, 2020.

[3] J. Atif, C. Hudelot, and I. Bloch. Explanatory reasoning for image understanding using formal concept analysis and description logics. *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, 44 :552–570, 2014.

[4] A. Bénel and C. Lejeune. Humanities 2.0 : documents, interpretation and intersubjectivity in the digital age. *Int. J. Web Based Communities*, 5 :562–576, 2009.

[5] R. Confalonieri and Tarek R. Besold. Trepan reloaded : A knowledge-driven approach to explaining black-box models. In *ECAI*, 2020.

[6] D. Conigliaro, R. Ferrario, C. Hudelot, and D. Porello. Integrating computer vision algorithms and ontologies for spectator crowd behavior analysis. In *Group and Crowd Behavior for Computer Vision*, 2017.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet : A large-scale hierarchical image database. In *CVPR*, 2009.

[8] Z. Ding, L. Yao, B. Liu, and J. Wu. Review of the application of ontology in the field of image object recognition. In *ICCMS 2019*, 2019.

[9] F. K. Dosilovic, M. Brčič, and N. Hlupic. Explainable artificial intelligence : A survey. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215, 2018.

[10] A. Gonzalez-Garcia, D. Modolo, and V. Ferrari. Do semantic parts emerge in convolutional neural networks ? *International Journal of Computer Vision*, 126 :476–494, 2017.

[11] B. Goodman and S. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Mag.*, 38 :50–57, 2017.

[12] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *ECCV*, 2016.

[13] Z. C. Lipton. The mythos of model interpretability. *Queue*, 16 :31 – 57, 2018.

[14] M. Losch, M. Fritz, and B. Schiele. Interpretability beyond classification output : Semantic bottleneck networks. *ArXiv*, abs/1907.10882, 2019.

[15] S. M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NIPS*, 2017.

[16] D. Marcos, S. Lobry, and D. Tuia. Semantically interpretable activation maps : what-where-how explanations within CNNs. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4207–4215, 2019.

[17] D. Martens and F. Provost. Explaining data-driven document classifications. *MIS Q.*, 38 :73–99, 2014.

[18] D. P. Papadopoulos, Y. Tamaazousti, F. Ofli, I. Weber, and A. Torralba. How to make a pizza : Learning a compositional layer-based gan model. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7994–8003, 2019.

[19] M. Tulio Ribeiro, S. Singh, and C. Guestrin. "why should i trust you ?" : Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[20] O. Ronneberger, P. Fischer, and T. Brox. U-Net : Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L-C. Chen. MobileNetV2 : Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[22] Md. Kamruzzaman Sarker, N. Xie, D. Doran, M. Raymer, and P. Hitzler. Explaining trained neural networks with semantic web technologies : First steps. *ArXiv*, abs/1710.04324, 2017.

[23] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-CAM : Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128 :336–359, 2019.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[25] W. Swartout, C. Paris, and J. Moore. Explanations in knowledge systems : design for explainable expert systems. *IEEE Expert*, 6 :58–64, 1991.

[26] D. West. The future of work : Robots, AI, and automation. 2018.

[27] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *2011 International Conference on Computer Vision*, pages 2018–2025, 2011.

[28] Q. Zhang, Y. Wu, and S. Zhu. Interpretable convolutional neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.

[29] M. Zhitomirsky-Geffet, E. S. Erez, and J. Bar-Ilan. Toward multiviewpoint ontology construction by collaboration of non-experts and crowdsourcing : The case of the effect of diet on health. *Journal of the Association for Information Science and Technology*, 68, 2017.

[30] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. Youngblood. Explainable AI for designers : A human-centered perspective on mixed-initiative co-creation. *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, 2018.

Hybridation de l'Answer Set Programming et de la théorie de Dempster Shafer

S. SONFACK SOUNCHIO[1], L. GENESTE[1], B. KAMSU FOGUEM[1]

[1] Université de Toulouse, Laboratoire Génie de Production

10 juin 2021

Résumé

Au cours des processus d'expertise, qui permettent d'explorer les différentes solutions d'un problème, il est utile de raisonner en intégrant à la fois un raisonnement non-monotone et l'incertitude. Nous proposons pour cela une approche hybridant la théorie de Dempster Shafer (pour l'incertitude) et l'approche Answer Set Programming (pour le raisonnement non monotone). Un choix peut ainsi être effectué, parmi les ensembles solution du programme logique, en utilisant conjointement une mesure de croyance et le niveau d'incohérence résultant de la base de connaissances.

Mots-clés

Représentation de la connaissance, Answer Set Programming, théorie des fonctions de croyance, incertitude.

Abstract

In practical situations it is useful to be able to reason about uncertain knowledge within a non-monotonic logic. This is, the case of expertise processes which finds all possible solutions of a cause. We propose for this purpose an approach that aims at combining the theory of belief functions and Answer Set Programming (ASP) by allowing decision-making with the generalized ordered weighted average.

Keywords

Knowledge représentation, Answer Set Programming, belief function theory, uncertainty.

1 Introduction

Les démarches d'expertise se caractérisent par des processus fortement exploratoires fondés sur un raisonnement à base d'hypothèses (intégrant donc de l'incertitude) et la manipulation de connaissances expertes. Elles sont mises en œuvre dans des contextes très variés comme la conception de systèmes, la résolution de problèmes industriels ou encore l'expertise médico-légale. Un référentiel normatif sur les démarches d'expertise a été défini, d'abord en France avec la norme NF X50-110 [19] "Qualité des activités d'expertise", puis au niveau européen avec la norme CSN EN 16775 "Activités d'expertise - Exigences générales pour les services d'expertise". Cependant, si ces normes définissent des principes et une description des processus d'expertise, elles ne fournissent pas d'outils pour modéliser les problèmes à expertiser et faciliter la prise de décision en ce qui concerne le choix des solutions du processus d'expertise.

L'espace de prise de décision humaine dans les processus d'expertise fait apparaître principalement deux types de défauts selon [29] : (1) les défauts qualitatifs résultant des limites de la représentation du monde réel ; (2) les défauts numériques liés à l'incertitude du dit monde.

Ainsi, prendre en compte ces deux inconvénients pour la prise de décision permettra de l'améliorer et de réduire les risques ou conséquences des décisions dans certains domaines [18] [22].

C'est dans cette optique de considérer ces défauts que nous proposons dans cet article une méthode permettant de déterminer une valeur de prise de décision à partir d'une représentation qualitative et quantitative de la connaissance d'un domaine donné. Ainsi, pour la représentation qualitative, nous nous sommes penchés sur la programmation par ensemble réponses (Answer Set Programming, ASP), qui au-delà de son expressivité par rapport aux langages basés simplement sur la logique classique, est non monotone et modélise bien le raisonnement intégrant le bon sens [5, 1], ce qui la rend appropriée pour raisonner avec une connaissance incomplète d'un domaine.

En outre, il est important de tenir compte de l'incertitude quantitative dans cette représentation, car selon [4], elle aide à surmonter certaines limites de l'ASP, qui ne peut exprimer de manière naturelle les croyances des littéraux, ni leur véracité quantifiée et même leur manque de connaissance. En général, l'ASP classique ne peut pas exprimer intuitivement la quantité d'incertitude de sa sémantique.

Bien que des approches pour apporter la quantification de l'incertitude en représentation de la connaissance à base de logique aient été proposées, le problème reste d'actualité [7]. Il est à noter que la plupart des méthodologies reposent uniquement sur les théories des probabilités ou des possibilités appliquées aux règles logiques [3, 7] [11]. Tandis que nous proposons une approche utilisant d'une part la théorie de Dempster Shafer qui est une généralisation des théories précédentes [21] et d'autre part la mesure de l'inconsistance de la connaissance représentée en ASP.

De cette façon, nous cherchons à repousser les limites des ensembles solutions (Answer Sets) en ce qui concerne la gestion des connaissances incertaines[34].

Le reste de ce document est structuré de la manière suivante : en section 2, nous présentons la programmation par ensemble réponses et la théorie de Dempster Shafer. En section 3 nous décrivons l'approche proposée pour calculer une valeur de décision, en combinant la programmation par ensemble réponses et la théorie de Dempster Shafer. Nous terminerons cette section par un exemple d'application illustrant la méthode.

2 Fondamentaux

2.1 Programmation par ensemble réponses (Answer Set Programming, ASP)

L'ASP est une approche de représentation de la connaissance basée sur un paradigme déclaratif reposant sur la logique de premier ordre. Elle utilise un solveur basé sur la recherche de modèles stables, semblable à celui de la programmation par contraintes [12, 27, 8]. Bien qu'utilisant une syntaxe similaire au langage Prolog, ce langage est adapté pour la résolution de problèmes d'ordre multiples (web sémantique, planification, théorie des graphes, bio-informatique, configuration) et tire ses racines du raisonnement non monotone [13]. La non-monotonicité de l'ASP est obtenue par une forme de négation, appelée *négation comme échec* qui peut s'assimiler au raisonnement par défaut [14]. Cela en fait un choix approprié pour le raisonnement de bon sens, proche du raisonnement humain.

2.1.1 Syntaxe ASP

La programmation en ensemble réponses peut se reposer aussi bien sur la logique propositionnelle que sur la logique du premier ordre [24]. De plus elle dispose d'extensions avec des règles de choix, de pondération ou de cardinalité [28].
Un programme ASP est constitué d'un ensemble de règles (r) se présentant sous la forme suivante :
$a \leftarrow b_1, \ldots, b_n, notb_{n+1}, \ldots, notb_m$, $0 \leq n \leq m$ où a et b_i sont atomes.
a est appelé la *tête* de la règle et est notée par $tete(r)$
$\{b_1, \ldots, b_n, b_{n+1}, \ldots, b_m\}$ forment le corps de celle-ci et se note $corps(r)$.
$corps(r)$ peut être divisé en deux parties :

— Les *littéraux positifs* $corps^+(r) = \{b_1, \ldots, b_n\}$
— Les *littéraux négatifs* $corps^-(r) = \{b_{n+1}, \ldots, b_m\}$

Il est possible d'avoir des règles sans corps ($a \leftarrow$, c'est-à-dire $corps(r) = \emptyset$) aussi appelés *fait*.
Si $corp^-(r) = \emptyset$, c'est-à-dire si la règle ne dispose pas de littéraux négatifs, on parle de *règle définie* (*definite rule*). Ainsi lorsqu'un programme est constitué uniquement de règles définies, il est appelé *programme défini* sinon on parle de *programme normal*
La négation *not* que nous retrouvons dans une règle est différente de la négation classique ($\neg$), que l'on retrouve en logique classique. En effet, dans le cadre de la programmation par ensemble réponses, cette négation est appelée : *négation par défaut* et a le sens de : "*on ne pense pas que*". C'est cette sémantique qui permet de faire un raisonnement par défaut.
Par définition, un programme de la programmation par ensemble réponses est constituée d'un ensemble de règles comme nous l'avons défini ci-dessus, mais dispose aussi d'instructions comme *règles de choix, règles pondérées, règles de cardinalité* qui ont pour rôle d'ajouter des contraintes sur l'ensemble réponse (*Answer Set*).

2.1.2 Sémantique de ASP

La signification des programmes est donnée par la sémantique du *modèle stable*, qui définit un ensemble d'atomes satisfaisant toutes les règles du programme. Cet ensemble d'atomes est appelé *ensemble réponse* (*Answer Set*) et correspond au *point fixe* de l'*opérateur de conséquence immédiate* (*single step operator*) 2.1.2
Un ensemble d'atomes est obtenu par extension M est un ensemble de réponses d'un programme ASP s'il satisfait aux conditions suivantes : [15, 20] :

— Si le programme ASP est un programme défini, c'est-à-dire constitué uniquement de règles définies, alors M est un ensemble minimal d'atomes qui satisfait à toutes les règles de ce programme ASP.
— Si le programme ASP est un programme normal (c'est-à-dire non défini), alors M coïncide avec l'ensemble réponse de la réduction connue sous le nom de *réduction Gelfond-Lifschitz* du programme à sa forme définie.

Lorsqu'un programme ASP a un ensemble réponse (*Answer Set*), on dit qu'il est **consistant** et **inconsistant** dans le cas contraire.
La **réduction Gelfond-Lifschitz** est un processus de transformation d'un programme normal en un programme défini (sans règle négative), par rapport à un ensemble d'atomes. Elle est basée sur la forme instanciée du programme de départ, c'est-à-dire que la nouvelle forme obtenue du programme ne contient pas de variable. Cette réduction se déroule comme suit : étant donné un programme arbitraire, P et un ensemble d'atomes M, la réduction de P par rapport à M (P^M) est le programme défini obtenu en :

1. Supprimant toutes les règles qui ont un littéral *not* dans leur corps et dont le dit littéral se trouve dans M ;
2. Supprimant tous les littéraux *not* dans le corps des règles restantes.

Après cette réduction, un modèle M du programme défini P^M est le plus petit ensemble *close* (ne contenant pas de variables libres) par rapport à P^M. Cela correspond à un *modèle de Herbrand minimal*, qui est également un *point fixe* de l'*opérateur de conséquence* T_P.

$$T_P : 2^{\mathbb{X}} \rightarrow 2^{\mathbb{X}}$$
$$A \longmapsto T_P(A) = \{a | a \leftarrow b_1, \ldots, b_n \in P \wedge b_i \in A, i = 1, \ldots, n\}$$

$\mathbb{X}$ est l'ensemble d'atome de P.

2.2 Théorie de Dempster Shafer

La théorie de Dempster Shafer, également connue sous le nom de théorie des preuves ou théorie des fonctions de croyance, est un outil généralisant le raisonnement dans un contexte d'incertitude.

2.2.1 Concepts de base

Cette théorie est élaborée autour de la notion de *fonction de masse* qui, étant donnée une question, associe un poids lié aux évidences disponibles sur les différentes hypothèses associées [23] [36] [17] [26]. Elle se formalise de la manière suivante :
Soit $\Theta = \{\theta_1, \ldots, \theta_n\}$ un ensemble d'hypothèses de réponse à une question, cet ensemble est aussi appelé *cadre de discernement* (*frame of discernment, FOD*).
Soit 2^Θ l'ensemble des partitions de Θ, $2^\Theta = \{A | A \subseteq \Theta\}$.
Une fonction de masse m permettant la distribution de masses de probabilité sur l'ensemble des partitions est :

$$\begin{aligned} m : 2^\Theta &\to [0,1] \\ A &\longmapsto m(A) \\ \sum \{m(A)/A \subseteq \Theta\} &= 1. \end{aligned}$$

$m(\emptyset) = 0$ est la forme normalisée de la fonction de distribution et correspond à l'hypothèse du monde clos.
Si $\forall A \subseteq \Theta$ si $m(A) > 0$ alors A est un *ensemble focal* (*focal set*) pour m.
À partir de la distribution, $m(A)$, il est possible de déduire :

- La *fonction de croyance* $Bel()$:
  $\begin{cases} Bel(A) = \sum_{B \subseteq A} m(B) \\ m(\emptyset) = 0 \end{cases}$
  $Bel(A)$ est interprété comme le degré de croyance que la vérité réside en A.
- La fonction de plausibilité $Pl()$ d'une hypothèse A est la quantité de croyance non strictement engagée dans le complément de $\bar{A}$
  $\begin{cases} Pl : 2^\Theta \to [0,1] \\ Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(\overline{\mathrm{A}}), A \subseteq \Theta \end{cases}$

3 Approche d'hybridation ASP-DST

La méthodologie que nous proposons comporte deux parties. Dans un premier temps, elle combine le programme ASP et les fonctions de croyance, en définissant une distribution de masse d'évidence sur l'ensemble des atomes de la base de Herbrand du programme. Cette distribution représente le niveau de véracité attribué aux atomes de la base de Herbrand et permettra ainsi d'évaluer la croyance d'un modèle.
Par la suite, nous utilisons la croyance de l'ensemble réponse et l'incohérence induite par les connaissances utilisées, pour calculer une valeur de décision en se basant sur une moyenne pondérée ordonnée.

3.1 Distribution d'évidence sur la base de Herbrand

Soit P un programme ASP défini, et B_P la base de Herbrand obtenue à partir de P.
B_P sera considérée comme le *FOD* de la fonction de distribution de masse, car tout modèle de P est un modèle de Herbrand minimal, qui est sous-ensemble de B_P.
Soit $2^{B_P} = \{A | A \subseteq B_P\}$ l'ensemble des partitions de B_P.
Soit m la fonction de distribution de masse d'évidence sur l'ensemble des partitions de la base de Herbrand.

$$\begin{aligned} m : 2^{B_P} &\to [0,1] \\ A &\longmapsto m(A) \\ \sum \{m(A)/A \subseteq B_P\} &= 1. \end{aligned}$$

De cette fonction de distribution, nous pouvons évaluer la croyance d'un sous-ensemble A de B_P ensemble solution d'un programme P : $Bel(A) = \sum_{B \subseteq A} m(B)$. Cela permet par conséquent de pouvoir évaluer la croyance d'un modèle du programme P.
Étant donné qu'il est possible d'avoir plusieurs ensembles modèles pour un programme ASP normal donné, nous ne nous intéressons plus au modèle minimal, mais plutôt au modèle ayant la croyance la plus élevée.

3.2 Inconsistance des programmes ASP

Les bases de connaissances construites à la main ou par l'automatisation sont souvent inexactes, ce qui signifie qu'elles ne reflètent pas, dans une certaine mesure, la réalité, et il est important de savoir à quel point elles s'écartent de celle-ci, car leur utilisation peut conduire à de mauvaises décisions [35]. En effet, cette mauvaise qualité des bases de connaissances est souvent due à des données erronées, à des informations manquantes ou à l'incohérence des schémas [10]. Cependant, malgré les approches utilisées pour les construire, il est encore difficile d'éviter ces problèmes lors de la formalisation des bases de connaissances et cela peut entraîner l'existence d'éléments d'informations contradictoires dans celle-ci [31].Ce problème de la représentation de la connaissance est généralisé sous le nom d'inconsistance.
En général, l'inconsistance peut se définir comme la présence de contradictions dans une base de connaissances, par rapport au formalisme de représentation [6]. Pour ce qui est de la représentation basée sur le formalisme du langage ASP, elle peut se traduire par tous les ensembles solutions sont inconsistant [33].
À fin de pouvoir quantifier ces problèmes rencontrés dans les bases de connaissances, différentes fonctions de mesure d'inconsistance ont été élaborées pour certaines méthodes de représentation de connaissances, comme la logique[30] et leur définition a été étendue à la programmation par ensemble réponses. À cet effet, la mesure de l'incohérence peut être exprimée par une fonction I positive, avec la sémantique de représenter la gravité de l'inconsistance dans

la base de connaissances, telle que I augmente avec l'incohérence.
Pour le cas de la représentation des connaissances basée sur ASP, [33] a défini des fonctions de mesure d'incohérence qui supportent le postulat de rationalité sans être *monotone*. Cette inconsistance est définie comme suite :
$I_{\pm}(P) = min\{|A| + |D|$ où A, D représentent respectivement les règles qui ont été ajoutées et réduites du programme P tel que $(P \cup A) - D$ est consistant $\}$.
La mesure de l'inconsistance des bases de connaissances construite à partir de la logique est généralement d'une complexité de Co-NP ou NP. Mais certaines fonctions permettent de réduire cette complexité, comme par l'exemple l'utilisation d'une mesure portée sur les ensembles réponses ou l'emploie de valeurs booléennes :

— $I_{\#}(P)$ = $min\{k|M$ est $k - inconsistant\}$, M est un ensemble solution du programme P et un ensemble solution est k-inconsistant, si il existe exactement k atomes et leurs opposés dans l'ensemble solution ;
— la mesure drastique [32]
$I_d(K) = \begin{cases} 1 \rightarrow inconsistance \\ 0 \end{cases}$
qui retourne 1 si le programme est inconsistant et 0 dans le cas contraire.

3.3 Moyenne pondérée ordonnée de la programmation des ensembles de réponses

Cette section exprime un moyen de calculer une valeur unique sur laquelle on peut prendre une décision à partir d'une représentation de la connaissance d'un problème avec un programme ASP.
Soit P un programme ASP normal. Soit X un ensemble solution du programme P et $Bel(X)$ sa croyance (voir section 3.1).
Soit $I(P) \in [0, 1]$ l'incohérence normalisée du programme P. Étant donné ces valeurs, un moyen approprié de combiner les métriques (croyance et incohérence) est l'utilisation d'une fonction d'agrégation, par exemple à l'aide d'une moyenne pondérée ordonnée (**OWA**) [9].
Cet opérateur peut être étendu en **GOWA**, qui est une forme généralisée de **OWA** [16]. Nous exploitons cet opérateur en faisant usage de la transformation qui range les valeurs de l'inconsistance et de la croyance dans l'ordre croissant :
$GOWA_{\alpha,\lambda}(I(P), Bel(X)) = (\alpha_1 * max^{\lambda}(I(P), Bel(X)) + \alpha_2 * min^{\lambda}(I(P), Bel(X)))^{\frac{1}{\lambda}}$
avec $w = (w_1, w_2) \in [0, 1]^2$, et $w_1 + w_2 = 1$:
w_1, w_2 représentent la confiance que l'on a respectivement sur la formalisation de la représentation de la connaissance et l'ensemble solution émanant de cette représentation. La figure 1 ci-dessous résume notre approche mettant en contribution la programmation par l'ensemble et la théorie de Dempster Shafer.

Example d'illustation
Soit P un programme ASP avec les règles suivantes :
$oiseau(X) : -vole(X), plume(X), not\ anormal(X).$
$\{vole(X); nage(X)\} : -oiseau(X).$

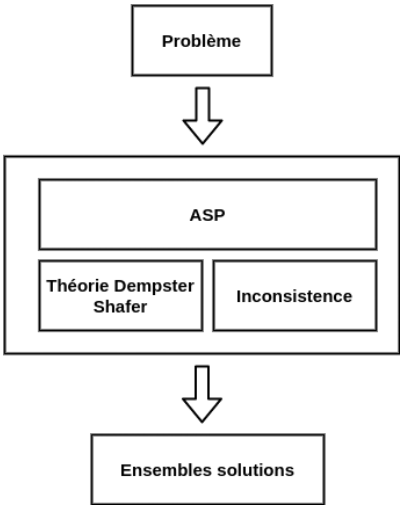

FIGURE 1 – Approche DST-ASP

$oiseau(tweety).$

— Pour améliorer cette base de connaissances (retirer les inconsistances) nous allons :
Ajouter la contrainte selon laquelle un oiseau ne peut pas avoir les aptitudes de nager et voler simultanément. La base de connaissances devient :
$oiseau(X) : -vole(X), plume(X), not\ anormal(X).$
$\{vole(X); nage(X)\} : -oiseau(X).$
$: -nage(X), vole(X).$
$oiseau(tweety).$
— La base de Herbrand associée à ce programme est :
$B_P = \{vole(tweety), oiseau(tweety)$
$anormal(tweety), plume(tweety), nage(tweety)\}.$

— Pour réunir *ASP* et *DST*, une assignation de croyance de base (BBA) doit être définie sur 2^{H_P}. Comme cette assignation peut devenir difficile, sur tout avec une taille importante de H_P (complexité NP), nous allons faire usage d'une fonction de raffinement.
De ce fait, il faut :

1. Utiliser les connaissances ou l'expérience passées pour regrouper la base Herbrand en ensembles disjoints.
2. Attribuer une valeur de preuve aux éléments individuels de la base Herbrand.
3. Définir une affectation de croyance sur le nouvel ensemble qui constitue une partition de la base Herbrand.

Les croyances que nous obtenons par rapport à la distribution 1 sont :

1. $X1 = \{oiseau(tweety)\}$ avec une croyance de :
$Bel(X1) = 0.2$
2. $X2 = \{oiseau(tweety), vole(tweety)\}$
$Bel(X2) = m(\{oiseau(tweety)\}) +$

{ vole(tweety) } : 0.1	{ oiseau(tweety)} : 0.2
{ nage(tweety)} : 0.1	{ plume(tweety)} : 0.1
{ vole(tweety) , nage(tweety)} :0.1	{ vole(tweety) , oiseau(tweety)} : 0.2
{ vole(tweety) , plume(tweety)} :0.1	{nage(tweety) , oiseau(tweety)} : 0.1

TABLE 1 – Distribution de masses

$m(\{vole(tweety)\}) + m(\{oiseau(tweety), vole(tweety)\})$
$Bel(X2) = 0.2 + 0.1 + 0.2 = 0.5$

3. $X3 = \{oiseau(tweety), nage(tweety)\}$
$Bel(X3) = m(\{oiseau(tweety)\}) + m(\{nage(tweety)\}) + m(\{oiseau(tweety), nage(tweety)\})$
$Bel(X3) = 0.2 + 0.1 + 0.1 = 0.4$

— Notre représentation de connaissance a une inconsistance de :
$I_{\pm} = 1$ Nous normalisons cette valeur par $\frac{I_{\pm}}{|A|+|P|}$ où A représente les règles ajoutées et P le programme initial sans ajout ni réduction.
$I = 0.25$

— De ce qui précède, nous pouvons calculer les valeurs de décision pour les deux modèles $X1$, $X2$ et $X3$: En utilisant GOWA

1. $X1 = \{oiseau(tweety)\}$ La croyance de $X1$ par rapport à la distribution m est :
$Bel(X1) = 0.2$
$w_1 = 0.5, w_2 = 0.5$
$GOWA_{w,1}(P, X) = 0.5 * 0.25 + 0.5 * 0.2 = 0.22$
$I1 = 0.22$
2. $X2 = \{oiseau(tweety), vole(tweety)\}$
La croyance de $X2$ par rapport à la distribution m est :
$Bel(X2) = 0.5$
$w_1 = 0.5, w_2 = 0.5$
$GOWA_{w,1}(P, X) = 0.5 * 0.25 + 0.5 * 0.5 = 0.37$
$I2 = 0.37$
3. $X3 = \{oiseau(tweety), nage(tweety)\}$
La croyance de $X3$ par rapport à la distribution m est :
$Bel(X3) = 0.4$
$w_1 = 0.5, w_2 = 0.5$
$GOWA_{w,1}(P, X) = 0.5 * 0.25 + 0.5 * 0.4 = 0.32$
$I3 = 0.32$

L'exemple nous montre qu'il serait judicieux de choisir la solution $X2$ ou $X3$, au lieu de $X1$, car elles ont des valeurs de décision supérieures à celle de $X1$. Cette différence de valeurs peut s'expliquer par le fait que la croyance du modèle $X1$ est faible comparée à celle des modèles $X3$ et $X2$.

De cette example la solution $X2 = \{oiseau(tweety), vole(tweety)\}$ est meilleure vu qu'elle a la plus grande valeur de décision.

4 Comparaison et Discussion

4.1 Comparaison

Nous allons dans un prémier temps comparer notre approche hybride avec les deux modèles mise en contribution de façon individuelle et par la suite avec une la proposition de Al Machot et al [2].
En effet, pour un même problème entre l'approche hybride et l'approche ASP toute seule, la différence fondamentale est que cette dernière ne donne aucun moyen de choisir une solution parmi l'ensemble des modèles trouvés pour le problème, ce qui peut être embarrassant pour la prise de décision. Par ailleurs pour la même condition précédente, l'approche Dempter Shaffer toute seule pourra à la limite fournir des valeurs de croyance, sans qu'on ne sache quelles sont les solutions du problème. Le tableau 2 ci-dessous récapitule la comparaison entre l'approche hybride et les deux autres approches prises seules.
En résumé, l'approche hybride que nous proposons permet de combler les limites des méthodes ASP et DSP. Par conséquent, elle permet de résoudre un problème et sélectionner la meilleure des solutions.

Méthode	Solution au problème	Choix d'une solution
ASP	oui	non
DST	non	oui
Hybridation	oui	oui

TABLE 2 – Comparaison ASP, DST et Hybridation

Pour ce qui est de la comparaison avec l'approche élaborée par Al Machot et al. [2], il en ressort que :

— Nous proposons une généralisation de la combinaison entre la programmation par ensemble et la théorie de Dempster shafer alors que les travaux de Al Machot et al. présentent une façon et un cas particulier de cette intégration et s'intéressent à la combinaison des sources différentes.
— Dans l'approche que nous présentons, nous tenons compte de l'inconsistance, qui peut considérablement influencer la prise de décision comparé à la seule croyance. Par contre cet aspect de la connaissance n'est pas considéré par ces auteurs.

4.2 Discussion

Nous proposons une hybridation de programmation par l'ensemble (ASP) et de la théorie de Demspter Shafer qui permet de sélectionner la meilleure solution au cours d'un processus d'expertise. En effet ce processus permet d'explorer toutes les pistes de solutions possibles d'un problème et de ce fait il devient import d'avoir une value de décision pour faciliter le choix d'une possibilité par rapport aux autres. Cette approche se résume aux étapes suivantes :

— Dans un premier temps de pouvoir évaluer la croyance liée à un ensemble solution que représente les différentes possibilités d'un processus d'expertise.
— Par la suite, nous calculons l'incohérence de la base de connaissances ou des ensembles solutions.
— En fin les deux valeurs précédemment calculées nous permettent d'avoir une valeur de décision, qui permet de choisir un ensemble solution en tenant compte de l'inconsistance et la croyance.

Pour l'implémentation, nous avons développé un prototype en langage Python, qui utilise les librairies py_dempster_shafer [1] et clyngor [2]. Techniquement nous utilisons une structure de données de type dictionnaire pour associer la distribution de masse aux éléments de la base de Herbrand et par la suite ce dictionnaire est exploité pour le calcul de croyance et la valeur de décision.
En raison de la complexité que l'on peut avoir lorsque la base de Herbrand est important ou qu'il devient difficile de définir une fonction de distribution d'évidence, une approche qui reste dans la même démarche, que la précédente consistera à utiliser la fonction de support simplifiée, qui est une fonction de croyance basée sur le support d'un sous-ensemble [25] de la dite base.
En effet elle consiste à fixer un support $s \in [0, 1]$ représentant la croyance pour un sous-ensemble $A \subset H_P$ et tout sous-ensemble de H_P le contenant. Elle se décrit comme suit

$$S(B) = \begin{cases} 0, A \nsubseteq B \\ s, A \subseteq B \neq H_P \\ 1, B = H_P \end{cases}$$

Pour illustrer, utilisons l'exemple précédent et supposons que la personne accorde un support de $s = 0.6$ au sous ensemble $\{oiseau(tweety), vole(tweety)\}$. Cela revient aux calculs de croyance suivants :

— $X1 = \{oiseau(tweety)\}$
$X1 \subseteq \{oiseau(tweety), vole(tweety)\}$
$Bel(X1) = 0.6$
— $X2 = \{oiseau(tweety), vole(tweety)\}$
$X2 \subseteq \{oiseau(tweety), vole(tweety)\}$
$Bel(X2) = 0.6$
— $X3 = \{oiseau(tweety), nage(tweety)\}$
$X3 \nsubseteq \{oiseau(tweety), vole(tweety)\}$
$Bel(X3) = 0$

5 Conclusion

Dans ce travail, nous avons dans un premier temps présenté la programmation par ensemble réponses qui est activement utilisée dans la représentation des connaissances et pour le raisonnement non-monotone. Par la suite, nous avons présenté une synthèse de la théorie de Dempster Shafer utilisée pour représenter des connaissances incertaines. Enfin, nous avons défini une méthodologie de calcul d'une valeur décisionnelle, qui premièrement combine les deux théories précédentes pour un meilleur choix du modèle en fonction de la croyance. Cette croyance est par la suite combinée à la valeur d'incohérence de la base de connaissances au travers de la moyenne pondérée ordonnée généralisée pour avoir une valeur de prise de décision.
L'approche proposée dans ce travail permet de prendre une décision non seulement en se focalisant sur l'ensemble solution de la programmation par ensemble, mais en tenant compte de la croyance et l'incohérence.
Pour ce qui est des difficultés liées à la complexité, nous proposons l'utilisation d'un simple fonction d'assignation basée sur un support d'évidence pou la difficulté de distribution d'évidence et l'utilisation d'inconsistance drastique pour la complexité liée à la mesure d'inconsistance.
En effet, cette approche facilite la prise de décision, car elle est numérique et de plus permet de prendre une meilleure décision parce qu'elle tient compte de la croyance et l'incohérence émanant du modèle de représentation du problème.
En comparaison avec l'approche proposée par Al Machot et al [2] montre que notre méthode est général et tient compte de l'inconsistance, ce qui n'est pas le cas des travaux de ces auteurs. De plus notre méthode offre un mécanisme de décision sur les solutions d'un problème, que ne dispose pas ASP et DST prisent individuellement.
Pour la suite de ce travail, nous allons dans un premier temps définir un algorithme qui permettra de trouver le modèle avec la plus grande conviction, basée sur la croyance et l'incohérence. Par la suite développer un outil basé sur cet algorithme et qui intègre les deux librairies du langage Python que nous avons utilisé.

1. https ://github.com/reineking/pyds
2. https ://github.com/aluriak/clyngor

Références

[1] Erdi Aker, Volkan Patoglu, and Esra Erdem. Answer set programming for reasoning with semantic knowledge in collaborative housekeeping robotics. *IFAC Proceedings Volumes*, 45(22) :77–83, 2012.

[2] Fadi Al Machot, Heinrich C Mayr, and Suneth Ranasinghe. A hybrid reasoning approach for activity recognition based on answer set programming and dempster–shafer theory. In *Recent Advances in Nonlinear Dynamics and Synchronization*, pages 303–318. Springer, 2018.

[3] Chitta Baral. Logic programming and uncertainty. In *International Conference on Scalable Uncertainty Management*, pages 22–37. Springer, 2011.

[4] Kim Bauters, Steven Schockaert, Martine De Cock, and Dirk Vermeir. Semantics for possibilistic answer set programs : uncertain rules versus rules with uncertain conclusions. *International journal of approximate reasoning*, 55(2) :739–761, 2014.

[5] Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12) :92–103, 2011.

[6] Mark Burgin and CNJ de Vey Mestdagh. Consistent structuring of inconsistent knowledge. *Journal of Intelligent Information Systems*, 45(1) :5–28, 2015.

[7] Federico Cerutti and Matthias Thimm. A general approach to reasoning with probabilities. *International Journal of Approximate Reasoning*, 111 :35–50, 2019.

[8] Thomas Eiter, Giovambattista Ianni, and Thomas Krennwallner. Answer set programming : A primer. In *Reasoning Web International Summer School*, pages 40–110. Springer, 2009.

[9] Ali Emrouznejad and Marianna Marra. Ordered weighted averaging operators 1988–2014 : A citation-based literature survey. *International Journal of Intelligent Systems*, 29(11) :994–1014, 2014.

[10] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek. Predicting completeness in knowledge bases. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 375–383, 2017.

[11] Laurent Garcia, Claire Lefèvre, Odile Papini, Igor Stephan, and Eric Würbel. Possibilistic asp base revision by certain input. 2018.

[12] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Answer set solving in practice. *Synthesis lectures on artificial intelligence and machine learning*, 6(3) :1–238, 2012.

[13] Jeroen Janssen, Steven Schockaert, Dirk Vermeir, and Martine De Cock. *Answer Set Programming for Continuous Domains : A Fuzzy Logic Approach*, volume 5. Springer Science & Business Media, 2012.

[14] Antonis C Kakas. Default reasoning via negation as failure. In *Foundations of Knowledge Representation and Reasoning*, pages 160–178. Springer, 1994.

[15] Michael Kaminski. A note on the stable model semantics for logic programs. *Artificial intelligence*, 96(2) :467–479, 1997.

[16] Fateme Kouchakinezhad and Alexandra Šipošová. Ordered weighted averaging operators and their generalizations with applications in decision making. *Iranian Journal of Operations Research*, 8(2) :48–57, 2017.

[17] Liping Liu and Ronald R Yager. Classic works of the dempster-shafer theory of belief functions : An introduction. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 1–34. Springer, 2008.

[18] Peter Lucas and Linda Van Der Gaag. *Principles of expert systems*. Addison-Wesley Wokingham, 1991.

[19] Huver Loisel Peyrouty Pineau Tuffery M. Peyrouty, Chanay ; Fourniguet. Recommandations pour l'application de la norme nf x 50-110 :2003. Technical report, Association Française de Normalisation, 2011.

[20] Pierre Marquis, Odile Papini, and Henri Prade. Panorama de l'intelligence artificielle. *Ses Bases Méthodologiques, ses Développements*, 2, 2014.

[21] Arnaud Martin, Anne-Laure Jousselme, and Christophe Osswald. Conflict measure for the discounting operation on belief functions. In *2008 11th International conference on information fusion*, pages 1–8. IEEE, 2008.

[22] Steve Pye, Francis GN Li, Arthur Petersen, Oliver Broad, Will McDowall, James Price, and Will Usher. Assessing qualitative and quantitative dimensions of uncertainty in energy modelling for policy support in the united kingdom. *Energy research & social science*, 46 :332–344, 2018.

[23] Thomas Reineking. *Belief functions : theory and algorithms*. PhD thesis, Universität Bremen, 2014.

[24] Fabrizio Riguzzi. *Foundations of Probabilistic Logic Programming*. River Publishers, 2018.

[25] Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

[26] Glenn Shafer. Probability judgment in artificial intelligence. In *Machine Intelligence and Pattern Recognition*, volume 4, pages 127–135. Elsevier, 1986.

[27] Yi-Dong Shen and Thomas Eiter. Determining inference semantics for disjunctive logic programs (extended abstract).

[28] Patrik Simons, Ilkka Niemelä, and Timo Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138(1-2) :181–234, 2002.

[29] Philippe Smets. Imperfect information : Imprecision and uncertainty. In *Uncertainty management in information systems*, pages 225–254. Springer, 1997.

[30] Matthias Thimm. On the expressivity of inconsistency measures. *Artificial Intelligence*, 234 :120–151, 2016.

[31] Matthias Thimm. Inconsistency measurement. In *International Conference on Scalable Uncertainty Management*, pages 9–23. Springer, 2019.

[32] Matthias Thimm and Johannes P Wallner. On the complexity of inconsistency measurement. *Artificial Intelligence*, 275 :411–456, 2019.

[33] Markus Ulbricht, Matthias Thimm, and Gerhard Brewka. Measuring inconsistency in answer set programs. In *European Conference on Logics in Artificial Intelligence*, pages 577–583. Springer, 2016.

[34] Yi Wang and Joohyung Lee. Handling uncertainty in answer set programming. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[35] Michael Wick, Sameer Singh, Ari Kobren, and Andrew McCallum. Assessing confidence of knowledge base content with an experimental study in entity resolution. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 13–18, 2013.

[36] Ronald R Yager and Liping Liu. *Classic works of the Dempster-Shafer theory of belief functions*, volume 219. Springer, 2008.

L'ontologie E-Phy, une base de connaissances pour le catalogue des produits phytopharmaceutiques autorisés en agriculture en France

Syphax Bouazzouni,[1,2] Clement Jonquet[1,3]

[1] LIRMM, Univ. de Montpellier, CNRS, France
[2] Ecole Nationale Supérieure d'Informatique, Alger, Algérie
[3] MISTEA, Univ. de Montpellier, INRAE, Institut Agro, France

gs_bouazzouni@esi.dz, jonquet@lirmm.fr

Résumé

Le développement de ressources sémantiques (ontologies, vocabulaires, graphes de connaissances, etc.) est une activité clé pour faciliter l'intégration et l'interopérabilité des données en agriculture. Bien souvent, les catalogues ou les référentiels officiels ne sont pas du tout "FAIR," et n'existent pas dans un format RDF, comme dans notre cas, en agriculture, le catalogue E-Phy, produit par l'ANSES, qui contient l'ensemble des produits phytopharmaceutiques et de leurs usages, des matières fertilisantes et des supports de culture autorisés en France. *Dans ce travail, nous détaillons notre démarche pour formaliser le catalogue E-Phy sous forme d'une base de connaissances OWL constituée d'un modèle ontologique, de ses instances et d'alignements vers d'autres ontologies. Nous montrons les points difficiles rencontrés dans ce processus, et les limites de la modélisation actuelle restée rétro-compatible avec la base de données d'origine. Nous illustrons également la valeur ajoutée de l'ontologie E-Phy via des requêtes SPARQL qui valorisent la sémantique et les alignements et permettent des interrogations impossibles sur les données d'origine.*

Mots-clés

Ontologie, développement de ressource sémantique, RDFisation, agriculture, ANSES, E-Phy, produits-phytosanitaires ou phytopharmaceutiques

Abstract

The development of semantic resources (ontologies, vocabularies, knowledge graphs, etc.) is a key activity to facilitate data integration and interoperability in agriculture. Often, catalogs or official reference lists do not respect the FAIR principles, and do not exist in an RDF format, as in our case, in agriculture, the E-Phy catalog, produced by ANSES, which contains all the plant protection products (phytosanitary) and their uses, fertilizers and growing media authorized in France. *In this work, we detail our approach to formalize the E-Phy catalog in the form of an OWL knowledge base consisting of an ontological model, instances and alignments to other ontologies. We show the various issues encountered in this process, and the limitations of the current model, which is still backward compatible with the original database. We also highlight, with a few SPARQL queries, the added value of the E-Phy ontology's semantics and alignments with queries impossible on the original data.*

Keywords

Ontology, semantic resource development, RDFisation, agriculture, ANSES, E-Phy, phytosanitary or phytopharmaceutical products

1 Introduction

De nombreux référentiels ou catalogues officiels existent dans le domaine de l'agriculture, en France ou dans le reste du monde. Ces référentiels sont produits par des organismes différents le plus souvent accrédités pour les maintenir, et avec des formats, des processus de production, et de maintenance variés. Par exemples, le *Catalogue officiel des espèces et variétés de plantes cultivées en France* (GEVES), le *catalogue officiel des variétés de vigne* (FranceAgriMer). Dans ce travail nous nous intéressons au *catalogue des produits phytopharmaceutiques et de leurs usages, des Matières Fertilisantes et des Supports de Culture autorisés en France* (ANSES)[1] et à son sous ensemble le *guide des produits de protection des cultures utilisables en agriculture biologique en France* (ITAB)[2] qui décrivent des intrants dont les usages sont eux même décrits dans le *catalogue des usages phytopharmaceutiques* (Ministère de l'Agriculture). Ces référentiels sont indispensables en agriculture mais ils sont pratiquement inutilisables lorsqu'il s'agit de les utiliser pour décrire ou structurer des données : localisation variées, formats hétérogènes et pas exploitables par une machine, pas d'identifiant, maintenance, etc. Ils contiennent une connaissance très riche qui pourrait être mieux représentée pour ensuite être mieux exploitée dans le cadre d'applications ou de recherches en agriculture.

Dans le cas du catalogue des produits phytopharmaceutiques, une première étape a été franchie avec la mise à disposition, depuis 2016, des référentiels produits par l'ANSES[3] sur la plateforme nationale d'ouverture et de partage des données publiques data.gouv.fr. On y trouve le catalogue sous forme d'export de données de l'application web E-Phy créé par l'ANSES pour accéder/rechercher le catalogue en ligne. Ce

[1] www.anses.fr/fr/content/registre-des-amm-de-produits-phyto-et-mfsc
[2] www.itab.asso.fr/downloads/com-intrants/guide-protection-plantes6.pdf
[3] Agence nationale de sécurité sanitaire de l'alimentation, de l'environnement et du travail.

catalogue est proposé sur la plateforme sous deux formats CSV ou XML qui sont des formats libres largement utilisés sur le web et qui ont l'avantage d'être très simples à utiliser, qui structurent un minimum les données, mais qui souffrent aussi terriblement d'un manque de sémantique (pas d'identifiant, pas de hiérarchie ou de typage). L'absence de contraintes d'intégrité engendre également de nombreuses erreurs (de saisie, de valeurs, ou de valeur nulle).

C'est avec l'objectif d'avoir une meilleure structuration des données et pour permettre une meilleure réutilisation et interopérabilité que la prochaine étape serait maintenant de construire une base de connaissances de la base de données E-Phy en l'encodant avec les technologies du web sémantique (e.g., SKOS, OWL) et en l'enrichissant d'alignements avec d'autres ontologies ou vocabulaires standards en agronomie. Ainsi, il s'agit de faire passer les données d'E-Phy du niveau 3 étoiles du fameux modèle des données ouvertes et liées de Berners-Lee [1], au niveau 5 étoiles où les données sont identifiées par des URI et publiées sur le web dans un format ouvert, non-propriétaire, riche et standard tel que RDF.

La démarche de création de ressource sémantique RDF a été largement décrite dans la littérature que ça soit dans le domaine de l'agronomie –par exemple, la définition d'une ontologie pour les stades phénologiques des plantes cultivées [2] –ou dans d'autres domaines comme le médical –par exemple MuEVo, un vocabulaire multi-expertise (patient/médecin) dédié au cancer du sein extrait de données de forum [3]. Parfois, des méthodes spécifiques sont utilisées, comme la méthode Linked Open Terms [4] ou DataLift[1]. Mais dans le cas des démarches de "RDFisation" de données préexistantes, le plus compliqué est l'instanciation des éléments des ressources d'origine et leur alignement à d'autres ressources. Pour cela divers outils et projets existent pour faciliter la transformation des données tabulaires ou XML en données sémantiquement structurées et interconnectées. Comme exemples nous pouvons citer Any23[4], Triplify / Sparqlify [5], Open Refine[5] ou Cellfie.[6] Ces méthodes et outils permettent de faire le mapping entre les éléments des fichiers d'origine vers un modèle défini par une étude de l'existant, cependant leur utilisation n'évite pas un prétraitement pour s'adapter à la structure de la ressource et préparer le dit mapping.

Nous nous sommes alors orientés vers une méthodologie en trois étapes: étude de l'existant en créant un diagramme de classe UML de la ressource d'origine, création de la structure de l'ontologie avec Protégé [6] et instanciation/alignement avec une phase de prétraitement avec Talend,[7] puis le plug-in Cellfie de Protégé.

Dans cette article, nous présentons la démarche que nous avons suivi pour transformer le catalogue des produits phytopharmaceutiques de l'ANSES en une base de connaissances OWL dont le modèle ontologique est celui de la base de données E-Phy et les instances sont les produits qui y sont listés, alignées avec d'autres ressources sémantiques tel que le thésaurus French Crop Usage pour les cultures et l'ontologie CHEBI pour les familles chimiques. Cependant considérant que nos organismes ne sont pas des autorités pour ce catalogue, notre modèle de données est volontairement "bridé"[8] pour être complètement rétro-compatible avec la base d'origine de façon à facilement mettre à jour notre ontologie à partir de nouveaux exports de l'ANSES.

Nous illustrons, à travers des requêtes SPARQL, comment l'ontologie E-Phy permet de répondre à des requêtes impossibles sur les données d'origines car elles valorisent la sémantique des ontologies alignées. Par exemple, avec l'utilisation de la hiérarchie des cultures du thésaurus French Crop Usage pour obtenir tous les produits utilisables pour une famille de culture donnée. Nous pouvons également obtenir avec une requête SPARQL une vue (i.e., un sous-ensemble de triplets) de l'ontologie E-Phy qui représente le *catalogue des usages phytopharmaceutiques* produit par l'ITAB à partir des données de l'ANSES.

L'article est organisé comme suit : Section 2, nous commençons par une présentation détaillée du catalogue avec une description des fichiers sources, de ses contenus et de sa structure. Section 3, nous présentons la méthodologie suivie. Section 4, nous présentons nos résultats avec le détail technique, l'ontologie produite et les difficultés rencontrées et les requêtes SPARQL. Finalement, la section 5 conclut et donne quelques perspectives.

2 Présentation du Catalogue E-Phy

Le catalogue E-Phy contient l'ensemble des données des produits (produits phytopharmaceutiques, matières fertilisantes et supports de culture, adjuvants, produits mixtes et mélanges) couverts par une Autorisation de Mise sur le Marché (AMM) ou un permis de commerce parallèle. En France, les décisions d'AMM et de permis sont délivrées par l'ANSES depuis juillet 2015. L'agence partage ce catalogue via l'application web E-Phy (www.ephy.anses.fr) qui est un reflet de l'état actuel des autorisations de produits et qui permet de faire des recherches pour retrouver un produit par numéro AMM, usage ou composition. Ces données servent principalement aux professionnels du secteur pour savoir si un produit est autorisé ou pour connaître les substances actives, le titulaire des autorisations, et les usages possibles d'un produit.

2.1 Description des fichiers sources

Les fichiers publiés mensuellement sur la plateforme data.gouv.fr[9] sont un export du catalogue en CSV et XML. Les fichiers proposés au format CSV, sont vérifiés et offrent une équivalence complète avec les données publiées sur le site E-Phy et sont au nombre de neuf :

— La liste des produits autorisés ou retirés,
— La liste des usages des produits (hors MFSC),

[4] http://any23.apache.org
[5] https://openrefine.org
[6] https://github.com/protegeproject/cellfie-plugin
[7] https://www.talend.com/fr/

[8] C'est-à-dire que si nous avions dû concevoir un modèle de données "de zéro", nous n'aurions pas forcément modélisé cela. Par exemple, dans une ontologie on évite les classes qui sont des unions de concepts (ici MFSC).
[9] www.data.gouv.fr/fr/datasets/donnees-ouvertes-du-catalogue-e-phy-des-produits-phytopharmaceutiques-matieres-fertilisantes-et-supports-de-culture-adjuvants-produits-mixtes-et-melanges

— La liste des usages des produits (hors MFSC) autorisés,
— La liste des phrases de risque des produits,
— La liste des substances actives,
— La liste des conditions d'emploi des produits,
— La liste des classes et des mentions danger des produits (hors MFSC),
— La liste des usages des MFSC et produits mixtes,
— La liste des compositions des MFSC et produits mixtes.

Quant aux fichiers XML, ils sont au nombre de six : un fichier XML contenant les données et cinq autres de type XSD faisant office de description de la structure des données.

2.2 Description des caractéristiques du catalogue

Le catalogue E-Phy contient principalement des intrants. Les intrants sont tous *les différents produits apportés aux terres et aux cultures, qui ne proviennent ni de l'exploitation agricole, ni de sa proximité. Les intrants ne sont pas naturellement présents dans le sol, ils y sont rajoutés pour améliorer le rendement des cultures*.[10] Parmi ces produits :

— Les *PPPs* (Produit PhytoPharmaceutiques) sont des préparations destinées à protéger les végétaux et les produits de culture.
— Les *MFSCs* (Matières Fertilisantes et Supports de Culture) sont des produits destinés à assurer ou à améliorer la nutrition des végétaux, ainsi que les propriétés des sols. Les supports de culture sont destinés à servir de milieu de culture à certains végétaux.
— Les *adjuvants* sont des substances qui renforcent l'action des produits phytosanitaires en augmentant le pouvoir d'absorption du produit par la plante, un insecte, le bois, etc.
— Les *produits mixtes* sont composés soit d'une matière fertilisante ou d'un support de culture et d'un produit phytopharmaceutique, de façon à avoir un double effet.
— Les *mélanges* se composent de plusieurs produits phytopharmaceutiques rendus solubles et bénéficiant chacun d'une autorisation de mise sur le marché à titre individuel.

Le catalogue contient, comme indiqué sur la Figure 1, majoritairement des PPPs. Dans sa version d'octobre 2020, que nous avons utilisé, il en compte 13087 pour un total de 14093 intrants.

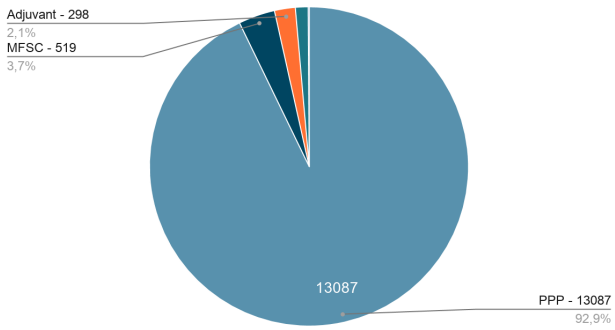

Fig. 1. Répartition des types d'intrants dans E-Phy.

[10] https://www.agriculture-nouvelle.fr/qu-est-ce-qu-un-intrant

En plus des intrants le catalogue contient d'autres informations utiles telles que :

— Les *substances actives*, qui constituent le principe actif des produits, ce sont elles qui agissent sur les nuisibles. Les substances actives sont homologuées au niveau européen; la version que nous avons utilisée en compte 1247.
— *Le catalogue des usages*, auquel E-Phy fait référence, est publié sous forme de note de service au Bulletin officiel du ministère chargé de l'agriculture.[11] Ce dernier n'existe pas non plus sous forme RDF. La version que nous avons utilisée compte 1540 usages.
— Les *cultures préconisées* pour lesquelles un MFSC ou un produit mixte peut être utilisé ; ils sont au nombre de 144 dans la version actuelle.

3 Méthodologie

Afin d'atteindre notre objectif d'avoir une ressource RDF représentant le catalogue E-Phy et dont le processus de création soit automatisé et reproductible, nous avons analysé sa structure, son contenu, essayé de comprendre le sens de chacun de ces éléments afin d'en avoir une vue d'ensemble. L'objectif était de pouvoir détecter des axes d'amélioration que ça soit côté modélisation, OWL nous permettant d'exprimer des aspects non envisageables dans le modèle relationnel de base, ou côté alignement avec d'autres ressources existantes. Durant cette première étape nous sommes parti d'un diagramme de classe UML strictement identique au modèle des données sources auquel nous avons ajouté petit à petit des modifications afin de le rendre compatible avec une approche orientée ontologie. Le tout en gardant une rétrocompatibilité avec le modèle original pour ne pas trop éloigner les données et également afin de pouvoir automatiser la re-création de l'ontologie et de la base de connaissance à chaque mise à jour des données source. Dans une deuxième étape, nous avons construit une ontologie OWL qui implémente le modèle de données et qui servira de structure d'accueil pour les instances du catalogue.

Après finalisation du modèle ontologique, la prochaine étape a été de préparer l'alignement. Cela consiste à rechercher les ressources cibles candidates pour établir des correspondances avec les classes et les instances de l'ontologie E-Phy et à évaluer pour chacune leurs taux de couverture des données afin de décider de la pertinence de l'alignement et des stratégies à suivre en cas d'éléments manquants.[12] Ces correspondances/alignements peuvent ensuite être utilisés pour diverses tâches d'intégration de données. Nous avons mis en place une méthode semi-automatique avec une validation humaine –faites par les auteurs– pour certain des termes à aligner (éléments chimique, substances, etc) pour lesquels de nombreuses correspondances possibles ont été identifiées en utilisant des portails d'ontologies / vocabulaires de références comme BioPortal et AgroPortal. Il fallait alors sélectionner les plus pertinentes et veiller à garder la cohérence sémantique

[11] https://daaf.reunion.agriculture.gouv.fr/Catalogue-national-des-usages

[12] Pour favoriser la valorisation des alignements, il est fréquent et logique de chercher à identifier la ressource sémantique la plus répandue/standard qui recouvre au mieux l'ontologie à aligner.

(e.g., liée à la hiérarchie d'origine). Nous avons utilisé `skos:exactMatch` ou `skos:closeMatch` pour encoder les alignements des classes de l'ontologie.
Finalement, l'étape d'instanciation (c.-à-d l'importation des données dans l'ontologie) clôture le processus, comme indiqué sur la Figure 3 et détaillée ci-après.

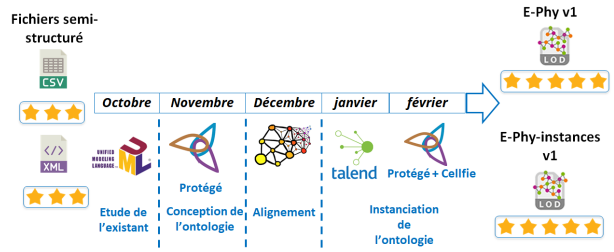

Fig. 2. Organisation dans le temps de la démarche de RDFisation.

La première phase est la préparation des données en fichiers Excel prêts à l'import à partir des fichiers CSV d'origine. Pour ces opérations de prétraitement, nous avons utilisé l'outil d'ETL Talend, en exécutant des scripts automatisés et reproductibles. Nous avons nettoyé les doublons et réorganisé les fichiers pour utiliser Cellfie, le plugin de Protégé qui crée les instances RDF en se basant sur des règles de mapping. Nous avons également créé le patron pour les URI en réutilisant les identifiants existants ou, lorsque nécessaire, en créant certains identifiants à partir d'un label (si unique) (e.g., *"ColorantBleuBrillant(acideBlue9)"* pour une substance) ou avec un numéro unique (e.g., *"usage_10"* pour un usage)). Nous avons également manipulé les données comme suit :

- Création des doses (e.g., *"dose min par apport : 2.0 L/ha"* pour un MFSC) en éclatant le champ texte d'origine en deux parties la valeur et l'unité;
- Scinder les colonnes de type liste en plusieurs lignes avec une colonne clé commune (e.g., la colonne "Variant" qui a comme valeur une liste de variants séparés par le caractère |: "bromoxynil | bromoxynil octanoate | bromoxynil butyrate | bromoxil (octanoate, heptanoate)");
- Éclater les identifiants des usages en trois parties, pour obtenir la portée d'usage, la méthode d'application et le groupe de nuisible (e.g., scinder "Ananas*Trt Part.Aer.*Act. Floraison" en "Ananas" , "Trt Part.Aer." et "Act. Floraison");
- Ajouter les alignements (détails section 4.1.3).

La deuxième phase de l'instanciation utilise les fichiers produits précédemment pour créer des instances dans l'ontologie (modèle) et la transformer en base de connaissances. Pour cela nous avons utilisé Protégé et son plugin Cellfie qui utilise le language *Mapping Master*[13] pour définir les mappings du contenu d'une feuille de calcul vers un triplet RDF.
Par exemple, la règle suivante permet de créer un individu de la classe *"Substance"* avec la valeur de la colonne B comme URI et label.

Individual: @B*
types: Substance
annotations:
rdfs:label @B*

Une phase de validation/contrôle de l'instanciation a été faite à postériori en comparant le nombre d'éléments de chaque type dans les fichiers sources et dans l'ontologie avec respectivement Talend et des requêtes SPARQL.

4 Résultats

4.1 La base de connaissances E-Phy

4.1.1 Etude de l'existant

Le but était de construire un diagramme de classe UML représentant la donnée d'origine, au total nous avons obtenu 18 classes (Table 2 et Figure 2).

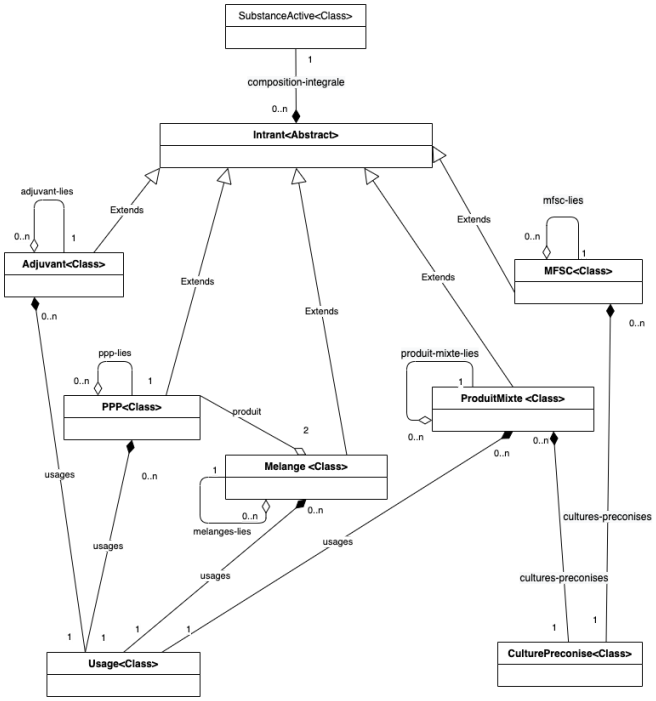

Fig. 3. Schéma UML simplifié d'une partie de la structure de la base de données du catalogue E-Phy.

Table 1. Description des principales classes du modèle UML.

Nom de la classe	*Exemple de propriétés*
Intrant (parent des classes PPP, MFSC, Adjuvant, Mélanges et Produit-mixte)	titulaire, type-produit, etat-produit, numero-AMM, nom-produit, type-commercial, mentions-autorisees, date-première-autorisation
Usage (utilisé par les classes PPP, Adjuvant, Mélanges et Produit-mixte)	id, identifiant-usage, stade-cultural-min, stade-cultural-max, etat-usage, dose-retenue
culture-preconise (utilisé par les classes MFSC et Produit-mixte)	id, type-culture, culture-commentaire , etat, date-decision

[13] www.github.com/protegeproject/mapping-master

Substance (utilisé par intrant)	identifiant, famille-chimique, nom-produit, etat-produit, variants, numero-cas

A noter que beaucoup de propriétés définies dans les fichiers XSD ne sont jamais utilisées dans les données, tel que *produit-importé* d'Intrant, identifiant-*usage-groupe -organisme-nuisible, identifiant-usage-portée-usage*, *identifiant-usage-methode-application* d'Usage ou encore *autres-noms* de Substance. Nous avons noté aussi des éléments existants seulement dans le XML tel que les familles chimiques et ou a contrario seulement dans les CSV tel que les numéro-cas des substances.

4.1.2 Construction de l'ontologie

L'ontologie (ephy-v1.owl) obtenue à cette étape représente seulement la structure du catalogue E-Phy obtenu à partir des fichiers CSV/XML d'octobre 2020, les données en elle-même seront ajoutées après comme instances dans une base de connaissances séparée (ephy-full-v1.owl) qui importera (`owl:imports`) la structure. Nous avons explicitement séparé la structure des individus pour qu'elle puisse être utilisée individuellement au besoin, entre autres pour la vue des données ITAB (section 4.3). Pour chaque classe de l'ontologie, nous avons déclaré un label français (`rdfs:label`) et tant que possible une définition (`skos:definition`). La métadonnée de l'ontologie a été également décrite en utilisant les propriétés recensées dans MOD [7]. MOD intègre au total 23 vocabulaires de métadonnées existants standards (e.g, Dublin Core, OMV, DCAT, VoID) pour la description et la publication d'une ontologie. La Table 3 détaille quelques métriques sur l'ontologie créée.

Table 2. Métriques de l'ontologie E-Phy de structure

Métrique	***Valeur***
Nombre de classes	22
Nombre de propriétés objets (object properties) utilisées	28
Nombre de propriétés de données (data properties) utilisées	31
Nombre de propriétés d'annotations (annotation properties) utilisées	50
Nombre d'axiome au total	364

4.1.3 Alignement

La liste des ressources sémantiques que nous avons identifiées pour établir des correspondances avec l'ontologie E-Phy est décrite dans la Table 4 pour les instances et dans la Table 5 pour les classes.

Table 3. Ontologies cibles pour les instances.

Classe et ressource sémantique candidate	***Taux de couverture***	***Décision***
Alignement de famille chimique avec CHEBI *Chemical Entities of Biological Interest Ontology* est une classification structurée des composés chimiques d'importance biologique développée dans le cadre de l'OBO Foundry.	**83.33%** (115 familles chimiques trouvées et 23 non trouvées)	Aligner avec CHEBI lorsque possible en utilisant directement les URI des instances de la classe 'Chemical Entity' de CHEBI, sinon créer des nouvelles instances de cette classe dans notre espace de nom.
Alignement des unités avec UO *Units of Measurement Ontology* est un ensemble de d'unités métriques à utiliser avec entre autre avec PATO (Phenotypic Quality Ontology)	**18.35%** (29 unités trouvées et 129 non trouvées)	Abandonner l'alignement à UO car trop spécifique à la ressource E-Phy. Un travail plus significatif d'encodage des unités serait nécessaire.
Alignement des portées des usages et des types de culture avec FCU French Crop Usage est un thesaurus des types de cultures, organisés en fonction de leur usage, développé par INRAE et basé sur les définitions du Larousse Agricole.	**98,8%** (171 cultures trouvées et 2 cultures non trouvées). Les non trouvées sont en fait des cas spéciaux (e.g. "autres cultures").	Aligner avec FCU lorsque possible en utilisant les URI des instances (`skos:Concept`). Solliciter les auteurs de FCU pour leur proposer les cultures manquantes et pertinentes (ce qui nous a fait passer d'un taux de 63% à 98%).

Table 4. Ontologies cibles pour les classes.

Classes candidate	***Équivalents***
Culture (`skos:exactMatch`)	• crop - AnaEE Thesaurus • crop - Agronomy Ontology • crops - AGROVOC • crops - NALT
Saison d'application (`skos:closeMatch`)	• seasons - NALT, • season - ENVO, • seasons - AGROVOC
Intrant (`skos:exactMatch`)	• farm inputs - AGROVOC • farm inputs - NALT
Adjuvant (`skos:exactMatch`)	• adjuvants (AGROVOC) • Adjuvants (CHEBI) • adjuvants (NALT)
MFSC (`skos:exactMatch`)	• fertilizer - Agronomy Ontology • fertilizers (NALT) • fertilizers (AGROVOC)
PPP (`skos:exactMatch`)	• phytosanitary - AnaEE Thesaurus • pesticides (AGROVOC) • pesticides (Nalt) • pesticide (CHEBI)
Substance (`skos:exactMa`	• chemical substance - chebi • chemical substances -NALT)

tch)	• chemical substances (AGROVOC)
usage (skos:closeMatch)	• uses (AGROVOC)
restrictions (skos:closeMatch)	• use restrictions (AGROVOC)
dose (skos:exactMatch)	• dose specification (OBI) • dose (GEMET) • dosage (AGROVOC)

La recherche des ressources sémantiques candidates a été faite en utilisant les outils de recommandations (Recommender Service) des plateformes NCBO BioPortal [8] et AgroPortal [9]. Cette étape a été compliquée pour plusieurs raisons :

- Il fallait trouver les synonymes des labels qui ne matche pas directement.
- Les termes d'origine étant en français, il a fallu passer par une phase de traduction en anglais, non triviale pour les noms scientifiques.
- Le cas "Autres", utilisé très fréquemment dans les cultures (684 fois) est impossible à aligner.
- Le manque de cohérence des données d'origine, e.g., dans les cultures on peut trouver des traitements ou des types de terrains.
- Des granularités différentes dans les termes utilisés, avec par exemple des familles de culture et des cultures dans les mêmes champs/propriétés.
- Le cas des éléments trop spécifiques à la ressource d'origine, tels que les unités, qui font qu'il n'y a plus trop d'intérêt à faire l'alignement, vu le faible taux d'équivalence.

A terme, il nous faudrait une validation des alignements par des experts du sujets, entre autres pour les familles chimiques dont les noms peuvent avoir plusieurs écritures et variantes. Également, les Substances pourraient être également alignées à CHEBI mais leur nombre élevé (1274) nécessiterait la mise en place d'une méthode automatique plus fiable.

4.1.4 Instanciation

La difficulté de cette étape vient principalement du fait que les fichiers CSV ne contenaient pas toutes les données et que certaines ne se trouvaient que dans le XML (e.g., familles chimique d'une substance, nombre d'apport min et max de culture préconisé). Ces données ont dû être extraites séparément à partir du fichier XML. Les erreurs de saisie ont aussi représenté un problème significatif, tels que le nom d'un même élément écrit différemment (e.g., *"Pois écossées frais"* et "*Pois écossés frais*" ou *"Trt Part.Aer. "* sans point final et "*Trt Part.Aer*" avec).

L'utilisation de Cellfie était ensuite relativement simple une fois les fichiers préparés en amont. La Table 5 donne des exemples de règles Cellfie utilisées pour l'instanciation des substances.

Table 5. Instanciation des substances.

Fichier source	"substance_active_v3_Windows-1252.csv" et "famille_chimique.csv"	
Nom du fichier produit	**Nom de ces colonnes**	**Règle Cellfie utilisée**
subtance_active_simple.xlsx	A famille_chimique B Nom_substance_active C Etat_d_autorisation	`Individual:@B* types: Substance annotations: rdfs:label @B* facts: etat @C*, familleChimique @A*`
	Explication	Créer un individus si non existant de la classe "Substance" avec la valeur de la colonne "B" comme URI et label et la colonne" C","A" comme valeur des propriétés etat et familleChimique respectivement
substances_active_variants.xlsx	A Nom_substance_active B Variant	`Individual: @A* facts: variant @B*`
	Explication	Créer un individus de la classe "Substance" si non existant avec la valeur de la colonne "A" comme URIet la colonne" B" comme valeur de la propriété variant

La base de connaissances obtenue contient l'intégralité des données du catalogue E-Phy (CSV et XML), apporte en plus un alignement avec CHEBI et FCU et enrichissent les données existantes d'informations extraites des champs textuels comme : les portées d'usage, les méthodes d'application, les groupes de nuisibles, les valeurs et unités des doses, les substances et les teneurs des substances actives. La Table 6 synthétise le nombre d'instances créées dans la base de connaissances.

Table 6. Métriques sur la base de connaissances E-Phy.

Métrique	***Valeur***
Nombre total d'individus	127 280
Nombre de propriétés d'objets (object properties) utilisées	291 162
Nombre de propriété de données (data properties) utilisées	377 764
Nombre de propriétés d'annotations (annotations properties) utilisées	72 880
Nombre total d'axiomes	996 702

Le code des scripts de traitement et de transformation ainsi que la documentation nécessaire pour reproduire une nouvelle version de l'ontologie E-Phy est publiquement disponible : https://github.com/d2kab/E-Phy-Ontology

Le dépôt GitHub contient également les fichiers intermédiaires ainsi que les diagrammes UML des données sources. L'ontologie E-Phy (plus exactement la base de connaissances) est mise à disposition[14] sur AgroPortal à l'URL : http://agroportal.lirmm.fr/ontologies/E-PHY [15]

4.2 Requêtes SPARQL

Pour illustrer la valeur ajoutée de l'ontologie E-Phy nous avons préparé quelques requêtes SPARQL qui valorisent la sémantique et les alignements et permettent des interrogations impossibles sur les données d'origine. Elles correspondent à des cas d'usage pressenti pour l'ontologie E-Phy.

Par exemple, la requête suivante exploite l'héritage et les synonymes du thesaurus FCU pour obtenir tous les produits utilisables pour la famille de culture "Arboriculture fruitière" à partir de son synonyme "verger" :

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX cropusage: <http://ontology.irstea.fr/cropusage/>
PREFIX ephy: <http://www.d2kab.org/ontologies/ephy#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT distinct ?pl ?ptypel ?fcuL ?fcud ?fcuInfo
WHERE {
 SERVICE <http://ontology.irstea.fr/cropusage/sparql>
 {
        ?fcuP a skos:Concept.
        {
          ?fucP skos:altLabel ?fcuPsyn.
          filter(?fcuPsyn = "verger"@fr)
        }
        UNION
        {
          ?fucP skos:prefLabel ?fcuPL.
          filter(?fcuPL = "verger"@fr)
        }
        ?fcu skos:broader* ?fcuP ;
             skos:definition ?fcud;
             skos:prefLabel ?fcuL;
             rdfs:seeAlso ?fcuInfo
 }
 {
    ?p rdfs:label ?pl.
    ?p a ?ptype.
    ?ptype rdfs:label ?ptypel.

   {
       ?iu ephy:porteeUsage ?fcu.
       ?u ephy:identifiantUsage ?iu.
       ?p ephy:usage ?u.
   }
   UNION
   {
       ?cp ephy:typeCulture ?fcu.
       ?p ephy:culturePreconise ?cp.
   }
}}
```

Exemple de résultats :

Table 7. Exemple d'un résultat de la requête utilisant l'alignement avec FCU .

pl (Label du produit)	"KORI FEUILLE"
ptypel (Type du produit)	"Matières fertilisantes et supports de culture" @fr
fcuL (Label de la culture dans FCU)	"arboriculture fruitière" @fr
fcud (Définition de la culture selon FCU)	"cultures des arbres et arbustes qui produisent des fruits comestibles; Les fruits peuvent subir une transformation et être mangés sous forme de jus de fruits alcoolisés ou non." @fr
fcuInfo (URL extérieure pour avoir plus d'information)	https://fr.wikipedia.org/wiki/Arbre_fruitier @fr

D'autres exemples de requêtes permettent d':

- Exploiter les informations de CHEBI pour obtenir la définition de toutes les familles chimiques qui composent un produit.
- Exploiter la structure de l'ontologie pour obtenir tous les produits utilisables pour une culture donnée ou produit par une société spécifique (ou les deux).
- Exploiter la structure de l'ontologie pour obtenir tous les produits efficaces contre un groupe d'organismes nuisibles (à améliorer à l'avenir en liant les agresseurs à une base de référence).

Ces requêtes sont disponibles dans le dépôt GitHub.

4.3 Une vue pour l'AB: l'ontologie ITAB

La ressource ITAB est un autre 'catalogue' sous le nom de *Guide des produits de protection des cultures utilisables en Agriculture Biologique en France* produit par L'Institut Technique de l'Agriculture Biologique (ITAB)[16] qui maintient, de manière distincte de l'ANSES, une liste des produits de protection des cultures, utilisables dans le cadre de la production biologique. Cependant, tous les produits du guide ITAB (version de septembre 2014 en ligne) sont inclus dans la base de données E-Phy dénotés avec la propriété "mentions-autorisées=utilisable en agriculture biologique". Etant donné que le guide ITAB est une ressource en soi, d'intérêt pour les professionnels de l'agriculture biologique, nous avons produit une autre base de connaissances, sous la forme d'une vue (i.e., un sous ensemble de triplets) de la base de connaissances E-Phy. L'ontologie ITAB est ainsi une base de connaissances qui contient le même modèle ontologique que l'ontologie E-Phy mais ne contient que les instances explicitement déclarées pour l'agriculture biologique (ou les instances reliées).

En RDF, pour obtenir la vue ITAB, il suffit de faire une requête SPARQL qui extrait les triplets pertinents de la base de connaissances E-Phy :

```
SELECT ?x ?pp ?v
WHERE { ?p rdf:type/rdfs:subClassOf ephy:Intrant;
```

[14] Les URIs proposées ne sont pas déréférençables et seront sans doute amenées à changer.

[15] La version actuelle d'AgroPortal ne permet pas de visualiser les instances, même si elles sont bien stockées dans le portail et accessibles globalement ou pour une classe via l'API REST : http://data.agroportal.lirmm.fr/ontologies/E-PHY/instances

[16] www.itab.asso.fr/downloads/com-intrants/guide-protection-plantes6.pdf

```
        ephy:mentionAutoriser ?ma;
        (!rdf:null)* ?x.
    ?x ?pp ?v
    filter(?ma = "Utilisable en agriculture biologique")
}
```

Il ne reste plus qu'à ajouter ces triplets dans une nouvelle ontologie (itab-v1.owl) qui importe l'ontologie E-Phy et lui attribuer un identifiant et des métadonnées propres. La base de connaissance ITAB est disponible sur AgroPortal comme une vue de l'ontologie E-PHY à http://agroportal.lirmm.fr/ontologies/ITAB

5 Conclusions et perspectives

A partir des données d'origine (CSV et XML) nous avons pu construire l'ontologie E-Phy (version 1), une base de connaissances (modèle ontologique et instances) du catalogue des produits phytopharmaceutiques et de leurs usages, des matières fertilisantes et des supports de culture autorisés en France. Notre méthodologie a été automatisée (sauf l'étape alignement) et est facilement reproductible pour reconstruire l'ontologie quand les données originales seront mises à jour. Elle a permis d'identifier plusieurs erreurs dans les données : erreurs structurelles (e.g., un produit sans AMM ou deux produits avec le même AMM, ce qui devrait être totalement impossible), erreurs de saisie, des erreurs de valeurs, valeurs générique (e.g., "Autres", "Traitement généraux", "Tous"), des champs vides ou encore des éléments qui n'ont pas d'identifiant prédéfini. Nous travaillons actuellement sur une synthèse que nous fournirons à l'ANSES dans une démarche de qualité. Actuellement, en restant totalement rétrocompatible avec les données d'origine, nous n'avions pas l'autorité (ou l'expertise) pour changer ou corriger les erreurs, ou pour changer de manière significative le modèle de données. La RDFization que nous avons faite permet à minima de détecter plus facilement les problèmes et de les corriger. Cependant, notre démarche se veut également incitative pour encourager l'ANSES à adopter les technologies du web sémantique de manière native dans le développement du catalogue. Ainsi, l'agence pourrait assigner des URIs pérennes dont elle serait responsable.

Parmi les améliorations envisageables, dans la version actuelle, il s'agirait d'aller plus loin dans l'alignement en ajoutant les substances avec CHEBI et restructurer l'utilisation des unités dans les doses avec une ontologie des unités de mesures (UO, QUDT, etc.). Un autre axe d'amélioration serait de RDFizer le *catalogue national des usages phytopharmaceutiques* (actuellement disponible sous forme d'un document PDF seulement) produit par la direction générale de l'alimentation. Ainsi l'ontologie E-Phy pourrait se reposer formellement sur ce catalogue pour aller plus loin dans la représentation des usages.

Dans le cadre du projet ANR "Des Données aux Connaissances en Agronomie et Biodiversité (D2KAB – www.d2kab.org) nous prévoyons d'utiliser l'ontologie E-Phy pour annoter sémantiquement des occurrences de produits ou d'intrants dans les Bulletins de Santé du Végétal qui sont utilisés par les agriculteurs pour faire de la veille sanitaire pour leurs cultures. En partenariat avec la SME Elzeard, l'ontologie E-Phy sera utilisée pour construire un graphe de connaissances exploité dans une application pour la gestion des itinéraires culturaux à destination des maraîchers.

Remerciements

Ce travail a été réalisé dans le cadre du projet ANR D2KAB (ANR-18-CE23-0017) lors d'un stage soutenu par l'Institut de Convergence en Agriculture Numérique, #DigitAg (ANR-16-CONV-0004). Nous remercions le *service des systèmes d'informations des produits réglementés* de l'ANSES pour les renseignements sur la base E-Phy. Nous remercions également C. Roussey (INRAE) pour son aide sur les alignements avec FCU, S. Aubin (INRAE) et O. Corby (INRIA) pour son aide avec l'extraction de la vue ITAB.

Références

[1] T. Berners-lee, "Linked Data - Design Issues," *Design Issues*, 2006. https://www.w3.org/DesignIssues/LinkedData.html.

[2] C. Roussey, X. Delpuech, F. Amardeilh, S. Bernard, C. Jonquet. Semantic Description of Plant Phenological Development Stages, starting with Grapevine. *14th international conference on Metadata and Semantics Research Conference (MTSR)*, Dec 2020, Madrid, Spain. pp.257-268, ff10.1007/978-3-030-71903-6_25ff..

[3] S. Eholié, M. D. T. Nzali, S. Bringay, and C. Jonquet, "MuEVo, un vocabulaire multi-expertise (patient/médecin) dédié au cancer du sein," Jun. 2016.

[4] M. Poveda-Villalón, "A reuse-based lightweight method for developing linked data ontologies and vocabularies," in *Lecture Notes in Computer Science*, 2012, vol. 7295 LNCS, pp. 833–837, doi: 10.1007/978-3-642-30284-8_66.

[5] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller, "Triplify - Light-weight Linked Data publication from relational databases," in *WWW'09 - Proceedings of the 18th International World Wide Web Conference*, 2009, pp. 621–630, doi: 10.1145/1526709.1526793.

[6] J. H. Gennari *et al.*, "The evolution of Protégé: An environment for knowledge-based systems development," *Int. J. Hum. Comput. Stud.*, vol. 58, no. 1, pp. 89–123, Jan. 2003, doi: 10.1016/S1071-5819(02)00127-1.

[7] B. Dutta, A. Toulet, V. Emonet, and C. Jonquet, "New generation metadata vocabulary for ontology description and publication," in *Communications in Computer and Information Science*, Nov. 2017, vol. 755, no. 755, pp. 173–185, doi: 10.1007/978-3-319-70863-8_17.

[8] N. F. Noy *et al.*, "BioPortal: Ontologies and integrated data resources at the click of a mouse," *Nucleic Acids Res.*, vol. 37, no. SUPPL. 2, 2009, doi: 10.1093/nar/gkp440.

[9] C. Jonquet *et al.*, "AgroPortal: A vocabulary and ontology repository for agronomy," *Comput. Electron. Agric.*, vol. 144, pp. 126–143, Jan. 2018, doi: 10.1016/j.compag.2017.10.012.

Covid-on-the-Web: Graphe de Connaissances et Services pour faire Progresser la Recherche sur la COVID-19

F. Michel, F. Gandon, V. Ah-Kane, A. Bobasheva, E. Cabrio, O. Corby,
R. Gazzotti, A. Giboin, S. Marro, T. Mayer, M. Simon, S. Villata, M. Wincker.

University Côte d'Azur, Inria, CNRS, I3S (UMR 7271), France

franck.michel@cnrs.fr

Résumé

Le projet Covid-on-the-Web permet aux chercheurs d'accéder à la littérature relative à la famille des coronavirus, de l'interroger et d'en extraire des connaissances. Il s'aligne sur des besoins concrets formulés par des instituts de santé et de recherche. Ainsi, il adapte, combine et étend des outils destinés à traiter, analyser et enrichir le corpus CORD-19 qui rassemble plus de 100 000 articles scientifiques relatifs aux coronavirus. Ce jeu de données comprend deux principaux graphes de connaissances décrivant (1) 113 millions de mentions d'entités nommées liées au Web de données, et (2) les arguments extraits à l'aide d'ACTA, un outil d'extraction et de visualisation de graphes argumentatifs. Nous fournissons également plusieurs outils de visualisation et d'exploration basés sur la plateforme Corese, la bibliothèque MGExplorer, ainsi que des Notebooks Jupyter.

Mots-clés

COVID-19, arguments, visualisation, entités nommées, données liées.

Abstract

The Covid-on-the-Web project allows scientists to access, query and extract knowledge from the literature on the coronavirus family. It is aligned with concrete needs formulated by health and research institutes. Thus, it adapts, combines and extends tools designed to process, analyze and enrich the CORD-19 corpus, that gathers 100,000+ scientific articles related to the coronaviruses. This dataset comprises two main knowledge graphs describing (1) 113 million mentions of named entities linked to the Web of data, and (2) arguments extracted using ACTA, a tool for extraction and visualization of argumentative graphs. We also provide several visualization and exploration tools based on the Corese platform, the MGExplorer library, and Jupyter Notebooks.

Keywords

COVID-19, arguments, visualization, named entities, linkeddata.

1 Des données sur la COVID-19 vers des données ouvertes liées

En mars 2020, alors que la maladie infectieuse respiratoire COVID-19 nous obligeait à rester confinés, l'équipe de recherche Wimmics [1] a décidé de se joindre aux efforts de nombreux scientifiques du monde entier qui mettent à profit leur expertise et leurs ressources pour lutter contre la pandémie et en atténuer ses effets dévastateurs. Nous avons lancé un nouveau projet nommé *Covid-on-the-Web* visant à faciliter l'accès, la recherche et la compréhension de la littérature scientifique biomédicale relative au COVID. À cette fin, nous avons adapté, réorienté, combiné et utilisé des outils pour publier, aussi exhaustivement et rapidement que possible, un maximum de données liées relatives aux coronavirus.

En quelques semaines, nous avons déployé plusieurs outils afin d'analyser le *COVID-19 Open Research Dataset* (CORD-19) [18] qui compte plus de 100 000 articles scientifiques relatifs à la famille des coronavirus. D'une part, nous avons adapté la plateforme ACTA, [2] conçue initialement pour aider les cliniciens dans l'analyse des essais cliniques et la prise de décision [11], en permettant l'extraction automatique et la visualisation des graphes argumentatifs. D'autre part, notre expertise dans la gestion des données extraites à l'aide de graphes de connaissances, qu'elles soient génériques ou spécialisées, et leur intégration dans le projet HealthPredict [8, 9], nous ont permis d'enrichir le corpus CORD-19 avec différentes sources. Nous avons utilisé DBpedia Spotlight [5], Entity-fishing [3] et NCBO BioPortal Annotator [10] afin d'extraire les entités nommées des articles du corpus CORD-19, et les désambiguïser en regard des ressources des données ouvertes liées venant de DBpedia, Wikidata et BioPortal. En utilisant la plateforme Morph-xR2RML, [4] nous avons transformé le résultat en un jeu de données RDF que nous avons publié via un point d'accès SPARQL public. En parallèle, nous avons intégré les plateformes Corese [5] [4] et MGExplorer [3] pour mani-

1. https://team.inria.fr/wimmics/
2. http://ns.inria.fr/acta/
3. https://github.com/kermitt2/entity-fishing
4. https://github.com/frmichel/morph-xr2rml/
5. https://project.inria.fr/corese/

puler des graphes de connaissances et permettre leur visualisation et leur exploration sur le web.

Le projet Covid-on-the-Web (représenté dans la Figure 1) a ainsi conçu et mis en place un pipeline (workflow) intégré facilitant l'extraction et la visualisation des informations issues du corpus CORD-19 par la production et la publication d'un graphe de connaissances de données liées enrichi en permanence. Nous pensons que notre approche, qui intègre des structures argumentatives et des entités nommées, est pertinente dans le contexte actuel. En effet, alors que de nouvelles recherches liées à la COVID-19 sont publiées chaque jour, les résultats sont activement débattus, et de nombreuses controverses voient le jour (sur l'origine de la maladie, son diagnostic, son traitement...) [2]. Les chercheurs ont donc besoin d'outils pour les aider à étayer ou écarter certaines hypothèses, traitements ou explications. L'exploitation conjointe de structures argumentatives et de raisonnement basé sur les entités nommées peut aider à répondre aux besoins de ces utilisateurs et ainsi réduire les zones d'ombres liées à la maladie.

Cet article est un résumé long traduit et mis à jour de l'article [14] que nous avons publié à ISWC 2020 et dans lequel nous dressons un bilan ainsi qu'une comparaison avec les travaux connexes (non reproduits dans cet article).

Le reste de cet article est organisé de la manière suivante. Dans la section 2, nous détaillons le pipeline d'extraction mis en place pour traiter le corpus CORD-19 et générer des données RDF. Puis, la section 3 détaille les caractéristiques du jeu de données et des services mis à disposition pour l'exploiter. Les sections 4 et 5 présentent les outils d'exploitation et de visualisation, et traitent des applications futures et de l'impact potentiel de notre jeu de données.

2 De CORD-19 au jeu de données Covid-on-the-Web

Le *COVID-19 Open Research Dataset* [18] (CORD-19) est un corpus rassemblant des articles scientifiques liés au SARS-Cov-2 et à la famille des coronavirus. Les créateurs de CORD-19 ont traité plus de 100 000 articles et les ont convertis en documents JSON tout en nettoyant les citations et les références bibliographiques.

Cette section décrit comment nous avons exploité ce jeu de données pour *(1)* établir des liens significatifs entre les articles du corpus CORD-19 et le Web de données au moyen des entités nommées, et *(2)* extraire un graphe d'arguments découverts dans les articles, tout en reposant sur les normes du Web sémantique et les pratiques des données liées.

2.1 Construction du graphe de connaissances des entités nommées CORD-19

Le graphe de connaissances des entités nommées CORD-19 (CORD19-NEKG), décrit les entités nommées identifiées et désambiguïsées dans les articles du corpus CORD-19 à l'aide de trois logiciels : DBpedia Spotlight [5] pour désambiguïser et lier les ENs à DBpedia dont nous avons utilsé les modèles pré-entraînés [6] ; Entity-fishing [7] pour désambiguïser les ENs en regard à Wikidata ; et NCBO BioPortal Annotator [8] [10] qui permet d'annoter du texte biomédical et désambiguïser les ENs en regard des ontologies se trouvant sur BioPortal.

Listing 1 – Représentation de l'entité nommée "réaction en chaîne par polymérase" (PCR) comme une annotation du résumé d'un article présente de la position 235 à 238.

```
@prefix covidpr: <http://ns.inria.fr/covid19/property/>.
@prefix dct:     <http://purl.org/dc/terms/>.
@prefix oa:      <http://www.w3.org/ns/oa#>.
@prefix schema:  <http://schema.org/>.

[] a oa:Annotation;
  schema:about <http://ns.inria.fr/covid19/f74923b3...>;
  dct:subject "Engineering", "Biology";
  covidpr:confidence   "1"^^xsd:decimal;
  oa:hasBody <http://wikidata.org/entity/Q176996>;
  oa:hasTarget [
    oa:hasSource
    <http://ns.inria.fr/covid19/f74923b3...#abstract>;
    oa:hasSelector [
      a oa:TextPositionSelector, oa:TextQuoteSelector;
      oa:exact "PCR"; oa:start "235"; oa:end "238" ]];
```

Pour assurer sa réutilisabilité, CORD19-NEKG s'appuie sur des vocabulaires largement répandus, adaptés à la représentation des articles et des entités nommées en RDF. Nous présentons ci-dessous les principaux concepts de cette modélisation. De plus amples détails sont disponibles sur le dépôt Github du projet. [9]

Les métadonnées (ex. titre, auteurs, DOI) et le contenu des articles sont décrits à l'aide de DCMI, [10] FRBR-aligned Bibliographic Ontology (FaBiO), [11] Bibliographic Ontology, [12] FOAF [13] et Schema.org. [14] Les entités nommées sont représentées comme des annotations à l'aide du Web Annotation Vocabulary. [15] Un exemple d'annotation est donné dans le Listing 1. Le corps de l'annotation (`oa:hasBody`) correspond à l'URI de la ressource liée à l'entité détectée. Le morceau de texte reconnu comme l'entité nommée est la cible de l'annotation (`oa:hasTarget`). Celle-ci indique la partie de l'article dans laquelle l'entité a été reconnue (titre, résumé ou corps de l'article), et en donne sa position. Chaque annotation est accompagnée d'informations de provenance exprimées à l'aide de l'ontologie PROV-O, [16] indiquant la source en cours de traitement, l'outil utilisé pour extraire l'entité, le degré de confiance de l'annotateur sémantique, ainsi que l'auteur de l'annotation.

6. https://downloads.dbpedia.org/repo/dbpedia/spotlight/spotlight-model/
7. https://github.com/kermitt2/entity-fishing
8. http://data.bioontology.org/documentation
9. https://github.com/Wimmics/covidontheweb
10. https://www.dublincore.org/specifications/dublin-core/dcmi-terms/
11. https://sparontologies.github.io/fabio/current/fabio.html
12. http://bibliontology.com/specification.html
13. http://xmlns.com/foaf/spec/
14. https://schema.org/
15. https://www.w3.org/TR/annotation-vocab/
16. https://www.w3.org/TR/prov-o/

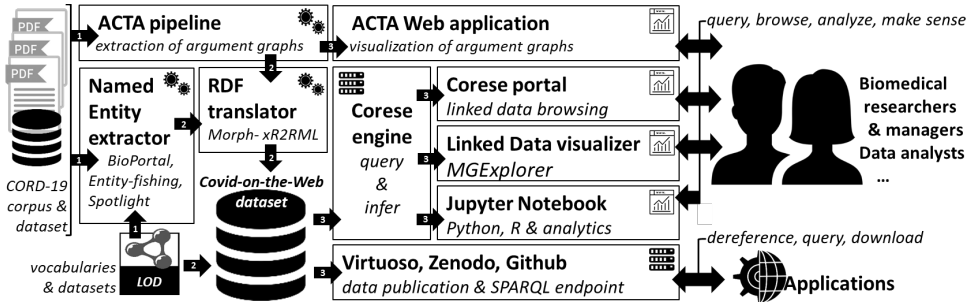

FIGURE 1 – Vue d'ensemble de Covid-on-the-Web : pipeline, ressources, services et applications.

2.2 Construction d'un graphe de connaissances d'argumentation

Argumentative **C**linical **T**rial **A**nalysis (ACTA) [11] est un outil conçu pour analyser les composants argumentatifs des essais cliniques, s'appuyant sur la méthode PICO. [17] Développé à l'origine comme un outil de visualisation interactif pour aider les cliniciens dans l'analyse des essais cliniques, nous avons adapté ACTA pour annoter le corpus CORD-19. ACTA va bien au-delà de la simple recherche par mot-clés en extrayant la ou les affirmations (claims) principales énoncées dans un article, ainsi que les preuves (evidences) supportant cette affirmation, et les éléments PICO correspondants.

Dans le contexte des essais cliniques, une *affirmation* est une déclaration finale faite par l'auteur sur le résultat de l'étude. Elle décrit généralement la relation d'un nouveau traitement par rapport aux traitements existants. Par conséquent, une observation ou une mesure est une *preuve* qui soutient ou attaque un autre composant argumentatif. Les observations comprennent les effets secondaires et les résultats obtenus. Deux types de relations peuvent exister entre les composants argumentatifs. La relation dit d'*attaque* tient lorsqu'un composant contredit la proposition de la composante cible, ou déclare que les effets observés ne sont statistiquement pas significatifs. La relation dit de *support* s'applique à toutes les déclarations ou observations justifiant la proposition du composant cible.

Chaque résumé du corpus CORD-19 a été analysé par ACTA [18] et le résultat représenté en RDF afin de générer le graphe de connaissances d'argumentation CORD-19 (CORD19-AKG). Le pipeline se compose de quatre étapes : *(1)* la détection des composants argumentatifs, c.-à-d. détecter les affirmations et les preuves, *(2)* la prédiction des relations existant entre ces composants, *(3)* l'extraction des éléments PICO, et *(4)* la production de la représentation RDF des arguments et des éléments PICO.

Détection de composants argumentatifs. Il s'agit d'une tâche d'étiquetage séquentiel où, pour chaque mot, le modèle prédit si le mot fait partie d'un composant ou non. Nous associons l'architecture BERT [6] à un LSTM (un réseau récurrent à mémoire court et long terme) [17] et un champ aléatoire conditionnel (conditional random fields) pour effectuer de la classification des unités lexicales (tokens). Les poids de BERT sont initialisés à partir des poids de SciBERT [1], ce qui permet une meilleure représentation textuelle des articles scientifiques utilisés dans un corpus tel que CORD-19. Le modèle optimisé (fine-tuned) sur un jeu de données annoté avec des composants argumentatifs obtient une f_1-score de 0,90 [12].

Classification des relations. Les composants argumentatifs extraits à partir de l'étape précédente sont ensuite évalués conjointement pour définir leurs inter-relations. Déterminer quelles relations existent entre les composants est une tâche d'étiquetage séquentiel à trois classes, où la séquence est constituée d'une paire de composants, et où la tâche est d'apprendre la relation entre eux, c.-à-d. *support, attaque* ou *aucune relation*.

Le transformeur SciBERT est utilisé pour créer la représentation vectorielle du texte en entrée, auquel on ajoute une couche linéaire afin de déterminer les relations. Le modèle est optimisé sur un jeu de données comportant les relations d'argumentation dans le domaine médical et obtient une f_1-score de 0,68 [12].

Détection des éléments PICO. Nous utilisons la même architecture que pour la détection des composants. Le modèle est entraîné sur le corpus EBM-NLP [16] afin de prédire la/le population / patient / problème (P de PICO), l'intervention (I de PICO) [19] et les critères de jugement (O de PICO) pour une entrée donnée. L'évaluation de la détection des éléments PICO obtient une f_1-score de 0.734 [11]. Chaque composant argumentatif est annoté avec les éléments PICO qu'il contient. Pour faciliter les requêtes structurées, les éléments PICO sont rattachés à leurs concepts UMLS (Unified Medical Language System) à l'aide de ScispaCy [15].

Graphe de connaissances d'argumentation. Le *graphe de connaissances d'argumentation CORD-19* (CORD19-AKG) s'appuie sur l'Argument Model Ontology (AMO), [20]

17. PICO est un cadre utilisé pour répondre aux questions de soins de santé dans la pratique fondée sur des preuves. Il signifie : patients/population (P), intervention (I), control/comparison (C) et outcome (O).

18. https://github.com/Wimmics/CovidOnTheWeb/tree/master/src/acta

19. L'étiquette d'intervention (I de PICO) et de comparaison (C de PICO) sont traitées comme appartenant à une même classe.

20. http://purl.org/spar/amo/

Listing 2 – Exemple des composants argumentatifs et de leur relation.

```
@prefix prov:   <http://www.w3.org/ns/prov#>.
@prefix schema: <http://schema.org/>.
@prefix aif:    <http://www.arg.dundee.ac.uk/aif#>.
@prefix amo:    <http://purl.org/spar/amo/>.
@prefix sioca:  <http://rdfs.org/sioc/argument#>.

<http://ns.inria.fr/covid19/arg/4f8...>
 a amo:Argument;
 schema:about <http://ns.inria.fr/covid19/4f8...>;
 amo:hasEvidence
 <http://ns.inria.fr/covid19/arg/4f8.../0>;
 amo:hasClaim <http://ns.inria.fr/covid19/arg/4f8.../6>.

<http://ns.inria.fr/covid19/arg/4f8.../0>
 a amo:Evidence, sioca:Justification, aif:I-node;
 prov:wasQuotedFrom <http://ns.inria.fr/covid19/4f8...>;
 aif:formDescription "17 patients discharged in
                      recovered condition...";
 sioca:supports
 <http://ns.inria.fr/covid19/arg/4f8.../6>;
 amo:proves <http://ns.inria.fr/covid19/arg/4f8.../6>.
```

le SIOC Argumentation Module (SIOCA) [21] et l'Argument Interchange Format. [22] Chaque argument identifié par ACTA est modélisé comme un `amo:Argument` dont les composants argumentatifs, affirmations et preuves, sont reliés. Les affirmations et les preuves sont elles-mêmes reliées par des relations de support ou de réfutation (avec respectivement les propriétés `sioca:supports`/`amo:proves` et `sioca:challenges`).

Le Listing 2 dresse un exemple des composants de ce graphe. Les éléments PICO (non illustrés dans le Listing 2) sont décrits comme des annotations des éléments argumentatifs dans lesquels ils ont été identifiés, de manière très similaire aux entités nommées (Listing 1). La différence est que le corps (body) de l'annotation contient les identifiants UMLS des concepts (CUI) et de types (TUI).

2.3 Génération du jeu de données

D'un point de vue technique, le corpus CORD-19 se compose d'un document JSON par article scientifique. La génération du jeu de données RDF Covid-on-the-Web implique donc deux étapes principales : (1) traiter chaque document du corpus pour en extraire les entités nommées et les arguments, et (2) traduire les résultats de ces deux traitements en un jeu de données RDF unifié et cohérent. L'ensemble du pipeline est décrit dans la Figure 1.

Extraction des entités nommées. Pour chaque article du corpus, DBpedia Spotlight, Entity-fishing et BioPortal Annotator produisent chacun un document JSON allant de 100 KB à 50 MB chacun. Ces documents ont été chargés dans une base de données MongoDB, et prétraités pour filtrer les données inutiles ou invalides (ex. les caractères non valides) ainsi que pour supprimer les entités nommées de moins de trois caractères. Ensuite, chaque document a été traduit en RDF tel que décrit dans la section 2.1 en utilisant Morph-xR2RML, [23] une implémentation pour MongoDB du langage de transformation xR2RML [13]. Les trois annotateurs sémantiques ont été déployés sur une Precision Tower 5810 équipée d'un CPU à 3,7 GHz et de 64 Go de RAM.

Pour que les fichiers générés par Annotator+ conservent une taille manipulable, nous avons désactivé les options relatives à la négation (`negation`), à la détection du patient impliqué dans une expression médicale (`experiencer`), à la temporalité (`temporality`), à la hiérarchie des concepts identifiés (`display_links`) et aux informations sur les vocabulaires requêtés (`display_context`). Nous avons activé l'option `longest_only`, ainsi que l'option de lemmatisation (`lemmatize`) pour améliorer les capacités de détection. MongoDB et Morph-xR2RML ont été déployés sur une autre machine équipée de 8 cœurs et de 48 Go de RAM.

Extraction du graphe d'arguments. Seuls les résumés de plus de dix mots ont été traités par ACTA pour garantir des résultats significatifs. Au total, 44 153 documents ont répondu à ce critère. ACTA a été déployé sur un nœud dual-Xeon de 2,8 GHz avec 96 Go de RAM.

Comme pour l'extraction des entités nommées, les documents JSON en sortie ont été chargés dans MongoDB et traduits dans la représentation RDF décrite dans la section 2.2 en utilisant Morph-xR2RML. La traduction en RDF a été effectuée sur la même machine que celle décrite ci-dessus (celle hébergeant MongoDB et Morph-xR2RML).

3 Publication et interrogation du jeu de données Covid-on-the-Web

Le jeu de données Covid-on-the-Web est composé de deux principaux graphes RDF, à savoir le graphe de connaissances des entités nommées CORD-19 et le graphe de connaissances d'argumentation CORD-19. Un troisième graphe décrit les métadonnées et le contenu des articles CORD-19. Le Tableau 1 synthétise la quantité de données en termes de documents JSON et de triples RDF produits.

Description du jeu de données. Conformément aux meilleures pratiques en matière de publication de données [7], nous fournissons une description détaillée du jeu de données Covid-on-the-Web lui-même. Celle-ci comprend notamment (1) des informations relatives aux licences, aux contributeurs et la provenance décrites avec DCAT, [24] et (2) aux vocabulaires, aux liens entre jeux de données et les informations pour accéder aux données avec VOID. [25]

Accessibilité des données. Le jeu de données RDF est identifié via un DOI, téléchargeable depuis la plateforme Zenodo et accessible au moyen d'un endpoint SPARQL public. Chaque URI peut être déférencée avec négociation de contenu.

Le dépôt Github du projet fournit une documentation exhaustive incluant des détails relatifs aux licences, aux représentations RDF, aux graphes nommés et aux ontologies chargées dans l'endpoint. Ces informations sont résumées dans le Tableau 2.

21. `http://rdfs.org/sioc/argument#`
22. `http://www.arg.dundee.ac.uk/aif#`
23. `https://github.com/frmichel/morph-xr2rml/`
24. `https://www.w3.org/TR/vocab-dcat/`
25. `https://www.w3.org/TR/void/`

TABLE 1 – Volume de données de Covid-on-the-Web.

Type de données	Données JSON	Ressources produites	Triplets RDF
Métadonnées sur les articles et le contenu	15 GB	n.a.	3.72 M
Graphe de Connaissances des Entités Nommées CORD-19 (CORD19-NEKG)			
ENs identifiées par DBpedia Spotlight (titres, résumés)	87 GB	4.1 M	65.4 M
ENs identifiées par Entity-fishing (titres, résumés, corps)	52 GB	66.1 M	1.16 G
ENs identifiées par BioPortal Annotator (titres, résumés)	378 GB	43 M	104.4 M
Graphe de Connaissances d'Argumentation CORD-19 (CORD19-AKG)			
Composants preuves / affirmations (résumés)	112 MB	119 k	7.47 M
Éléments PICO		515 k	
Données totales pour Covid-on-the-Web (en incluant les métadonnées sur les articles et le contenu)			
	532 GB	113 M entités nommées 119 k preuves/affirmations 515 k éléments PICO	1.36 G

TABLE 2 – Accessibilité de Covid-on-the-Web.

CovidOnTheWeb DOI	10.5281/zenodo.4247134
Données RDF	`https://doi.org/10.5281/zenodo.4247134`
Endpoint SPARQL	`https://covidontheweb.inria.fr/sparql`
Documentation	`https://github.com/Wimmics/CovidOnTheWeb`
Espace de nommage	`http://ns.inria.fr/covid19/`
CovidOnTheWeb URI	`http://ns.inria.fr/covid19/covidontheweb-1-2`

Reproductibilité. Conformément aux principes de la science ouverte, tous les scripts, fichiers de configuration et de traduction en RDF impliqués dans notre pipeline sont fournis dans le dépôt Github du projet selon les termes de la licence Apache 2.0, de sorte que n'importe qui peut relancer l'ensemble de la chaîne de traitement (de l'extraction des articles au chargement des fichiers RDF dans Virtuoso OS).

Licences. Les données produites pour Covid-on-the-Web sont dérivées du corpus CORD-19, et en tant que telles différentes licences s'appliquent à ces dernières. Les métadonnées sur les articles ainsi que le contenu des articles traduits en RDF de CORD-19 sont publiés sous les mêmes termes que la licence de CORD-19. [26]

Les résultats de l'extraction des articles, qu'il s'agisse des ENs (CORD19-NEKG) ou des composants argumentatifs (CORD19-AKG), sont publiés selon les termes de la licence d'attribution Open Data Commons 1.0 (ODC-By). [27]

Maintenance. Chaque semaine de nouvelles recherches sont publiées au sujet du Covid-19. La valeur de Covid-on-the-Web, ainsi que des autres jeux de données s'attaquant à cette problématique, réside dans leur habilité à pouvoir intégrer ces nouveaux résultats au fur et à mesure de leur publication. À cette fin, nous avons pris soin de produire un pipeline documenté et reproductible, et nous avons déjà effectué une telle mise à jour, validant ainsi notre démarche. À moyen terme, nous avons l'intention d'améliorer la fréquence des mises à jour en considérant à la fois *(1)* l'importance des mises à jour de CORD-19 (nombre de nouveaux articles), et *(2)* les besoins définis par l'expression de nouveaux scénarios d'application (voir Section 5). En outre, nous avons déployé un serveur permettant d'héberger un endpoint SPARQL qui bénéficie d'une infrastructure à haute disponibilité et d'un support 24 heures sur 24, 7 jours sur 7.

26. `https://ai2-semanticscholar-cord-19.s3-us-west-2.amazonaws.com/2020-03-13/COVID.DATA.LIC.AGMT.pdf`

27. `http://opendatacommons.org/licenses/by/1.0/`

4 Visualisation et utilisations du jeu de données

Notre projet s'est également attaché à explorer les moyens de visualiser les données et d'interagir avec elles. Nous avons pour cela développé un outil appelé *Covid Linked Data Visualizer* [28] qui comprend une interface web hébergée sur un serveur node.js, un moteur de transformation basé sur Corese Semantic Web factory [4], et la bibliothèque graphique MGExplorer [3].

Par le biais de l'interface web, les utilisateurs peuvent utiliser des requêtes SPARQL prédéfinies ou écrire leurs propres requêtes. Des formulaires HTML servent à spécifier certains critères de recherche tels que la date de publication des articles. Par la suite, le moteur de transformation convertit les résultats des requêtes SPARQL dans le format JSON attendu par la bibliothèque graphique. L'exploration du graphe résultant est possible grâce à MGExplorer, une bibliothèque qui englobe un ensemble de techniques de visualisation spécialisées, chacune d'entre elles permettant de se concentrer sur un type de relation particulier.

La Figure 2 illustre certaines de ces techniques : le diagramme de graphe (gauche) montre une vue d'ensemble des nœuds et de leurs relations ; ClusterVis (en haut à droite) est une vue basée sur les clusters, qui permet de comparer les attributs des nœuds tout en préservant la représentation des relations entre eux ; IRIS (en bas à droite) est une vue égocentrique qui permet d'afficher tous les attributs et les relations d'un nœud donné. L'originalité de ces techniques de visualisation est d'offrir aux utilisateurs différents modes d'interaction qui peuvent les aider à explorer, classer et analyser l'importance des publications.

28. `http://covid19.i3s.unice.fr:8080`

Lors d'une réunion avec des organismes liés au domaine de la santé et de la recherche médicale (Inserm et INCa), un expert nous a indiqué un exemple de requête que les chercheurs aimeraient faire sur un tel jeu de données : "Identifier les articles qui mentionnent à la fois un type de cancer et un virus de la famille des coronavirus". En prenant en considération cette requête, nous avons utilisé Covid Linked Data Visualizer et affiché les résultats à l'aide de la bibliothèque MGExplorer (Figure 2).

Nous avons également créé plusieurs Notebooks Jupyter, Python et R [29] pour montrer que ces résultats peuvent être convertis en Dataframes (des structures de données tabulaires utilisées en analyse des données) afin de procéder à de la fouille de données (Figure 3).

5 Impact potentiel et exploitation

À notre connaissance, le jeu de données Covid-on-the-Web est le premier à intégrer dans un seul et même ensemble cohérent des ENs, arguments et éléments PICO . Nous pensons qu'il pourra servir de base pour des applications du Web sémantique, pour des algorithmes d'analyse comparative ou pour des défis.

Les ressources et les services que nous proposons, liés à la littérature concernant la COVID-19, sont intéressants pour les organismes et les instituts de santé puisqu'ils permettent d'extraire et d'analyser efficacement les informations sur une maladie encore relativement inconnue et pour laquelle la recherche est en constante évolution. Dans une certaine mesure, il est possible de croiser les connaissances pour mieux appréhender ce sujet et, en particulier, pour initier des recherches sur des voies inexplorées. Nous espérons également que l'ouverture des données et du code permettra aux contributeurs de faire progresser l'état actuel des connaissances sur cette maladie dont l'impact sanitaire est mondial.

En plus d'être interopérables avec les graphes de connaissances majeurs utilisés au sein de la communauté du Web Sémantique, les visualisations que nous offrons au moyen de MGExplorer et de Notebooks Jupyter montrent le potentiel de ces technologies dans d'autres domaines, à titre d'exemple, dans les domaines biomédicaux et médicaux.

Documentation / Tutoriels. Nous avons conservé les documents méthodologiques que nous avons suivis afin de pouvoir justifier nos choix de conception : La documentation technique des algorithmes et des représentations RDF, [30] les meilleures pratiques dans l'élaboration et publication des données (FAIR, Cool URIs, données liées à cinq étoiles, etc.) et des documentations destinées aux utilisateurs finaux (par exemple, les Notebooks Jupyter).

Scénarios, modèles d'utilisateurs et requêtes types. Nos ressources sont basées sur des outils génériques que nous avons adaptés à la problématique de la COVID-19. En adoptant une approche orientée utilisateur, nous les avons conçues selon trois principaux scénarios identifiés par une analyse des besoins des instituts biomédicaux avec lesquelles nous collaborons : *(Scénario 1)* Aider les cliniciens à obtenir des graphes argumentatifs pour analyser les essais cliniques et prendre des décisions fondées sur des données ; *(Scénario 2)* Aider les médecins en milieu hospitalier à collecter les valeurs biologiques (par exemple, le cholestérol) à partir d'articles scientifiques, afin de déterminer si leurs patients sont dans les normes ou non ; *(Scénario 3)* Aider les chefs de mission d'un institut du cancer à identifier les articles scientifiques traitant du cancer et des coronavirus afin d'élaborer des programmes de recherche pour étudier les liens entre eux.

La généricité des outils que nous avons développés nous permet de les appliquer à un panel plus large de scénarios, et nos partenaires dans le domaine biomédical nous incitent déjà à réfléchir à des scénarios liés à d'autres questions que la COVID-19.

Outre les scénarios décrits ci-dessus, nous établissons également des modèles d'utilisateurs représentatifs (sous la forme de personas), dont le but est de nous aider à identifier les besoins, l'expérience, les comportements et les objectifs de nos utilisateurs.

Nous avons également reçu diverses demandes des utilisateurs potentiels que nous avons interrogés. Ces demandes servent à préciser et à tester notre graphe de connaissances et nos services. À des fins de généricité, nous avons élaboré une typologie à partir des demandes collectées, en utilisant des dimensions telles que : demande prospective vs. rétrospective ou demande descriptive (demande de description) vs. explicatives (demande d'explication) vs. argumentatives (demande d'argumentation). Voici des exemples de ces requêtes :

(Demandes descriptives prospectives) Quels types de cancers sont susceptibles d'apparaître chez les victimes de la COVID-19 au cours des prochaines années ? Chez quelles catégories de patients ? Etc.

(Demandes rétrospectives descriptives) Quels types de cancers sont apparus chez les victimes de [SARSCoV1 | MERS-CoV] au cours des [2|3|n] années suivantes ? Quel était le taux d'occurrence ? Chez quels types de patients ? Etc. Quelles sont les différentes séquelles liées aux coronavirus ? Quels sont les patients guéris de la COVID-19 qui ont une fibrose pulmonaire ?

(Demandes rétrospectives explicatives) Le [SARS-CoV1 | MERS-CoV] peut-il induire le cancer ? La [malignité | progression du cancer] est-elle directement induite par une infection au coronavirus ? Ou était-elle indirectement causée par les [inflammations | modifications métaboliques] liées à une infection ? Quelles séquelles liées aux coronavirus sont responsables du plus fort potentiel de malignité ?

(Demande rétrospective argumentatives) Quelles sont les preuves que le [SARS-CoV1 | MERS-CoV] provoque le cancer ? Quelles expériences ont démontré que la fibrose pulmonaire observée chez les patients guéris de la COVID-19 était causée par la COVID-19 ?

Ces requêtes sont une brève illustration d'une liste réelle (mais non exhaustive) de questions avancées par les utilisateurs. Certaines questions peuvent trouver une réponse en

29. https://github.com/Wimmics/covidontheweb/tree/master/notebooks

30. https://github.com/Wimmics/covidontheweb/blob/master/doc/01-data-modeling.md

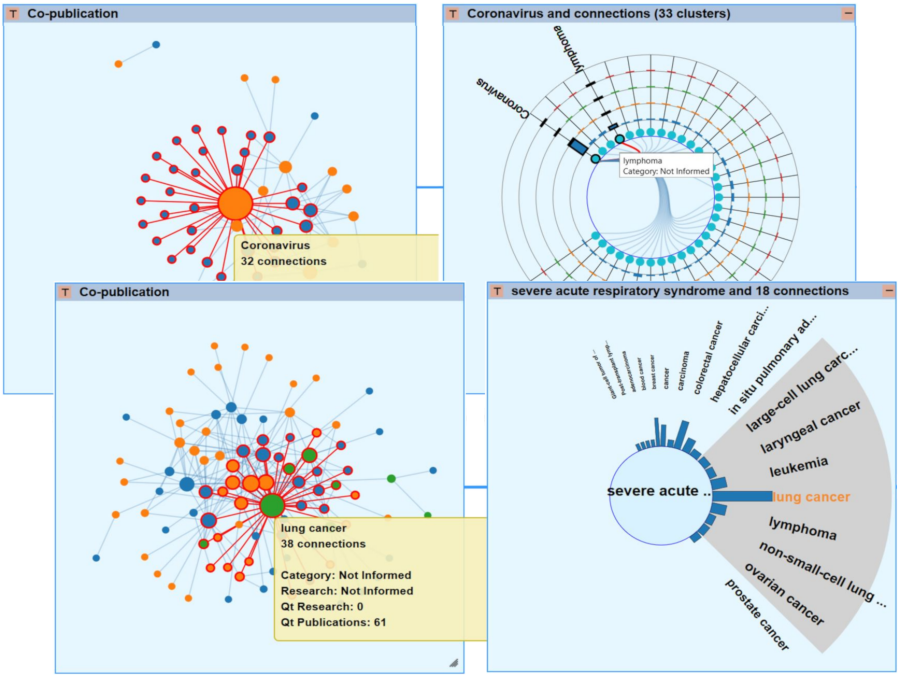

FIGURE 2 – Covid Linked Data Visualizer : visualisation des articles qui mentionnent à la fois un type de cancer (points bleus) et un virus de la famille des coronavirus (points orange).

montrant la corrélation entre les composants (par exemple, les types de cancer), d'autres nécessitent de représenter des tendances (par exemple, le cancer susceptible de se produire au cours des prochaines années) et l'analyse d'attributs spécifiques (par exemple, des détails sur les changements métaboliques causés par la COVID-19). La réponse à ces questions complexes requiert l'exploration du corpus CORD-19, et pour cela nous offrons une variété d'outils d'analyse et de visualisation. Ces requêtes et la typologie générique seront réutilisées dans d'autres extensions ainsi que d'autres projets.

Le Covid Linked Data Visualizer (présenté dans la Section 4) permet l'exploration visuelle du jeu de données Covid-on-the-Web. Les utilisateurs peuvent inspecter les éléments du graphe généré par une requête SPARQL (en positionnant la souris sur les éléments) ou explorer le graphe de façon itérative en chaînant les visualisations et en utilisant l'une des techniques d'interaction disponibles (que ce soit par IRIS, ClusterVis, etc.). Ces techniques de visualisation sont destinées à aider les utilisateurs à comprendre les relations présentes au sein des résultats. Par exemple, les utilisateurs peuvent lancer une requête pour visualiser un réseau de co-auteurs ; puis se servir de IRIS et ClusterVis pour comprendre qui collabore ensemble et sur quelles thématiques. Ils peuvent également lancer une recherche pour trouver des articles mentionnant la COVID-19 et divers types de cancer. Enfin, le mode avancé permet d'ajouter de nouvelles requêtes SPARQL mettant en œuvre d'autres chaînes d'exploration de données.

6 Conclusion et perspectives

Nous avons décrit dans cet article les données et logiciels déployés par le projet Covid-on-the-Web. Nous avons adapté et combiné des outils pour traiter, analyser et enrichir le corpus CORD-19 afin de permettre aux chercheurs dans le domaine biomédical d'accéder plus aisément à la littérature relative à la COVID-19, de l'interroger et de lui donner sens.

Nous avons conçu et publié un graphe de connaissances des données liées décrivant les entités nommées mentionnées dans les articles de CORD-19 et les graphes d'argumentation qu'ils incluent. Nous avons également publié le pipeline mis en place pour générer ce graphe de connaissances, afin de *(1)* continuer à l'enrichir et de *(2)* faciliter la réutilisation et l'adaptation du jeu de données et du pipeline.

Au-delà de ce graphe de connaissances, nous avons également développé, adapté et déployé plusieurs outils fournissant des visualisations de données liées, des méthodes d'exploration et des Notebooks pour les spécialistes dans les sciences des données. Par nos interactions avec des instituts dans le domaine de la santé et de la recherche médicale (entretiens, observations, tests d'utilisateurs), nous continuons de nous assurer que notre approche est guidée par et alignée sur les besoins de la communauté biomédicale. Nous avons montré que notre jeu de données permet d'effectuer des recherches documentaires et fournir des visualisations adaptées aux besoins des experts. De plus, notre démarche, dès ses prémices, s'est attachée à répondre aux objectifs de la science ouverte et reproductible ainsi qu'aux principes FAIR.

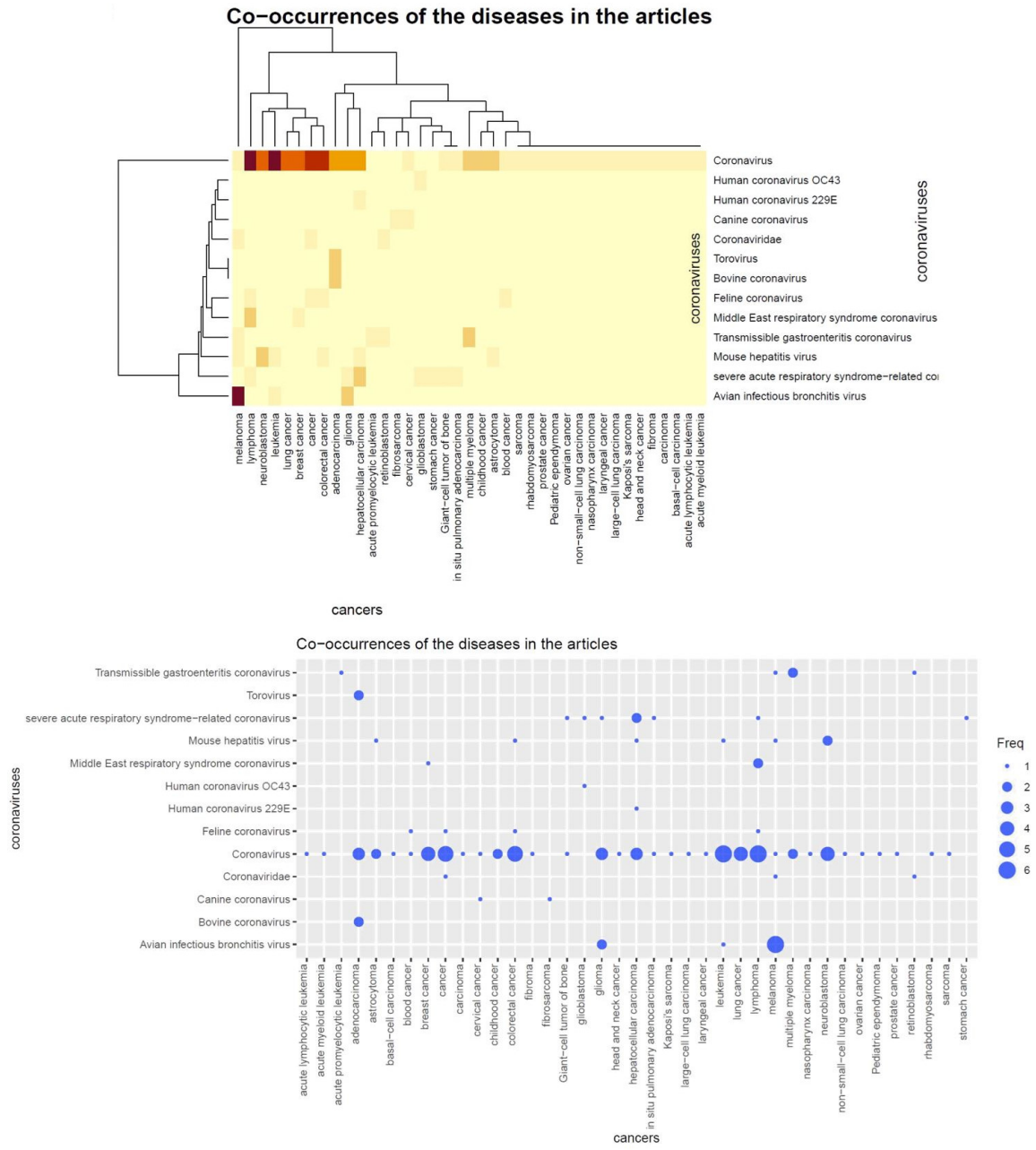

FIGURE 3 – Deux représentations différentes sous Jupyter Notebook du nombre d'articles qui co-mentionnent les types de cancer et les virus de la famille des coronavirus.

Depuis l'émergence de la COVID-19, le rythme effréné auquel les nouvelles recherches ont été publiées et les bases de connaissances ont évolué pose des problèmes critiques. Par exemple, de nouvelles versions de CORD-19 étaient publiées chaque semaine (désormais ce rythme peut être journalier), ce qui remet en question la capacité à suivre les dernières avancées. Par ailleurs, l'extraction et la désambiguïsation des ENs sur notre première version du jeu de données avaient été réalisées à l'aide de modèles préentraînés produits avant la pandémie, donc avant même la création de l'entité SARS-Cov-2 dans Wikidata. Par conséquent, à moyen terme, nous avons l'intention de nous engager dans un objectif de maintenance pérenne visant à ingérer régulièrement de nouvelles données, suivre l'évolution des connaissances et mettre régulièrement à jour nos extracteurs. Étant donné qu'il n'existe pas de jeu de données de référence de CORD-19 qui aurait été manuellement annoté et qui pourrait donc servir de référence (gold standard), il est difficile d'évaluer la qualité des modèles utilisés pour extraire les ENs et les structures argumentatives. Pour pallier ce problème, nous travaillons sur la mise en œuvre de curation de contenu (data curation), et la découverte auto-

matisée de motifs et de règles d'association qui pourraient être utilisés pour détecter les erreurs dans l'extraction des ENs, permettant ainsi de proposer un contrôle qualité des données.

Références

[1] I. Beltagy, K. Lo, and A. Cohan. SciBERT : A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, 2019.

[2] M. Bersanelli. Controversies about COVID-19 and anticancer treatment with immune checkpoint inhibitors. *Immunotherapy*, 12(5) :269–273, April 2020.

[3] R. Cava, C. Freitas, and M. Winckler. Clustervis : visualizing nodes attributes in multivariate graphs. In *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, pages 174–179. ACM, 2017.

[4] O. Corby, R. Dieng-Kuntz, and C. Faron-Zucker. Querying the semantic web with Corese search engine. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, volume 16, page 705, Valencia, Spain, 2004.

[5] J. Daiber, M. Jakob, C. Hokamp, and P. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 121–124, 2013.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 4171–4186, 2019.

[7] B. Farias Lóscio, C. Burle, and N. Calegari. Data on the Web Best Practices. *W3C Recommandation*, 2017.

[8] R. Gazzotti, C. Faron-Zucker, F. Gandon, V. Lacroix-Hugues, and D. Darmon. Injecting domain knowledge in electronic medical records to improve hospitalization prediction. In *The Semantic Web - 16th European Conference, ESWC, Portorož, Slovenia, June 2-6, 2019, Proceedings*, volume 11503 of *Lecture Notes in Computer Science*, pages 116–130. Springer, 2019.

[9] R. Gazzotti, C. Faron-Zucker, F. Gandon, V. Lacroix-Hugues, and D. Darmon. Injection of automatically selected DBpedia subjects in electronic medical records to boost hospitalization prediction. In *SAC '20 : The 35th ACM/SIGAPP Symposium on Applied Computing, online event, March 30 - April 3, 2020*, pages 2013–2020. ACM, 2020.

[10] C. Jonquet, N. Shah, and M. Musen. The open biomedical annotator. *Summit on translational bioinformatics*, 2009 :56, 2009.

[11] T. Mayer, E. Cabrio, and S. Villata. ACTA a tool for argumentative clinical trial analysis. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6551–6553, 2019.

[12] T. Mayer, E. Cabrio, and S. Villata. Transformer-based argument mining for healthcare applications. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, 2020.

[13] F. Michel, L. Djimenou, C. Faron-Zucker, and J. Montagnat. Translation of Relational and Non-Relational Databases into RDF with xR2RML. In *Proceeding of the 11th International Conference on Web Information Systems and Technologies (WebIST)*, pages 443–454, Lisbon, Portugal, 2015.

[14] F. Michel, F. Gandon, V. Ah-Kane, A. Bobasheva, E. Cabrio, O. Corby, R. Gazzotti, A. Giboin, S. Marro, T. Mayer, M. Simon, S. Villata, and M. Winckler. Covid-on-the-web : Knowledge graph and services to advance COVID-19 research. In *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, volume 12507 of *Lecture Notes in Computer Science*, pages 294–310. Springer, 2020.

[15] M. Neumann, D. King, I. Beltagy, and W. Ammar. ScispaCy : Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August 2019. Association for Computational Linguistics.

[16] B. Nye, J. Li, R. Patel, Y. Yang, I. Marshall, A. Nenkova, and B. Wallace. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proc. 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 197–207, 2018.

[17] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv :1402.1128*, 2014.

[18] L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Eide, K. Funk, R. Kinney, Z. Liu, W. Merrill, P. Mooney, D. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. Stilson, A. Wade, K. Wang, C. Wilhelm, B. Xie, D. Raymond, D. Weld, O. Etzioni, and S. Kohlmeier. Cord-19 : The covid-19 open research dataset. *ArXiv*, abs/2004.10706, 2020.

Assister l'édition manuelle de données RDF à l'aide du raisonnement à partir de cas

N. Lasolle[1,2], O. Bruneau[1], J. Lieber[2], E. Nauer[2] et S. Pavlova[2]

[1] Université de Lorraine, CNRS, Université de Strasbourg, AHP-PReST, F-54000 Nancy, France
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

[1] prénom.nom@univ-lorraine.fr
[2] prénom.nom@loria.fr

Résumé

Les technologies du Web sémantique fournissent des outils pour structurer, exploiter et enrichir des corpus historiques tels que le corpus de la correspondance d'Henri Poincaré. Cependant, le processus d'édition de données RDF *est un processus qui est souvent manuel et peut sembler fastidieux pour les contributeurs. Cet article introduit un système de suggestions qui s'appuie sur le raisonnement à partir de cas pour assister l'édition de données* RDF*. Ce système se décline en quatre versions qui sont comparées au travers d'une double évaluation.*

Mots-clés

Web sémantique, raisonnement à partir de cas, édition de données RDF, humanités numériques, transformation de requêtes SPARQL, corpus historique.

Abstract

Semantic Web technologies provide a way to structure, to exploit and to enhance historical corpora data such as the Henri Poincaré correspondence corpus. However, RDF *data editing is often manual and may seem tedious for the concerned user. This article introduces a suggestion system which uses case-based reasoning to assist manual* RDF *data editing. This system is available in four versions which are compared to each other through a double evaluation.*

Keywords

Semantic Web, case-based reasoning, RDF data editing, Digital Humanities, SPARQL query transformation, historical corpus.

1 Introduction

Dans le contexte des bases RDF, le processus d'édition des données nécessite fréquemment une intervention humaine. Cette tâche peut rapidement s'avérer fastidieuse pour les personnes concernées avec un risque d'erreur important. Ce constat a notamment été porté lors de l'édition du corpus de la correspondance d'Henri Poincaré. Ce corpus historique rassemble des échanges d'ordre scientifique, administratif et privé qui forment une source d'information importante pour les historiens. Le cœur du corpus est constitué d'un ensemble d'environ 2100 lettres envoyées ou reçues par Henri Poincaré qui sont liées à des documents divers (article, rapport, compte-rendu, etc.), des personnalités (de France et d'ailleurs), des lieux et institutions, etc. Des technologies du Web sémantique (RDF, RDFS et SPARQL) ont été utilisées pour exploiter et enrichir ce corpus historique. Les lettres sont accessibles sur le site `http://henripoincare.fr`[1]. Chaque lettre comporte une numérisation du document original[2], une transcription, un apparat critique ainsi qu'un ensemble de méta-données. Ces dernières permettent à la fois la description physique de la lettre (expéditeur, destinataire, date de rédaction, etc.) et de son contenu (thèmes scientifiques abordés, personnes et institutions citées, etc.). Ce site a été créé grâce au système de gestion de contenus $Omeka\ S$ [4] qui permet à des institutions (musées, archives, etc.) d'éditer, de publier et de rendre accessibles des corpus.

Lors de l'édition du corpus de la correspondance d'Henri Poincaré, plusieurs types d'erreurs ont été identifiés. *L'erreur de duplication* se produit lorsque qu'un utilisateur insère des données qui existent déjà dans la base. *L'erreur d'ambiguïté* se produit lorsque qu'un utilisateur ne dispose pas de suffisamment d'informations pour distinguer des éléments. Par exemple, si une recherche est effectuée sur la base de la chaîne "Henri Poincaré", différents types de ressources peuvent être retournés. En effet, la réponse attendue la plus plausible devrait se référer au célèbre scientifique, mais ce terme se réfère également à différents instituts et écoles et, depuis 1997, un prix de physique mathématique a été créé en sa mémoire. *L'erreur de frappe* se produit lorsqu'un utilisateur souhaite écrire un mot existant pour désigner une ressource spécifique, mais qu'il commet une erreur lors de la saisie de l'identifiant. Si elle n'est pas remarquée, une erreur de ce type peut conduire à la création d'une nouvelle ressource dans la base au lieu de faire référence à une ressource existante. Outre ces possibles erreurs, la charge cognitive associée à l'utilisation d'un système d'annotation ne doit pas être négligée. Selon le vo-

1. Sur ce site, d'autres éléments sont disponibles : les travaux de Poincaré, une iconographie et une bibliographie de ce mathématicien.
2. Certaines numérisations ne sont pas disponibles pour des raisons en lien avec les droits d'auteur.

lume du corpus à annoter, ce processus pourrait être un projet à long terme (plusieurs années). Maintenir la motivation des contributeurs lors de l'exécution des tâches associées est donc primordial.

Cet article présente un outil pour assister l'édition manuelle de données RDF. Quelques notions préliminaires sur le Web sémantique sont données (section 2). Ensuite, le système de suggestions, décliné en quatre versions est présenté (section 3). L'outil s'appuie notamment sur un mécanisme de raisonnement à partir de cas qui s'intéresse aux liens entre la ressource en cours d'édition et des ressources déjà éditées dans la base de connaissances. Une double évaluation a été réalisée pour mesurer la pertinence des suggestions proposées au regard de cas d'éditions réels (section 4). Cet article présente également des travaux liés autour de l'édition de données RDF et des systèmes de recommandation (section 5). Une conclusion et une présentation de perspectives de recherche sont données en section 6. Ce travail de recherche a été préalablement présenté lors de la conférence internationale du Web sémantique (ISWC 2020) [9].

2 Préliminaires sur le Web sémantique

Cette section présente les technologies du Web sémantique utilisées pour exploiter les données de ce corpus : le modèle RDF, le langage de représentation de connaissances RDFS et le langage d'interrogation de graphes SPARQL.

2.1 Le modèle RDF

Resource Description Framework (RDF [10]) est un modèle de représentation de données fondé sur l'utilisation de graphes orientés et étiquetés. C'est un standard développé et maintenu par le *World Wide Web Consortium* (W3C) et qui est fortement utilisé par la communauté du Web sémantique. Un graphe RDF est composé de trois types de nœuds : des *ressources nommées*, des *ressources anonymes* et des *littéraux*. Une *ressource nommée* est identifiée par un *Internationalized Resource Identifier* (IRI) et permet de décrire une classe (p. ex. `Personne`, `Mathématicien`, `Lettre`, etc.), une propriété (p. ex. `expéditeur`, `destinataire`, etc.) ou une instance (p. ex. `henriPoincaré`, `lettre11`, etc.) [3]. Une *ressource anonyme* représente une ressource qui n'est pas explicitement identifiée (*nœud vide*). Une telle ressource est désignée par l'utilisation d'un point d'interrogation préfixant un nom de variable (p. ex. `?x`, `?personne`, etc.). Un littéral correspond à une valeur constante d'un type donné (entier, chaîne de caractères, date, etc.).

Il est possible de définir des relations entre les nœuds du graphe par l'utilisation de propriétés décrivant les ressources les composant. Ces relations sont caractérisées par des triplets de la forme $\langle sujet\ \ prédicat\ \ objet \rangle$. Le sujet représente la ressource (nommée ou anonyme) à décrire. Le prédicat est une propriété qui décrit cette ressource. L'objet est la valeur associée à la propriété et peut être une ressource nommée, une ressource anonyme ou un littéral.

2.2 RDF Schema

RDF Schema (RDFS [5]) est un langage de représentation de connaissances qui étend le modèle RDF. Plusieurs propriétés sont utilisées pour structurer les ressources : `rdfs:subclassof` (resp. `rdfs:subpropertyof`) permet de créer une hiérarchie entre des classes (resp. propriétés). Par exemple, le triplet $\langle$`Lettre rdfs:subclassof Document`$\rangle$ indique que le concept de `Lettre` est plus spécifique que le concept de `Document`. `rdfs:domain` (resp. `rdfs:range`) s'applique à une propriété et permet d'ajouter une contrainte à propos du type de la ressource se trouvant en position de $sujet$ (resp. $objet$) au sein d'un triplet. Un graphe RDF constitue une base de données à partir de laquelle de nouvelles données peuvent être engendrées. La déduction RDFS désigne l'application de règles d'inférence RDFS afin d'inférer de nouvelles données. Ce mécanisme est utilisé pour mener des raisonnements au sein des graphes RDF.

2.3 SPARQL

SPARQL est le langage recommandé par le W3C pour interroger des graphes RDF [12]. Une forme courante de requêtes SPARQL est constituée d'une clause SELECT contenant une ou plusieurs variables suivie par une clause WHERE composée d'un *patron de graphe* divisé en un ou plusieurs *patrons de triplet* et d'une possible clause FILTER qui permet d'ajouter des contraintes pour les valeurs littérales associées aux propriétés. Considérons la requête informelle suivante :

$\mathcal{Q} =$ « Donner les lettres rédigées par Henri Poincaré entre 1885 et 1890 et ayant pour thème la géométrie »

Cette requête peut s'exprimer de la façon suivante en utilisant le langage SPARQL :

$\mathtt{Q} =$

```
SELECT ?l
WHERE
   {
     ?l type Lettre .
     ?l expéditeur henriPoincaré .
     ?l thème géométrie .
     ?l dateDeRédaction ?y .
     FILTER(YEAR(?y) >= 1885
        AND YEAR(?y) <= 1890)
   }
```

3 Un système de suggestions pour assister l'édition de données RDF

Comme décrit dans l'introduction de ce chapitre, le processus d'édition est un travail fastidieux qui justifie la nécessité de créer un éditeur dédié pour assister l'utilisateur. Ce système devrait permettre une mise à jour interactive efficace d'une base RDFS, en visualisant les faits déjà édités et en fournissant des suggestions adaptées au contexte d'édition.

3. Dans un souci de lisibilité, les ressources nommées ne sont pas représentées en utilisant des IRI complets dans ce document.

Pour répondre à cette problématique, quatre versions d'un moteur de suggestions ont été mises en place :

Le système basique assiste l'utilisateur en proposant un mécanisme d'autocomplétion dans laquelle les suggestions sont classées par ordre alphabétique. Les suggestions proposées ne dépendent ni du problème d'édition courant ni des connaissances définies dans l'ontologie.

Le système déductif bénéficie de l'utilisation des connaissances de l'ontologie pour le classement des suggestions fournies à l'utilisateur.

Le système à base de cas s'appuie sur des problèmes d'édition similaires au problème courant.

Le système combiné combine les deux précédentes méthodes.

3.1 Le système déductif

La notion de *question d'annotation* est ici introduite : cela correspond à un triplet pour lequel 1, 2 ou les 3 éléments sont inconnus, et pour lequel un champ est en cours d'édition. Ce champ est représenté en utilisant un cadre autour d'une variable existentielle (c'est-à-dire $\boxed{\texttt{?p}}$, $\boxed{\texttt{?o}}$). Par exemple, $\langle s \ \boxed{\texttt{?p}} \ \texttt{?o} \rangle$ correspond à un type de question d'annotation pour lequel le sujet est connu, le prédicat est actuellement en cours d'édition et l'objet est inconnu. Il existe douze types de questions d'annotation différents (voir [7] pour plus de détails). Pour chacun d'entre eux, les connaissances liées aux domaines et co-domaines des propriétés peuvent être utilisées pour classer les valeurs candidates pour le champ cible.

Soit la question d'annotation $\langle \texttt{lettre11 expéditeur } \boxed{\texttt{?o}} \rangle$ qui est du type $\langle s \ p \ \boxed{\texttt{?o}} \rangle$. L'objectif est ici de fournir des suggestions appropriées en classant des valeurs potentielles pour l'objet. Pour cette version du système de suggestions, les premières suggestions sont les ressources des classes `Personne` et `Institution` car ces classes font partie du co-domaine de la propriété `expéditeur`. Cette connaissance est donc utilisée pour favoriser les instances de ces classes. Cependant, les ressources qui n'appartiennent pas explicitement à ces classes sont toujours proposées parce que RDFS fonctionne selon l'hypothèse du monde ouvert [4].

Il existe différentes règles qui peuvent être utilisées pour classer les valeurs potentielles et dont les applications dépendent du type de la question d'annotation. Pour répondre à une question d'annotation, un score est calculé pour chaque valeur candidate `?v` en fonction du nombre de règles qui ont récupéré cette valeur. La liste finale des suggestions est classée en fonction de ce score par ordre décroissant. Pour les valeurs potentielles ayant le même nombre, l'ordre alphabétique est utilisé.

4. Si un fait n'est pas affirmé, cela ne signifie pas qu'il soit faux. Dans cette situation, il peut exister une ressource r qui est destinée à représenter une personne (resp. une institution) mais est telle que le triplet $\langle r$ `a Personne`$\rangle$ (resp. $\langle r$ `a Institution`$\rangle$) ne peut être inféré par l'état actuel de la base RDFS. Par conséquent, r peut également être suggérée, bien que plus loin dans la liste de suggestions.

3.2 Le système à base de cas

L'utilisation de la déduction RDFS apporte une première amélioration au système de suggestions en utilisant les connaissances liées aux domaines et co-domaines des propriétés définies dans la base. Toutefois, dans certaines situations, cela ne suffit pas pour proposer les ressources les plus appropriées relatives à la question d'annotation courante.

À titre d'exemple, considérons un triplet en cours d'édition pour lequel le sujet est une instance de `Lettre` (`lettre2100`), le prédicat est `destinataire` et pour lequel des suggestions concernant le champ d'objet sont attendues. Étant donné que la classe `Personne` fait partie du co-domaine de la propriété `destinataire`, le système favorisera les instances de cette classe dans la liste de suggestions. Mais le problème est qu'il y a de nombreuses instances de cette classe dans la base [5], et il n'y a aucune garantie que la valeur appropriée sera parmi les premières suggestions de la liste. En effet, pour les valeurs ayant le même score, c'est l'ordre alphabétique qui est utilisé. Une autre façon d'obtenir un classement pertinent de la liste de suggestions est d'utiliser le raisonnement à partir de cas : dans la situation actuelle, des éléments d'information provenant de situations similaires peuvent être réutilisés.

3.2.1 Préliminaires sur le raisonnement à partir de cas

Le raisonnement à partir de cas (RàPC [13]) vise à résoudre des problèmes à l'aide d'une *base de cas* `BC`, c'est-à-dire un ensemble fini de cas, où un cas représente un problème passé avec une solution associée. Un cas est souvent défini comme un couple ordonné $(\texttt{x}, \texttt{y})$ où `x` est un problème et `y` est une solution de `x`. Un cas $(\texttt{x}^s, \texttt{y}^s)$ de la base de cas est appelé *cas source*, avec $\texttt{x}^s$ représentant un problème source et $\texttt{y}^s$ la solution de $\texttt{x}^s$. L'entrée d'un système de raisonnement à partir de cas est un problème appelé le *problème cible* et désigné par $\texttt{x}^{\texttt{cible}}$.

Le processus de RàPC peut être décomposé en quatre étapes [1]. (1) Un cas $(\texttt{x}^s, \texttt{y}^s) \in \texttt{BC}$ jugé similaire à $\texttt{x}^{\texttt{cible}}$ est sélectionné (*remémoration*). (2) Ce cas $(\texttt{x}^s, \texttt{y}^s)$ est utilisé pour résoudre le problème $\texttt{x}^{\texttt{cible}}$ (*réutilisation*). La solution proposée $\texttt{y}^{\texttt{cible}}$ peut être égale à $\texttt{y}^s$ (réutilisée telle que) ou adaptée pour tenir compte de différence entre $\texttt{x}^s$ et $\texttt{x}^{\texttt{cible}}$. (3) Le couple $(\texttt{x}^{\texttt{cible}}, \texttt{y}^{\texttt{cible}})$ est évalué pour vérifier que $\texttt{y}^{\texttt{cible}}$ résout correctement $\texttt{x}^{\texttt{cible}}$ et, dans le cas contraire, $\texttt{y}^{\texttt{cible}}$ peut être modifiée en conséquence (*révision*). (4) Enfin, le cas nouvellement formé $(\texttt{x}^{\texttt{cible}}, \texttt{y}^{\texttt{cible}})$ est ajouté à `BC` si cet ajout est jugé approprié (*mémorisation*).

3.2.2 Explication du mécanisme

Dans le contexte de ce système de suggestions, un problème d'annotation $\texttt{x}^{\texttt{cible}}$ est composé d'une question d'annotation et d'un contexte. Pour les questions d'annotation du type $\langle s \ p \ \boxed{\texttt{?o}} \rangle$, il est défini comme suit :

$$\texttt{x}^{\texttt{cible}} = \boxed{\begin{array}{l}\textbf{question :} \ \langle subj^{\texttt{cible}} \ pred^{\texttt{cible}} \ \boxed{\texttt{?o}} \rangle \\ \textbf{contexte :} \ \text{l'ensemble des triplets liés à } subj^{\texttt{cible}}\end{array}}$$

5. Au moment de la rédaction de cet article, il y a environ 1800 personnes définies dans la base de données.

Pour l'exemple lié à la lettre 2100, cela donne $\mathtt{x}^{\mathtt{cible}} =$

question :	⟨lettre2100 destinataire ?o⟩
contexte :	⟨lettre2100 expéditeur henriPoincaré⟩ ⟨lettre2100 thème écolePolytechnique⟩ ⟨lettre2100 cite paulAppell⟩

La base de cas correspond à la base RDF $\mathcal{D}_{\text{HP}}$. Un cas source est donné par un triplet $\langle subj^{\mathtt{s}}\ pred^{\mathtt{s}}\ obj^{\mathtt{s}}\rangle$ de $\mathcal{D}_{\text{HP}}$, considéré parmi tous les triplets de $\mathcal{D}_{\text{HP}}$, et qui, en lien avec $\mathtt{x}^{\mathtt{cible}}$, peut être décomposé en un problème $\mathtt{x}^s$ et une solution $\mathtt{y}^s$:

$$\mathtt{x}^s = \begin{array}{|ll|}\hline \textbf{question :} & \langle subj^{\mathtt{s}}\ pred^{\mathtt{s}}\ \boxed{?o}\rangle \\ \textbf{contexte :} & \text{la base } \mathcal{D}_{\text{HP}} \\ \hline\end{array}$$

et la solution $\mathtt{y}^s = obj^{\mathtt{s}}$. Pour les besoins de cet exemple, considérons un extrait $\mathcal{D}_{ex}$ de la base du corpus de la correspondance Henri Poincaré $\mathcal{D}_{\text{HP}}$ composé des lettres relatives aux instances suivantes de la classe `Personne` : `göstaMittagLeffler`, `alineBoutroux`, `eugénieLaunois`, `felixKlein` et `henriPoincaré`. Comment ordonner la liste composée de ces 5 ressources ? Pour proposer une solution à ce problème, la méthode consiste à récupérer et utiliser les cas qui correspondent le mieux au problème d'annotation actuel. À chaque cas source $\mathtt{x}^s$ est associée une valeur $\mathtt{y}^s$ qui est utilisée comme solution candidate de $\mathtt{x}^{\mathtt{cible}}$. Un score est calculé pour le classement des solutions candidates. Ce score correspond au nombre de lettres similaires ayant cette valeur associée à la propriété `destinataire`. Une requête initiale SPARQL $\mathcal{Q}$ générée à partir de $\mathtt{x}^{\mathtt{cible}}$ est définie pour calculer ce score. Pour l'exemple courant, cela donne :

$\mathcal{Q} =$ "Donner, pour chaque valeur candidate, `?o`,
le nombre de lettres ayant cette valeur associée
à la propriété `destinataire`
où les lettres ont `écolePolytechnique`
comme thème, citent `paulAppell` et
ont été rédigées par `henriPoincaré`"

L'exécution de $\mathcal{Q}$ sur $\mathcal{D}_{ex}$ ne retourne aucun résultat. En effet, il est rare de trouver deux lettres différentes ayant exactement le même contexte. La question est donc de trouver une méthode pour retrouver les cas les plus similaires. Cette question peut être traitée en utilisant un mécanisme de transformation de requêtes SPARQL. Un système a déjà été conçu pour gérer des règles de transformation et s'est révélé utile dans différents contextes, notamment pour la recherche dans le corpus de la correspondance d'Henri Poincaré et dans le système de cuisine *Taaable* [6].

Les règles sont configurées par l'utilisateur et peuvent être générales ou dépendantes d'une application. À chaque règle est associé un coût, correspondant à un coût de transformation de la requête. Deux règles sont considérées dans l'exemple courant :

- $r_{\mathtt{échange}}$: échange de l'expéditeur et du destinataire de la lettre (coût de 2) ;
- $r_{\mathtt{genObj}}$: généralise une instance de classe en position d'objet en remplaçant cette instance par un nœud anonyme du type de cette classe (coût de 3).

Un arbre de recherche peut être parcouru par coût croissant à partir de la requête initiale en appliquant successivement une ou plusieurs règles de transformation. Un coût maximum est défini pour limiter la profondeur d'exploration de l'arbre de recherche. Pour cette application, ce coût maximum est fixé à 10. À la profondeur 1, l'application de la règle $r_{\mathtt{échange}}$ sur $\mathcal{Q}$ génère la requête $\mathcal{Q}_1$ avec un coût de 2 (la partie modifiée de la requête est soulignée) :

$\mathcal{Q}_1 =$ "Donner, pour chaque valeur candidate, `?o`,
le nombre de lettres ayant cette valeur associée
à la propriété `expéditeur`
où les lettres ont `écolePolytechnique`
comme thème, citent `paulAppell` et
ont été reçues par `henriPoincaré`"

Le résultat de l'exécution de $\mathcal{Q}_1$ sur $\mathcal{D}_{ex}$ est : [{`eugénieLaunois` : 2}, {`alineBoutroux` : 1}]. Trois applications de la règle $r_{\mathtt{genObj}}$ existent à la profondeur 1, chacune d'entre elles pour un coût de 3. La première s'applique pour la personne citée, en remplaçant `paulAppell` par toute instance de la classe `Mathématicien` (parce que Paul Appell appartient à cette classe), la deuxième s'applique à l'expéditeur de la lettre, et la dernière s'applique au thème `écolePolytechnique`. Les requêtes générées sont $\mathcal{Q}_2$, $\mathcal{Q}_3$ et $\mathcal{Q}_4$:

$\mathcal{Q}_2 =$ "Donner, pour chaque valeur candidate, `?o`,
le nombre de lettres ayant cette valeur associée
à la propriété `destinataire`
où les lettres ont `écolePolytechnique`
comme thème, citent un `mathématicien` et
ont été rédigées par `henriPoincaré`"

$\mathcal{Q}_3 =$ "Donner, pour chaque valeur candidate, `?o`,
le nombre de lettres ayant cette valeur associée
à la propriété `destinataire`
où les lettres ont `écolePolytechnique`
comme thème, citent `paulAppell` et
ont été rédigées par un `mathématicien`"

$\mathcal{Q}_4 =$ "Donner, pour chaque valeur candidate, `?o`,
le nombre de lettres ayant cette valeur associée
à la propriété `destinataire`
où les lettres ont un thème lié
à `l'éducation`, citent `paulAppell` et
ont été rédigées par `henriPoincaré`"

L'exécution de $\mathcal{Q}_2$ sur $\mathcal{D}_{ex}$ donne : [{`eugénieLaunois` : 138}, {`alineBoutroux` : 4}]. Les exécutions de $\mathcal{Q}_3$ et $\mathcal{Q}_4$ ne retournent aucun résultat. Le coût maximum ayant été fixé à 10, il est possible de continuer l'exploration de l'arbre sur les différentes branches afin de réordonner la liste de suggestions.

À la profondeur 2, l'application de $r_{\mathtt{genObj}}$ sur $\mathcal{Q}_2$ (généralisation du thème) génère la requête :

$\mathcal{Q}_{21} =$ "Donner, pour chaque valeur candidate, `?o`,
le nombre de lettres ayant cette valeur associée
à la propriété `destinataire`
où les lettres ont un thème lié
à `l'éducation`, citent un `mathématicien` et
ont été rédigées par `henriPoincaré`"

L'exécution de $\mathcal{Q}_{21}$ sur $\mathcal{D}_{ex}$ retourne : [{`eugénieLaunois` : 280},

{göstaMittagLeffler : 74}, {alineBoutroux : 17}]. À la profondeur 3, l'application de $r_{\texttt{genObj}}$ sur $\mathcal{Q}_{21}$ (pour l'expéditeur) génère la requête :

$\mathcal{Q}_{211} =$ "Donner, pour chaque valeur candidate, ?o, le nombre de lettres ayant cette valeur associée à la propriété destinataire où les lettres ont un thème lié à l'éducation, citent un mathématicien et ont été rédigées par un mathématicien"

L'exécution de $\mathcal{Q}_{211}$ sur $\mathcal{D}_{ex}$ donne : [{eugénieLaunois : 305}, {henriPoincaré : 219}, {göstaMittagLeffler : 141}, {felixKlein : 25}, {alineBoutroux : 21}]. Les autres applications possibles de la règle (en tenant compte du coût maximum) génèrent des requêtes déjà proposées par d'autres combinaisons ou qui donnent les mêmes ressources mais avec un coût plus élevé. La liste finale des suggestions est classée par le coût de transformation minimal requis. Pour les ressources ayant le même coût minimal, le score lié à l'exécution de la requête associé à ce coût est utilisé (par ordre décroissant). Pour l'exemple courant, cela donne, pour les 5 premières suggestions, du numéro 1 au numéro 5 : eugénieLaunois, alineBoutroux, göstaMittagLeffler, henriPoincaré[6] et felixKlein. Le reste des suggestions est composé de toutes les ressources de la base classées par ordre alphabétique. Cette approche constitue l'étape de remémoration du modèle de RàPC. L'étape de réutilisation est une approche de réutilisation en tant que telle : il n'y a pas d'adaptation des ressources proposées. Après cela, l'utilisateur choisit la ressource appropriée, qui peut être considéré comme une étape de *révision*. Ensuite, le triplet édité est inséré dans la base de connaissances (étape de *mémorisation*).

3.3 Le système combiné

La dernière version du système combine l'utilisation de la déduction RDFS avec le RàPC. Elle tire parti à la fois des connaissances sur les ressources similaires à celle en cours d'édition et des domaines et co-domaines des propriétés utilisées lors de l'édition. Les ressources trouvées en utilisant le RàPC sont en tête de la liste de suggestions. Pour les autres ressources, le calcul du score tel que présenté dans la section 3.1 est appliqué. Considérons l'exemple présenté ci-dessus, dans lequel la question d'annotation était ⟨lettre2100 destinataire ?o⟩. En utilisant le système de raisonnement à partir de cas, les cinq premières suggestions sont des ressources qui semblent pertinentes compte tenu du contexte actuel d'édition et en recherchant des objets similaires dans la base de données. Mais pour le reste des suggestions, seul l'ordre alphabétique est utilisé pour le classement. Pour y remédier, il est possible d'utiliser le co-domaine de la propriété destinataire (comme expliqué dans la section 3.1) pour classer la deuxième partie de la liste de suggestions. Sachant que la classe Personne fait partie du co-domaine de la propriété expéditeur, toutes les instances de cette classe seront plus hautes dans la liste de suggestions que celles des autres classes (p. ex. Article, Revue, Adresse).

6. Cette suggestion pourrait être supprimée si le système sait que le destinataire d'une lettre ne peut être son expéditeur.

4 Évaluation

L'objectif de cette évaluation est de comparer l'efficacité des différentes versions du système pour des situations d'annotation concrètes. La première évaluation est humaine, par l'intermédiaire d'un utilisateur qui teste et compare les quatre versions du système au travers d'un outil Web. Une deuxième évaluation est gérée par un programme dédié et fournit des mesures objectives. Différentes classes existent dans la base de connaissances du corpus de la correspondance d'Henri Poincaré (Lettre, Personne, Article, etc.) mais pour cette évaluation, l'accent est mis sur l'édition des lettres. Les deux évaluations se concentrent sur un sous-ensemble de 7 propriétés parmi les plus fréquemment utilisées lors de l'édition de lettres : expéditeur définit l'expéditeur ; destinataire définit le destinataire ; thème donne l'un des thèmes ; archivéeÀ précise le lieu d'archivage ; aPourRéponse donne une lettre de réponse à la lettre actuelle ; répondÀ donne une lettre à laquelle répond la lettre actuelle ; cite fait référence à une personne mentionnée dans la transcription de la lettre.

4.1 Évaluation humaine

4.1.1 Un outil Web pour éditer les données RDF

Une interface Web a été développée pour utiliser et comparer les différentes versions du système de suggestions. Cet outil propose à un utilisateur un mécanisme d'autocomplétion qui utilise le système de suggestions pour fournir des valeurs. L'interface est commune à toutes les versions du système. L'outil permet la visualisation et la mise à jour des bases de données RDF. Trois champs sont disponibles pour définir les valeurs du sujet, du prédicat et de l'objet. L'utilisation de préfixes a été mise en place pour améliorer la lisibilité de l'outil. La liste des espaces de noms existants et des préfixes associés est accessible dans un tableau récapitulatif. Lors de l'édition d'un triplet, l'éditeur affiche le *contexte* associé qui est rafraîchi à chaque fois que la valeur du champ sujet est mise à jour. Lorsqu'un nouveau triplet est créé et inséré dans la base de données, il est ajouté au contexte actuel. L'interface complète associée à plusieurs cas d'utilisation fait l'objet d'une vidéo de démonstration accessible en ligne[7].

4.1.2 Méthodologie

Cette évaluation implique un utilisateur unique qui est l'une des personnes chargées de l'édition du corpus de la correspondance d'Henri Poincaré. Il n'avait aucune expérience préalable avec cet outil au moment où il a conduit l'évaluation. L'ensemble de test est composé de 10 lettres qui ont été choisies au hasard parmi un ensemble de 30 lettres inédites provenant de la correspondance d'Henri Poincaré.

7. https://videos.ahp-numerique.fr/videos/watch/0d544e5b-b4be-423e-9497-216f29ab44f3

Cet ensemble constitue un véritable cas d'annotation par rapport aux lettres déjà éditées dans la base du corpus. Les éléments du corpus d'évaluation ont été édités en utilisant Omeka S avant le début de l'évaluation, de manière à veiller à ce qu'aucune version du système ne souffre d'être la première à être évaluée. Pour chaque version du système, l'utilisateur édite en une fois les 10 mêmes lettres en utilisant l'interface fournie. Les versions sont présentées dans un ordre aléatoire et inconnu. Avant de passer à la version suivante, la base RDF est réinitialisée pour correspondre à l'état initial.

Après avoir édité l'ensemble des lettres pour une version du système, l'utilisateur est invité à compléter un questionnaire pour fournir un retour d'expérience pour cette version. Cette enquête insiste sur l'appréciation de l'efficacité du mécanisme d'autocomplétion – mais un retour d'expérience sur l'interface utilisateur est également attendu. Pour chaque propriété, l'utilisateur est invité à attribuer une note en utilisant une *échelle de Likert* [2], de 1 (pas du tout pertinent) à 7 (très pertinent) pour caractériser la pertinence des suggestions fournies pour les questions d'annotation liées à cette propriété.

4.1.3 Résultats et analyse

Il ressort de cette évaluation que le système combinant déduction RDFS et raisonnement à partir de cas est perçu comme le plus efficace, et ce pour toutes les propriétés de l'évaluation. Les scores moyens de toutes les évaluations pour les différentes propriétés sont indiqués dans le tableau 1. Le système classique est la version qui a obtenu le score le plus bas. Il a été perçu comme « n'aidant pas à l'annotation », mais ne causant pas de problèmes à l'utilisateur. Le système déductif et le système à base de cas ont obtenu des notes moyennes élevées. Toutefois, dans les situations où la recherche de cas sources conduit à un ensemble de cas vide, le moteur de RàPC utilise uniquement l'ordre alphabétique pour le classement de la liste des suggestions, et peut fournir des ressources non pertinentes. Cela a causé de la frustration à l'utilisateur et explique pourquoi le score moyen du système à base de cas est inférieur à celui du système déductif. La combinaison des deux systèmes est une bonne méthode pour éviter ces situations. En outre, l'interface associée à l'outil a permis d'éviter les erreurs décrites dans l'introduction : elle empêche l'insertion de triplets qui existent déjà dans la base du corpus (*erreur de duplication*), le type de la ressource sélectionnée est toujours visible (*erreur d'ambiguïté*), et l'utilisation d'étiquettes simplifie la gestion des ressources pour l'utilisateur (*erreur de frappe*).

4.2 Évaluation automatique

4.2.1 Méthodologie

L'objectif de l'évaluation automatique est de comparer les performances des différentes versions de l'outil par des mesures. Les mesures choisies sont liées au rang de la valeur attendue $\texttt{rang}(qa)$ où qa est la question d'annotation actuelle. $\texttt{rang}(qa) = 1$ signifie que la valeur associée est la première dans la liste de suggestions. En d'autres termes, plus le rang moyen est petit, meilleure est la version.

Le graphe RDF de la correspondance d'Henri Poincaré $\mathcal{G}_{\text{HP}}$ a été utilisé comme un ensemble de tests. Ce graphe est formé par l'union de la base de faits $\mathcal{D}_{\text{HP}}$ et de l'ontologie $\mathcal{O}_{\text{HP}}$: $\mathcal{G}_{\text{HP}} = \mathcal{D}_{\text{HP}} \cup \mathcal{O}_{\text{HP}}$. Au moment de la rédaction de cet article, le graphe est composé d'environ 220 000 triplets. Pour cette évaluation, l'application des règles d'inférence RDFS mentionnées dans la section 2 a été considérée. Un ensemble de 100 lettres est aléatoirement extrait de l'ensemble existant des lettres éditées. Pour chaque lettre de cet ensemble, les triplets formant son contexte sont utilisées pour simuler des questions d'annotation dont la réponse est déjà connue. Pour chaque triplet, l'édition des 3 champs (sujet, prédicat et objet) est considéré dans un ordre aléatoire afin d'inclure différents types de questions d'annotation dans l'évaluation. Pour chaque question d'annotation qa, les quatre systèmes de suggestion sont appelés pour fournir une liste de suggestions ordonnée. Le rang de la valeur attendue $\texttt{rang}(qa)$ dans la liste est enregistré pour chaque version et est ajouté au multi-ensemble $\texttt{Rangs}(\mathit{système})$ correspondant. À l'issue de l'évaluation, des mesures relatives aux éléments de $\texttt{Rangs}(\mathit{système})$ sont calculées. Ces mesures correspondent au pourcentage de questions annotées dont la valeur escomptée a été donnée parmi les n premières propositions.

4.2.2 Résultats et analyse

Les résultats de cette évaluation sont présentés dans le tableau 2 pour chaque version du système. Différentes valeurs de n ont été choisies (5, 10 et 15) mais l'évolution de l'efficacité des versions reste la même dans toutes les situations. Cela montre que le système combiné fournit les meilleurs résultats pour les différentes questions d'annotation liées à cette évaluation parce qu'il suggère plus souvent la valeur appropriée. Il est donc plus susceptible d'aider l'utilisateur pendant le processus d'édition. Bien qu'il n'ait pas été utilisé comme mesure lors de l'évaluation, le temps de calcul a été pris en considération. Il correspond au temps nécessaire pour fournir la liste de suggestions pour une question d'annotation. En effet, le temps de réaction aux demandes doit être pris en compte dans un système d'interaction humaine d'autant plus que ce système utilise un mécanisme d'autocomplétion pour lequel un utilisateur ne s'attend pas à une latence. Le temps de calcul est plus important lorsqu'on utilise le système combiné mais cela reste suffisamment bas pour ne pas avoir d'impact sur l'utilisateur.

5 Discussion et travaux proches

5.1 Les éditeurs de données RDF

La méthode de RàPC présentée dans la section 3 est inspirée du système UTILIS [8]. Ce système introduit l'idée de rechercher des ressources similaires à celle qui est éditée pour suggérer des valeurs qui pourraient être appropriées au problème d'édition courant. Mais la forme de relaxation de requêtes proposée est différente car elle s'appuie principalement sur des règles de généralisation indépendantes du domaine d'application. Le moteur présenté dans cet article permet aux utilisateurs de définir leurs propres règles pour

TABLEAU 1 – Score moyen (sur une échelle de 1 à 7) associé a la pertinence des suggestions pour les différentes versions du système.

	Système classique	Système déductif	Système à base de cas	Système combiné
Score moyen	3,4	5,7	5,3	7

TABLEAU 2 – Mesures du rang lié aux suggestions pour les quatre versions du système.

	Système classique	Système déductif	Système à base de cas	Système combiné
$rang \leqslant 15$	11,3%	22,11%	49,0%	49,5%
$rang \leqslant 10$	7,1%	21,15%	43,2%	43,2%
$rang \leqslant 5$	2,7%	19,23%	33,6%	34,7%

exploiter des connaissances spécifiques à un domaine d'application. Combiné à l'utilisation des connaissances d'une ontologie RDFS, ce système propose des suggestions adaptées aux différents types de questions d'annotation.
Protégé [11] est l'un des outils les plus fréquemment utilisés pour éditer les données du Web sémantique. Lors de l'édition d'une instance d'une classe spécifique, Protégé s'appuie sur les domaines et co-domaines des propriétés pour faire des suggestions de valeurs de prédicats et d'objets. Mais ces suggestions ne s'appuient pas sur les triplets déjà édités présents dans la base. D'autres approches existent pour faciliter l'édition des bases RDF, plusieurs d'entre elles sont fondées sur le traitement automatique des langues. L'outil d'édition GINO [3] propose l'utilisation d'un langage naturel guidé et contrôlé qui permet à l'utilisateur de préciser des phrases correspondant à des faits RDF. L'idée principale est que les principes du Web sémantique ne sont parfois pas facilement appréhendés par les non spécialistes, et devraient donc être intégrés dans un système plus convivial. La syntaxe de ce langage est proche de la syntaxe de l'anglais (par exemple « There is a mount named Everest », « The height of mount Everest is 29 029 feet », etc.). Un mécanisme de suggestion propose des classes, des instances et des propriétés pour compléter l'annotation en cours. Le principal défi de ce système concerne l'interprétation de la demande de l'utilisateur afin de construire des triplets à partir de phrases.

5.2 Les systèmes de recommandation

Plus généralement, l'outil présenté dans cet article pourrait être défini comme un système de recommandation. Ces systèmes visent à aider l'utilisateur en présentant des informations susceptibles de l'intéresser. Différents systèmes de recommandation, tels que celui présenté ici, s'appuient sur le RàPC [14]. Il existe une grande variété de méthodes, et l'outil présenté dans cet article pourrait bénéficier de plusieurs d'entre elles. À titre d'exemple, l'implication de l'utilisateur dans le mécanisme de proposition de suggestions est envisagée. L'explicabilité de l'outil pourrait être renforcée car il peut être important de comprendre pourquoi certaines ressources ont été favorisées pour un contexte d'édition particulier. Ce système pourrait également bénéficier de l'utilisation d'un système de retour d'utilisation fondé sur des préférences, ce qui pourrait améliorer les résultats de l'outil dans plusieurs situations, donner à l'utilisateur le sentiment d'être inclus et renforcer ainsi la perception positive de l'outil. D'autre part, le mécanisme de transformation de requêtes utilisé ici pourrait être réutilisé dans d'autres systèmes de recommandation.

6 Conclusion et perspectives

Un processus d'annotation manuelle a été choisi pour éditer les données relatives aux ressources du corpus de la correspondance d'Henri Poincaré. Ce processus a été identifié comme étant fastidieux pour les utilisateurs chargés de l'édition. Pour remédier à ce problème, un outil proposant un système de suggestions a été mis en place. Il s'agit d'un outil général pour l'édition des données RDF. Il utilise des déductions s'appuyant sur une ontologie RDFS combinée à du RàPC. Différentes versions du système ont été créées. La première version classe les valeurs potentielles en utilisant l'ordre alphabétique. La deuxième version tire parti des connaissances sur les domaines et co-domaines des propriétés de l'ontologie. La troisième version utilise du RàPC pour exploiter les connaissances à propos des ressources éditées similaires à celle en cours d'édition. La dernière version est une combinaison des deux précédentes. Deux évaluations différentes ont été réalisées. L'évaluation humaine a permis de comparer les différentes versions du système entre elles et avec le système d'annotation actuel (Omeka S). L'évaluation automatique a apporté des mesures en comparant, pour un ensemble sélectionné de questions d'annotation, les suggestions des différentes versions du système. La pertinence des suggestions et le temps de calcul ont été pris en compte. Comme expliqué dans la section 4, alors que les mesures calculées par l'évaluation automatique montrent que l'utilisation du RàPC seul a donné de meilleurs résultats que l'utilisation de la déduction RDFS seule, le système à base de cas est parfois insuffisant et peut fournir des ressources non pertinentes dans certaines situations. Pour les deux évaluations, les résultats montrent que la dernière version du système qui est la combinaison de l'utilisation de la déduction RDFS avec celle du RàPC est la plus efficace.Toutefois, dans certaines situations, ce système présente certaines limites. Par exemple, considérons la question d'annotation présentée dans la section 3. Les troisième et quatrième versions du système proposent `henriPoincaré` comme réponse plausible bien qu'il soit déjà défini comme l'expéditeur de la lettre en

cours d'édition. Une façon de traiter cette question serait d'utiliser certaines connaissances du domaine en tant que contraintes d'intégrité. Un autre point observé lors des évaluations humaine et automatique est que l'ordre d'édition des différentes propriétés affecte grandement l'efficacité du moteur de suggestion. En effet, certaines valeurs de propriétés donnent plus d'informations sur la ressource que d'autres, et donc le fait de remplir ces valeurs en premier devrait améliorer le classement des suggestions. Le principal défi consiste alors à trouver le meilleur ordre d'édition pour les propriétés de la base. Ceci constitue une perspective de recherche. Un autre axe de recherche est lié à l'utilisation d'une logique plus expressive que RDFS telle que OWL-DL. Une logique contenant une forme de négation permettrait de supprimer certaines valeurs de la liste des valeurs potentielles. Toutefois, une telle extension pourrait affecter le temps de calcul et sa mise en œuvre devrait donc être étudiée.

Remerciements

Ce travail a bénéficié d'une aide de l'État, gérée par l'Agence Nationale de la Recherche, au titre du projet Investissements d'Avenir Lorraine Université d'Excellence, portant la référence ANR-15-IDEX-04-LUE.

Références

[1] A. Aamodt et E. Plaza. Case-based Reasoning : Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1) :39-59, 1994.

[2] I. E. Allen et C. A. Seaman. Likert scales and data analyses. *Quality progress*, 40(7) :64-65, 2007.

[3] A. Bernstein et E. Kaufmann. GINO–a guided input natural language ontology editor. *International semantic web conference*, pages 144-157. Springer, 2006.

[4] C. Boulaire et R. Carabelli. Du digital naive au bricoleur numérique : les images et le logiciel Omeka. É. Cavalié, F. Clavert, O. Legendre et D. Martin, éditeurs, *Expérimenter les humanités numériques. Des outils individuels aux projets collectifs*, chapitre 7, pages 81-103. Les Presses de l'Université de Montréal, Montréal, Québec, 2017.

[5] D. Brickley, R. V. Guha et B. McBride. RDF Schema 1.1, 2014. URL : `https://www.w3.org/TR/rdf-schema/`. Dernière consultation : février 2021.

[6] O. Bruneau, E. Gaillard, N. Lasolle, J. Lieber, E. Nauer et J. Reynaud. A SPARQL Query Transformation Rule Language — Application to Retrieval and Adaptation in Case-Based Reasoning. D. Aha et J. Lieber, éditeurs, *Case-Based Reasoning Research and Development. ICCBR 2017*, Lecture Notes in Computer Science, pages 76-91, Cham. Springer, 2017.

[7] O. Bruneau, N. Lasolle, J. Lieber, E. Nauer, S. Pavlova et L. Rollet. Applying and Developing Semantic Web Technologies for Exploiting a Corpus in History of Science : the Case Study of the Henri Poincaré Correspondence. *Semantic Web*, 12(2), 2021.

[8] A. Hermann, S. Ferré et M. Ducassé. An Interactive Guidance Process Supporting Consistent Updates of RDFS Graphs. A. ten Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles et N. Hernandez, éditeurs, *Knowledge Engineering and Knowledge Management*, pages 185-199, Berlin, Heidelberg. Springer Berlin Heidelberg, 2012.

[9] N. Lasolle, O. Bruneau, J. Lieber, E. Nauer et S. Pavlova. Assisting the RDF Annotation of a Digital Humanities Corpus Using Case-Based Reasoning. J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne et L. Kagal, éditeurs, *The Semantic Web - ISWC 2020*, pages 617-633, Cham. Springer International Publishing, 2020.

[10] F. Manola, E. Miller, B. McBride et al. RDF Primer, 2004. URL : `https://www.w3.org/TR/rdf-primer`. Dernière consultation : février 2021.

[11] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson et M. A. Musen. Creating Semantic Web Contents with Protégé-2000. *IEEE intelligent systems*, 16(2) :60-71, 2001.

[12] E. Prud'hommeaux. SPARQL Query Language for RDF. E. Prud'hommeaux et A. Seaborne, éditeurs, 2008. URL : `http://www.w3.org/TR/rdf-sparql-query/`. Dernière consultation : février 2021.

[13] C. K. Riesbeck et R. C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989.

[14] B. Smyth. Case-based recommendation. P. Brusilovsky, A. Kobsa et W. Nejdl, éditeurs, *The adaptive web*, pages 342-376. Springer, Berlin, 2007.