



HAL
open science

La centralité des ontologies, du Web Sémantique des utilisateurs au Web Sémantique des objets

Nathalie Jane Hernandez

► **To cite this version:**

Nathalie Jane Hernandez. La centralité des ontologies, du Web Sémantique des utilisateurs au Web Sémantique des objets. Intelligence artificielle [cs.AI]. Université de Toulouse, 2019. tel-02919961

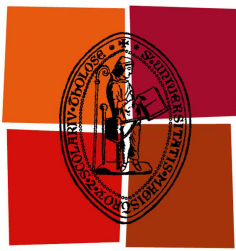
HAL Id: tel-02919961

<https://hal.science/tel-02919961>

Submitted on 24 Aug 2020

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

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



Université
de Toulouse

MANUSCRIT

En vue de l'obtention de l'

HABILITATION A DIRIGER DES RECHERCHES DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse - Jean Jaurès*

Présentée et soutenue le *vendredi 13 décembre 2019 (13/12/2019)* par :

NATHALIE HAEMMERLÉ-HERNANDEZ

**La centralité des ontologies, du Web Sémantique des utilisateurs au
Web Sémantique des objets**

JURY

FRÉDÉRIQUE LAFOREST
CHANTAL REYNAUD
ANDREA TETTAMANZI
MICHAEL MARISSA
JOSIANE MOTHE
NATHALIE
AUSSENAC-GILLES

Professeur
Professeur
Professeur
Professeur
Professeur
Directeur de recherche

Membre du Jury
Membre du Jury
Membre du Jury
Membre du Jury
Membre du Jury
Membre du Jury

École doctorale et spécialité :

MITT : Image, Information, Hypermédia

Unité de Recherche :

IRIT - Institut de Recherche en Informatique de Toulouse - UMR 5505

Rapporteurs :

Frédérique Laforest, Chantal Reynaud et Andrea Tettamanzi

Remerciements

Je tiens tout d'abord à remercier Frédérique Laforest, Chantal Reynaud et Andrea Tetta-
manzi d'avoir accepté d'être rapporteurs de cette HDR. Extrêmement admirative de la carrière
de Chantal, je suis très touchée qu'elle m'honore de sa présence dans mon jury d'HDR.

Je remercie également Michael Mrissa qui a pu se rendre disponible pendant sa délégation
en Slovénie.

Merci à Josiane Mothe qui m'a initiée à la recherche en encadrant mon DEA puis ma thèse.
Je suis ravie que cette HDR soit l'occasion de confronter de nouveau nos visions scientifiques.

Merci à Nathalie Aussenac-Gilles qui suit également mes travaux depuis leur début. Je
tiens à lui témoigner ma gratitude pour avoir accompagné mon premier encadrement de thèse
ainsi que mon évolution dans le monde de la recherche, que ce soit au sein de l'équipe Me-
lodi, dans les différents projets de recherche auxquels nous avons participé ou encore par le
mentorat de ce projet d'HDR.

Je tiens par dessus tout à adresser mes plus chaleureux remerciements aux étudiants que
j'ai eu la chance d'encadrer. Ce mémoire repose sur leurs travaux et n'existerait pas sans
eux. La richesse de nos échanges scientifiques alimente sans cesse mes réflexions sur le Web
Sémantique, sur l'I.A. et sur le monde de demain. Leur pertinence, leur disponibilité, leur
état d'esprit, leurs points de vue et leur faculté à s'adapter à mes particularités ont enrichi
plus généralement ma vie professionnelle. J'ai pris et prends encore un plaisir fou à partager
ces moments de vie avec eux. Ils m'ont rendu fier à plus d'une occasion.

Merci à Anis Tissaoui, Camille Pradel, Fabien Amarger, Rafik Abbes, Nicolas Seydoux, Rim
Teyeb et Elodie Thiéblin.

Merci aussi à Noor, Max, Pierre, Quentin, Gilles, Camille, Marie, Arona, Akila, Juan, Gre-
gory, Samuel.

Un énorme merci à ceux qui sont aujourd'hui mes amis et avec qui je veillerai à continuer de
partager les repas au On'Nador, les longueurs de piscine, les boissons qui bouillent, les sché-
mas évocateurs, le plaisir de la rédaction, la bienveillance, le CRI, les manifs, les concerts,
l'amour du Web, la politique. Une mention particulière à Caroline Desprat que je compte
parmi ces précieuses personnes même si je n'ai pas eu la chance de l'encadrer.

Je veux maintenant également remercier les chercheurs avec qui j'ai partagé le coencadre-
ment de ces personnes.

Philippe Laublet, tout d'abord, pour qui j'ai une pensée particulièrement émue. J'ai eu la
chance de découvrir le rôle d'encadrant à ses côtés et cette expérience a eu une influence
déterminante sur ma façon d'appréhender ma posture d'enseignant-chercheur.

Ollivier Haemmerlé pour sa rigueur et sa vision de l'intelligence artificielle qui ont influencé

nos encadrements communs. Je tiens également à le remercier pour son investissement pour la collectivité empreint d'équité et de tant de valeurs que je partage.

Jean-Pierre Chanet et Catherine Roussey pour leur enthousiasme et leur chaleur humaine. Merci à Catherine pour sa disponibilité et son recul, que ce soit sur l'ingénierie des connaissances, la taxonomie des plantes, les réseaux de capteurs ou les tourments de la vie.

Thierry Monteil et Khalil Drira qui m'ont permis de découvrir les défis de l'IoT et m'ont aussi apporté leur point de vue judicieux sur notre profession ainsi que sur les collaborations internationales.

Juliette Dibie et Patrice Buche avec qui il est extrêmement enrichissant de collaborer.

Jean-Paul Arcangeli, Sylvie Trouilhet et Jean-Michel Bruel avec qui j'espère n'être qu'au début d'une longue collaboration.

Mohand Boughanem et Karen Sauvagnat qui m'ont aidé à mettre en avant les points communs entre la recherche d'information et l'ingénierie des connaissances. La richesse de nos échanges a fait évoluer mon point de vue sur le Web.

Je tiens aussi à remercier plus généralement l'équipe IRIS qui m'a toujours accueillie à bras ouverts. Cette équipe représente pour moi l'exemple de la bienveillance.

Je tiens également à remercier plus particulièrement Karen qui, depuis nos thèses, est TOUJOURS LA. Quelle chance j'ai de l'avoir à mes cotés !

Merci à Fleur Mougin qui 16 ans plus tard est toujours à l'écoute pour me sortir des rundos puntos del pensamiento ! Prochaine étape : un projet de recherche en commun.

Merci à la communauté IC que je prends plaisir à retrouver lors des différents événements.

Merci à Dominique Laffly et aux palois pour ces collaborations mêlant pédagogie, recherche et voyages instructifs.

Plus localement, un merci tout particulier à Véronique Fabris et à Francine Soncourt qui, par leurs attentions de chaque instant, rendent la vie plus facile. Et merci à Manu Hallouin pour toutes ces discussions sur nos étudiants, la politique ou bien la vie, qui resserrent et permettent d'avancer.

Merci aux amis de toujours et aux fidèles entrevalais.

Un énorme MERCI à la dreamteam Marengo-Périole... Merci de mener ces combats qui donnent sens à la vie et foi en l'humanité. Plus égoïstement, merci de m'avoir soutenue pour mener à bien ce défi personnel. Une pensée particulière pour Tamar, sa famille et les personnes contre lesquelles notre société s'acharne au lieu d'œuvrer pour préserver leurs droits fondamentaux.

Que serait ma vie sans ma famille ...

Merci à mes parents pour leur soutien indéfectible, leurs valeurs, leur amour. Merci à ma soeur avec qui le fil reste toujours aussi tendu malgré les aléas de la vie.

Merci à mon mari pour sa disponibilité, ses encouragements et ses délicieux repas remplis d'amour. MMNAGB.

Merci à Quentin et Louison, mes beaux enfants.

Merci à Raphaël et Elsa d'avoir accepté et soutenu mon investissement dans ce projet d'HDR. Raph dont l'imagination digne d'un chercheur en herbe l'a conduit entre autres à inventer de nouveaux usages pour les ontologies allant même jusqu'à soigner les maux de la vie... Elsa dont le courage et la sensibilité nous rendent plus fortES. Tous les deux, qui partagent mon admiration pour TBL. Merci de me rendre la vie si belle.

Table des matières

1	Introduction	3
2	Contexte	7
2.1	Ontologies et Bases de Connaissances	7
2.1.1	Ressource Termino-Ontologique	9
2.1.2	Vocabulaire partagé	10
2.1.3	Vocabulaire partagé et lié	11
2.1.4	Graphe de connaissances	11
2.2	De l’IoT au SWoT en passant par le WoT	12
3	Réutilisation d’ontologies et de bases de connaissances	15
3.1	Introduction	15
3.2	Réutilisation de sources pour un besoin applicatif	17
3.2.1	Motivations et travaux existants	18
3.2.2	Transformation de modèles conceptuels guidée par une amorce de module	20
3.2.3	Fusion de bases de connaissances	22
3.2.4	Extension optimale	25
3.2.5	Évaluations	26
3.3	IoT-O : une ontologie modulaire de core-domain pour l’IoT	27
3.3.1	Motivations	27
3.3.2	Exigences pour une ontologie de l’IoT	28
3.3.3	Analyse de l’existant	29
3.3.4	Ontologies réutilisées pour IoT-O	30
3.3.5	IoT-O : une ontologie de core-domain	31
3.3.6	Utilisations d’IoT-O	33
3.4	Conclusion	33
4	Évolution d’ontologies guidée par des textes	36
4.1	Introduction	36
4.2	Évolution d’ontologies pour l’annotation de textes	37
4.2.1	Motivations et travaux existants	37
4.2.2	Aperçu de l’approche	38
4.2.3	Déclencher un changement	39
4.2.4	Représentation des changements	40
4.2.5	Influence sur les annotations	41
4.2.6	Évaluations	41
4.3	Recherche de documents vitaux hors corpus	41

4.3.1	Travaux connexes	42
4.3.2	Vue générale de l'approche	43
4.3.3	Sélection des documents vitaux	43
4.3.4	Sélection des phrases vitales	43
4.3.5	Détection de la nouveauté	44
4.3.6	Évaluations	45
4.4	Conclusion	46
5	Interrogation de bases de connaissances guidée par le besoin utilisateur	47
5.1	Introduction	47
5.2	Interrogation d'une base de connaissances à partir de patrons de requêtes	48
5.2.1	Motivations	48
5.2.2	Vue générale du système Swip	52
5.2.3	De la requête initiale à la requête pivot	53
5.2.4	De la requête pivot à la requête finale	55
5.2.5	Implementation tout en SPARQL	59
5.2.6	Aperçu de l'évaluation	60
5.3	Interroger plusieurs bases de connaissances	62
5.3.1	Motivations	62
5.3.2	Illustration de l'approche de réécriture	67
5.3.3	Évaluations	68
5.4	Conclusion	69
6	Le Web Sémantique pour les objets connectés	71
6.1	Introduction	71
6.2	Le Web Sémantique pour l'IoT : notre vision	72
6.2.1	Motivations	72
6.2.2	Les nœuds	73
6.2.3	Les fonctions	74
6.2.4	Adaptation des technologies du Web Sémantique aux contraintes de l'IoT	79
6.3	Distribuer les traitements sur des nœuds Fog	80
6.3.1	Motivation	80
6.3.2	Vue générale de l'approche	83
6.3.3	Les règles EDR et leur propagation	86
6.3.4	EDRT : une stratégie de propagation pour EDR	89
6.3.5	Évaluations	93
6.4	Conclusion	93
7	Analyses et perspectives	95
7.1	Des ontologies pour décrire	95
7.2	Des patrons ou formes pour façonner	96
7.3	Réutiliser des ontologies ?	97
7.3.1	Place de la "sémantique"	97
7.3.2	Faciliter la réutilisation de vocabulaires	98
7.4	Les ontologies pour gérer l'interopérabilité ?	99
7.4.1	L'interopérabilité au niveau de l'alignement	99
7.4.2	Un défi qui attend l'IoT	100
7.5	Re-décentraliser le Web Sémantique	100
7.6	Quelle place pour les utilisateurs finals dans l'IoT ?	101
7.6.1	Agir sur le réseau d'objets à partir de la langue naturelle	101
7.6.2	Les données personnelles	101

Liste des figures

2.1	Représentation simplifiée du modèle de RTO présenté dans Reymonet et al. (2007)	10
3.1	Partie haute du module Agronomic Taxon	18
3.2	Exemple de Taxons décrits à partir du module Agronomic Taxon	19
3.3	Processus de transformation d'un modèle conceptuel	21
3.4	Processus de fusion des bases de connaissances	22
3.5	Exemple de candidats sommets	23
3.6	Exemple de candidats arcs	24
3.7	Exemple d'incompatibilité entre deux candidats	25
3.8	Les patrons de conception SSO et AAE, structurant respectivement les ontologies SSN et SAN	31
3.9	Aperçu de l'architecture d'IoT-O	32
4.1	Types d'évolutions pour l'annotation à partir de RTO	38
4.2	TextViz et EvOnto dans Protégé	39
4.3	Présentation d'une stratégie dans EvOnto	40
5.1	Vue d'ensemble du processus d'interprétation de Swip.	53
5.2	Étapes de l'interprétation de la requête en langue naturelle.	54
5.3	Graphe requête pour l'exemple 1	56
5.4	Graphe requête pour l'exemple 2	56
5.5	Graphe requête pour l'exemple 3	56
5.6	Graphe requête pour l'exemple 4	57
5.7	Patron issu de la requête 5.3.	57
5.8	Patron proposé pour représenter la requête 5.4	57
5.9	Patron proposé pour représenter la requête 5.5	57
5.10	Patron correspondant à la fusion des patrons 5.8 and 5.9	58
5.11	Graphe requête issu de l'instantiation du patron 5.10 pour la requête de l'exemple 2.	58
5.12	Exemple de patron complexe, contenant un sous-patron répétable.	58
5.13	Exemple de patron complexe, contenant des sous-patrons optionnels et répétables.	58
5.14	Étapes de l'interprétation de la requête pivot.	59
5.15	Fragment de l'ontologie Agronomic Taxon	63
5.16	<i>Triticum durum</i> taxon représenté dans Agronomic Taxon	63
5.17	Fragment de l'ontologie de DBpedia	64
5.18	<i>Triticum durum taxon</i> dans DBpedia	64

5.19	Fragment de l'ontologie TAXREF-LD	65
5.20	<i>Letriticum durum taxon</i> représenté dans TAXREF-LD	65
5.21	Workflow global de l'approche	67
6.1	Patron des architectures SWoT trois-tiers	74
6.2	Scénario dans une usine connectée	81
6.3	Rôle de la base de connaissances dans le comportement d'un nœud d'EDR	84
6.4	Modules constituant une règle EDR	87
6.5	Vue générale de l'algorithme d'EDR	90
6.6	Exemple de règles déployées suivant EDRT	91

Liste des tableaux

3.1	Couverture des notions-clés de l’IoT dans les principales ontologies existant en 2016	30
3.2	Réutilisabilité des ontologies de l’IoT	30
4.1	Exemple de phrases vitales	44
5.1	Requête exprimée sur Agronomic Taxon pour retrouver les sous-taxons de <i>Triticum</i>	65
5.2	Requête initiale exprimée sur Agronomic Taxon pour retrouver les sous-taxons de <i>Triticum</i> et les requêtes pour DBpedia et TAXREF-LD obtenues par ré-écriture automatique.	68

Le développement du Web depuis les années 1990 puis l'avènement du Web Sémantique à l'orée du XXI^{ème} siècle ont conduit à la mise en place de gigantesques réservoirs de connaissance. Au travers du réseau Internet, cette connaissance est accessible aux êtres humains mais aussi à des agents logiciels qui, peu à peu, intègrent le moindre objet de notre quotidien. À l'échelle de l'Humanité, il est vraisemblable que ce partage universel de connaissances soit un jour comparé à la révolution qu'avait été en son temps l'invention de l'imprimerie.

Les formalismes de représentation de connaissances utilisés pour décrire les ressources (données, entités, documents) identifiables sur le Web Sémantique découlent des travaux de chercheurs en Intelligence Artificielle, notamment menés au cours de la deuxième moitié du XX^{ème} siècle. Les descriptions de ressources s'appuient sur des vocabulaires déréférencables publiés sur le Web : les "ontologies".

Un premier défi à relever pour permettre la mise en place du Web Sémantique a donc été de développer et de déployer des ontologies. Des travaux menés au début des années 2000 ont permis de définir des méthodes et des outils facilitant la construction de telles représentations de connaissances. Des centaines d'ontologies sont aujourd'hui largement diffusées. Cependant, comme elles sont généralement construites pour des besoins ciblés, leur réutilisation et leur évolution ne sont pas forcément triviales.

Une première partie de mes travaux a été menée dans le but de faciliter leur réutilisation pour de nouveaux usages. Les points suivants ont été plus particulièrement abordés :

- la réutilisation conjointe de ressources hétérogènes telles que les terminologies, les thésaurus et les bases de connaissances, afin de construire de nouvelles bases de connaissances pour un objectif applicatif donné ;
- la réutilisation d'ontologies pour définir des ontologies générales pour un domaine, appelées ontologies de core-domain.

Une autre problématique cruciale pour la mise en place du Web Sémantique réside dans la mise à jour des ontologies lorsque le domaine de connaissances qu'elles représentent évolue ou lorsque les ressources qu'elles servent à décrire changent. En effet, la quasi-totalité des informations présentes sur le Web traitent de domaines non figés. Nous avons défini différentes approches permettant :

- de mettre à jour les descriptions de textes reposant sur une ontologie lorsque celle-ci évolue ;
- d'identifier des documents du Web contenant de l'information nouvelle par rapport à la connaissance déjà représentée dans une base de connaissances.

Ces bases de connaissances, que l'on suppose à jour, sont mises à disposition dans des formalismes standardisés par le W3C, qui ne sont malheureusement que peu exploitables par des humains. Nous avons montré qu'il est possible de faciliter l'accès aux informations présentes sur le Web pour un utilisateur final par :

- la mise en place d'une approche visant à transformer une question exprimée en langage naturel en requête SPARQL, et ce de façon transparente pour l'utilisateur ;
- la définition de méthodes mettant en correspondance les descriptions proposées dans les différentes bases de connaissances présentes sur le Web.

Récemment, le développement de l'Internet des Objets (ou IoT, pour Internet of Things) a accentué la place des agents logiciels sur le Web. L'IoT consiste en un ensemble de technologies et de principes qui permettent le déploiement de réseaux d'objets et de services. De nombreuses perspectives en termes d'applications s'ouvrent, mais l'interopérabilité, la configuration des objets ou le passage à l'échelle en sont autant de freins.

Nous avons cherché à montrer comment les technologies du Web Sémantique pouvaient être utilisées et adaptées pour prendre en compte ces différents aspects en nous intéressant plus particulièrement :

- à la place que pouvaient trouver les technologies du Web Sémantique dans une architecture IoT ;

— au développement d’algorithmes de distribution du raisonnement au plus près des objets qui collectent les données.

La mise en place de l’ensemble de ces technologies et algorithmes dans le proche environnement des êtres humains, sans même que ces derniers ne le perçoivent, n’est pas sans répercussions sur des questions de respect de la vie privée et des droits humains. C’est une des perspectives que je brosse en conclusion.

CHAPITRE 1

Introduction

Il est de coutume de présenter l'Intelligence Artificielle comme étant l'héritière des philosophes logiciens de la Grèce antique. Plus prosaïquement, l'Intelligence Artificielle moderne est souvent "datée" du milieu du XX^{ème} siècle, avec les travaux désormais largement médiatisés d'Alan Turing. Parmi les champs d'investigation des chercheurs en Intelligence Artificielle, la représentation de connaissances a été un domaine particulièrement fructueux, à l'intersection de la logique, de la théorie des graphes, mais aussi de sciences sociales comme la linguistique. Depuis les réseaux sémantiques jusqu'aux logiques de description, des formalismes de représentation de connaissances aux propriétés théoriques de mieux en mieux étudiées ont vu le jour dans la deuxième moitié du XX^{ème} siècle.

De manière partiellement concomitante, la fin du XX^{ème} siècle a connu un bouleversement rapide dans l'accès à l'information avec le développement d'un média à ce point répandu que l'on confond parfois le support (l'Internet) avec ce service phare qu'il véhicule : le World Wide Web, plus communément appelé "le Web". Le Web, dans sa version originelle, permettait de lier entre elles des informations réduites à leur plus simple expression de texte mis en forme, compréhensibles quasi-exclusivement par les humains.

Dès que le Web a pris son essor et qu'il s'est transformé en un gigantesque réservoir de toutes sortes d'informations, de nombreux travaux de recherche en Intelligence Artificielle, et notamment en Représentation des connaissances, se sont attelés à en exploiter le potentiel. Ces travaux ont été menés dans deux directions complémentaires : d'une part, la masse d'informations "cachée" dans le web nécessitait de disposer de méthodes de fouille, de recherche d'informations, et d'autre part, les formats de représentation des connaissances devaient être adaptés afin de décrire de façon fine les connaissances représentées et d'autoriser du raisonnement sur ces grandes masses d'information distribuées.

Mes travaux s'inscrivent dans ce moment charnière de l'évolution de l'humanité où le stockage de données de manière massive laisse entrevoir des possibilités de rupture scientifique comme a pu en représenter en son temps l'avènement de l'imprimerie. Ce moment voit à l'échelle mondiale la constitution de vocabulaires riches appelés "ontologies" qui décrivent partiellement les connaissances qui peuvent être représentées sur le Web. Ce moment voit la nécessité de faciliter l'accès des humains à la gigantesque masse d'informations disponible, comme on leur a facilité l'accès, jadis, aux bibliothèques. Ce moment voit, enfin, l'arrivée de ce type de connaissances au cœur même des objets connectés qui se généralisent dans notre quotidien et qui, peu à peu, organisent leur communication au travers du réseau à l'aide de leur connaissance propre, sans intervention humaine...

Mon travail de recherche s'inscrit dans le domaine de l'Intelligence Artificielle, et plus particulièrement dans le domaine du Web Sémantique. Le Web Sémantique [Berners-Lee et al.](#)

(2001) est une extension du Web de la fin du XX^{ème} siècle, permettant l'intégration d'une représentation formelle de la connaissance, afin de la rendre directement exploitable par des algorithmes. Une application concrète du Web Sémantique est le Web de données liées (ou Linked Open Data) [Christian et al. \(2011\)](#), qui propose notamment de représenter des descriptions des données dans un format unique afin de favoriser leur réutilisation. Les représentations utilisées pour décrire les ressources (données, entités, documents) identifiables sur le Web Sémantique s'appuient sur des vocabulaires déréférencables que sont les "ontologies" publiées sur le Web. En plus de favoriser l'interopérabilité, les ontologies permettent également la déduction automatique de nouvelles informations à partir de la connaissance qu'elles représentent. Les descriptions de ressources ainsi que les ontologies sur lesquelles elles reposent constituent des bases de connaissances accessibles sur le Web. Le W3C, organisme en charge de la standardisation du Web, a proposé un certain nombre de recommandations pour mettre en œuvre le Web Sémantique¹. Les ontologies, au cœur du Web sémantique, ont pour but de faciliter l'accès et la manipulation des ressources Web par deux types d'agents : les agents logiciels et les agents humains. Les ontologies représentent la connaissance utile pour l'exploitation potentielle des ressources du Web par ces deux types d'acteurs.

Le premier défi auquel le Web sémantique a dû faire face est la mise en place d'une surchouche sémantique sur le Web, en développant et déployant des ontologies. Les travaux menés dans la première partie des années 2000, période pendant laquelle j'ai préparé ma thèse, ont permis de définir des méthodes et des outils facilitant la construction de telles représentations de connaissances [Cimiano \(2006\)](#). Comme en témoigne le LOV² qui répertorie et donne accès à des ontologies à partir du domaine de connaissances qu'elles couvrent et de la connaissance qu'elles représentent, des centaines d'ontologies sont aujourd'hui disponibles. Cependant, comme elles sont généralement construites pour des besoins ciblés, leur réutilisation et leur évolution n'est pas forcément triviale.

Le premier chapitre de mon manuscrit décrit les travaux qui ont été menés pour faciliter leur réutilisation pour de nouveaux usages. Les points suivants sont plus particulièrement abordés :

- la construction de bases de connaissances pour un besoin applicatif peut être facilitée par la réutilisation conjointe de ressources hétérogènes telles que les textes, les terminologies, les thésaurus et les bases de connaissances. Ceci présente un enjeu pour le Web Sémantique car réutiliser des ressources existantes permet de capitaliser l'expertise et les efforts de modélisations entrepris [Amarger et al. \(2016a\)](#), [Amarger et al. \(2016b\)](#) ;
- le besoin applicatif peut être considéré lors de la réutilisation de sources ;
- des ontologies peuvent être réutilisées pour définir des ontologies générales pour un domaine aussi appelées ontologies de core-domain [Seydoux et al. \(2016b\)](#).

Le deuxième chapitre se concentre sur une autre problématique cruciale pour la mise en place du Web Sémantique : la mise à jour de ces ontologies lorsque le domaine de connaissances qu'elles représentent évolue ou lorsque les ressources qu'elles servent à décrire changent. Ce type d'approche est d'autant plus essentiel que la quasi-totalité des informations présentes sur le Web traitent de domaines non figés, dont la connaissance évolue constamment. Dans ce chapitre sont décrites différentes approches permettant :

- d'exploiter les ressources textuelles pour un domaine donné afin de mettre à jour une base de connaissances lorsque le domaine représenté évolue [Tissaoui et al. \(2011b\)](#), [Ottens et al. \(2009\)](#) ;
- de mettre à jour les descriptions de textes reposant sur une ontologie lorsque celle-ci évolue [Tissaoui et al. \(2011a\)](#) ;

1. <https://www.w3.org/standards/semanticweb/>

2. <https://lov.linkeddata.es/dataset/lov/>

- d’identifier des documents du Web contenant de l’information nouvelle par rapport à la connaissance représentée dans une base de connaissances [Abbes et al. \(2015c\)](#), [Abbes et al. \(2015b\)](#).

Une fois les bases de connaissances disponibles et à jour, la connaissance et les informations qu’elles contiennent sont mises à disposition d’utilisateurs humains. Or les bases de connaissances sont représentées dans des formalismes du W3C essentiellement exploitables par des informaticiens. Le troisième chapitre de mon manuscrit montre qu’il est possible de faciliter l’accès aux informations présentes sur le Web pour un utilisateur final par :

- la mise en place d’une approche visant à transformer une question exprimée en langage naturel en requête SPARQL (langage du W3C consacré à l’interrogation de bases de connaissances) de façon transparente pour l’utilisateur [Pradel et al. \(2014\)](#) ;
- la définition de méthodes mettant en correspondance les descriptions proposées dans les différentes bases de connaissances présentes sur le Web (à l’image notamment du LOD³ qui ne cesse de croître). Aligner ces bases de connaissances permet de croiser les informations qu’elles contiennent pour des besoins spécifiques [Thiéblin et al. \(2017a\)](#).

Récemment, le développement de l’Internet des Objets a accentué la place des agents logiciels sur le Web. L’Internet a en effet évolué en un environnement dans lequel les objets au sens large du terme (smartphones, montres, serveurs, objets de la vie quotidienne bardés de toutes sortes de capteurs. . .) sont globalement connectés. Le terme IoT (Internet of Things, ou Internet des objets) fait référence à l’ensemble des technologies et principes qui permettent le déploiement de réseaux d’objets et de services. Ceux-ci ouvrent de nombreuses perspectives en termes d’applications, mais se heurtent à des difficultés en ce qui concerne l’interopérabilité, la configuration ou le passage à l’échelle.

Le domaine de l’IoT bouleverse les hypothèses souvent faites par les chercheurs du Web Sémantique. Une particularité essentielle est qu’un IoT est un réseau de machines dans lequel l’humain n’est pas intégré. Par exemple, les observations recueillies par des capteurs sont transmises directement à un autre objet ou service et ne sont pas interprétables par un humain sans traitement préalable. Dans la mesure où un humain ne peut pas re-contextualiser les échanges entre objets, les échanges doivent être compris par les machines. Par ailleurs, les objets connectés aux réseaux sont pour la plupart contraints (voire très contraints) en termes de capacité de calcul et de stockage. Mettre en place de la sémantique dans un IoT impose de prendre en compte cet aspect et d’adapter les formalismes habituellement utilisés pour représenter les descriptions et intégrer du raisonnement.

Le quatrième chapitre de mon manuscrit expose en quoi les technologies du Web Sémantique peuvent être utilisées et adaptées pour prendre en compte ces différents aspects. Plus particulièrement, il décrit :

- où peuvent se placer et à quoi peuvent servir les technologies du Web sémantique dans une architecture IoT [Seydoux et al. \(2017b\)](#) ;
- un algorithme pour distribuer le raisonnement au plus près des objets qui collectent les données [Seydoux et al. \(2019a\)](#).

Mon dernier chapitre est un retour sur les dix dernières années et l’importance capitale qu’elles ont eues sur la diffusion des connaissances devenue immédiatement possible et son corollaire, l’exploitation des données personnelles des utilisateurs à des fins parfois discutables. Je trace un certain nombre de questions de recherche qui pourraient m’animer dans les années qui viennent et qui, de toute façon, seront vraisemblablement au cœur de travaux de recherche à venir.

Les travaux présentés dans ce manuscrit ont été réalisés en partie dans le cadre de 6 thèses soutenues que j’ai co-encadrées :

3. <http://lod-cloud.net/>

- **Camille Pradel** *D'un langage de haut niveau à des requêtes graphes permettant d'interroger le Web Sémantique*, Thèse de l'Université de Toulouse (Contrat Doctoral, Ministère de l'Enseignement Supérieur), soutenue en décembre 2013 et co-encadrée avec Ollivier Haemmerlé (IRIT/UT2J).
- **Anis Tissaoui** *Annotation sémantique de corpus dynamiques*, Thèse de l'Université de Toulouse (Financement sur projet ANR), soutenue en décembre 2013 et co-encadrée avec Nathalie Aussenac-Gilles (IRIT/CNRS) et Philippe Laublet (Université Paris-Sorbonne).
- **Fabien Amarger** *Vers un système intelligent de capitalisation de connaissances pour l'agriculture durable : Construction d'ontologies agricoles par transformation de sources existantes et exploitation des éléments de connaissances consensuels*, Thèse de l'Université de Toulouse (co-financement CNRS/IRSTEA) soutenue en décembre 2015 et co-encadrée avec Ollivier Haemmerlé (IRIT/UT2J), Catherine Roussey et Jean-Pierre Chanet (chercheurs à l'IRSTEA - Clermont-Ferrand).
- **Rafik Abbas** *Filtrage et agrégation d'informations vitales relatives à des entités*, Thèse de l'Université de Toulouse (financement sur projet) soutenue en décembre 2015 et co-encadrée avec Karen Sauvagnat-Pinel (IRIT/UPS) et Mohamed Boughanem (IRIT/UPS).
- **Rim Teyeb** *Annotation automatique d'éléments multimedia*, Thèse en co-tutelle avec l'Ecole Nationale d'Ingénieurs de Sfax, Tunisie (financement sur poste d'enseignant-chercheur CDD en Tunisie) soutenue en juillet 2017 et co-encadrée avec Ollivier Haemmerlé (IRIT/UT2J), Mouna Torjmen et Mohammed Ben Jemaa (Enseignants-Chercheurs à Sfax).
- **Nicolas Seydoux** *Vers une gestion intelligente des données de l'internet des objets*, Thèse de l'Université de Toulouse (Contrat Doctoral, Ministère de l'Enseignement Supérieur) soutenue en novembre 2018 et co-encadrée avec Khalil Drira (LAAS/CNRS) et Thierry Monteil (LAAS/INSA toulouse).

Ils ont également été enrichis par l'encadrement des stages post-doctoraux de Marie Chagnoux, Fabien Amarger et Mohammad Noorani Bakerally et des stages de Master de Seif Eddine Kramdi, Fabien Amarger, Akila Ghuersedine, Arona Ba, Juan Yu.

2.1 Ontologies et Bases de Connaissances

La communauté travaillant sur la représentation des connaissances s'est emparée du terme *ontologie* au cours des années 1990 afin de désigner la composante terminologique des bases de connaissances, résultant le plus souvent d'un processus spécifique de modélisation.

Afin de préciser ce que nous entendons par représentation de connaissances, il nous semble important de donner notre définition des notions de *donnée*, d'*information* et de *connaissance*, qui sont parfois assimilées. Notre interprétation de ces différentes notions reprend celle qui est proposée dans [Rowley \(2007\)](#).

Une donnée est le résultat d'une mesure. Elle peut être collectée par un capteur, une personne, ou extraite d'une base de données. Une donnée n'est pas intrinsèquement porteuse de sens ; elle n'est ni vraie, ni fausse. Sa finalité est d'être transmise, stockée et surtout traitée. Une mesure de température issue d'une station météo est un exemple de donnée.

Les données peuvent être structurées pour produire de l'information. Cette information porte alors un sens, soit pour le système, soit pour l'humain. Les relevés de température à un endroit donné, à intervalle précis et sur une période donnée sont un exemple d'information.

L'information entre dans un processus de raisonnement lorsqu'elle est reçue dans un contexte donné et interprétée par un humain et/ou par un système sur la base des connaissances qu'il possède préalablement. L'énoncé "*le chauffage doit être désactivé lorsque la température dépasse 19 degrés*" est un exemple de connaissance. Dans le domaine de l'Intelligence Artificielle, l'information sert de fondement à des inférences pouvant mener à la génération de nouvelles informations. Un système dit "intelligent" pourra inférer l'information "*le chauffage doit être désactivé sur le campus*" à partir du traitement des relevés de température et de la connaissance précédemment énoncée. Dans [Kayser \(1997\)](#), une inférence est définie comme une façon générique de désigner l'ensemble des mécanismes par lesquels des entrées (perceptives ou non) sont combinées à des connaissances préalables afin d'obtenir des comportements élaborés.

La finalité des ontologies est d'une part de fournir un vocabulaire permettant de décrire un contexte associé à la donnée et, d'autre part, de définir une sémantique pour ce vocabulaire. À partir de la sémantique des constructeurs du langage formel utilisé pour représenter l'ontologie, la connaissance est utilisée pour interpréter l'information et mener par le biais d'inférences à la génération de nouvelles informations.

Sur le Web Sémantique, l'explication du contexte est possible en définissant pour chaque

ressource potentiellement descriptible une URI¹ ou IRI². Les descriptions s'expriment dans le langage RDF à partir d'*entités* définies dans des ontologies, qui sont elles-même représentées à l'aide des formalismes RDFS ou OWL. RDF (Resource Description Framework)³ est un langage permettant de décrire formellement les ressources du Web et leurs métadonnées. Ces descriptions sont exprimées sous forme de graphes, composés eux-mêmes de *triplets*. RDFS (RDF Schema)⁴ et OWL (Web Ontology Language)⁵ sont des formalismes permettant la représentations d'ontologies de différents niveaux d'expressivité.

Les entités qui constituent les ontologies sont des *classes*, des *propriétés d'objets*, des *propriétés de données*, des *instances* et des *valeurs*. Ces entités sont définies à partir d'axiomes. L'expressivité des axiomes est limitée au fragment de logique que l'ontologie implémente (comme par exemple *SHIN*, *SHIF*, *SHOIN*, *SROIQ*, logique du premier ordre à deux variables). Les variantes de OWL implémentent différents fragments de la logique du premier ordre.

Le Web Sémantique repose sur les principes fondationnels du Web et plus précisément sur les principes du Linked Open Data (LOD) [Bizer et al. \(2009\)](#)⁶ selon lesquels :

- chaque ressource est identifiée à partir d'une IRI déréréférençable ;
- quand l'IRI est déréréférencée, une ou plusieurs représentations de la ressource sont accessibles à partir de formats et de protocoles standardisés par le W3C ;
- les ressources sont connectées les unes aux autres à partir de liens permettant la navigation.

Dans nos travaux, nous faisons une distinction entre une ontologie et :

- **un schéma de tableau** qui est un schéma plat, composé des intitulés des colonnes et est instancié par des données tabulaires ;
- **un schéma de base de données relationnelle** qui impose que les données soient organisées comme des relations suivant le modèle relationnel. Le nom de chaque relation, ainsi que que le nom des attributs et leurs types sont représentés. Ce type de schéma inclut les notions de clés étrangères et primaires.
- **un schéma orienté document** comme les DTD, schémas XML et JSON, qui définissent la structure de documents XML ou JSON. Ils incluent les éléments, les attributs et leurs types. Les éléments peuvent être simples quand ils spécifient un type de données ou complexes quand ils spécifient des éléments imbriqués.
- **des modèles conceptuels** qui incluent les thésaurus et les taxonomies (utilisés pour structurer les termes d'un domaine), les modèles entités-associations (utilisés pour abstraire le schéma d'une base de données relationnelle) et les modèles UML (utilisés pour abstraire les programmes ou les bases de données orientées objets). Les entités de ces modèles décrivent la structure interne d'objets du domaine et peuvent être organisées hiérarchiquement. Les modèles peuvent également exprimer des relations ou associations avec différentes contraintes entre les entités.

Ces différents types de représentations peuvent être utilisés pour construire ou enrichir des ontologies, comme nous le verrons dans le chapitre 3.

Plusieurs sortes de modèles de représentation de connaissances développés autour de la notion d'ontologie ont été introduits depuis les années 2000. Nous les décrivons ci-dessous et indiquons leur utilisation dans nos travaux.

1. <https://www.ietf.org/rfc/rfc3986.txt>
2. <https://www.ietf.org/rfc/rfc3987.txt>
3. <https://www.w3.org/RDF/>
4. <https://www.w3.org/TR/rdf-schema/>
5. <https://www.w3.org/OWL/>
6. <https://www.w3.org/DesignIssues/LinkedData.html>

2.1.1 Ressource Termino-Ontologique

Un des objectifs du Web Sémantique étant de favoriser l’interaction entre humains et machines, la composante lexicale que nous appelons aussi composante terminologique d’une ontologie (à distinguer de la partie terminologique ou TBox d’une base de connaissances) a, de notre point de vue, une importance capitale. En effet, une représentation n’a d’intérêt que si elle permet d’expliciter le contexte associé à des informations liées à des données, à des documents ou exprimées directement par l’utilisateur d’un système. Bien que les URI et IRI indispensables à l’identification des ressources sur le Web puissent être assez explicites pour un humain, aucune obligation d’intelligibilité n’est imposée dans la spécification. Les propriétés d’annotations définies dans différentes ontologies pallient ce problème en comblant le fossé entre la modélisation ontologique d’un domaine et le lexique de ce domaine. `rdfs : label`⁷ issue du vocabulaire RDFS, la plus répandue de ces propriétés, permet d’associer à une ressource un terme du langage utilisé pour dénoter cette ressource. La forme lemmatisée du terme est généralement représentée pour faciliter l’identification de ses occurrences par des outils d’extraction d’information dans des sources (textes, requêtes, etc.). D’autres propriétés d’annotation ont été définies, comme `dc : title`⁸, issue de l’ontologie du Dublin Core⁹, qui permet de définir le titre d’une œuvre, ou `foaf : name`¹⁰, provenant de FOAF¹¹, qui permet d’exprimer le nom d’une personne. Bien qu’ayant une signification plus précise que `rdfs : label`, ces propriétés ont approximativement le même rôle : elles permettent d’exprimer une représentation lexicale d’une ressource.

Dans le cadre de nos travaux, l’accent est mis sur l’utilisation et l’exploitation de ces propriétés, que ce soit dans le but d’améliorer un processus de recherche d’information [Abbes et al. \(2015b\)](#), d’enrichir un système de questions/réponses [Pradel et al. \(2014\)](#) ou de favoriser l’interopérabilité dans l’IoT [Seydoux et al. \(2016a\)](#).

Certains travaux, dont des travaux que nous avons menés, visent à exploiter une composante terminologique plus riche que celle qui est représentable avec ces propriétés. C’est le cas notamment des travaux qui reposent sur les *ressources termino-ontologique* (RTO). Une ressource termino-ontologique peut être définie comme un modèle conceptuel comportant une composante conceptuelle, prenant la forme d’une ontologie, et une composante lexicale, ou terminologie, servant à décrire les termes dénotant les entités de l’ontologie [Reymonet et al. \(2007\)](#). Un intérêt de tels modèles est de permettre la représentation et la description (catégories grammaticales notamment) des variantes lexicales associées à un terme. L’exploitation des termes annotant les entités est alors moins dépendante des approches d’extraction d’information en charge d’identifier les occurrences des termes dans des sources. Un autre intérêt de ces modèles est de permettre non seulement la description des termes associés aux entités de l’ontologie, mais aussi celle des occurrences des termes dans les sources, ce qui est très utile pour la recherche d’information.

Dans le cadre du projet ANR DYNAMO, nous avons considéré que la description sémantique d’un document, également appelée son *annotation sémantique*, revient à associer des occurrences de termes définis dans une RTO à des documents. Pour une collection de documents donnée, le projet Dynamo fait l’hypothèse que l’on va concevoir une RTO adaptée pour réaliser l’annotation de la collection. La RTO repose alors sur un modèle dont une représentation simplifiée et un exemple sont donnés dans la figure 2.1. Dans cet exemple, le texte “Txt52” contient une occurrence du terme T37 sous la forme “moteur” désignant une instance de “Motorisation”. Cette occurrence est une instance du terme T37 dénotant le concept Motorisation.

7. https://www.w3.org/TR/rdf-schema/#ch_label

8. <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/2012-06-14/?v=terms#terms-title>

9. <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>

10. http://xmlns.com/foaf/spec/#term_name

11. <http://xmlns.com/foaf/spec/>

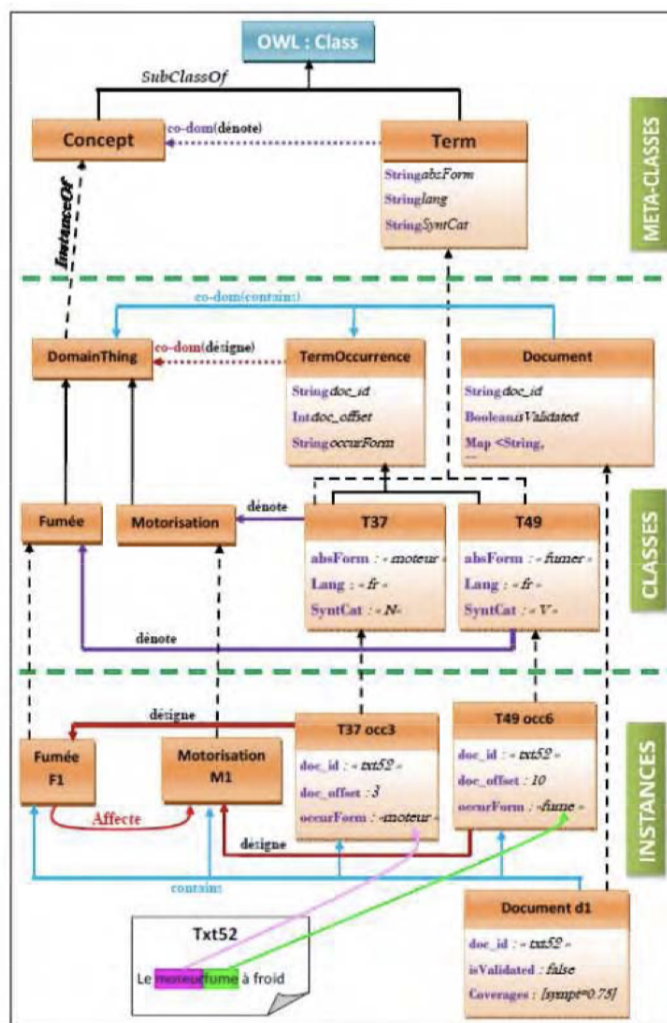


FIGURE 2.1 – Représentation simplifiée du modèle de RTO présenté dans Reymonet et al. (2007)

Afin de faciliter la conception de RTO, nous avons travaillé dans le projet ANR DAFOE Charlet et al. (2010) sur un éditeur destiné à ce type de ressources.

Initialement conçu pour représenter les thésaurus, le langage SKOS¹² permet également de représenter les termes et leurs occurrences. Depuis ces travaux, des modèles plus riches ont été proposés dans la littérature comme par exemple Buche et al. (2011) ou OntoLexlemon McCrae et al. (2017).

2.1.2 Vocabulaire partagé

Avec le développement du LOD, les termes *vocabulaire*, *vocabulaire partagé* ou *vocabulaire sémantique* sont aussi utilisés aujourd’hui pour désigner les ontologies. Comme cela est précisé par le W3C¹³, cette notion de vocabulaire fait référence aux classes, propriétés et types de données, souvent désignés comme des *entités* définies pour décrire des données. Ces vocabulaires, eux-même décrits grâce au vocabulaire défini par RDF, sont exprimés et publiés en suivant les principes du Web de données liées. Bien que “vocabulaire” et “ontologie” fassent référence aux mêmes types de représentations de connaissances, dans la pratique, les vocabulaires sont généralement peu axiomatisés.

Réutiliser un même vocabulaire ou des entités de ce vocabulaire pour décrire des données appartenant à deux bases de connaissances différentes permet de facilement lier les données.

12. <https://www.w3.org/2004/02/skos/>

13. <https://www.w3.org/standards/semanticweb/ontology.html>

Dans le cadre de nos travaux appliqués aux domaines de l’agriculture [Amarger et al. \(2016a\)](#) ou de l’IoT [Seydoux et al. \(2016a\)](#), nous avons veillé à favoriser la réutilisation de vocabulaires existants.

2.1.3 Vocabulaire partagé et lié

L’initiative du Linked Open Vocabularies (ou LOV) [Vandenbussche et al. \(2017\)](#) vise à recenser les vocabulaires correctement documentés et respectant les principes du LOD (descriptions des entités du vocabulaire à partir d’URI déréférencables, lien entre les entités d’autres vocabulaires) pour en faciliter la réutilisation.

Ancrés dans ces principes, les vocabulaires partagés peuvent en effet maintenant être qualifiés de *vocabulaires liés partagés*, car pour la plupart ils réutilisent, en les important et/ou en définissant des correspondances, d’autres vocabulaires.

L’*alignement d’ontologies* (ontology matching) est le processus qui consiste à définir des correspondances entre une ontologie source et une ontologie cible [Euzenat and Shvaiko \(2013\)](#).

Les correspondances et les alignements reposent sur la notion d’expression. Une *expression simple* est seulement composée de l’identifiant unique d’une entité (c’est à dire l’IRI d’une classe ou d’une instance sur le Web Sémantique).

Une *expression complexe* est composée d’au moins une entité sur laquelle un constructeur ou une fonction de transformation est appliqué. Un *constructeur* est un constructeur du langage dans lequel l’expression est formalisée (union, restriction de cardinalité par exemple pour OWL2). Une fonction de transformation est une fonction qui modifie la valeur d’un littéral (concaténation de chaînes de caractères, somme d’entiers par exemple).

Définition 1 Un alignement $A_{o_1 \rightarrow o_2}$ est bidirectionnel entre une ontologie source o_1 et une ontologie cible o_2 .

$A_{o_1 \rightarrow o_2}$ est un ensemble de correspondances, $A_{o_1 \rightarrow o_2} = \{c_1, c_2, \dots, c_n\}$.

Définition 2 Une *correspondance* c_i est un tuple (e_{o_1}, e_{o_2}, r, n) . e_{o_1} et e_{o_2} sont les *membres* de la correspondance. Ils correspondent à des expressions simples ou complexes d’entités appartenant respectivement à o_1 et o_2 :

- si la correspondance est **simple**, e_{o_1} et e_{o_2} sont toutes deux des expressions simples ;
- si la correspondance est **complexe**, au moins e_{o_1} ou e_{o_2} est une expression complexe ;
- r est une relation entre e_{o_1} et e_{o_2} choisie parmi équivalence (\equiv), plus général (\sqsupseteq), plus spécifique (\sqsubseteq), disjonction (\perp).
- les systèmes d’alignement assignent usuellement une valeur de confiance pour chaque correspondance, notée n .

2.1.4 Graphe de connaissances

Le terme *graphe de connaissances* a commencé à être utilisé par la société Google pour désigner l’utilisation de connaissances dans leurs approches de recherche d’information sur le Web. Ces dernières années, il est utilisé pour faire référence aux bases de connaissances suivant les technologies du Web Sémantique telles que DBpedia ou YAGO. Selon [Paulheim \(2017\)](#), un graphe de connaissances :

- décrit les entités du monde et leurs liens sous la forme d’un graphe ;
- décrit les classes possibles et leurs relations dans un schéma ;
- offre la possibilité de lier des entités données ;
- couvre différents types de domaines.

Les graphes de connaissances sur le Web Sémantique sont publiés en suivant les principes du Linked Open Data.

La base de connaissances que nous considérons dans notre travail est constituée d'une taxonomie de classes et de propriétés, des instances de classes, des littéraux et des relations exprimées entre ces entités sous la forme de triplets ; il est également possible d'assigner des étiquettes aux ressources.

2.2 De l'IoT au SWoT en passant par le WoT

Le *Web Sémantique des objets* (ou *SWoT* pour Semantic Web of Things) est situé à l'intersection des domaines de l'IoT et du Web Sémantique. Ce domaine a progressivement vu le jour après que l'IoT a intégré les protocoles et standards du Web. Nous revenons ici sur ces différents domaines.

Le terme *Internet of Things*, introduit en 1999 par Kevin Ashton [Ashton \(2009\)](#), faisait référence initialement à un réseau de tags RFID. Sa signification a largement évolué depuis et la définition proposée par l'UIT, Union internationale des télécommunications, en 2012 [itu \(2012b\)](#), correspond à la vision que nous avons à ce jour de ce terme : “[*The IoT is*] a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies”. Pour nous, IoT fait donc référence à l'ensemble des technologies et des domaines de recherche qui permettent le déploiement d'objets connectés. Par “objet”, nous entendons ce que l'UIT définit comme une “Thing” soit “an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks”. Un objet peut donc aussi bien correspondre à un objet connecté comme un capteur (capteur de température par exemple) ou un actionneur (un radiateur connecté), qu'à un service proposé par un objet (accéder à la valeur de la température ou baisser la température).

Une caractéristique forte de l'IoT est l'hétérogénéité. Cette hétérogénéité résulte tout autant de la diversité des types d'objets qui peuvent se trouver dans un système IoT que des technologies avec lesquelles ces objets sont conçus, déployés et communiquent, comme en témoignent les différentes analyses menées dans [Zanella et al. \(2014\)](#), [Barnaghi et al. \(2012\)](#), [Foteinos et al. \(2013\)](#) et [Szilagyi and Wira \(2016\)](#). D'autre part, l'intérêt grandissant pour l'utilisation d'objets connectés dans les activités humaines a entraîné l'émergence de nombreux domaines d'application pour ces technologies comme la domotique, la santé, l'agriculture, l'industrie, le commerce, etc. ce qui, là-encore, est générateur d'hétérogénéité.

La quantité croissante d'objets connectés déployés est aussi une caractéristique forte du domaine. D'après les estimations faites par Mozilla [moz \(2018\)](#), en 2020, 30 milliards d'objets¹⁴ seront connectés par des technologies machine à machine M2M. Ce chiffre est à comparer aux 900 millions d'objets connectés en 2009 [Rivera and van der Meulen \(2013\)](#). L'augmentation exponentielle du nombre d'objets corrélée aux volumes considérables de données qu'une partie d'entre eux vont produire impose de mettre en place des solutions adaptées à ces échelles.

Cependant, ces différents constats sont à pondérer par le fait que les applications reposant sur des réseaux d'objets sont limitées par la démarche suivie par l'industrie aujourd'hui, qui consiste à créer des silos en définissant des systèmes pour chaque domaine d'application [Desai et al. \(2015\)](#). Les systèmes “propriétaires” sont conçus pour des usages donnés et fonctionnent comme des boîtes noires. Pour interagir avec la majorité des objets connectés, l'utilisateur final doit utiliser une interface fournie par le constructeur. Cette démarche va à l'encontre du potentiel offert par l'IoT et accentue l'hétérogénéité des solutions qui se retrouve au niveau des applications qui interagissent avec les objets. L'utilisateur devrait pouvoir combiner les objets

14. <https://internethealthreport.org/2018/spotlight-securing-the-internet-of-things/>

connectés qu'il choisit et les développeurs d'applications devraient proposer des applications génériques qui s'adaptent à n'importe quel objet, quel qu'en soit le fournisseur.

Les principes du Web et ses standards ont été utilisés dans l'IoT à partir de l'année 2008 pour répondre au moins en partie à ces problèmes [Stirbu \(2008\)](#), [Guinard et al. \(2009\)](#). Le WoT a été défini par l'UIT dans [itu \(2012a\)](#) comme "*A way to realize the IoT where (physical and virtual) things are connected and controlled through the World Wide Web*". L'identification des objets et des données produites via des IRI et la communication entre les différentes ressources via des protocoles du Web est vue comme une façon d'uniformiser l'accès aux données et aux objets, qui se fait de façon très disparate dans les réseaux de communication IoT.

Le WoT est standardisé par un groupe de travail du W3C¹⁵. Le déploiement du WoT repose sur l'utilisation de protocoles tels que HTTP ou CoAP permettant aux objets, grâce à la négociation de contenu, d'accéder aux représentations de ressources dans un format choisi par l'objet.

Les objets sur le WoT sont identifiés via des IRI, mais les objets connectés peuvent ne pas être capables de communiquer en HTTP ou CoAP. Dans ce cas, le réseau IoT sous-jacent intègre un nœud qui joue le rôle d'intermédiaire (ou proxy), qui traduit les requêtes HTTP vers les protocoles spécifiques à l'IoT (EnOcean, Z-Wave, Phidget, Zigbeebut...). Une application IoT communique avec un réseau d'objets via un serveur, mais ce serveur n'est généralement pas connecté directement à tous les objets. Ce sont des passerelles qui sont déployées sur le réseau qui mettent en place les interfaces de communication.

Bien que le WoT favorise la communication entre objets en cherchant l'uniformisation des protocoles, il ne règle pas entièrement la question de l'interopérabilité. Un message doit pouvoir être échangé mais aussi compris par les deux machines. Dans cette optique, le SWoT a été développé à la fin des années 2000 dans l'objectif de développer l'interopérabilité sémantique au sein des réseaux IoT [Scioscia and Ruta \(2009\)](#), [Pfisterer et al. \(2011\)](#).

La notion de SWoT correspond à l'intégration des principes et technologies du Web Sémantique dans l'IoT. Nous expliquons la convergence de l'IoT et du Web Sémantique par une de leurs motivations communes consistant à déléguer le traitement de l'information aux machines. Les ontologies sont utilisées dans l'IoT pour décrire à la fois les objets des systèmes mais aussi l'environnement dans lequel ils sont déployés, ainsi que les informations produites par les systèmes.

Utiliser des vocabulaires déréréférencables pour l'IoT offre de nombreuses potentialités. L'accès aux descriptions et à leur sémantique au moment de l'exécution permet aux traitements d'être adaptés à la dynamique des réseaux IoT. La mobilité des objets implique que l'environnement du système n'est que très rarement statique. Le fait que les vocabulaires reposent sur des formalismes du Web Sémantique permet également de déployer des traitements qui ajoutent de la valeur aux informations générées et manipulées par le système en fonction du contexte dans lequel elles sont produites. Les mécanismes de raisonnement permettent d'explicitier de l'information qui est implicite au moment de la collecte des données.

Deux paradigmes de programmation existent sur le SWoT. Exploiter les ressources fournies par des serveurs sans se soucier des caractéristiques de ces machines est un intérêt de la programmation dans le nuage ou *cloud computing*. Le NIST (National Institute of Standards and Technology) définit ce paradigme dans [Mell and Grance \(2011\)](#) de la manière suivante : "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*". L'"élasticité" et l'accessibilité sur le réseau de ce type de modèles fait qu'ils sont largement utilisés pour déployer des applications pour le SWoT. Les données sont collectées par les objets et sont centralisées et traitées sur des serveurs du cloud.

15. <https://www.w3.org/WoT/WG/>

Les architectures évoluent également vers le paradigme du *Fog computing* [Bonomi et al. \(2012\)](#) ou programmation dans le brouillard, qui promeut le stockage et le traitement des données à la bordure des réseaux [Patel et al. \(2017\)](#). Le Fog computing est défini par l'open Fog consortium¹⁶ comme une “*system-level horizontal architecture that distributes resources and services [...] anywhere along the continuum from Cloud to Things*”.

Un panorama des applications relevant du SWoT et suivant l'un de ces deux paradigmes est présenté dans le chapitre 6.

16. <http://openfogconsortium.org/>

Réutilisation d'ontologies et de bases de connaissances

Sommaire

3.1	Introduction	15
3.2	Réutilisation de sources pour un besoin applicatif	17
3.2.1	Motivations et travaux existants	18
3.2.2	Transformation de modèles conceptuels guidée par une amorce de module	20
3.2.3	Fusion de bases de connaissances	22
3.2.3.1	Les candidats	23
3.2.3.2	Calcul de la confiance d'un candidat	24
3.2.4	Extension optimale	25
3.2.5	Évaluations	26
3.3	IoT-O : une ontologie modulaire de core-domain pour l'IoT	27
3.3.1	Motivations	27
3.3.2	Exigences pour une ontologie de l'IoT	28
3.3.2.1	Exigences fonctionnelles	28
3.3.2.2	Exigences non-fonctionnelles	29
3.3.3	Analyse de l'existant	29
3.3.4	Ontologies réutilisées pour IoT-O	30
3.3.5	IoT-O : une ontologie de core-domain	31
3.3.6	Utilisations d'IoT-O	33
3.4	Conclusion	33

3.1 Introduction

Depuis le début des années 2000, les nombreuses méthodologies [Aussenac-Gilles and Gandon \(2013\)](#) proposées pour guider la construction d'ontologies ont mené au développement de milliers d'ontologies couvrant de multiples domaines. Parallèlement aux avancées méthodologiques, les technologies du Web Sémantique sont maintenant suffisamment matures pour permettre la publication de ces ontologies et de données les utilisant, sur le Web de données liées. Cependant, le Web de données liées et le Web Sémantique doivent actuellement faire face au défi majeur qu'est la réutilisation effective de ces données et de ces ontologies. Identifier et réutiliser une ontologie ou un jeu de données pouvant être utiles pour un usage donné n'est pas une tâche aisée. Cette tâche est d'autant plus complexe qu'un grand nombre de ces ressources peuvent être publiées pour un même domaine.

Un point essentiel dans le processus de construction d’une ontologie est l’identification des besoins fonctionnels et non-fonctionnels auxquels elle doit répondre. Des méthodologies et bonnes pratiques ont été définies, notamment dans NeOn [del Carmen Suarez de Figueroa Baonza \(2010\)](#). NeOn propose de spécifier les *besoins fonctionnels* à partir de *questions de compétences*. Les questions de compétences sont les questions auxquelles les ontologies doivent permettre de répondre : l’idée sous-jacente est d’assurer que l’ontologie en cours de construction est “compétente” pour répondre aux requêtes qui pourraient être posées par le système auquel elle sera intégrée. Une ontologie est donc construite pour des besoins concrets, ce qui implique que, pour être réutilisée, les exigences imposées par le nouveau système dans lequel elle sera intégrée doivent être couvertes par les exigences fixées pour l’ontologie lors de sa création. Outre le fait que la documentation décrivant les exigences pour lesquelles l’ontologie a été construite est rarement disponible [Fernández-López et al. \(2019\)](#), il est souvent peu réaliste d’imaginer que la globalité de l’ontologie ait un intérêt pour une nouvelle application.

De façon analogue aux pratiques existant dans le domaine de l’ingénierie logicielle, la décomposition d’ontologies en modules permet de faciliter la réutilisation de la représentation réalisée, que ce soit pour les sous-parties d’ontologies ou de bases de connaissances. En réponse à la difficulté de maintenir et de réutiliser de “grandes ontologies” [Stuckenschmidt et al. \(2009\)](#), des *modules* sont considérés dès le processus de conception. Ils consistent à définir des ontologies plus petites mais interconnectées, répondant chacune à un sous-ensemble des spécifications ou questions de compétences que doit satisfaire l’ontologie globale. Les bonnes pratiques de conception d’ontologies reposent également sur des *patrons de conception* (Ontology Design pattern) [Gangemi \(2005\)](#). Les patrons de conception tirent profit des bonnes pratiques de modélisation et proposent une façon de représenter des notions transversales à plusieurs domaines. Construire des ontologies qui respectent des patrons de conception augmente leur réutilisabilité et favorise leur alignement, comme cela a été montré dans [Scharffe et al. \(2008\)](#).

Cependant, bien souvent, les ontologies ou modules existants ne sont pas adaptés aux nouveaux cas d’application considérés.

Différentes questions peuvent se poser quand on aborde la réutilisation :

- la tâche de réutilisation se limite-t-elle à la réutilisation à l’identique d’une ressource ?
- comment est-il possible de tirer profit, au moins partiellement, des efforts de modélisation qui ont mené à la publication d’ontologies et de bases de connaissances lorsque celles-ci répondent en partie aux besoins fonctionnels identifiés ?

La tâche de réutilisation est intuitivement perçue, notamment dans le domaine de l’informatique, comme le fait de pouvoir se servir de solutions disponibles “sur étagère”, directement exploitables pour de nouveaux besoins. Dans les travaux relatifs à la conception d’ontologies, la réutilisation consiste à prendre une ou des ontologies existantes et à les modifier pour satisfaire de nouvelles exigences [Uschold et al. \(1998\)](#). La réutilisation d’ontologies est définie dans [Bontas et al. \(2005\)](#) comme “*The process in which available (ontological) knowledge is used as input to generate new ontologies*”. La méthodologie NeOn qui inclut une activité de réutilisation dans le processus de construction (activité 2) [Fernández-López et al. \(2012\)](#), précise que la réutilisation comprend les sous-tâches d’*“élagage, enrichissement, traduction et adaptation de l’ontologie sélectionnée”*. [Katsumi and Grüninger \(2016\)](#) formalisent la tâche de réutilisation et définissent quatre opérations sur l’ontologie réutilisée : *as is*, *extraction*, *extension* et *combinaison*. L’opération *as is* consiste à réutiliser les théories axiomatisées représentées par l’ontologie telle quelle, *extraction* vise à extraire des axiomes de l’ontologie, *extension* enrichit l’ontologie par de nouveaux axiomes et *combinaison* complète l’ontologie avec les axiomes d’une autre ontologie.

Dans le cadre de nos travaux, nous avons abordé la réutilisation en suivant deux axes complémentaires. Le premier axe consiste à construire une *amorce de module* pour définir le périmètre du domaine de connaissance à représenter, ainsi que les exigences fonctionnelles auxquelles l’ontologie doit répondre, puis à réutiliser des modèles conceptuels et des bases de

connaissances existants pour spécialiser et instancier l’amorce de module. Nous avons dans ce travail-là considéré l’opération `extraction` sur les sources réutilisées puis `combinaison` dans le but de ne représenter dans la base de connaissances finale que les entités en lien avec le module et provenant de plusieurs sources.

Ces travaux ont été menés dans le cadre d’une collaboration étroite avec l’IRSTEA¹. Au plus près des besoins du domaine de l’agriculture, l’équipe Copain² est spécialisée dans la conception de systèmes d’information consacrés à la gestion agri-environnementale. Cette collaboration m’a permis d’enrichir ma vision de la réutilisation de connaissances en l’appliquant à un domaine où l’échange de connaissances est nécessaire pour soutenir le changement de pratiques des agriculteurs, en prenant notamment en compte l’impact sur l’environnement. Cette collaboration s’est concrétisée par le co-encadrement de la thèse de Fabien Amarger [Amarger \(2015\)](#) qui nous a permis de définir les travaux présentés dans ce chapitre et qui ont été notamment publiés dans [Amarger et al. \(2014\)](#), [Amarger et al. \(2015\)](#), [Amarger et al. \(2016a\)](#) et [Amarger et al. \(2016b\)](#).

Le deuxième axe selon lequel nous avons abordé la réutilisation est la construction d’ontologies pour un domaine émergent. C’est le cas de l’IoT, domaine ayant été défini au début des années 2000 [Ashton \(2009\)](#) et pour lequel l’utilisation d’ontologies est apparue très rapidement comme une solution naturelle pour faire face à l’hétérogénéité des formats, des protocoles, des plates-formes et des domaines d’applications, comme nous l’avons présenté dans le chapitre 2. Les premières ontologies constituées pour ce domaine [Avancha et al. \(2004\)](#), [Jurdak et al. \(2004\)](#), [Russomanno et al. \(2005\)](#), [Li et al. \(2015\)](#), [Hussein et al. \(2016\)](#) reposent sur des modèles conceptuels *ad hoc*. Dans d’autres travaux comme [Nachabe et al. \(2015\)](#), [Kibria \(2015\)](#) et [Pease et al. \(2017\)](#), seules des représentations graphiques des ontologies sont proposées. Un enjeu de ces dernières années a donc été de définir des ontologies possiblement réutilisables dans les différents domaines transversaux d’applications de l’IoT (la domotique, la santé, l’agriculture, etc.). Nous avons pour cela défini les exigences auxquelles une ontologie de l’IoT devait répondre, et réutilisé puis complété des modules existants pour les satisfaire. Nous avons pour ces travaux utilisé les opérations `as is` et `combinaison` sur les ontologies réutilisées.

Bien que relevant plutôt de l’ingénierie, nous avons fait le choix de présenter ici ces travaux car nous pensons que l’essor du Web Sémantique n’est possible que si les avancées théoriques s’accompagnent de production de ressources concrètes, utilisées dans des applications réelles au service des utilisateurs.

Ces travaux ont été menés dans le cadre d’une collaboration avec l’équipe SARA³ du LAAS⁴. Spécialisée dans le domaine de l’IoT, cette équipe contribue à proposer des architectures et des applications consacrées aux spécificités de la communication entre objets connectés. Leur approche orientée réseau est complémentaire de la nôtre, qui se concentre sur les données et les descriptions utiles pour la communication entre agents. Notre collaboration a notamment donné lieu au co-encadrement de la thèse de Nicolas Seydoux [Seydoux \(2018\)](#), dont l’ontologie IoT-O présentée dans ce chapitre est une contribution. Ces travaux ont notamment été publiés dans [Seydoux et al. \(2016a\)](#) et [Seydoux et al. \(2015\)](#).

3.2 Réutilisation de sources pour un besoin applicatif

Les travaux que nous présentons dans cette section ont pour finalité de construire une base de connaissances répondant à des exigences spécifiques à partir de sources existantes. Nous entendons par sources des modèles conceptuels ou des bases de connaissances, tels que nous les avons présentés dans le chapitre 2.

1. Institut national de recherche en sciences et technologies pour l’environnement et l’agriculture
2. Systèmes d’information communicants et agri-environnementaux
3. Services et Architectures pour Réseaux Avancés
4. Laboratoire d’analyse et d’architecture des systèmes

3.2.1 Motivations et travaux existants

Comme nous l’avons introduit précédemment, un module permet de répondre à un ensemble de besoins fonctionnels concernant un domaine de connaissances. Ce module peut ensuite être utilisé dans différents contextes par son importation dans des ontologies répondant à des besoins plus larges. Dans nos travaux, nous proposons de construire une *amorce de module* que nous appelons la *partie haute d’un module*. Cette représentation est construite en relation avec des experts, à partir de questions de compétences précisant les besoins fonctionnels auxquels l’ontologie doit répondre. Elle définit les notions générales (classes et propriétés) à représenter, ainsi que la sémantique de chacune d’elles pour décrire les exigences fonctionnelles. Nous cherchons ensuite à étendre ce module par spécialisation ou instantiation en exploitant des connaissances présentes dans des sources existantes.

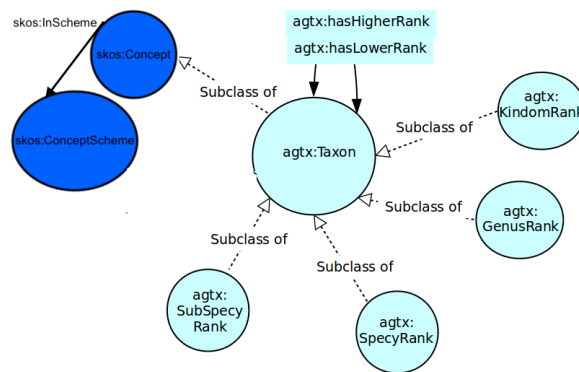


FIGURE 3.1 – Partie haute du module Agronomic Taxon

Pour illustrer ce que nous entendons par amorce ou partie haute d’un module, nous présentons dans la figure 3.1 la partie haute du module Agronomic Taxon Roussey et al. (2010). Ce module a été créé manuellement par des experts du domaine de l’agriculture à partir de questions de compétences, dans le but de capitaliser dans un système d’information reposant sur une ontologie les informations relatives à la taxonomie des organismes vivants⁵. De telles taxonomies permettent de regrouper, et ainsi de caractériser, les êtres vivants suivant différentes perspectives appelées *rangs*. Dans la modélisation choisie dans Agronomic Taxon, le rang le plus spécifique correspond à l’*espèce*. Les organismes vivants appartenant à différentes espèces peuvent être regroupés selon leur *genre*, chaque genre faisant partie d’un *règne*. Chacun des groupements d’êtres vivants est appelé *taxon*. Le *rang taxonomique* d’un taxon correspond à son type du regroupement (espèce, genre et règne).

Dans Agronomic Taxon, un taxon est représenté par une classe `agtx : Taxon` pour typer les êtres vivants qu’il décrit (comme le blé, les plantes, etc.). Ce module réutilise les concepts `skos : Concept` et `skos : ConceptScheme` du vocabulaire SKOS, qui permettent de représenter la notion générale de concept en l’adaptant à la représentation d’un Taxon et de décrire le fait que l’ensemble des taxons définis à partir du module appartiennent à un même schéma ou vocabulaire. Dans ce module, un vocabulaire existant n’a pas vocation à être réutilisé dans son intégralité, mais seulement un sous-ensemble de ses entités. La classe `agtx : Taxon` est spécialisée en différentes sous-classes permettant de représenter les différents rangs taxonomiques auxquels un taxon peut appartenir. Pour chaque taxon, les propriétés `agtx : hasHigherRank` et `agtx : hasLowerRank` permettent de définir les rangs dont il dépend.

La figure 3.2 montre un exemple de taxon défini à partir du module Agronomic Taxon.

L’utilisation de la partie haute du module est motivée par le fait qu’elle permet de prendre en compte les exigences conceptuelles qui devront être satisfaites par la base de connaissances

5. <https://www.aquaportail.com/definition-1066-taxon.html><https://fr.wikipedia.org/wiki/Taxon>

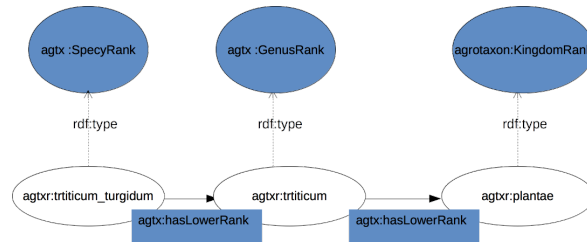


FIGURE 3.2 – Exemple de Taxons décrits à partir du module Agronomic Taxon

à l'issue du processus de réutilisation. Dans notre exemple, plusieurs modélisations peuvent être proposées pour classer les êtres vivants, comme nous le verrons dans la section 5 du chapitre 5.3.1. Pour répondre aux besoins des experts, nous allons chercher à représenter l'ensemble des taxons disponibles dans des sources existantes à partir des classes et propriétés définies dans Agronomic Taxon. Réutiliser une source selon un module va consister à spécialiser ou à instancier les éléments du module. Dans notre exemple, les sources seront utilisées pour proposer de nouvelles spécialisations des rangs taxonomiques et pour décrire des instances de taxons.

Nous positionnons maintenant nos travaux par rapport à ceux visant à réutiliser des modèles conceptuels ou des bases de connaissances pour les transformer.

La majorité des travaux visant à réutiliser des modèles conceptuels pour générer une base de connaissances cherchent à transformer l'ensemble de la source [Amarger et al. \(2013b\)](#). Or, chaque modèle conceptuel suivant des principes de modélisation spécifiques liés à l'utilisation envisagée du modèle ainsi qu'au format dans lequel il est représenté (SKOS, UML, OWL, etc.), des transformations ciblées sur les entités d'intérêt pour la réutilisation semblent plus adaptées. Certaines approches existantes comme [Villazon-Terrazas et al. \(2010\)](#) considèrent que les mêmes règles de transformation doivent être appliquées à l'ensemble des entités respectant certains critères (tous les termes plus spécifiques qu'un terme donné, toutes les occurrences de telle association, etc.). D'autres comme [Soergel et al. \(2004\)](#) proposent de spécifier les règles en fonction de caractéristiques des éléments de la source (transformation pour tel concept ayant telle association par exemple).

Dans le domaine de l'accès aux bases de données via des ontologies (Ontology-Based Data Access), différents langages ont été proposés pour définir des règles de transformation entre schémas de bases de données et RDF [Xiao et al. \(2018\)](#). [Hert et al. \(2011\)](#) en dresse un état de l'art. R2RML [Das et al. \(2012\)](#) est un format proposé par le W3C pour spécifier la façon dont les éléments du schéma de la BD doivent être exprimés en fonction de l'ontologie ciblée. Le langage RML [Dimou et al. \(2014\)](#) étend R2RML en prenant en compte d'autres types de formats pour la source, tels que XML schema, JSON, ou CSV pour les données tabulaires. Le langage xR2RML [Michel et al. \(2015\)](#) permet quant à lui d'exprimer des règles de transformation à partir d'éléments de formats mixtes dans la source, comme par exemple dans le cas où un objet JSON a pour valeur un tuple d'une BD. Récemment proposé, le langage D2RML [Chortaras and Stamou \(2018\)](#) repose sur R2RML et RML ; il permet d'établir des règles de transformation à partir d'expressions conditionnelles. Dans la veine de ces travaux, SPARQL-Generate [Lefrançois et al. \(2017\)](#) vise à étendre SPARQL 1.1 pour définir des règles de transformation pour les différents formats précédemment évoqués, mais aussi pour des données déjà représentées en RDF que l'on cherche à décrire avec un nouveau vocabulaire.

Pour ces différents langages, les règles de transformation sont définies manuellement. Établir l'ensemble des règles nécessaires à la transformation de la source est un travail fastidieux, davantage encore lorsque plusieurs sources sont à transformer.

Nous proposons de définir un algorithme de transformation prenant en compte uniquement les éléments de la source potentiellement en correspondance avec des entités du module. Le fait de ne considérer que les éléments de la source ayant un intérêt vis-à-vis de l'amorce du module réduit le nombre d'éléments à transformer. À partir de cette première étape de

transformation effectuée sur l'ensemble des sources disponibles, nous poursuivons automatiquement l'opération de combinaison en exploitant l'ensemble des bases de connaissances générées.

Les bases de connaissances ayant été transformées suivant un même module dans nos travaux, nous considérons que construire une base de connaissances à partir de bases de connaissances correspond à un processus de fusion. Nous nous appuyons sur la définition proposée par [Pottinger and Bernstein \(2003\)](#) selon laquelle :

En considérant deux modèles A et B et un alignement composé d'un ensemble de correspondances Map_{AB} établies entre ces deux modèles, le processus de fusion génère un troisième modèle représentant l'union sans doublon des modèles de A et B conformément aux correspondances de Map_{AB} .

Certains travaux comme [Dong et al. \(2014a\)](#) incluent le calcul de l'alignement dans la fusion, alors que d'autres considèrent l'alignement comme une entrée du processus [Raunich and Rahm \(2014\)](#). Comme il existe de nombreux systèmes d'alignement présentés et classifiés dans [Euzenat and Shvaiko \(2013\)](#) et [Thiéblin et al. \(2019\)](#), nous avons choisi de réutiliser des travaux existants et éprouvés.

Certaines approches concernant la fusion prennent en compte deux bases de connaissances. Les approches présentées dans [Raunich and Rahm \(2014\)](#) et [Guzmán-Arenas and Cuevas \(2010\)](#) proposent par exemple un processus asymétrique de fusion. Ce processus donne une priorité à un des modèles. Le fait de considérer une base de connaissances comme prioritaire sur une autre permet de faire un choix parmi les entités à conserver face à une situation d'ambiguïté. Puisque nous considérons que la fusion porte sur plus de deux sources (ce qui est plus réaliste à l'échelle du Web de données liées), nous généralisons les approches précédentes et priorisons les éléments à intégrer dans la base de connaissances finale en cherchant à évaluer la qualité des bases de connaissances dont les éléments sont extraits.

Certains travaux considèrent, comme nous le faisons, plusieurs sources dans leur processus de fusion. L'approche présentée dans [Dong et al. \(2014a\)](#) propose de fusionner à l'échelle du Web des triplets extraits de 4 sources non-ontologiques (textes, structures de documents HTML, tableaux et annotations posées manuellement) et d'une base de connaissances. Dans cette approche, une priorité est donnée aux triplets présents dans la base de connaissances. Une autre approche est présentée dans [Dong et al. \(2014b\)](#) : 12 techniques d'extraction d'information sont utilisées pour identifier dans des documents du Web les objets de triplets extraits d'une base de connaissances. Les résultats de ces extracteurs sont ensuite fusionnés et comparés aux valeurs des objets dans la base de connaissances. L'approche montre que le fait de favoriser l'information trouvée plusieurs fois dans les textes, tout en prenant en compte la qualité de ces textes et les caractéristiques de l'extracteur utilisé, améliore la pertinence des éléments extraits. Dans notre cas, nous ne souhaitons pas favoriser *a priori* une source par rapport aux autres, mais nous prenons en considération la qualité de chacune d'entre elles. Nous proposons plus précisément d'évaluer la qualité d'un élément d'une base de connaissances en considérant le nombre de sources dans lesquelles il apparaît, la qualité de chacune des sources, mais aussi la confiance accordée par l'aligneur lors de la mise en correspondance des éléments appartenant aux différentes sources.

À notre connaissance, aucune méthode n'exploite simultanément plusieurs bases de connaissances existantes. Or l'intérêt d'une telle approche est de comparer les éléments présents dans les différentes sources, afin de statuer sur leur réutilisation dans la base de connaissances finale. Nous faisons l'hypothèse que des éléments communs à plusieurs sources sont représentatifs du domaine et que leur présence dans la base de connaissances finale est souhaitable.

3.2.2 Transformation de modèles conceptuels guidée par une amorce de module

Le processus de transformation que nous proposons comporte quatre étapes :

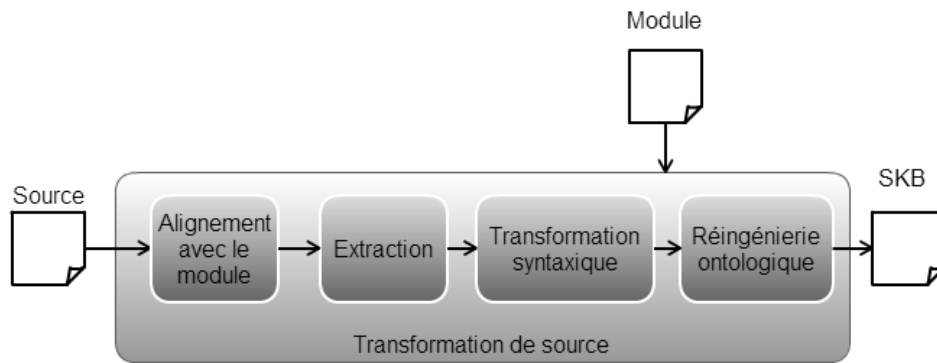


FIGURE 3.3 – Processus de transformation d’un modèle conceptuel

1. Alignement module/source : entre les classes du module et les éléments de la source, des correspondances simples d’équivalence sont établies (soit par un aligneur, soit manuellement). De cette manière, il est possible d’ancrer le processus de transformation à partir d’éléments de la source correspondant aux classes définies dans le module.
2. Extraction : en utilisant les correspondances, tous les éléments de la source et les éléments qui leur sont liés sont extraits. Par éléments liés, nous entendons les éléments liés par un ensemble d’associations pré-établies en fonction du type de la source (comme par exemple les associations UF/U, BT, NT, RT définies dans la norme de thesaurus iso :25964⁶).
3. Transformation : afin d’obtenir une description des éléments extraits à partir du vocabulaire défini dans le module, un patron de transformation est choisi par un expert et appliqué pour un sous-ensemble des éléments de la source mis en correspondance avec le module.
4. Réingénierie : l’expert du domaine a la possibilité de raffiner manuellement la base de connaissances produite.

Un patron de transformation utilisé dans l’étape 3 se présente sous la forme d’un ensemble de règles à appliquer pour transformer la source. Par exemple, dans un patron de transformation spécifique aux thesaurus⁷, la règle suivante est implémentée : tous les regroupements de termes dans un thesaurus mènent à la création d’une classe dont les termes sont représentés à partir du prédicat `rdfs : label`. L’utilisation du module dans la définition des règles de transformation permet également de spécifier la façon dont les éléments de la source doivent être décrits à partir d’entités du module. Il est par exemple possible de sélectionner un ensemble d’associations et d’indiquer qu’elles doivent mener à la description d’instances d’un certain type avec une propriété définie dans le module. Les patrons de transformation que nous avons définis reposent sur l’analyse des approches suivies dans la littérature pour extraire de la connaissance à partir des modèles conceptuels. Par exemple, pour les patrons de conception applicables à des thesaurus, nous avons utilisé les analyses faites dans [Amarger et al. \(2013b\)](#). Des patrons de transformation ont été définis pour les taxonomies, pour les thesaurus et pour les bases de connaissances.

Notre approche a été utilisée pour la transformation de trois modèles conceptuels en agriculture. Trois experts ont évalué la pertinence de chacune des entités des trois bases de connaissances obtenues en résultat de la transformation. Le détail de cette expérimentation est disponible dans [Amarger \(2015\)](#). L’analyse montre que ces éléments sont dans le périmètre des exigences fonctionnelles ciblées par le module, mais que la présence de chaque entité n’est pas toujours pertinente. L’approche présentée dans la section suivante vise justement à favoriser les entités à représenter dans la base de connaissances lorsque celles-ci peuvent être identifiées dans plusieurs bases de connaissances du domaine.

6. <https://www.iso.org/standard/53657.html>

7. <https://tinyurl.com/phss8kj>

3.2.3 Fusion de bases de connaissances

Nous proposons une approche permettant de fusionner plusieurs bases de connaissances lorsque celles-ci portent sur un même domaine et qu'elles sont le résultat d'un processus de transformation ayant permis de les structurer suivant les exigences fonctionnelles exprimées dans un même module.

Nous considérons une base de connaissances comme un graphe S orienté composé d'un ensemble de sommets et d'un ensemble d'arcs $S = (V_S, E_S)$ tels que :

- V_S est l'ensemble des sommets de S . Les sommets sont les classes, les individus et les littéraux de la base de connaissances ;
- E_S est l'ensemble des arcs de S . Les arcs sont toutes les propriétés utilisées pour lier les individus, les classes et les littéraux.

Nous appelons par la suite :

- *éléments de type sommet* les sommets des différentes bases de connaissances considérées ;
- *éléments de type arc* les arcs des différentes bases de connaissances considérées.

Notre approche génère des *candidats sommets* et des *candidats arcs* à intégrer dans la base de connaissances finale. Chaque candidat sommet (resp. arc) est issu d'un sous-ensemble des *éléments de type sommet* (resp. des *éléments de type arc*) considérés comme équivalents et dignes de confiance dans les différentes bases de connaissances.

Nous faisons l'hypothèse que la confiance accordée à un élément extrait des bases de connaissances est fonction de deux critères : (1) le nombre de bases de connaissances dans lesquelles un autre élément équivalent peut être identifié et (2) la qualité des bases de connaissances dans lesquelles ces éléments équivalents sont identifiés.

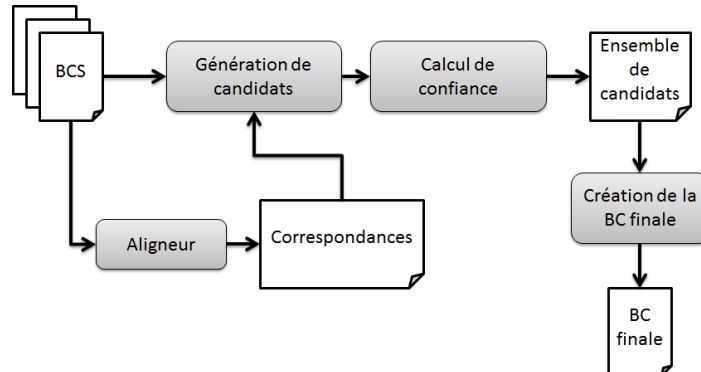


FIGURE 3.4 – Processus de fusion des bases de connaissances

Le processus se décompose en quatre étapes présentées dans la figure 3.4 :

1. **Alignement des bases de connaissances** : cette étape établit des alignements entre tous les couples de bases de connaissances considérés ;
2. **Génération de candidats** : à partir des correspondances, des candidats sont générés ;
3. **Calcul de la confiance** : un score de confiance est calculé et associé à chaque candidat ;
4. **Construction de la base de connaissances** : une sélection des candidats est effectuée à partir de leurs scores de confiance pour déterminer ceux qui appartiendront à la base de connaissances finale. Une fois qu'un candidat est sélectionné, une entité permettant de le représenter dans la base de connaissances finale est générée (choix de l'URI, choix des labels, description de la provenance, etc.).

Nous décrivons par la suite les deux phases principales du processus : la génération des candidats et le calcul de leurs scores de confiance.

En ce qui concerne la première étape d’alignement, nous considérons que la majorité des aligneurs établissent des correspondances d’équivalence entre paires de bases de connaissances [Euzenat and Shvaiko \(2013\)](#), [Thiéblin et al. \(2019\)](#). Une correspondance se représente par une arête entre deux sommets $\{oe_i \in V_{S_i} ; oe_j \in V_{S_j}\}$ pondérée par $valueE(oe_i, oe_j)$.

La correspondance satisfait les contraintes suivantes :

- $V_{S_i} \neq V_{S_j}$ car une correspondance est toujours établie entre deux sommets appartenant à des ensembles de sommets issus de bases de connaissances différentes (V_{S_i} et V_{S_j}) ;
- une correspondance est toujours établie entre deux sommets de même nature (soit des individus, soit des classes) ;
- $valueE$ est une application qui, à toute arête définie comme correspondance, associe une valeur de confiance comprise entre 0 et 1 fournie par l’aligneur telle que $valueE(oe_i, oe_j) = valueE(oe_j, oe_i)$.

Ces correspondances sont exploitées dans l’étape de génération des candidats et dans le calcul de leur score de confiance.

La création de la base de connaissances finale, dernière étape du processus, est réalisée en générant une nouvelle ressource pour chaque candidat sélectionné à partir des processus décrits ci-dessous. Nous tenons à préciser que, pour chacune des ces ressources, un lien de provenance est décrit à partir du vocabulaire Prov-O⁸ de façon à indiquer de quelle source les entités composant le candidat sont extraites. Plus de détails sur cet aspect sont donnés dans [Amarger \(2015\)](#).

3.2.3.1 Les candidats

Deux types de candidats sont générés : les candidats sommets et les candidats arcs.

Candidat sommet Les candidats sommets sont générés en exploitant les correspondances établies entre éléments de type sommet.

Un candidat sommet $CandS = (V_{CandS}, E_{CandS}, valueE)$ est un graphe non-orienté connexe dont les sommets sont des éléments de type sommet de même nature (classe ou instance) provenant de bases de connaissances différentes et dont les arêtes sont les correspondances issues des alignements entre les n bases de connaissances.

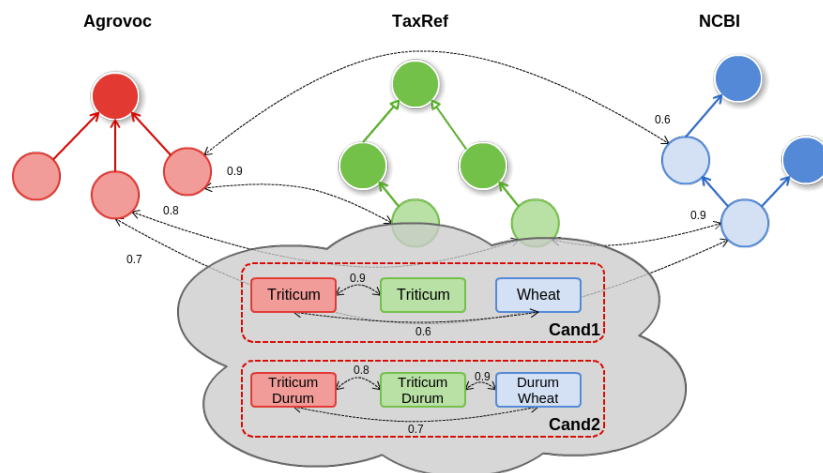


FIGURE 3.5 – Exemple de candidats sommets

La figure 3.5 présente deux candidats liant des individus issus de trois bases de connaissances. Les deux candidats représentent donc des entités potentielles de la base de connaissances finale, chaque élément de type sommet extrait des trois bases de connaissances étant représenté dans la figure à partir d’un de ses labels.

8. <https://www.w3.org/TR/prov-o/>

Candidat arc Un candidat arc $CandA = (V_{CandA}, E_{CandA})$ est un graphe biparti orienté étiqueté. Un candidat arc se construit à partir d’au moins un candidat sommet. Un candidat arc peut lier deux candidats sommets, comme le candidat *arc2* de la figure 3.6. Un candidat arc peut aussi lier un candidat sommet à un sommet du module comme le candidat *arc1*. Tous les arcs de E_{CandA} sont étiquetés par la même étiquette d’arc. Dans l’exemple de la figure 3.6, les arcs du candidat *arc1* sont étiquetés par *instanceOf*.

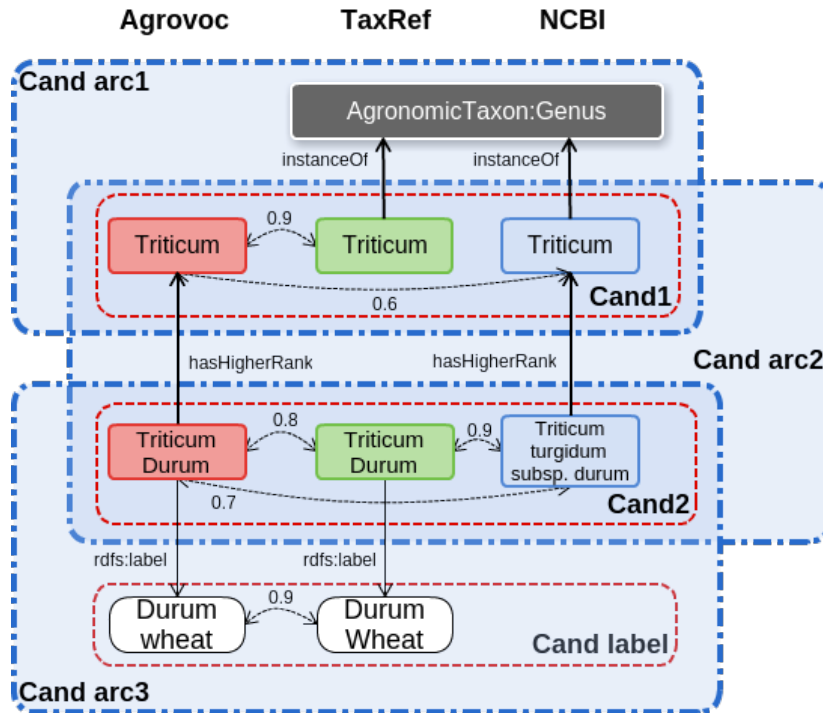


FIGURE 3.6 – Exemple de candidats arcs

3.2.3.2 Calcul de la confiance d’un candidat

Une fois les candidats générés, nous leur affectons un score de confiance établi à partir de deux critères. Le premier critère vise à favoriser les candidats contenant le plus grand nombre de sommets issus de différentes bases de connaissances (autrement dit identifiés par l’aligneur comme étant équivalents). Le deuxième critère cherche à prendre en compte la qualité des bases de connaissances dont les éléments sont issus.

Nous avons défini trois fonctions de calcul de confiance pour un candidat, que nous avons comparées lors de l’étape d’évaluation. Ces fonctions sont présentées en détail dans [Amarger et al.](#). Nous en donnons ici une présentation intuitive.

Fonction Trust Likelihood Inspirée des fonctions de vraisemblance, cette fonction évalue la confiance d’un candidat en calculant le ratio entre le nombre de bases de connaissances impliquées dans le candidat et le nombre total de bases de connaissances considérées dans le processus de fusion. Cette fonction ne prend en compte que l’ensemble des sommets du candidat V_{CandS} et le nombre de bases de connaissances alignées n .

Fonction Trust Degree La fonction Trust Degree prend en compte non seulement le nombre d’éléments composant un candidat mais aussi le degré de fiabilité de la correspondance qui a mené à la prise en compte de l’élément dans le candidat. Une variante de cette fonction a également été proposée pour prendre en compte le fait que le score de confiance d’un candidat arc peut être influencé par la confiance des candidats sommets reliés par ce candidat arc.

Fonction Trust Choquet Nous cherchons également à prendre en compte la qualité des bases de connaissances dont les sommets d'un candidat sont issus en donnant une importance relative à la qualité d'une base de connaissances par rapport à celles des bases de connaissances dont les autres éléments sont extraits. Par exemple, un élément extrait d'une base de connaissances jugée de bonne qualité aura plus d'importance dans un candidat composé d'éléments issus de bases de connaissances de moins bonne qualité que dans un candidat dont les éléments sont extraits de bases de connaissances dont la qualité est comparable.

Nous avons pour cela proposé une fonction reposant sur l'intégrale de Choquet, utilisée pour la prise de décision sur un ensemble de critères [Grabisch and Roubens \(2000\)](#). Elle permet de pondérer l'intérêt de chaque sous-ensemble de critères au lieu de pondérer chaque critère indépendamment des autres, comme le ferait une somme pondérée.

3.2.4 Extension optimale

Les candidats générés par la méthode précédente sont composés d'éléments extraits de différentes bases de connaissances identifiés comme équivalents par un aligneur.

Nous souhaitons prendre en considération le fait que lors d'un processus classique d'alignement simple entre deux bases de connaissances BC_A et BC_B , plusieurs correspondances d'équivalence peuvent être proposées pour une entité de BC_A . Or il est peu vraisemblable que l'entité de BC_A soit équivalente à l'ensemble des entités identifiées dans BC_B . Le processus d'alignement génère cet ensemble de correspondances comme étant des correspondances potentielles dont la valeur de confiance est fournie par l'aligneur. Dans les travaux présentés dans cette section, nous faisons l'hypothèse que lorsqu'une entité de BC_A est mise en correspondance avec n entités de BC_B , $n - 1$ correspondances ne sont pas pertinentes. Ceci implique qu'un élément d'une base de connaissances ne peut pas se trouver dans différents candidats qui seront retenus pour faire partie de la base de connaissances finale. Nous allons chercher à identifier dans l'ensemble des candidats générés à l'étape précédente les candidats qui partagent au moins un élément commun. Nous appelons ces candidats des *candidats incompatibles*. Nous générons ensuite des sous-ensembles de candidats compatibles. Nous appelons ces sous-ensembles des *extensions*. Finalement, nous identifions l'extension qui permet de conserver un maximum de candidats.

La figure 3.7 donne un exemple de candidats incompatibles.

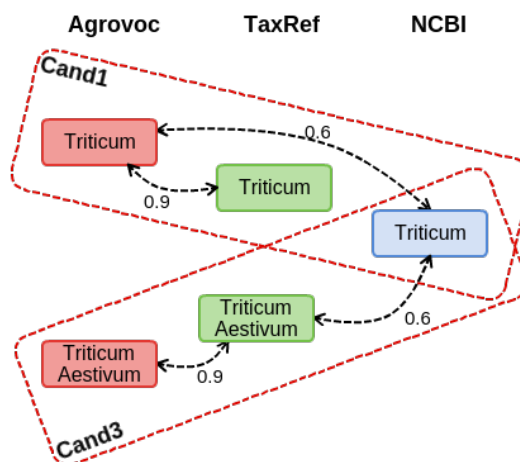


FIGURE 3.7 – Exemple d'incompatibilité entre deux candidats

Nous définissons le graphe non-orienté qui représente l'ensemble des incompatibilités $G_{inc} = \{V_{inc}, E_{inc}\}$ tel que V_{inc} sont des candidats et E_{inc} des arcs. Le graphe complémentaire à ce graphe (le graphe de compatibilité), liant uniquement les candidats qui ne sont pas incompatibles, est ensuite généré.

Nous cherchons finalement à générer les cliques maximum du graphe de compatibilité pour générer les extensions. Chaque clique est un sous-ensemble de candidats compatibles. Pour résoudre le problème classique d'énumération de cliques maximales (Maximal Clique Enumeration), plusieurs approches existent, dont notamment celle présentée dans [Bron and Kerbosch \(1973\)](#) pour laquelle des optimisations ont été proposées, comme dans [Tomita et al. \(2006\)](#) et [Eppstein and Strash \(2011\)](#). Dans notre approche, nous ne cherchons pas à générer l'ensemble des cliques possibles mais à trouver l'extension optimale. Pour cela, nous procédons en deux temps. Nous utilisons le solveur GLPK⁹, qui met en œuvre l'algorithme de Branch and Bound en utilisant comme fonction objective le choix de l'extension contenant le plus de candidats. Nous obtenons ainsi une extension parmi toutes les extensions possibles. Nous demandons ensuite à un expert, via une interface spécifique, de valider chacun des candidats de l'extension proposée et nous ajoutons itérativement le résultat de la validation comme contrainte à notre solveur pour générer une nouvelle extension.

L'ajout de contraintes au problème permet de converger vers une solution optimale, autrement dit vers une extension dont tous les candidats sont validés.

Le temps nécessaire pour la validation par l'expert des candidats est, dans le pire des cas, égal en nombre d'interactions au nombre de candidats générés. En d'autres termes, dans le pire des cas, l'expert aura à valider tous les candidats s'ils sont tous compatibles les uns avec les autres. Dès qu'une incompatibilité apparaît, ce nombre d'interactions est fortement diminué.

3.2.5 Évaluations

Les travaux présentés dans cette section ont été implémentés dans le système Muscka¹⁰ et reposent sur l'utilisation de l'aligneur LogMap¹¹. L'évaluation n'a pas été une tâche facile car il n'existe pas à notre connaissance de jeux de données composés de différentes bases de connaissances dont la fusion a été validée pour une application donnée. Nous avons donc mis en place deux cadres expérimentaux que nous avons entièrement conçus. L'un d'eux porte sur la conception d'une base de connaissances répondant à un besoin réel (besoin identifié par nos partenaires de l'IRSTEA) et consistant à représenter les taxons du blé. Ce cadre d'évaluation a été utilisé pour illustrer les différentes contributions présentées dans cette section. Trois experts ont été sollicités pour évaluer, via une interface Web que nous avons conçue, la pertinence des éléments de 3 modèles conceptuels pour la finalité visée. Nous avons ensuite évalué notre approche en analysant si les candidats retenus dans la base de connaissances finale avaient été générés à partir d'éléments jugés pertinents par les experts. L'autre cadre expérimental a été défini à partir de la tâche QA4OA (Question Answering for Ontology Alignment) de la campagne OAEI (Ontology Alignment Evaluation Initiative) 2014¹². Cette tâche n'a pas pour objectif d'évaluer les systèmes d'alignement en comparant les correspondances générées mais en évaluant la capacité des systèmes à fournir des alignements qui, utilisés dans un processus de reformulation de requêtes SPARQL, permettent de retrouver des ressources jugées préalablement pertinentes. La tâche fournit 7 bases de connaissances, 18 requêtes et des alignements de référence entre chaque paire de bases de connaissances. Pour construire la partie haute du module ontologique, nous avons considéré les 18 requêtes fournies dans la tâche comme définissant les besoins fonctionnels auxquels la base de connaissances finale devait répondre. Nous avons ensuite utilisé les 7 bases de connaissances et les paires d'alignements fournies comme entrées de notre approche et nous avons comparé le contenu de la base de connaissances obtenue par notre approche avec la liste des ressources jugées pertinentes pour chaque requête. Les détails de ces deux évaluations sont donnés dans [Amarger \(2015\)](#). Nous donnons ici les principaux points à retenir de ces évaluations :

9. <https://www.gnu.org/software/glpk/>

10. <https://github.com/Murloc6/Muskca>

11. <http://www.cs.ox.ac.uk/isg/projects/LogMap/>

12. <https://www.cs.ox.ac.uk/isg/projects/Optique/oa4qa/.dde>

- favoriser les candidats dont les éléments sont présents dans le plus de bases de connaissances est pertinent dans la mesure où il existe un certain consensus dans le domaine sur le type de candidats considérés. Par exemple, dans le cas des taxons, les instances sont extraites avec une F-mesure proche de 1. Cependant, en ce qui concerne le typage des taxons par leur rang taxonomique, l’approche est moins fiable car les bases de connaissances considérées ne représentent pas la même information ;
- prendre en compte les valeurs de confiance des correspondances dans le calcul du score d’un candidat améliore la fiabilité du score du candidat ;
- le score d’un candidat arc est plus représentatif lorsqu’il est calculé en fonction du score des candidats sommets qu’il relie ;
- prendre en compte à la fois la qualité de la base de connaissances et son importance localement pour le candidat permet d’améliorer le score d’un candidat, même si cette fonction nécessite de nombreux paramétrages. Cette considération est à relativiser en fonction du type de candidat. Une extension possible de ces travaux consisterait à évaluer la qualité d’une source par type d’éléments (classe, individu, relations spécifiques comme `rdfs : labels`) ;
- l’approche proposée pour générer l’extension optimale de candidats permet de diminuer de 40% le nombre de candidats à évaluer comparativement à une validation manuelle de l’ensemble des candidats ;
- l’utilisation du score calculé à partir de la fonction *trust_choquet* dans un processus de validation des candidats d’une extension a permis de diminuer de moitié le nombre d’interactions nécessaires par un expert pour obtenir une extension optimale. Ceci permet de montrer l’intérêt du calcul de confiance d’un candidat et la considération des incompatibilités dans le processus de validation.

3.3 IoT-O : une ontologie modulaire de core-domain pour l’IoT

Cette section décrit des travaux dans lesquels la tâche de réutilisation n’a pas consisté à extraire des éléments de sources exprimées dans des formalismes divers mais à réutiliser ou à combiner des ontologies existantes, dans l’optique de rendre l’ontologie résultante plus facilement réutilisable.

Ces travaux ont été réalisés dans l’objectif de représenter la connaissance nécessaire afin de faciliter l’interopérabilité sur des réseaux d’objets connectés. Dans ce contexte, les ontologies ont pour but de fournir un vocabulaire pour décrire les objets et les données qu’ils produisent ainsi que l’environnement dans lequel ils évoluent et sur lequel ils agissent.

3.3.1 Motivations

Comme nous l’avons introduit dans le chapitre 2, l’interopérabilité est une problématique centrale de l’IoT et a motivé l’apparition du SWoT. La notion d’interopérabilité se décompose en différents niveaux, chacun dépendant du précédent et illustré ici pour l’IoT [Gyrard et al. \(2015\)](#) :

- **l’interopérabilité technique** impose que les objets soient capables de communiquer en échangeant des messages à partir de mêmes protocoles, comme par exemple Bluetooth, WiFi ou Ethernet ;
- **l’interopérabilité syntaxique** nécessite que les messages échangés par les objets soient exprimés dans un même format (JSON ou XML par exemple) pour qu’ils soient en capacité de les manipuler. Ceci passe généralement par la maîtrise du schéma des données par l’objet ;

- **l’interopérabilité sémantique** nécessite que les objets associent la même signification aux termes des messages échangés. Les objets sont alors capables de partager une certaine “compréhension” du message.

Les standards sont reconnus comme une façon de favoriser l’interopérabilité dans les domaines technologiques où différentes parties prenantes doivent se mettre d’accord pour faire face à la diversité des protocoles et formats proposés. Centrés initialement sur l’interopérabilité syntaxique, les standards de l’IoT tels que oneM2M¹³ et le WoT du W3C¹⁴ intègrent depuis quelques années l’interopérabilité sémantique à leurs préoccupations. Les membres de ces deux groupes ont écrit conjointement en 2016 un article relatif à l’intérêt de considérer ce niveau d’interopérabilité dans l’IoT [Murdock and al. \(2016\)](#). Même si cette initiative est à relativiser compte tenu du nombre de standards existant dans l’IoT¹⁵, elle marque un certain engouement pour le SWoT. La première étape dans la mise en place de l’interopérabilité sémantique est la conception d’ontologies adaptées.

Le début des années 2010 a été marqué par la proposition d’ontologies dans différents travaux académiques. Malheureusement, il est à noter qu’une partie d’entre elles ne respectent pas les bonnes pratiques définies dans le domaine du Web Sémantique. Les représentations utilisées par exemple dans les premiers travaux [Avancha et al. \(2004\)](#), [Jurđak et al. \(2004\)](#), [Russomanno et al. \(2005\)](#), mais aussi dans des travaux plus récents comme [Li et al. \(2015\)](#), [Hussein et al. \(2016\)](#) et [Wang et al. \(2017\)](#) reposent sur des modèles conceptuels *ad hoc*. Dans d’autres travaux, seules des représentations graphiques des ontologies sont proposées. Dans [Nachabe et al. \(2015\)](#), [Kibria \(2015\)](#) ou encore [Pease et al. \(2017\)](#), les ontologies ne sont pas représentées dans les formalismes du W3C et ne sont pas disponibles en ligne. Dans l’ensemble de ces travaux, la représentation proposée n’est pas directement réutilisable.

Dans l’idée de pouvoir répertorier un maximum d’ontologies consacrées à l’IoT en intégrant également ces initiatives qui ne respectent pas l’ensemble des critères définis pour le LOV, le LOV4IoT¹⁶ a été proposé [Gyrard et al. \(2015\)](#). Ce portail répertorie les ontologies en les classifiant par domaines d’application. La volonté est de favoriser leur réutilisation.

Pour dresser un état des lieux de la capacité des ontologies existantes à répondre aux besoins d’interopérabilité sémantique, nous avons défini, par le biais de la méthodologie NeOn, les exigences auxquelles devait répondre une ontologie pour être réutilisable dans un système IoT, indépendamment du domaine d’application du système.

Nous avons défini deux types d’exigences : des *exigences fonctionnelles*, correspondant à la nature de la connaissance que l’ontologie doit représenter et des *exigences non-fonctionnelles*, faisant référence à la structure de l’ontologie et à l’approche de conception suivie.

3.3.2 Exigences pour une ontologie de l’IoT

3.3.2.1 Exigences fonctionnelles

Les besoins ont été définis par l’analyse du domaine de l’IoT. Pour être réutilisable dans ce domaine, l’ontologie doit couvrir les différentes notions-clés qui permettent de décrire les objets, les systèmes et l’environnement dans lequel ils évoluent. Nous distinguons principalement les notions :

- d’**objets connectés** ou d’**agents logiciels** qui constituent les deux composants principaux d’un système d’IoT, chacun pouvant correspondre à des objets physiques ou virtuels. Ces éléments sont principalement de deux types, listés ci-dessous :
 - les **capteurs** sont des objets capables d’acquérir des données, une **observation** décrit le contexte d’acquisition de cette donnée ainsi que la donnée elle-même ;

13. <http://www.onem2m.org/>

14. <https://www.w3.org/WoT/WG/>

15. <https://blogs.eclipse.org/post/benjamin-cabe/key-trends-iot-developer-survey-2018>

16. <https://lov4iot.appspot.com/?p=ontologies>

- les **actionneurs** sont des objets qui sont capables d’agir sur le monde physique et la notion d’**action** représente ce qu’ils sont capables de faire. Ces concepts représentent la connaissance que le système peut avoir sur ses propres capacités à agir sur l’environnement et à le faire évoluer ;
- de **service** : les objets connectés peuvent être vus comme des fournisseurs et des consommateurs de services. Les services déployés sur les objets ou utilisés par les objets font partie de leurs caractéristiques ;
- d’**énergie** : les objets connectés effectuent des traitements. La plupart de ces objets étant physiques, une modélisation complète du système doit inclure la description de leur consommation d’énergie ;
- de **cycle de vie** : que ce soient les données, les objets ou les services, les composants de l’IoT suivent un cycle de vie à des échelles différentes. Les objets sont allumés ou éteints, les services déployés ou mis à jour, les données peuvent être “périmées”... L’évolution de leur état représentée à partir du cycle de vie est un élément important pour les systèmes IoT.

3.3.2.2 Exigences non-fonctionnelles

Afin de favoriser sa réutilisabilité, il est important selon nous de mettre l’accent sur les caractéristiques suivantes de l’ontologie :

- **la modularisation** qui est une notion essentielle à prendre en compte pour faciliter la réutilisabilité de l’ontologie. Ce critère est particulièrement important dans le domaine de l’IoT car les systèmes IoT sont déployés dans différents domaines d’applications. Capturer la connaissance utile pour chacun d’entre eux dans une seule ontologie est impossible. Une ontologie reposant sur différentes ontologies en fonction de besoins spécifiques de l’application visée par le système est donc plus adaptée ;
- **l’utilisation de patrons de conception** favorisant, comme nous l’avons déjà mentionné, la réutilisation de bonnes pratiques de modélisation ainsi que l’alignement ;
- **la réutilisation de vocabulaires existants** qui, lorsque la description d’une entité dans un vocabulaire existant correspond à celle souhaitée, évite de rédéfinir certaines notions et favorise l’interopérabilité. En effet, les observations ou les objets dont la description est fondée sur un même vocabulaire seront interopérables même si leurs descriptions ont été générées par des systèmes différents ;
- **l’alignement à des ontologies de haut niveau** définissant les concepts abstraits indépendamment des domaines. Elles permettent d’articuler des ontologies de domaines différents ce qui est intéressant pour des domaines vastes comme celui de l’IoT ;
- **la prise en compte des critères de validation du LOV**, permettant de partager des vocabulaires : lorsqu’une ontologie y est publiée, sa visibilité est augmentée et ses critères de validation sont des garanties pour sa réutilisabilité.

3.3.3 Analyse de l’existant

Les exigences préalablement énoncées ont été analysées sur les ontologies existant en 2016.

Il s’agit de SSN¹⁷, SAREF¹⁸, iot-ontology¹⁹, IoT-lite²⁰, Spitfire²¹, IoT-S²², SA²³ et l’ontologie de oneM2M²⁴. Ces ontologies de l’IoT ont été sélectionnées car des informations

17. <http://purl.oclc.org/NET/ssnx/ssn>

18. <http://sites.google.com/site/smartappliancesproject/ontologies>

19. <http://ai-group.ds.unipi.gr/kotis/ontologies/IoT-ontology>

20. <http://iot.ee.surrey.ac.uk/fiware/ontologies/iot-lite>

21. <http://sensormeasurement.appspot.com/ont/sensor/spitfire.owl>

22. <http://personal.ee.surrey.ac.uk/Personal/P.Barnaghi/ontology/OWL-IoT-S.owl>

23. http://sensormeasurement.appspot.com/ont/sensor/hachem_onto.owl

24. http://www.onem2m.org/ontology/Base_Ontology/

	Actuator	Action	Service	Sensor	Observation	Energy	Lifecycle	Device	Software agent
iot-ontology	*	*	**	(**)	(**)		(*)	(**)	**
saref	*	*	**	*		**		**	**
OWL-IoT-S			(**)	(**)	(**)		(*)	(**)	
SA	*		*	(**)	(**)	(**)	(**)	(**)	
iot-lite	*		*	(*)				(*)	
spitfire				(*)	(*)	**		(*)	
ssn				**	**		*	**	
oneM2M			**					*	
IoT-O	**	**	(**)	(**)	(**)	(**)	(**)	(**)	*

TABLEAU 3.1 – Couverture des notions-clés de l’IoT dans les principales ontologies existant en 2016

	Structured by ODP	Modular	Reuses external ontologies	Aligned with upper ontologies	One the LOV	Available online
iot-ontology			*	**	N	Y
saref		**	*		Y	Y
OWL-IoT-S	(*)	*	**	*	N	Y
SA	(*)	*	**	**	N	N
iot-lite					N	Y
spitfire			*	**	Y	N
ssn	**	**	*	**	Y	Y
oneM2M					N	Y
IoT-O	(**)	**	**	**	Y	Y

TABLEAU 3.2 – Réutilisabilité des ontologies de l’IoT

les concernant étaient disponibles sur le Web ou sur le LOV4IoT en 2016. Les ontologies liées à des domaines d’application de l’IoT (domotique, agriculture, smart cities...) n’ont pas été considérées, car nous les avons jugées trop spécifiques.

Le tableau 3.1 résume l’analyse faite sur ces ontologies du point de vue des exigences conceptuelles. * signifie que le concept est superficiellement représenté (spécialisation de gros grain, peu voire pas de propriétés), ** indique que l’exigence est couverte, (*) et (**) indiquent que ce n’est pas l’ontologie elle-même mais une ontologie importée par l’ontologie analysée qui couvre l’exigence. Nous avons pu observer qu’aucune ontologie ne couvre l’ensemble des concepts. Le niveau de détails sur chaque concept est variable. Un élément notable est que les notions d’actionneur et d’action, pourtant au cœur des systèmes d’IoT, ne sont que partiellement décrites. SAREF et iot-ontology se limitent à représenter le concept d’actionneur comme un objet qui modifie une propriété. Ceci est beaucoup moins expressif que ce qui est représenté pour les capteurs, notamment dans SSN qui permet d’exprimer le fait qu’un capteur observe une propriété, mais aussi qu’un capteur génère des observations qui peuvent elles-mêmes être décrites. Ce constat nous a amenés à proposer une ontologie pour représenter les actionneurs et à définir une ontologie couvrant l’ensemble des concepts-clés.

Le tableau 3.2 montre que les bonnes pratiques du Web Sémantique ne sont que partiellement respectées par ces ontologies. La réutilisation de vocabulaires existants est rare, à l’exception de SSN qui, portée par le W3C et référencée sur le LOV, bénéficie d’une grande visibilité. Dans la majorité des ontologies, les concepts-clés présents sont redéfinis et mis en correspondance avec les autres ontologies du domaine. Pour SAREF, par exemple, au moment de notre étude, les correspondances étaient définies dans un document texte, ce qui ne facilitait pas leur réutilisation.

3.3.4 Ontologies réutilisées pour IoT-O

Face au constat qu’aucune ontologie ne couvrait les exigences que nous avons identifiées, nous avons proposé l’ontologie IoT-O²⁵. Suivant la méthodologie NeOn, les ontologies existantes couvrant les notions-clés à représenter ont été analysées. Nous avons choisi de réutiliser :

25. <http://www.irit.fr/recherches/MELODI/ontologies/IoT-O.owl>

- SSN²⁶ qui, comme nous l’avons déjà évoqué, est largement utilisée et représente clairement les notions de capteur et d’observation ;
- MSM²⁷, une ontologie consacrée à la représentation de services permettant de représenter les services REST et WSDL (contrairement à OWL-S²⁸) ;
- PowerOnt, une ontologie également utilisée par SAREF pour représenter la consommation énergétique des objets ;
- Lifecycle²⁹, qui est un vocabulaire permettant de représenter les états des machines. Nous l’avons ensuite étendu dans l’IoT-lifecycle³⁰, ontologie contenant des classes et des propriétés propres à l’IoT ;
- DUL³¹, une ontologie de haut niveau spécialisée pour définir les concepts-clés.

Aucune ontologie ne couvrant précisément le domaine des actionneurs, nous avons conçu et développé l’ontologie SAN³² qui permet de définir ce concept de la même façon que SSN permet de le faire pour un capteur. Les actionneurs sont des objets qui transforment un signal donné en entrée en une action sur l’environnement à l’exact inverse des capteurs qui réagissent à une action sur l’environnement en produisant un signal. SAN repose sur le patron de conception AAE³³, que nous avons proposé en nous inspirant de SSO, patron décrit dans Janowicz and Compton (2010). La figure 3.8 montre une représentation de AAE et SSO. SSN représente l’état du monde à partir d’un stimulus qui est converti par le capteur en observation abstraite. SAN est complémentaire : elle modélise la transformation abstraite faite par l’actionneur à partir de ses effets sur le monde réel, menant à la représentation de l’évolution que l’actionneur apporte à son environnement.

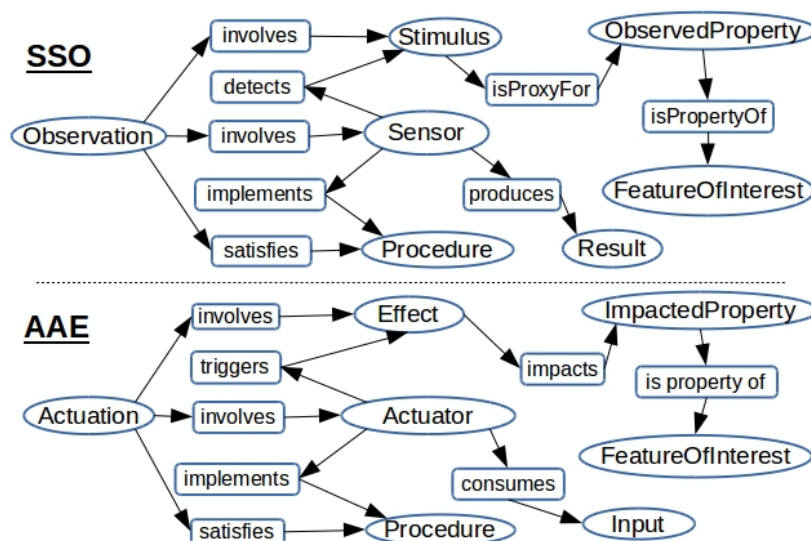


FIGURE 3.8 – Les patrons de conception SSO et AAE, structurant respectivement les ontologies SSN et SAN

3.3.5 IoT-O : une ontologie de core-domain

IoT-O, l’ontologie de core-domain que nous proposons, est composée de plusieurs modules. Son architecture est présentée dans la figure 3.9. Dans cette figure, les ressources créées appa-

26. <http://purl.oclc.org/NET/ssnx/ssn>

27. <http://iserve.kmi.open.ac.uk/ns/msm>

28. <https://www.w3.org/Submission/OWL-S/>, spécialisée dans la représentation de WSDL services

29. <http://vocab.org/lifecycle/schema>

30. <https://www.irit.fr/recherches/MELODI/ontologies/IoT-Lifecycle>

31. <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>

32. <https://www.irit.fr/recherches/MELODI/ontologies/SAN>

33. <http://ontologydesignpatterns.org/wiki/Submissions:Actuation-Actuator-Effect>

raissent en rouge, les noms de ressources modifiées sont soulignés et les flèches expriment les dépendences. Les flèches en trait plein sont des imports, les flèches en pointillés correspondent à des concepts réutilisés sans import.

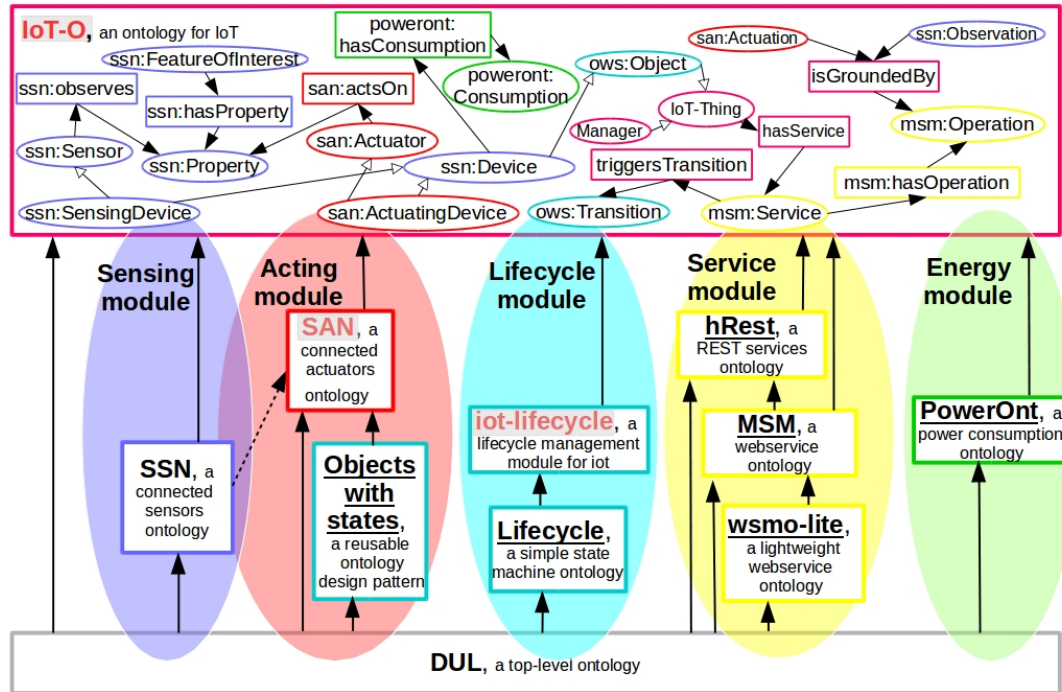


FIGURE 3.9 – Aperçu de l’architecture d’IoT-O

Cinq modules ont été développés pour l’ontologie IoT-O :

- **Sensing** décrit les données collectées par le système. Ses classes principales sont définies dans SSN : `ssn : Sensor` et `ssn : Observation`. `ssn : Device` et les propriétés (`ssn : Operating-Range`, `ssn : Deployment...`) permettent de décrire le capteur ;
- **Acting** décrit la façon dont le système peut interagir avec le monde physique. Ses classes principales sont définies dans SAN : `san : Actuator` et `san : Actuation`. Le module réutilise également les classes SSN qui ne sont pas spécifiques aux capteurs comme `ssn : Device` ;
- **Lifecycle** représente le cycle de vie des objets et leurs usages. Ses classes principales sont `lifecycle : State` et `lifecycle : Transition` ;
- **Service** représente les interfaces des services Web. Ses classes principales sont définies dans MSM : `msm : Service` et `msm : Operation`. Les services produisent et consomment des `msm : Messages`, et les services RESTful sont décrits avec l’ontologie hRest³⁴ ;
- **Energy** : ce module est décrit à partir de la classe `poweront : PowerConsumption` de PowerOnt ; il permet d’exprimer les profils de consommation.

L’ontologie IoT-O³⁵ lie les différents modules qui la composent à partir de classes et relations. Comme la plupart des entités sont dans les modules, elle définit 14 nouvelles classes (sur 1126 en incluant tous les modules), 18 nouvelles object properties (sur 249) et 4 nouvelles data properties (sur 78). La classe principale de IoT-O est `iot-o : IoT_Thing`, qui peut être soit un `ssn : Device` soit un `iot-o : SoftwareAgent`. La consommation d’énergie est liée à `ssn : Device`

34. <http://www.wsmo.org/ns/hrests>

35. <http://www.irit.fr/recherches/MELODI/ontologies/IoT-0.owl>

par l'association lifecycle : State et poweront : PowerConsumption. `iot-o : IoT_Thing` est un fournisseur de `msm : Service`, et `msm : Operation` peut avoir `iot-o : ImpactOnProperty` sur `ssn : Property`, ce qui permet de lier les concepts abstraits de services à l'environnement physique.

3.3.6 Utilisations d'IoT-O

Respectant l'ensemble des critères requis, l'ontologie IoT-O est disponible sur le LOV ³⁶.

IoT-O a été utilisée dans plusieurs contextes. Elle a tout d'abord été utilisée pour mettre en place la gestion intelligente d'un appartement connecté. L'approche mise en œuvre est présentée dans [Seydoux et al. \(2016a\)](#). Elle est également utilisée pour décrire automatiquement et quotidiennement sur un portail spécifique les données produites par plusieurs centaines de capteurs installés dans le bâtiment ADREAM du LAAS [Seydoux \(2018\)](#). Le portail OPA (Open Platform for ADREAM) ³⁷ propose les données de ces capteurs aux formats CSV et RDF (ces descriptions reposant sur IoT-O).

Les données du bâtiment ADREAM sont également publiées sur une plate-forme européenne. FIESTA-IoT ³⁸ est un projet européen visant à fédérer des données hétérogènes issues de différentes villes ou de bâtiments intelligents. L'idée sous-jacente est de permettre à des utilisateurs finals de se servir des données collectées par différents réseaux IoT pour mener des expérimentations [Sánchez et al. \(2018\)](#). De façon à prendre en compte le fait que chaque réseau d'IoT fournissant des données repose sur des technologies différentes imposant leur propre format de données, les données soumises à la plate-forme de fédération FIESTA doivent être décrites avec une ontologie choisie dans le cadre du projet : l'ontologie FIESTA-IoT ³⁹. Dans notre cas, les données sont déjà décrites avec IoT-O. Nous avons donc défini un ensemble de correspondances simples entre IoT-O et cette ontologie et nous exploitons quotidiennement cet alignement dans un processus de transformation de données à partir de l'approche définie dans [Euzenat et al. \(2008\)](#). Les descriptions transformées sont ensuite versées dans l'entrepôt fédéré.

3.4 Conclusion

Dans ce chapitre, nous avons montré qu'il était possible de construire de nouvelles bases de connaissances en réutilisant des ontologies ou bases de connaissances existantes par les opérations extraction, combinaison ou as is.

Nous avons dans un premier temps proposé une approche permettant de transformer un modèle conceptuel pour un besoin applicatif spécifique défini à partir d'une amorce de module ontologique. Nous avons également proposé une approche permettant d'étendre ce module en extrayant des éléments présents dans plusieurs bases de connaissances. Notre approche permet de tirer profit des éléments de ces bases de connaissances en prenant en compte différents critères dont nous pouvons pondérer l'importance : la qualité des sources dont les éléments sont extraits et la confiance avec laquelle les correspondances entre éléments ont été établies par un aligneur. Plusieurs pistes d'amélioration de l'approche sont envisagées. Nous en évoquons deux :

- **utilisation d'alignements complexes** : notre approche repose sur l'existence d'alignements simples entre les sources car les systèmes d'alignement les plus répandus génèrent des correspondances simples. Or, comme expliqué dans [Thiéblin et al. \(2019\)](#), ces correspondances ne couvrent qu'une partie des correspondances qui peuvent être établies entre bases de connaissances. De plus, nous faisons l'hypothèse que si plusieurs

36. <http://lov.okfn.org/dataset/lov/vocabs/ioto>

37. <https://syndream.laas.fr:8082/>

38. fiesta-iot.eu

39. <https://github.com/fiesta-iot/ontology>

correspondances sont établies pour un élément d'une source et les éléments d'une autre source, seule l'une d'entre elles est valide car les différentes correspondances établissent de potentielles relations d'équivalence. Une autre approche pourrait consister à affiner ces correspondances simples en identifiant celles qui expriment effectivement la relation d'équivalence. Des approches comme celles qui ont été proposées dans [Spiliopoulos et al. \(2008\)](#) et [Zong et al. \(2015\)](#) pourraient être utilisées ;

- **amélioration du score de qualité des sources** : dans nos propositions, le score de qualité d'une source est fixé empiriquement. Des approches telles que celles présentées dans [Debattista et al. \(2016\)](#), [Zaveri et al. \(2016\)](#) ou [Debattista et al. \(2018\)](#) pourraient être utilisées pour aider à la définition de ce score. Nous pourrions également reprendre les scores déjà définis pour certaines bases de connaissances sur le LOD à partir des travaux présentés dans [Debattista et al. \(2018\)](#).

Nous avons dans un deuxième temps présenté IoT-O, une ontologie de core-domain pour l'IoT. IoT-O repose sur différentes ontologies existantes mais aussi sur l'ontologie SAN que nous avons également proposée. Développée à partir d'un patron de conception, SAN définit un vocabulaire qui permet de décrire les objets de type actionneurs qui sont capables, à partir d'un signal, d'agir sur l'environnement dans lequel le réseau IoT est déployé. IoT-O est publiée sur le LOV. Elle a été utilisée pour décrire sur une plate-forme de données ouvertes les objets déployés dans le bâtiment ADREAM du LAAS, ainsi que les données qu'ils collectent. Une application de gestion intelligente de l'appartement connecté présent dans le bâtiment a également été proposée. Cette application, qui n'est pas décrite dans ce mémoire mais dans [Seydoux et al. \(2016b\)](#), exploite une base de connaissances reposant sur l'ontologie IoT-O. Pour ces exemples de réutilisation, l'ontologie IoT-O est spécialisée pour définir, à partir de la modélisation proposée, les connaissances spécifiques aux applications.

Depuis la construction d'IoT-O, plusieurs initiatives ont été menées pour proposer des ontologies réutilisables pour l'IoT. La démarche suivie correspond à ce que nous avons initié en développant IoT-O. Des ontologies générales pour le domaine de l'IoT sont en effet proposées dans l'objectif de fixer la représentation des notions-clés et de les spécialiser par la suite pour des applications spécifiques. Les deux groupes de standardisation précédemment évoqués n'ont commencé à avoir des propositions concrètes qu'à partir de 2017. C'est en effet au cours de cette année-là que le groupe d'intérêt du W3C consacré au WoT a proposé le premier document spécifiant la description de Thing⁴⁰ et que la première distribution de oneM2M intégrant des fonctionnalités sémantiques a été proposée. Il est intéressant de noter que chacun de ces groupes de standardisation a proposé sa propre ontologie, ce qui peut sembler aller à l'encontre de l'interopérabilité et donner l'impression que le grand nombre de standards dans le domaine peut mener à la création de nombreuses ontologies.

Comme nous le présentons maintenant, les ontologies pour l'IoT proposées depuis la proposition d'IoT-O en reprennent le principe de modularité.

SSN et SOSA : la première version de SSN (Semantic Sensor Network ontology)⁴¹ a été proposée par le W3C en 2011 [Lefort and Henson \(2011\)](#) comme une synthèse des modèles destinés aux capteurs et aux observations qu'ils produisent. Ses concepteurs ont délibérément laissé de côté les aspects contextuels des observations comme le temps, la localisation, ainsi que les aspects spécifiques des domaines d'application car ils ont vocation à être décrits par d'autres vocabulaires à définir en fonction de l'application visée. Cette ontologie est devenue rapidement un standard de fait sur le SWoT. Cette ontologie ayant été construite comme un module respectant les bonnes pratiques du Web Sémantique, ses réutilisations ont été très nombreuses [Haller et al. \(2019\)](#). L'augmentation du nombre de systèmes de l'IoT intégrant des actionneurs ainsi que l'identification de problèmes de modélisation dans SSN ont conduit à sa mise à jour en 2017. Rebaptisée SOSA SSN⁴², elle est maintenant définie à partir d'un

40. <https://www.w3.org/TR/2017/WD-wot-thing-description-20170914/>

41. <http://www.w3.org/ns/ssn/>

42. <http://www.w3.org/ns/sosa/>

module cœur destiné aux capteurs, observations et actionneurs et d'une extension contenant l'alignement de ce module cœur à l'ancien vocabulaire défini dans SSN.

L'ontologie WoT : l'ontologie WoT⁴³ est consacrée à la représentation du modèle de description d'objet (Thing Description) proposé par le groupe du W3C au groupe consacré au Web des Objets. Elle définit les termes associés à l'architecture du Web of Things et se limite aux interactions entre objets sur le Web, laissant volontairement de côté des éléments comme les caractéristiques physiques des objets et de leur déploiement, déjà représentables avec d'autres ontologies. La dernière version de travail proposée en 2019 importe l'ontologie Hypermedia Controls Ontology⁴⁴ qui permet de décrire les liens hypermedia sur le Web.

iot.schema.org : schema.org est un vocabulaire proposé pour décrire les ressources usuellement publiées sur le Web. Il a été étendu pour couvrir des sous-domaines spécifiques comme celui de l'IoT. iot.schema.org est décrit à partir d'une représentation en HTML⁴⁵ ou présentée et discutée via github⁴⁶.

Afin de garantir l'interopérabilité sémantique à partir d'IoT-O, cette ontologie a été alignée avec l'ontologie FIESTA-IoT dans le cadre du projet FIESTA. Nous avons également proposé un alignement avec l'ontologie SoSA SSN⁴⁷ soutenue par le W3C. Cet alignement a été pris en compte dans l'analyse de l'usage de l'ontologie SoSA SSN présentée dans [García-Castro et al. \(2019\)](#). Nous comptons poursuivre cette démarche et aligner IoT-O avec les ontologies précédemment citées et les ontologies qui pourraient être proposées à l'avenir.

43. <https://www.w3.org/2019/wot/td>

44. <https://www.w3.org/2019/wot/hypermedia>

45. <https://iot.schema.org/>

46. <https://github.com/schemaorg/schemaorg/issues/1272>

47. <https://www.irit.fr/recherches/MELODI/ontologies/iot-o-sosa.html>

 Évolution d'ontologies guidée par des textes

Sommaire

4.1	Introduction	36
4.2	Évolution d'ontologies pour l'annotation de textes	37
4.2.1	Motivations et travaux existants	37
4.2.2	Aperçu de l'approche	38
4.2.3	Déclencher un changement	39
4.2.4	Représentation des changements	40
4.2.5	Influence sur les annotations	41
4.2.6	Évaluations	41
4.3	Recherche de documents vitaux hors corpus	41
4.3.1	Travaux connexes	42
4.3.2	Vue générale de l'approche	43
4.3.3	Sélection des documents vitaux	43
4.3.4	Sélection des phrases vitales	43
4.3.5	Détection de la nouveauté	44
4.3.6	Évaluations	45
4.4	Conclusion	46

4.1 Introduction

Définies en tant que représentations consensuelles et stables, les ontologies sont utilisées aujourd'hui comme des modèles de connaissances d'un domaine pour des applications spécifiques. Cependant, les contextes dans lesquels elles sont utilisées sont en évolution régulière. Ces évolutions peuvent être liées au domaine de connaissances que les ontologies représentent aussi bien qu'aux exigences des applications dans lesquelles elles sont utilisées. Modifier les ontologies en fonction de ces évolutions présente plusieurs difficultés, tant sur un plan pratique que théorique. Pratiquement, il n'est pas toujours évident d'identifier un besoin de changement, ni l'influence que ce changement va avoir sur le comportement des systèmes utilisant l'ontologie. Théoriquement, l'évolution remet en question la représentation de la connaissance du domaine et suppose de maîtriser la sémantique du changement sur la connaissance déjà représentée.

Depuis 2002 environ, l'évolution des ontologies a fait l'objet de nombreuses recherches. Comme cela est souligné dans l'état de l'art présenté dans [Zablith et al. \(2015\)](#), différentes tâches sont en lien avec l'évolution :

- la détection d’un besoin de changement ;
- la suggestion d’un changement spécifique ;
- l’évaluation de l’impact d’un changement avant sa validation ;
- l’implémentation d’un changement et ses répercussions au niveau de l’ontologie elle-même et au niveau des applications qui l’utilisent.

Dans ce chapitre, nous montrons comment nous avons abordé la notion d’évolution dans nos travaux.

Nous avons proposé d’une part une approche mettant en œuvre les différentes tâches que nous venons d’énumérer dans le cas où le domaine de connaissances et les besoins d’une application spécifique évoluent. Nous nous sommes concentrés sur une application visant à annoter (c’est à dire décrire sémantiquement) des documents textuels. Ces travaux ont été réalisés dans le cadre du projet ANR DYNAMO durant lequel la thèse d’Anis Tissaoui que nous avons co-encadrée a été menée [Tissaoui \(2013\)](#). Ils ont conduit à différentes publications dont [Tissaoui et al. \(2011a\)](#) et [Tissaoui et al. \(2013\)](#).

Nous avons également cherché à répondre spécifiquement à la première tâche listée en proposant une approche visant à identifier des phrases dans des documents du Web pouvant révéler un besoin de changement dans une base de connaissances. À l’intersection des domaines de l’ingénierie des connaissances et de la recherche d’information, cette approche présente l’originalité de prendre en compte les informations sur une instance représentées dans une base de connaissances pour analyser les phrases d’un document mentionnant cette même instance. Ces travaux ont été réalisés dans le cadre d’une collaboration avec l’équipe IRIS¹ de l’IRIT. Spécialisée dans la recherche d’information, cette équipe a apporté son expertise en agrégation de contenus. Le co-encadrement de la thèse de Rafik Abbes [Abbes \(2015\)](#) a notamment mené aux travaux présentés dans ce chapitre et publiés dans [Abbes et al. \(2015a\)](#) et [Abbes et al. \(2015b\)](#).

4.2 Évolution d’ontologies pour l’annotation de textes

L’annotation ou la description sémantique de documents textuels consiste à décrire le contenu de ces documents à partir d’entités provenant d’ontologies. Ces descriptions sont ensuite exploitées par des systèmes de Recherche d’Information afin de faciliter l’accès aux documents [Ciravegna et al. \(2002\)](#).

4.2.1 Motivations et travaux existants

Dans [Klein and Fensel \(2001\)](#), l’évolution d’ontologies est considérée comme “*the ability to handle changes in ontologies by creating and managing different variants of it*”. Les différentes variantes de l’ontologie, également appelées versions, sont liées par des relations. Ces relations entre deux versions successives expriment non seulement des relations conceptuelles et logiques entre les entités des deux versions mais aussi et surtout les relations de changement qui représentent ce qui a été transformé [Klein et al. \(2002\)](#). Le premier type de relations établit des correspondances (telles que définies dans le chapitre 2) alors que le deuxième spécifie les opérations de changement (comme, par exemple, ajouter un domaine à une propriété, fusionner deux classes, etc.) à appliquer sur la version initiale de l’ontologie pour obtenir la nouvelle. Ces relations sont soit identifiées lorsque les changements sont effectués via des outils intégrant la gestion de l’évolution d’ontologies comme Protégé² ou KAON [Stojanovic et al. \(2002\)](#), soit identifiés *a posteriori* via des outils faisant la différence entre deux ontologies [Völkel et al. \(2005\)](#), [Noy and Musen \(2002\)](#), [Papavassiliou et al. \(2009\)](#), [Zeginis et al. \(2007\)](#). Les opérations de changement sont classifiées comme étant simples ou complexes en fonction

1. <https://www.irit.fr/departement/gestion-des-donnees/iris/>

2. <https://protege.stanford.edu/>

du type d'entités sur lesquelles le changement s'applique. Selon [Hartung et al. \(2011\)](#) : “*simple changes refer to the addition, modification or deletion of individual schema constructs, while complex changes refer to multiple such constructs and may be equivalent to multiple simple changes*”. Ces deux types de changements sont aussi appelés *low level/high level operations* dans [Papavassiliou et al. \(2009\)](#), *elementary/composite changes* dans [Stojanovic et al. \(2002\)](#) ou *atomic/complex changes* dans [Stuckenschmidt and Klein \(2003\)](#).

Les travaux que nous proposons présentent deux originalités. En premier lieu, nous nous intéressons à l'évolution d'ontologies à composantes lexicales, ou ressources termino-ontologiques (RTO), dans lesquelles les termes ont un statut à part entière à côté des composants habituels d'une ontologie. Ensuite, nous intégrons dans le processus d'évolution le fait que l'ontologie soit utilisée pour une application précise : l'annotation sémantique de documents. Une hypothèse forte de notre travail est que la prise en compte des conséquences d'une modification de l'ontologie sur l'ontologie elle-même et sur les annotations, et ce dès la proposition de cette modification, facilite l'implémentation de ce changement. Nous cherchons à éviter de devoir recommencer inutilement l'annotation des documents, ou de dégrader la qualité des annotations existantes.

Dans nos travaux, nous avons considéré l'environnement d'annotation automatique de documents TextViz [Reymonet et al. \(2010\)](#) qui s'appuie sur les termes d'une RTO pour annoter les documents. Ce système, exploitant la composante lexicale de l'ontologie, a été choisi dans le cadre du projet ANR DYNAMO dont ces travaux font partie. Dans ce projet, nous avons considéré qu'annoter sémantiquement un document revient à associer des occurrences de termes définis dans une RTO à des documents, tel que nous l'avons présenté dans le chapitre 2.

4.2.2 Aperçu de l'approche

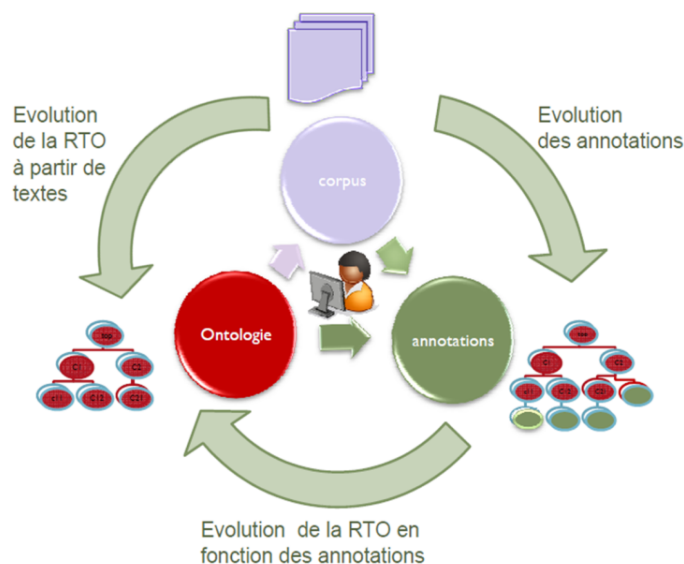


FIGURE 4.1 – Types d'évolutions pour l'annotation à partir de RTO

L'approche que nous avons proposée prend en compte l'évolution à différents niveaux. D'une part, toute évolution de la RTO implique de détecter les annotations devenues inconsistantes à cause des changements de la RTO, mais également de les corriger en vue de garantir la consistance de la base d'annotations. D'autre part, les évolutions de la base de documents et donc des annotations produites ou à produire peuvent révéler la nécessité d'évolutions de la RTO, qui en retour impliquent le maintien de consistance de l'ensemble des annotations.

Notre approche repose sur l'hypothèse selon laquelle l'évolution d'ontologies dans le cadre d'une application spécifique se fait dans un même environnement logiciel intégrant des exten-

sions permettant la gestion des évolutions de l'ontologie ainsi que la gestion des annotations réalisées à partir des ontologies. Ces extensions permettent de visualiser et d'agir sur l'ontologie et l'application l'utilisant. Nous avons pour cela considéré l'environnement Protégé qui, en plus de l'interface habituelle de visualisation et de gestion d'ontologies, est enrichi par l'extension TextViz permettant de visualiser et de manipuler les annotations de documents et l'extension EvOnto qui permet de gérer conjointement les évolutions de la RTO et des annotations. La figure 4.2 présente une capture d'écran de Protégé avec ces deux extensions. La partie de gauche permet de visualiser les documents et leurs annotations, la partie de droite est consacrée à l'évolution de la RTO et des annotations des documents.

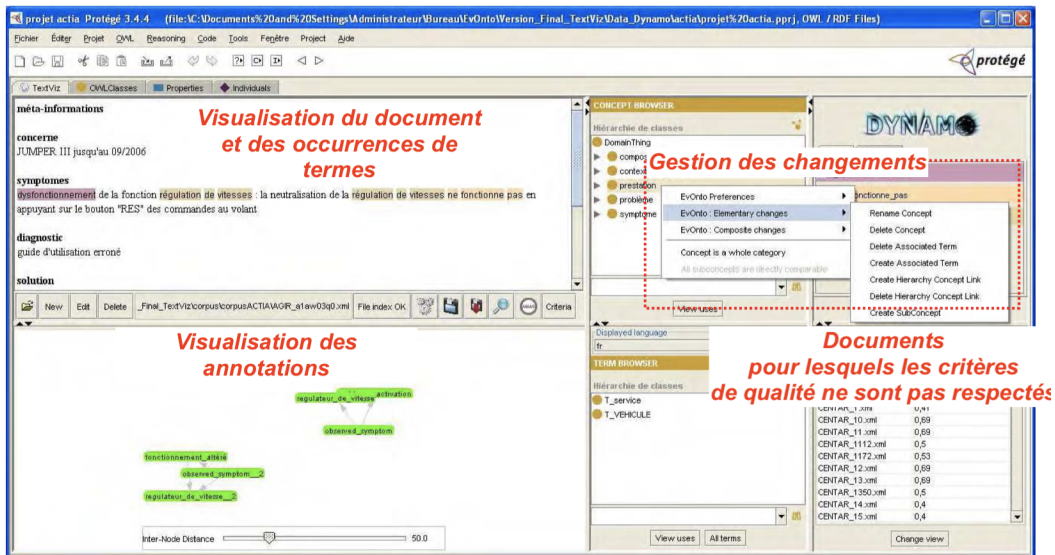


FIGURE 4.2 – TextViz et EvOnto dans Protégé

Nous décrivons ci-dessous les principes généraux de l'approche sur laquelle l'extension EvOnto repose.

4.2.3 Déclencher un changement

Alors que la plupart des travaux font l'hypothèse que les besoins de changement sont déclenchés à partir d'une analyse de l'ontologie elle-même (besoin d'ajout, de suppression ou de modifications d'entités), nous considérons une raison supplémentaire pouvant entraîner sa modification : les cas dans lesquels l'annotation d'un document ne peut pas être faite correctement.

L'ajout de nouveaux documents peut mettre en évidence des termes des documents, pertinents pour leur annotation mais absents de la RTO. Dans ce cas, les annotations générées peuvent être incomplètes (des parties de textes ne peuvent pas être décrites par des entités) ou fausses (les entités reconnues ne sont pas les bonnes).

Pour repérer rapidement et systématiquement tous les documents mal annotés, nous proposons de définir des critères permettant de détecter *a priori* une annotation de qualité. Ces critères sont propres au corpus et au processus d'annotation. Un exemple de critère est la proportion de termes du texte qui doivent correspondre à une occurrence de termes de la RTO. Les occurrences étant décrites à partir de la RTO, il est possible de calculer ce critère à partir de la base de connaissances. L'expert peut également indiquer des concepts et/ou des propriétés qui doivent être trouvés dans chaque annotation. Ce critère se traduit par un patron générique d'annotation. Ces deux types de critères sont définis par un expert du domaine à partir de requêtes SPARQL. Lorsque les critères définis par l'expert ne sont pas respectés, le document est signalé à l'utilisateur. Celui-ci peut alors décider d'effectuer un

changement dans la RTO. Bien que ce processus soit manuel, il permet de guider la détection du besoin de changement à partir de besoins liés à l'application d'annotation.

4.2.4 Représentation des changements

Par le biais de l'éditeur, l'utilisateur apporte des modifications à l'ontologie. Chaque modification peut avoir de nombreuses conséquences. La représentation du changement a pour but de présenter les conséquences avant d'en assurer la mise en œuvre. Pour représenter les changements sur une RTO, nous avons recensé l'ensemble des modifications que l'on peut y apporter et nous avons proposé un vocabulaire pour les décrire [Tissaoui et al. \(2011a\)](#). Les changements peuvent être élémentaires (ajouter ou effacer un terme, un concept ou une relation), ou complexes (associer un terme à un concept ou à une propriété, déplacer ou fusionner des concepts). Cette typologie élargit celle de [Stojanovic \(2004\)](#) par la prise en compte de la composante lexicale de la RTO qui implique de considérer les termes mais aussi les liens de dénotations entre termes et occurrences de termes. Pour chaque type d'évolution dans la RTO, nous avons défini un ensemble de changements supplémentaires qui doivent être effectués pour préserver la consistance de la RTO et des annotations. Lorsque plusieurs effets sont possibles pour un même changement, un choix doit être effectué soit arbitrairement par le système de gestion des changements, soit par l'utilisateur. La manière de faire intervenir l'utilisateur pour effectuer le choix d'un de ces effets (au cas par cas, une fois pour toutes, ...) revient à fixer une stratégie d'évolution. Pour chaque type de changement, nous avons identifié différentes stratégies d'évolution qui permettent d'anticiper les conséquences possibles d'un changement avant de le rendre effectif. Par exemple, les termes associés à un concept supprimé peuvent être associés au père du concept supprimé, à un autre concept choisi par l'utilisateur ou supprimés de la RTO. Les stratégies sont présentées à l'utilisateur avec leurs conséquences. La figure 4.3 montre une capture d'écran présentant les stratégies proposées lorsqu'un utilisateur veut fusionner deux concepts.

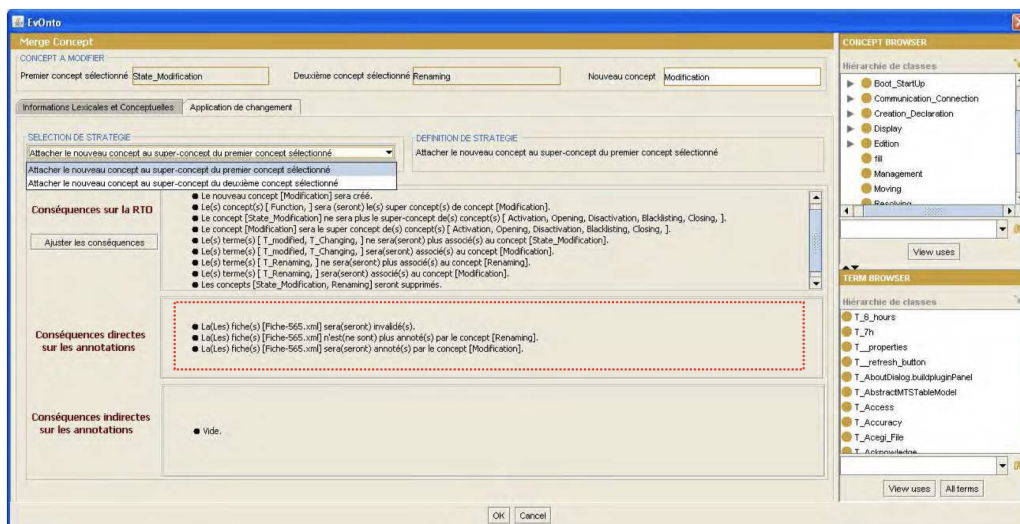


FIGURE 4.3 – Présentation d'une stratégie dans EvOnto

Les stratégies d'évolution sont adaptables. Elles reposent sur un ensemble de conséquences prédéfinies pour chaque type de changement, mais l'utilisateur peut ajuster les conséquences entité par entité s'il le juge nécessaire. Pour ce faire, différentes informations sont présentées à l'utilisateur. Ces informations comprennent les éléments de RTO dont la définition dépend de l'entité modifiée ainsi que les documents annotés par l'élément modifié et les documents annotés par les entités de la RTO affectés par la modification. De façon à prendre en compte la diversité des raisons motivant un changement, pour un même type de changement, l'utilisateur choisit la conséquence qu'il souhaite mettre en œuvre en fonction des informations présentées.

L’interface met à jour les informations présentées en fonction des conséquences choisies afin de guider l’utilisateur dans l’ajustement de la stratégie. Une fois une stratégie sélectionnée, le changement est répercuté sur la RTO et les annotations.

4.2.5 Influence sur les annotations

Lorsque l’utilisateur a choisi de faire évoluer la RTO, ce changement est répercuté sur les annotations. Pour gérer l’évolution conjointe des annotations, nous proposons un processus en deux temps : détecter les annotations qui impliquent au moins une entité de la RTO qui a évolué suite au changement, puis modifier ces annotations devenues incohérentes. Pour détecter les annotations à modifier, l’approche recherche dans la base de connaissances les descriptions de documents qui comportent des entités de la RTO qui ont été modifiées. Par exemple, un triplet RDF où apparaît un concept qui a été supprimé doit clairement être remis en cause. Une fois les annotations inconsistantes identifiées, elles doivent être corrigées. Nous proposons à l’utilisateur soit de relancer le processus d’annotation automatique sur les documents concernés, soit de modifier manuellement les annotations.

4.2.6 Évaluations

L’approche a été évaluée dans le cadre du projet ANR DYNAMO pour annoter des fiches relatives au diagnostic automobile. Nous avons sollicité 5 personnes habituées à construire des ontologies et nous leur avons demandé de faire évoluer une RTO utilisée pour annoter une centaine de documents en réalisant une liste imposée de changements. Après une phase de prise en main de l’interface EvOnto, nous leur avons demandé d’effectuer ces changements à partir de notre approche. Nous leur avons également demandé d’effectuer ces changements en utilisant uniquement Protégé et TextViz sans assistance spécifique pour l’évolution. Nous leur avons finalement demandé de remplir un questionnaire de satisfaction. Nous avons pu mettre en avant que :

- l’approche permettait de mettre en œuvre tous les changements nécessaires ;
- les informations présentées pour mettre en place les différentes stratégies étaient utiles ;
- l’approche permettait d’effectuer plus rapidement les changements que sans assistance.

4.3 Recherche de documents vitaux hors corpus

Dans nos travaux, nous avons considéré que l’extraction de connaissances à partir de documents textuels était une approche couramment utilisée pour la constitution de bases de connaissances [Petasis et al. \(2011\)](#). Or ces approches ne guident que très peu le processus de constitution du corpus de documents. Il est soit arbitrairement défini comme dans le cas des pages Wikipedia utilisées pour mettre à jour DBpedia, soit constitué manuellement [Augenstein et al. \(2012\)](#), [Exner and Nugues \(2012\)](#). Dans le contexte de la mise à jour de bases de connaissances, la tâche d’identification des textes à partir desquels extraire la connaissance n’est pas triviale. D’une part, certaines approches d’extraction de connaissances à partir de textes analysent les documents dans leur intégralité, or la connaissance sur une instance donnée est souvent décrite dans quelques phrases du document seulement. D’autre part, lorsque l’on considère en particulier certaines instances comme des instances de type événement (catastrophe naturelle, industrielle, mouvement social...) dont la connaissance établie peut évoluer fréquemment au cours de la période englobant la date de l’événement, les textes sur lesquels ces approches sont appliquées doivent relater des informations nouvelles et à jour.

Nous avons proposé une approche permettant d’aider à l’identification de documents du Web apportant de la connaissance nouvelle sur des instances de bases de connaissances. Nous identifions plus particulièrement dans un flux de documents du Web les informations opportunes et pertinentes sur ces instances, que nous appelons *informations vitales*. Les phrases

contenant des informations vitales sont appelées *phrases vitales* et par extension, les documents les contenant sont appelés *documents vitaux*.

Détecter en temps réel les phrases vitales est une tâche complexe qui soulève plusieurs problématiques pouvant être vues comme des étapes du processus :

- comment détecter le fait qu’un document relate une information nouvelle sur une instance ?
- étant donné un document vital, comment extraire les phrases vitales relatant les informations vitales ?
- comment détecter le fait que deux phrases vitales relatent la même information ?

L’étape de détection de documents vitaux est cruciale. Elle nécessite la mise en place d’un processus complexe que nous avons détaillé dans [Abbes et al. \(2013\)](#), [Abbes et al. \(2015d\)](#). Dans ce mémoire, nous nous concentrons sur l’identification de phrases vitales. Nous nous focalisons sur des instances de type événement et analysons le flux de documents uniquement dans les périodes de déroulement de ces événements. Par conséquent, les documents mentionnant le ou les termes associés à l’instance dans la base de connaissances contiennent potentiellement des informations sur l’instance. Concernant l’étape d’extraction des phrases vitales, les travaux de l’état de l’art se fondent généralement sur la présence de mots spécifiques. Ces mots sont soit choisis manuellement, soit sélectionnés automatiquement. Dans notre travail, nous proposons d’exploiter les termes associés aux descriptions de l’instance dans la base de connaissances. Afin de détecter la nouveauté d’une phrase par rapport à une autre, nous exploitons également les descriptions de l’instance contenues dans la base de connaissances.

4.3.1 Travaux connexes

Accélérer la mise à jour des bases de connaissances est une problématique actuelle dont le premier enjeu est d’identifier un besoin d’évolution. L’analyse de documents desquels extraire la connaissance à mettre à jour est une solution pour identifier ce besoin [Zablith et al. \(2015\)](#). La phase d’identification de ces documents est souvent laissée aux concepteurs de la base dans les travaux d’ingénierie de connaissances. Cependant, lorsque la base de connaissances comporte des instances très souvent mentionnées sur le Web, il est dommage de ne pas tirer profit de ces informations. Le module *DBpedia Live*, par exemple, vise à mettre à jour quasiment en temps réel la base de connaissances lorsque les *infoboxes* des pages Wikipedia sont modifiées [Lehmann et al. \(2014\)](#). Cependant, comme cela est souligné dans [Frank et al. \(2012\)](#), un temps de latence est constaté entre la publication d’une information sur le Web et les mises à jour de Wikipedia, puis de DBpedia.

Pour aider à l’identification d’informations vitales dans des documents du Web, nous nous sommes tournés vers les travaux de recherche d’information qui s’intéressent à ces questions. La campagne d’évaluation TREC a notamment lancé la tâche *Knowledge Base Acceleration (KBA)* [Frank et al. \(2012, 2013\)](#) dont l’objectif est d’évaluer la capacité des systèmes à mettre à jour quasiment en temps réel les bases de connaissances. Plusieurs méthodes ont été proposées comme [Wang et al. \(2013\)](#), [Abbes et al. \(2013\)](#) et [Abbes et al. \(2015d\)](#) afin d’identifier, en temps réel, les documents vitaux relatant des informations nouvelles sur des instances de type personne, organisation et établissement. Cependant, malgré leur utilité, ces méthodes renvoient des documents entiers, ce qui oblige les éditeurs de la base de connaissances ou les outils d’extraction à parcourir tous leurs contenus pour chercher les nouvelles informations vitales. En outre, elles ne traitent pas le problème de redondance entre les documents, c’est à dire qu’elles renvoient tous les documents apportant des informations vitales même s’ils contiennent des informations vitales redondantes.

D’autres approches se sont intéressées à identifier à partir d’un flux de documents Web les phrases vitales relatives à des événements largement connus, comme les catastrophes naturelles [Aslam et al. \(2013\)](#), [McCreadie et al. \(2014\)](#). [Liu et al. \(2013\)](#) s’appuient sur

des données d'apprentissage pour apprendre les mots importants permettant d'identifier les phrases vitales. [Xu et al. \(2013\)](#) utilisent un classifieur afin de détecter les phrases contenant de nouvelles informations. [Zhang et al. \(2013\)](#) sélectionnent les phrases contenant les mots les plus représentatifs selon leur fréquence d'occurrence.

4.3.2 Vue générale de l'approche

Notre approche a pour but de détecter en temps réel les phrases relatant de nouvelles informations vitales relatives à une instance donnée d'une base de connaissances, à partir d'un flux de documents issus du Web. Ces phrases vitales doivent être pertinentes (concerner l'instance), exhaustives (couvrir les différentes informations publiées sur l'instance), non redondantes (relatées une seule fois) et émises sans trop de latence.

Formellement, considérons un flux continu F composé de documents d ayant chacun une date de publication $t(d)$ et une séquence de phrases s_j , tels que $0 \leq j < l(d)$ où $l(d)$ désigne la longueur du document d en nombre de phrases. Soient h_0, h_1, \dots, h_m des instants séparés par un intervalle de temps constant (par exemple une heure). Nous désignons par F_{h_i} l'ensemble de documents du flux tels que $\forall d \in F_{h_i}, h_{i-1} \leq t(d) < h_i$.

L'algorithme que nous proposons repose sur :

1. la sélection des documents vitaux D_{h_i} par rapport à I en utilisant comme requête le ou les termes associé(s) à l'instance dans la base de connaissances ;
2. la sélection des phrases vitales candidates ;
3. la vérification de la nouveauté des phrases candidates par rapport aux phrases déjà sélectionnées ($\in V(I)$).

4.3.3 Sélection des documents vitaux

Nous n'analysons que la période "chaude" durant laquelle les informations vitales sur une instance donnée sont publiées dans le flux de documents du Web. Dans ce travail, nous supposons que cette période est connue. À chaque instant h_i , nous analysons les nouveaux documents apparus dans le flux entre h_{i-1} et h_i et nous attribuons à chacun d'eux un score de vitalité par rapport à l'instance I considérée. Ce score est calculé par la probabilité que le ou les mots composant le ou les termes désignant l'instance I soient générés par une modèle de langue probabiliste estimé à partir du document analysé [Zhai and Lafferty \(2001\)](#). Les h documents ayant obtenu le meilleur score, h étant fixé empiriquement, sont sélectionnés afin d'être analysés dans l'étape suivante.

4.3.4 Sélection des phrases vitales

Dans cette étape, nous analysons les phrases contenues dans les documents sélectionnés. Pour chaque phrase, nous devons décider si elle est vitale (relatant une information pertinente et opportune) par rapport à l'instance à surveiller I . Notre intuition est qu'une phrase doit être considérée comme vitale si :

- elle est à proximité des termes désignant I ;
- elle contient des termes désignant des entités sémantiquement liées à l'instance I .

La proximité d'une phrase par rapport à l'instance I peut refléter sa pertinence. Une phrase mentionnant l'instance a plus de chance de parler de celle-ci. Nous traduisons ainsi la proximité entre une phrase s_j et l'instance I en un score calculé selon l'équation suivante :

$$score_proximité(s_j, I) = \frac{1}{|I.label|} \sum_{t \in I.label} \sum_{dist=0}^{dmax} e^{-dist * match_s(t, s_{j+dist}, s_{j-dist})} \quad (4.1)$$

$I.label$ est un terme désignant l'instance I . $|I.label|$ est le nombre de mots qu'il contient.
 $match_s(t, s_x, s_y)$ est égal à 1 si t est contenu dans l'une des phrases s_x et s_y , 0 sinon.
 $dmax$ est la distance maximale à considérer (calculée en nombre de phrases).

Nous considérons uniquement les phrases à proximité d'occurrences de termes désignant l'instance I en favorisant celles qui sont proches de l'ensemble des mots composant ces termes.

En plus de la proximité, nous supposons que pour une instance I , il existe un ensemble de mots "importants" qui peuvent refléter la vitalité d'une phrase. Nous appelons ces mots des *mots déclencheurs*. Nous posons l'hypothèse selon laquelle les instances typées par le même concept dans la base de connaissances partagent les mêmes mots déclencheurs. Afin d'identifier ces mots déclencheurs, nous proposons d'exploiter toutes les annotations (descriptions en langage naturel) qui ont pu être renseignées sur des instances du type considéré. Nous considérons comme étant une annotation le texte associé à une instance par les propriétés d'annotation de OWL, ou les propriétés du Dublin Core, ou encore le résumé associé dans DBpedia par la propriété `dbpedia-owl : abstract`. Par exemple, les mots tels que *effets, force, tempête, blessés, dommages* pourront être très utiles pour décrire les instances de type *ouragan* comme ils sont présents dans les annotations associées aux instances *ouragan Sandy* et *ouragan Isaac*.

Formellement, soient $X(I) = \{A(I_1), A(I_2), \dots, A(I_m)\}$ l'ensemble des m mots extraits des annotations associées aux instances de même type que I . Nous pondérons les mots t par l'équation suivante :

$$\omega(t) = \frac{\sum_{i=1}^m TF(t, A(I_i))}{IIF(t)} \quad (4.2)$$

$TF(t, A(I_i))$ est le nombre d'occurrences du terme t dans l'annotation $A(I_i)$

$IIF(t) = \log(\frac{m+1}{IF(t)})$ est un facteur utilisé pour donner la priorité aux termes se trouvant dans la plupart des annotations des instances de même type que l'instance I

$IF(t)$ est le nombre d'instances du même type dont l'annotation contient le terme t

Les k premiers mots seront considérés comme des mots déclencheurs pour l'instance I , k étant fixé empiriquement.

Pour qu'une phrase soit considérée comme une phrase vitale candidate, il faut :

- que le score de proximité de la phrase soit $> \tau_p$,
- qu'elle contienne un mot déclencheur.

4.3.5 Détection de la nouveauté

Les phrases sélectionnées à l'étape précédente pourraient contenir des informations vitales redondantes déjà émises. Afin d'éliminer la redondance, nous comparons chaque phrase vitale candidate à toutes les phrases vitales déjà ajoutées incrémentalement à l'ensemble $V(I)$. Détecter la nouveauté n'est pas une tâche facile. Comme le montre le tableau 4.1, les deux phrases s_1 et s_2 contiennent un grand nombre de chaînes de caractères en commun, mais relatent deux informations différentes. Inversement, les phrases s_2 et s_3 sont divergentes textuellement mais portent la même information.

n°	Date	Texte
s1	26 Oct. 2012 - 07 :27	Hurricane Sandy leaves 21 people dead in Caribbean
s2	26 Oct. 2012 - 20 :50	Hurricane Sandy leaves 41 people dead in Caribbean
s3	30 Oct. 2012 - 06 :17	Hurricane Sandy is continuing to head north from the Caribbean where it has killed a total of 41 people in the Caribbean

TABLEAU 4.1 – Exemple de phrases vitales

Dans notre approche, nous considérons qu'une phrase vitale candidate s_j est nouvelle par rapport aux phrases déjà émises $V(I)$ si son texte est divergent et/ou présente une instance liée nouvelle (non détectée dans les phrases précédentes $V(I)$). Formellement, s_j est nouvelle si elle respecte la fonction de nouveauté suivante :

$$est_nouvelle(s_j, V(I)) = texte_divergent(s_j, V(I)) \circ instance_liée_nouvelle(s_j, V(I)) \quad (4.3)$$

$$texte_divergent(s_j, V(I)) = \begin{cases} faux & \text{si } \exists s_k \in V(I), \cos(s_j, s_k) > \tau_n(V(I)) \\ vrai & \text{sinon} \end{cases} \quad (4.4)$$

$$instance_liée_nouvelle(s_j, V(I)) = \begin{cases} vrai & \text{si } \exists x \in IL(s_j, I), \forall s_k \in V(I) x \notin IL(s_k, I) \\ faux & \text{sinon} \end{cases} \quad (4.5)$$

$IL(s_i, I)$ est l'ensemble des instances liées reconnues dans la phrase s_i . Dans notre méthode, nous proposons de prendre en compte les propriétés définies dans l'ontologie pour le concept par lequel l'instance I que nous considérons est typée. Nous cherchons à identifier dans la phrase des instances ou des valeurs potentiellement liées sémantiquement à l'instance. Pour cela, nous considérons les termes associés dans l'ontologie aux domaines et co-domaines de ces propriétés.

$\tau_n(V)$ est un seuil de nouveauté textuelle. Au fur et à mesure que l'ensemble de phrases vitales $V(I)$ s'enrichit, le risque de redondance augmente, d'où l'idée de faire décroître le seuil τ_n selon une fonction gaussienne :

$$\tau_n(V(I)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{|V(I)|^2}{\delta^2}} \quad (4.6)$$

Le paramètre σ a un impact sur la similarité, et le paramètre δ contrôle le taux de décroissance du seuil. $|V(I)|$ est le nombre de phrases de l'ensemble $V(I)$.

Le symbole \circ de l'équation 4.3 peut être un opérateur **ET** pour faire que le système privilégie la précision en limitant la redondance (dans ce cas, la phrase s_3 , ne présentant aucune instance liée nouvelle par rapport aux phrases s_1 et s_2 , sera considérée comme redondante malgré le fait que son texte diverge), ou bien un opérateur **OU** pour favoriser le rappel (dans ce cas, la phrase s_2 , bien qu'elle ne diverge pas trop par rapport à s_1 , sera considérée comme nouvelle car elle contient un nouvelle valeur).

4.3.6 Évaluations

Trouver un cadre expérimental pour évaluer notre approche n'est pas chose aisée car il n'existe pas à notre connaissance de jeux de données constitués d'une base de connaissances dont plusieurs versions sont disponibles pas plus que les corpus de documents desquels la connaissance a été extraite pour mettre en place les différentes versions. Bien qu'étant une tâche de Recherche d'Information, la tâche *Temporal Summarization* (TS) de la campagne d'évaluation TREC est à notre sens la plus adaptée.

Le but de cette tâche est de concevoir des systèmes capables de surveiller les événements en détectant à la volée toutes les nouvelles informations publiées dans un flux qui comporte 500 millions de documents en anglais, issus de différentes sources du Web (presse, blogs, etc.) et associées à des dates de publication. Les systèmes doivent extraire les phrases contenant des informations vitales tout en évitant la redondance.

Les thèmes à considérer dans le cadre de cette tâche correspondent à des événements d'actualité tels que des manifestations, des accidents ou des catastrophes naturelles. En analysant ces thèmes, nous avons remarqué qu'ils correspondaient à des instances de DBpedia pour lesquelles les termes les désignant sont définis. Nous considérons donc que les instances

pour lesquelles nous souhaitons identifier des phrases vitales dans notre approche sont les thèmes de cette tâche. Le terme ou les termes définis dans DBpedia pour désigner le topic sont utilisés pour constituer la requête dans notre approche.

Dans la tâche, les informations vitales à retrouver sont extraites à partir des différentes mises à jour des pages Wikipedia de ces événements. Ces informations ayant été rajoutées manuellement dans la page de Wikipedia, nous considérons qu'elles auraient également mené à une mise à jour manuelle de la connaissance représentée dans DBpedia sur l'instance. Cette référence nous paraît donc pertinente pour évaluer notre approche.

Les différentes expérimentations décrites dans [Abbes et al. \(2015b\)](#) ont montré que les mots saillants récupérés automatiquement à partir des annotations associées aux instances du même type améliorent la détection de phrases vitales comparativement à l'approche dans laquelle seul un vocabulaire spécifique à la nouveauté est considéré. L'approche permet également de récupérer ces informations vitales dès que celles-ci sont disponibles dans les documents.

4.4 Conclusion

Dans ce chapitre, nous avons présenté deux approches permettant de faciliter le processus d'évolution d'ontologies.

La première considère l'évolution dans le cas où une ressource termino-ontologique est utilisée pour décrire des documents textuels. Nous avons défini une typologie de changements pouvant être opérés sur ce type de ressources ainsi que différentes stratégies représentant l'impact de chacun de ces changements sur les autres entités de la RTO et sur les descriptions. Nous avons implémenté ces stratégies dans le cadre d'une extension à l'éditeur d'ontologies Protégé et nous avons montré l'intérêt d'une telle approche pour faire évoluer une RTO destinée à l'annotation de fiches de diagnostic automobile dans le cadre du projet ANR DYNAMO. Au cours de ce projet, nous avons également contribué à l'évaluation d'un système multi-agent pour l'évolution de RTO. Non présentée dans ce manuscrit mais décrite dans [Ottens et al. \(2009\)](#), cette approche repose sur l'originalité de faire représenter au système multi-agents adaptatif non seulement la RTO mais aussi le processus de construction de l'ontologie. La RTO évolue suite à l'ajout de documents dans le système et à des interactions avec l'utilisateur.

La deuxième axe correspond à une approche permettant d'identifier dans des documents publiés sur le Web des phrases apportant des informations nouvelles sur une instance décrite dans une base de connaissances. Cette approche repose sur l'identification à partir de la base de connaissances de termes caractéristiques de l'instance qui permettent d'identifier les documents et les phrases apportant une information nouvelle sur l'instance. Ce type d'approche est utile non seulement pour aider à mettre à jour les documents collaboratifs décrivant des instances telles que les pages Wikipedia, mais également afin de mettre à jour les bases de connaissances elles-mêmes. Pour étendre l'intérêt de l'approche que nous avons proposée, nous envisageons d'extraire la connaissances de ces phrases vitales à partir d'extracteurs tels que ceux décrits dans [Zablith et al. \(2015\)](#) ou [Gerber et al. \(2013\)](#) pour enrichir les bases de connaissances de nouveaux triplets.

 Interrogation de bases de connaissances guidée par le besoin utilisateur

Sommaire

5.1	Introduction	47
5.2	Interrogation d'une base de connaissances à partir de patrons de requêtes	48
5.2.1	Motivations	48
5.2.1.1	Interpréter le besoin en information	48
5.2.1.2	Utilisabilité des interfaces en langue naturelle	50
5.2.1.3	Approches existantes	51
5.2.2	Vue générale du système Swip	52
5.2.3	De la requête initiale à la requête pivot	53
5.2.3.1	La requête pivot	53
5.2.3.2	Étapes de transformation	54
5.2.4	De la requête pivot à la requête finale	55
5.2.4.1	Patrons de requêtes	55
5.2.4.2	Génération de la requête formelle	58
5.2.5	Implementation tout en SPARQL	59
5.2.6	Aperçu de l'évaluation	60
5.3	Interroger plusieurs bases de connaissances	62
5.3.1	Motivations	62
5.3.2	Illustration de l'approche de réécriture	67
5.3.3	Évaluations	68
5.4	Conclusion	69

5.1 Introduction

L'objectif premier du Web tel qu'il a été défini par Tim Berners-Lee était de rendre accessible l'information aux utilisateurs finals [Berners-Lee \(1989\)](#). Le développement des bases de connaissances disponibles sur le Web de Données liées a rendu nécessaire la mise en place d'interfaces de requêtage dont la prise en main soit aussi intuitive que possible pour les utilisateurs. Il est en effet très difficile pour un utilisateur final de maîtriser les vocabulaires à utiliser pour formuler des requêtes. En l'absence de dispositif d'interfaçage, l'utilisateur devrait en effet connaître le langage SPARQL¹ [Perez et al. \(2006\)](#), mais aussi les ontologies qui ont été utilisées pour décrire les données qu'il souhaite retrouver sur le LOD. La difficulté

1. <https://www.w3.org/TR/rdf-sparql-query/>

est encore plus grande lorsque l'on cherche à satisfaire le besoin en information en tirant profit de différentes bases de connaissances qui peuvent contenir des informations le satisfaisant. En effet, la requête doit être exprimée en suivant les vocabulaires spécifiques à chacune d'entre elles.

En s'inspirant de ce qui avait été proposé pour simplifier l'accès au contenu des bases de données, certaines approches proposées pour faciliter l'accès aux connaissances contenues sur le Web Sémantique proposent de rendre transparente la génération des requêtes SPARQL en utilisant des formulaires. D'autres considèrent que les requêtes peuvent être générées à partir de mots-clés.

Pour ce qui nous concerne, lorsque nos travaux sur le sujet ont commencé, nous avons choisi de favoriser l'interaction avec l'utilisateur en utilisant la langue naturelle. L'engouement du grand public pour les assistants vocaux que nous connaissons aujourd'hui nous porte à penser que ce choix était pertinent.

Nos travaux visent à proposer un moyen de traduire en SPARQL une requête utilisateur exprimée sous forme d'une question en langue naturelle. Contrairement aux approches par mots-clés, nous ne considérons pas seulement les entités des ontologies utilisées dans la phrase mais également la construction syntaxique de celle-ci. Le postulat principal sur lequel reposent nos travaux est que, dans le cas d'applications réelles, les requêtes soumises sont très majoritairement des variations autour d'un ensemble de familles de requêtes typiques. Notre approche diffère des approches existantes dans le sens où nous guidons le processus d'interprétation par l'utilisation de patrons de requêtes prédéfinis qui représentent ces familles de requêtes. Notre objectif est de pouvoir proposer à l'utilisateur des requêtes qui ont potentiellement un intérêt pour son besoin en information.

Notre approche a été construite incrémentalement. Dans [Comparot et al. \(2010\)](#), nous avons proposé une première version de l'approche visant à exprimer en termes de graphes conceptuels [Sowa \(1983\)](#), [Chein and Mugnier \(1992\)](#) des requêtes exprimées par mots-clés. Ces travaux ont ensuite été poursuivis dans le cadre de la thèse de Camille Pradel, puis dans le cadre des stages de Master de Fabien Amarger et Elodie Thiéblin. Dans [Pradel et al. \(2011a\)](#), nous avons étendu ce travail afin de prendre en compte des relations exprimées par l'utilisateur entre les mots-clés de la requête. Dans [Pradel et al. \(2012\)](#), nous l'avons adapté aux langages du Web Sémantique pour pouvoir l'évaluer sur les bases de connaissances disponibles sur le LOD. Nous avons proposé un processus d'interprétation de la requête en langue naturelle en lieu et place des requêtes mots-clés que nous traitions jusque là et avons comparé notre système aux systèmes existants dans [Pradel et al. \(2013\)](#). Dans [Amarger et al. \(2013a\)](#), nous avons intégré la prise en compte de requêtes nécessitant l'utilisation d'agrégats. Plus récemment, nous avons considéré la possibilité d'étendre une requête posée initialement sur une seule base de connaissances à partir d'éléments appartenant à d'autres bases de connaissances, en nous fondant sur la notion de correspondances complexes présentée dans le chapitre 2.

Ce chapitre présente une synthèse de ces travaux, qui ont notamment débouché sur le développement du système Swip.

5.2 Interrogation d'une base de connaissances à partir de patrons de requêtes

5.2.1 Motivations

5.2.1.1 Interpréter le besoin en information

Comprendre le besoin en information des utilisateurs du Web est une tâche difficile car de multiples facteurs sont à prendre en compte. De nombreux travaux en recherche d'information ont cherché à caractériser les besoins en information des utilisateurs du Web. Pour

ce faire, certains auteurs ont analysé les “logs” de requêtes de moteurs de recherche généralistes disponibles sur le Web, cherchant ainsi à mettre en évidence les caractéristiques de ces besoins sur une grande masse de données. [Spink et al. \(2001\)](#) ont mené un tel travail et ont comparé leurs résultats à ceux qui avaient été présentés dans [Jansen et al. \(1998, 2000\)](#) et [Silverstein et al. \(1999\)](#). Il en ressort que les termes des requêtes varient considérablement, à l’exception de certains termes très spécifiques. Ces termes reflètent les thèmes populaires sur la période pendant laquelle les logs ont été recueillis. Deux autres observations ont été faites par les auteurs de l’étude. La première se fonde sur la cooccurrence des termes dans les requêtes et montre la possibilité d’identifier des requêtes populaires et des thèmes récurrents. La deuxième est une analyse qualitative des requêtes. C’est la plus marquante pour nos propositions. Les requêtes ont pu être classifiées en 11 catégories principales, chacune correspondant à un ensemble de thèmes reliés sémantiquement.

Une autre étude [Downey et al. \(2008\)](#) analyse les performances des moteurs de recherche par rapport aux besoins en information fréquents (identifiés en comparant les couples <requête-page consultée> d’une semaine à l’autre dans une barre de recherche de navigateur) et ceux qui sont plus rares. Ils mettent en évidence qu’environ 70% des requêtes sont fréquentes et soulignent par ailleurs que pour satisfaire un besoin en information rare (qui n’a pas été ou a été très peu exprimé par d’autres utilisateurs), les moteurs de recherche ne sont pas très performants. Selon eux, la meilleure façon de procéder pour un utilisateur consiste alors à poser une question plus générale, puis à affiner sa recherche. Les auteurs préconisent de suggérer à l’utilisateur des reformulations à partir de patrons identifiés dans les requêtes fréquentes. Les auteurs de [Hollink et al. \(2013\)](#) analysent des sessions de recherche sur des logs de Yahoo!. Lorsqu’ils ciblent leur analyse sur les requêtes portant sur le domaine des films, ils observent que des patrons de requêtes peuvent être identifiés pour caractériser l’enchaînement des requêtes et prédire si la requête va aboutir à la satisfaction du besoin de l’utilisateur. Ces patrons sont constitués à partir de connaissances que l’on peut extraire du Web de données en utilisant l’entité identifiée dans la première requête posée par l’utilisateur.

Plusieurs analyses mettent donc en évidence des régularités et familles de besoins en information dans les requêtes exprimées par les utilisateurs finals, ce qui constitue une des hypothèses ayant motivé notre travail.

Parallèlement, des travaux en recherche d’information ont montré que pour exprimer des requêtes précises, l’expression du besoin par mots-clés avait des limites. Les auteurs de [Bendersky and Croft \(2009\)](#) ont analysé des requêtes “longues”, composées de plus de 4 mots-clés. Ils mettent en évidence les mauvaises performances des moteurs de recherche sur ce type de requêtes et concluent qu’exprimer précisément le besoin en information des utilisateurs nécessite, dans ces cas-là, de formuler des questions en langue naturelle. Cette idée est également développée dans [Dror et al. \(2013\)](#), qui proposent de générer automatiquement une question en langue naturelle lorsque l’utilisateur n’est pas satisfait par les réponses d’un moteur de recherche d’information classique. Cette question est ensuite posée sur des systèmes de questions-réponses communautaires, tels que Yahoo! answer.

Nous avons également cherché à vérifier l’intuition selon laquelle les requêtes pouvaient être regroupées par familles en fonction du type de besoin qu’elles expriment, en analysant des requêtes exprimées en langue naturelle pour interroger des bases de connaissances.

Nous avons considéré trois cadres d’évaluation : un jeu de requêtes “maison” et deux jeux issus de campagnes d’évaluation. Au début de notre travail, nous avons construit notre propre cadre d’évaluation. 160 requêtes distinctes concernant le cinéma d’un point de vue culturel et artistique ont été recueillies auprès de 24 personnes, chaque requête étant composée d’un ensemble de mots-clés et d’une phrase en langue naturelle exprimant le besoin en information. Ce jeu de données est décrit et utilisé dans [Pradel et al. \(2011a\)](#), [Pradel \(2012\)](#).

Parmi les 160 requêtes à notre disposition, 120 ont été sélectionnées aléatoirement et utilisées pour construire manuellement six patrons de requêtes. L’évaluation du système a été menée sur les 40 requêtes restantes. Nous avons constaté que 37 des 40 requêtes de test sont

couvertes par l'ensemble de patrons construits, c'est-à-dire que leur graphe requête peut être généré à partir du graphe d'au moins un des six patrons de l'ensemble. Nous avons considéré ce taux de couverture des requêtes de test par les patrons de 92,5% comme suffisamment élevé pour corroborer notre hypothèse et nous avons poursuivi nos travaux dans cette direction.

Ces observations nous ont poussé à proposer un mécanisme permettant l'expression de requêtes utilisateur en langue naturelle et leur interprétation s'appuyant sur des patrons de requêtes prédéfinis.

5.2.1.2 Utilisabilité des interfaces en langue naturelle

La pertinence d'utiliser des interfaces en langue naturelle a été considérée à plusieurs reprises dans la littérature, sans aboutir à une conclusion définitive, [Dekleva \(1994\)](#), [Desert \(1993\)](#), [Thompson et al. \(2005\)](#). Dans [Kaufmann and Bernstein \(2010\)](#), les auteurs évaluent l'utilisabilité d'interfaces permettant d'accéder à des connaissances par le biais de systèmes fondés sur l'utilisation de la langue naturelle. Ils identifient trois problèmes principaux :

- l'*ambiguïté linguistique*, autrement dit la multiplicité d'interprétations d'une même expression en langue naturelle est le problème le plus évident ;
- la *barrière d'adaptation*, qui constate la complexité d'adaptation d'un système d'interrogation du domaine d'application pour lequel il a été conçu à un autre domaine. Or les interfaces en langue naturelle les plus performantes sont celles qui ont été conçues pour un domaine d'application clairement délimité ;
- le *problème d'habitabilité* qui réside dans le fait que les utilisateurs sont parfois désarmés face à une trop grande liberté dans l'expression des requêtes : l'utilisateur peut par exemple exprimer des requêtes allant au delà des capacités du système.

La principale contribution de [Kaufmann and Bernstein \(2010\)](#) concerne ce dernier problème : l'*habitability hypothesis* stipule qu'une interface en langue naturelle, pour présenter les meilleures performances d'utilisabilité, doit imposer une certaine structure à l'utilisateur afin de l'aider pendant le processus de formulation de la requête. Afin d'appuyer leur hypothèse, les auteurs décrivent l'étude d'utilisabilité qu'ils ont menée avec de vrais utilisateurs. Dans cet objectif, ils ont développé quatre interfaces qu'ils ont évaluées avec 48 personnes.

Les auteurs rompent avec la classification dichotomique habituelle, séparant les approches entièrement en langue naturelle des approches fondées sur des requêtes formelles. Ils proposent de situer chaque système de requêtes sur un *continuum de formalisation* selon la nature de l'interfaçage avec l'utilisateur. Les interfaces entièrement en langue naturelle sont à une extrémité de ce continuum alors que les approches totalement formelles sont à l'autre extrémité. Les quatre systèmes développés dans le cadre de cette étude ont naturellement été positionnés sur ce continuum : (i) *NLP-Reduce* [Kaufmann et al. \(2007\)](#) est une interface en langue naturelle naïve et indépendante du domaine, dans laquelle les requêtes sont exprimées à l'aide de mots-clés, de fragments de phrases ou de phrases complètes en anglais ; (ii) *Querix* [Kaufmann et al. \(2006\)](#) est une interface indépendante du domaine dans laquelle les requêtes doivent être exprimées sous la forme de questions entièrement en anglais, commençant par "Which...", "What...", "How many...", "How much...", "Give me...", ou "Does..."; (iii) *Ginseng Bernstein et al. (2005)* est un moteur de recherche qui oblige l'utilisateur à formuler sa requête en langue naturelle sous forme d'une phrase respectant une grammaire donnée ; (iv) *Semantic Cristal* est un outil graphique qui aide l'utilisateur à construire sa requête formelle.

Les résultats de l'étude ont montré que les utilisateurs se sentaient beaucoup plus à l'aise avec Querix et son langage d'interrogation, ce qui tend à confirmer l'hypothèse d'habitabilité. Cependant, il est important de noter que les commentaires des utilisateurs et leur perception de l'exactitude des réponses ne reflètent pas la correction des systèmes étudiés. En effet, NLP-reduce a obtenu de meilleures performances sur des critères objectifs, comme le temps moyen d'exécution ou la qualité des résultats. Des interactions supplémentaires avec l'utili-

sateur réalisées par Querix (désambiguïsation de mot explicite, feedback en langue naturelle) semblent améliorer la confiance des utilisateurs dans le système.

Malgré ces résultats, nous estimons au moment où ces travaux ont été menés qu'un problème important allait être de pouvoir exploiter des requêtes exprimées oralement au moyen d'appareils mobiles. Les utilisateurs s'adressent aujourd'hui et de plus en plus à leurs terminaux comme ils le feraient à un ami. Dans ce contexte, l'interprétation de requêtes en langue naturelle pure est incontournable.

Nous retenons néanmoins de cette étude la nécessité d'interagir avec l'utilisateur, en lui procurant notamment des feedbacks. Ce point de vue est renforcé par les travaux de [Damljanovic et al. \(2013\)](#) qui exploitent le feedback et un dialogue de clarification afin d'améliorer l'expérience utilisateur.

5.2.1.3 Approches existantes

Dans cette section, nous présentons les approches proposées pour formuler des requêtes en langue naturelle au moment où nos travaux étaient menés.

Nous nous en tenons ici à la classification proposée dans [Kaufmann and Bernstein \(2010\)](#), et présentons les approches suivant le continuum de formalisation, de la plus formelle à la moins formelle, en commençant par les langages de requêtes eux-mêmes.

À une extrémité du continuum se situent les langages de requête formels comme SPARQL ². Les systèmes présentés dans cette section résultent en des requêtes exprimées dans ces langages. Ces langages présentent de manière évidente des contraintes d'utilisabilité qui les rendent inadaptés aux utilisateurs finals. Exprimer des requêtes formelles implique en effet de connaître et de respecter la syntaxe du langage utilisé, de comprendre un modèle de graphe et, ce qui est le plus contraignant, de connaître le schéma de données de la base de connaissances interrogée. Certains travaux visent à étendre le langage SPARQL et son mécanisme d'interrogation de multiples façons : en prenant en compte des mots-clés et des jokers quand l'utilisateur ne connaît pas exactement le schéma qu'il interroge [Elbassuoni et al. \(2010\)](#), en utilisant des expressions régulières pour exprimer des chemins de propriétés [Alkhateeb et al. \(2009\)](#), ou en ajoutant des capacités de navigation dans des expressions régulières imbriquées [Pérez et al. \(2010\)](#). Mais, là encore, toutes ces approches nécessitent la connaissance du langage SPARQL par l'utilisateur.

Très proches sont les approches assistant l'utilisateur pendant la formulation des requêtes exprimées dans ces langages. Des interfaces très légères comme *Flint*³ et *SparQLed*⁴ mettent en œuvre des fonctionnalités très simples, comme la coloration syntaxique ou l'autocomplétion. D'autres approches reposent sur des interfaces graphiques, comme [Athanasios et al. \(2004\)](#) pour les requêtes RQL, [Clemmer and Davies \(2011\)](#); [Russell and Smart \(2008\)](#) pour des requêtes SPARQL ou [CoGui \(2009\)](#) pour les graphes conceptuels. Même si ces interfaces graphiques sont utiles et rendent la formulation de la requête moins fastidieuse, nous pensons qu'elles ne conviennent pas aux utilisateurs car elles ne dépassent pas les limites d'utilisabilité des langages de requêtes de graphes formels évoquées précédemment.

Sewelis [Ferré and Hermann \(2011\)](#), [Ferré et al. \(2011\)](#), [Ferré and Hermann \(2012\)](#) combine la recherche par facettes et le requêtage, les paradigmes les plus populaires dans l'interrogation sémantique de données, tandis que d'autres approches comme *squall2sparql* [Ferré \(2012\)](#) définissent une langue naturelle contrôlée dont la traduction en SPARQL ne présente pas de difficulté. Avant de pouvoir utiliser ces systèmes correctement, les utilisateurs doivent néanmoins se familiariser avec, ce qui n'est pas simple.

D'autres travaux visent à générer automatiquement – ou semi-automatiquement – des requêtes formelles à partir de requêtes utilisateurs exprimées librement en mots-clés ou en langue naturelle. Notre travail se situe dans cette famille d'approches. L'utilisateur exprime

2. <http://www.w3.org/TR/rdf-sparql-query/>

3. <http://openuplabs.tso.co.uk/demos/sparqleditor>

4. <http://sindicetech.com/sindice-suite/sparqlled/>

son besoin en information de manière intuitive, sans avoir à connaître le langage de requêtes ou les formalismes de représentation de connaissances utilisés dans le système. Certains travaux ont été proposés dans le but d’exprimer des requêtes formelles en différents langages comme SeREQ [Lei et al. \(2006\)](#) ou SPARQL [Zhou et al. \(2007\)](#), [Tran et al. \(2009\)](#), [Cabrio et al. \(2012\)](#). Dans ces systèmes, la génération de requêtes passe par les étapes suivantes : (i) appariement des mots-clés aux entités sémantiques définies dans la base de connaissances, (ii) construction de graphes requêtes liant les entités détectées en explorant la base de connaissances, (iii) classement des graphes requêtes ainsi construits, (iv) sélection par l’utilisateur de la bonne requête. Ces approches s’attèlent à plusieurs problématiques principales : optimiser la première étape en s’appuyant sur des ressources externes comme WordNet ou Wikipedia [Lei et al. \(2006\)](#), [Wang et al. \(2008\)](#), optimiser les mécanismes d’exploration de connaissances utilisés dans la construction des graphes requêtes [Zhou et al. \(2007\)](#), [Tran et al. \(2009\)](#), améliorer le score de classement des requêtes [Wang et al. \(2008\)](#), et améliorer l’identification des relations en utilisant des patrons morpho-syntaxiques [Cabrio et al. \(2012\)](#).

Autosparql [Lehmann and Bühmann \(2011\)](#) étend la catégorie précédente : après une interprétation élémentaire de la requête en langue naturelle, le système converse avec l’utilisateur en lui demandant des exemples positifs et négatifs (éléments qui figurent ou non dans la liste des réponses attendues), afin d’affiner l’interprétation initiale en utilisant un algorithme d’apprentissage. Une étude complète des interfaces en langue naturelle permettant d’interroger le Web Sémantique a été faite dans [Lopez et al. \(2011\)](#) et plus récemment dans [Höffner et al. \(2017\)](#).

Dans la plupart des approches susmentionnées, le graphe de requête final est obtenu en parcourant la base de connaissances et en cherchant à relier les entités qui ont été identifiées dans la requête de l’utilisateur. Cette étape est guidée par certaines heuristiques et peut générer un graphe de requête incohérent ou dépourvu de sens du point de vue de l’utilisateur. Pour notre part, nous proposons de définir et d’exploiter des patrons de requêtes pour guider l’étape de construction de la requête. Ces patrons, que nous proposons de construire manuellement mais qui pourraient l’être de manière automatique, garantissent que les requêtes graphes générées sont toujours cohérentes pour l’utilisateur.

En ce qui concerne la tâche d’évaluation, seuls quelques travaux ont été proposés pour le moment. [Elbedweihy et al. \(2012\)](#) décrit une évaluation de l’utilisabilité et de la satisfaction des utilisateurs pour les approches de requêtes de recherche sémantique, selon les points de vue des experts et des utilisateurs occasionnels. Nous pouvons également mentionner l’initiative QALD (Question Answering over Linked Data), à laquelle nous avons participé avec notre système Swip lors des campagnes de 2011 et 2013.

5.2.2 Vue générale du système Swip

Dans le système Swip (pour Semantic Web Interface using Patterns), le processus d’interprétation de la requête repose sur 2 étapes principales présentées dans la figure 5.1. La première étape, *l’interprétation de la requête en langue naturelle*, consiste à identifier les entités nommées qu’elle contient ainsi que les dépendances qui les lient. Ces dépendances, identifiées entre les chaînes de caractères reconnues dans la requête en langue naturelle, sont explicitement représentées dans ce que nous appelons une *requête pivot*.

La deuxième étape, la *formalisation de la requête pivot*, repose sur la mise en correspondance des patrons de requêtes avec la requête pivot. En sortie de cette étape, nous obtenons des interprétations possibles de la requête utilisateur qui sont classées par rapport à leur pertinence vis-à-vis de la requête initiale, puis sont présentées à l’utilisateur à partir d’une reformulation en langue naturelle du besoin identifié. L’utilisateur peut alors sélectionner la requête qui correspond à son besoin en information.

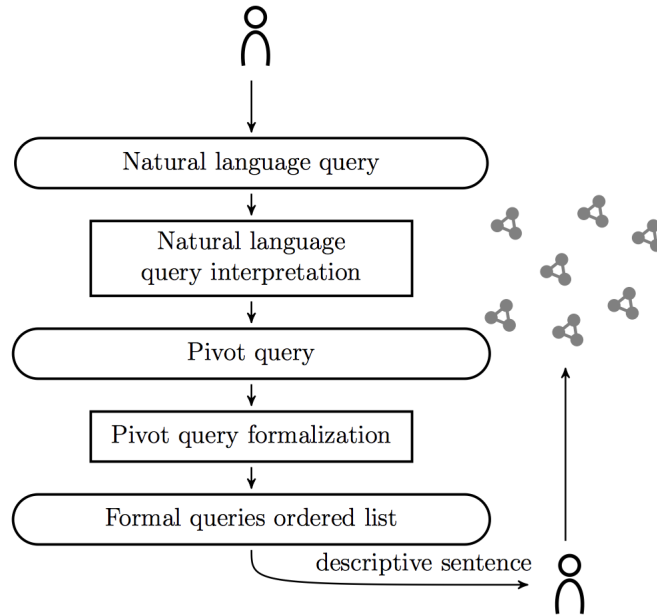


FIGURE 5.1 – Vue d’ensemble du processus d’interprétation de Swip.

5.2.3 De la requête initiale à la requête pivot

5.2.3.1 La requête pivot

Le requête pivot générée à l’issue de la première étape se situe à mi-chemin entre la requête initiale en langue naturelle et la requête formelle visée. Cette requête est générée à partir des mots-clés identifiés dans la requête initiale et de relations identifiées entre eux. Nous utilisons cette représentation intermédiaire pour deux raisons principales. Elle facilite tout d’abord la prise en compte du multilinguisme en proposant ce format intermédiaire : un module spécifique de traduction de la requête en langue naturelle vers le langage pivot peut être spécifié pour les différentes langues que l’on souhaite considérer. Dans nos travaux, nous avons par exemple proposé deux modules : un pour le français et un pour l’anglais. Elle permet de plus de spécifier la sémantique des relations portant sur les mots-clés. Ce type d’informations est ignoré par les approches reposant uniquement sur des mots-clés : il n’est pas possible, avec une simple liste de mots-clés, de différencier une requête cherchant les réalisateurs des films dans lesquels joue Clint Eastwood d’une requête cherchant les acteurs des films réalisés par Clint Eastwood.

La requête pivot est une représentation intermédiaire dont nous avons présenté la grammaire dans [Pradel et al. \(2012\)](#). Nous donnons ici une présentation intuitive de ses caractéristiques.

Le langage pivot est une extension d’une simple liste de mots-clés, permettant la déclaration explicite de relations entre eux. Une requête pivot est composée d’une ou plusieurs sous-requêtes. Une sous-requête contient un ou plusieurs éléments de requête. Un élément de requête peut être vu comme un mot-clé (comme "actor"), un type de littéral (comme `date`), ou un littéral typé (comme `date<2013-04-05>`). Le symbole optionnel “?” devant un élément de requête indique que l’élément est le *focus* de la requête, c’est à dire que l’utilisateur attend des résultats spécifiques en lien avec ce mot-clé. Une requête pivot est composée d’une conjonction de sous-requêtes pouvant être de différents types :

- sous-requête unaire comme ?"actor" qui demande la liste des acteurs présents dans une base de connaissances ;
- sous-requête binaire qui qualifie un mot-clé avec un autre mot-clé ; la requête ?"actor" :

"`married.`" demande la liste des acteurs mariés ;

- sous-requête ternaire qui qualifie par le biais d'un troisième mot-clé la relation entre les deux autres : la requête `"actor": "married to"= "Penelope Cruz"` recherche tous les acteurs qui sont/ont été mariés à Penelope Cruz.

Le type de la requête est indiqué à la fin de la requête pivot à partir de mots réservés : `COUNT` (resp. `ASK`) indique une requête de comptage (resp. dichotomique), aucun mot réservé indiquant une requête liste.

Exemple Une traduction possible de la requête *"When was the album Abbey Road released ?"* est `"Abbey Road": "album". "Abbey Road": "release"= ?date` (ou sous une forme factorisée `"AbbeyRoad": "album"; "release"= ?date`).

5.2.3.2 Étapes de transformation

La transformation d'une requête langue naturelle en requête pivot repose sur l'utilisation d'un analyseur de dépendances syntaxiques qui produit un arbre dont les nœuds représentent des mots de la phrase et les arcs représentent les dépendances grammaticales entre les mots. Comme cela est illustré sur la figure 5.2, la transformation se fait en plusieurs étapes.

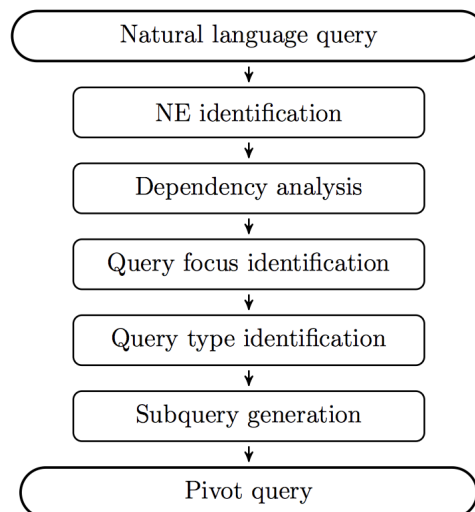


FIGURE 5.2 – Étapes de l'interprétation de la requête en langue naturelle.

Identification des entités nommées Avant d'analyser la requête, la première étape consiste à identifier dans la phrase des références aux entités de la base de connaissances. Ces entités sont identifiées à partir des termes définis dans la base de connaissances pour les désigner. Les chaînes de caractères correspondant à ces termes sont ensuite considérées comme un tout lors de la phase suivante, qui analyse les dépendances syntaxiques. Par exemple, dans la phrase *"Who are the actors of Underworld : Awakening"*, *"Underworld : Awakening"* est considéré comme une entité nommée, alors que les éléments de ponctuation auraient pu mener à considérer qu'il s'agissait de termes indépendants. Cette étape est cruciale dans le cas où les termes désignant les entités sont longs, comme les noms de groupes de musique ou les titres de films.

Analyse de dépendance Un arbre de dépendances est généré à partir de la requête en langue naturelle et les entités préalablement identifiées par un analyseur externe.

Identification du focus de la requête Dans un premier temps, le focus de la requête est identifié, autrement dit le ou les éléments de la requête qui seront considérés comme des variables dans la clause SELECT de la requête SPARQL finale. Pour ce faire, nous avons défini un algorithme permettant de prendre en compte trois cas. Tout d’abord, l’arbre de dépendances est analysé pour identifier un terme lié par une relation de dépendance à un adverbe interrogatif; le lemme de ce mot est alors considéré comme le focus. Par exemple, dans la phrase “*Which films did Ted Hope produce?*”, le focus de la requête sera “*film*”. Si aucune dépendance n’a été identifiée pour l’adverbe interrogatif, nous définissons le focus de la requête par un mot-clé prédéfini dépendant du type de résultats attendus par l’adverbe interrogatif. Par exemple pour la requête “*When did Charles Chaplin die?*”, le focus sera “*date*”. Pour la requête “*Who played in the Artist?*”, le focus est “*person*”. Si aucun adverbe interrogatif n’est identifié dans la phrase, nous cherchons un ensemble prédéfini d’amorces indiquant le type de réponses attendues comme pour les requêtes commençant par “*List*”, “*List all*”, “*I am looking for*”, “*What are*”, “*Give me*”... Dans ce cas, le focus sera le lemme du nom dépendant de l’amorce. Dans la phrase “*List all the films in which Charles Chaplin acted*”, le focus sera “*film*”.

Identification du type de la requête Si aucun focus n’a été identifié, nous considérons que la requête est de type dichotomique. Ceci est par exemple le cas pour “*Was Jean Dujardin involved in the film The Artist?*”.

Dans le cas où un focus a été identifié, nous recherchons l’expression indiquant la nécessité de considérer des agrégats, comme par exemple lorsque l’expression “*How many*” introduit la requête. Cette étape est décrite dans [Amarger et al. \(2013a\)](#).

Génération des sous-requêtes L’arbre de dépendances est de nouveau parcouru pour prendre en compte les différentes propositions de la phrase et les possibles sous-requêtes composant la requête pivot.

Un ensemble de règles définies dans [Pradel et al. \(2011b\)](#) ont été proposées pour extraire les sous-requêtes à partir de la catégorie grammaticale des nœuds et de la nature des dépendances. Nous en illustrons certaines ici. Si deux nœuds font partie du même syntagme nominal, une sous-requête binaire est construite à partir du lemme du nom jouant le rôle de la tête et du lemme du nom jouant le rôle d’expansion. Pour la requête “*Give me all the films of Jean Dujardin*”, la requête binaire générée sera “*?film: "Jean Dujardin"*”. Si trois nœuds font partie de la même clause composée d’un triplet sujet-verbe-complément, une sous-requête ternaire est générée à partir du lemme des nœuds. Pour la clause “*Who played in the Artist?*”, la sous-requête ternaire sera “*?person: "play"= "the Artist"*”. Si un nœud est un pronom relatif, le mot-clé utilisé pour représenter ce nœud sera le lemme du nom auquel il fait référence. Pour la requête “*Who played in a film in which Jean Dujardin also played?*”, la sous-requête pour la seconde clause sera “*film: "play"= "Jean Dujardin"*”, la sous-requête pour la première étant “*person: "play"= "film"*”. Notons que l’usage du même mot-clé permet d’exprimer que, dans les deux sous-requêtes, il s’agit du même film.

Pour chaque nœud, chaque règle est appliquée ce qui signifie que pour une même clause, plusieurs sous-requêtes peuvent être générées. Par exemple, pour la clause “*Was Jean Dujardin involved in the film The Artist?*”, les deux sous-requêtes “*film: "The Artist"*” et “*Jean Dujardin: "involved"= "The Artist"*” sont produites.

Les règles peuvent paraître simples mais nous avons observé que, dans la pratique, la structure des requêtes exprimées par les utilisateurs est généralement simple.

5.2.4 De la requête pivot à la requête finale

5.2.4.1 Patrons de requêtes

Nous présentons ici intuitivement la notion de patrons.

Les exemples 1, 2, 3 et 4 correspondent à des requêtes simples que des utilisateurs pourraient poser sur le domaine du cinéma.

Exemple 1 *Who plays in the movie The Artist ?*

Exemple 2 *Who is the director of the movie The Artist ?*

Exemple 3 *Which movies were directed by Michel Hazanavicius ?*

Exemple 4 *Which movies were directed by the Coen brothers ?*

En considérant qu'ils sont définis pour une base de connaissances du cinéma, les figures 5.3, 5.4, 5.5 et 5.6 représentent des graphes de requêtes qui sont des formalisations possibles de ces exemples. Lors de la génération de ces graphes, la variable `?res` représente la ressource recherchée ; pendant la phase de projection⁵ de ce graphe sur la base de connaissances, l'ensemble des valeurs possibles pour cette variable seront considérées comme étant des réponses à la requête.

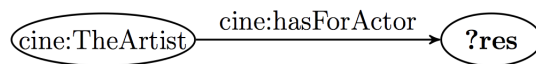


FIGURE 5.3 – Graphe requête pour l'exemple 1

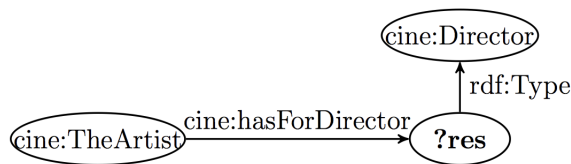


FIGURE 5.4 – Graphe requête pour l'exemple 2

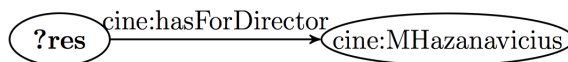


FIGURE 5.5 – Graphe requête pour l'exemple 3

Le patron présenté dans la figure 5.7 est un patron très simple, couvrant les différentes requêtes concernant les acteurs jouant dans un film. Le nœud c_1 est appelé un *élément qualifiant* car il fait référence à la classe `cine:Movie` de la base de connaissances. L'intuition derrière notre approche est qu'une requête pourra être générée en identifiant le patron (autrement dit la famille de requêtes) adapté à la requête, puis en substituant l'élément qualifiant par le nom du film spécifiquement référencé dans la requête. Nous appelons cette étape *instanciation du patron*. L'instanciation du patron 5.7 pour la requête 1 mènera à la création du graphe 5.3, en substituant l'élément qualifiant par la ressource `cine:TheArtist`.

Comme cela est illustré dans la figure 5.7, chaque patron est également composé d'une phrase descriptive. Cette phrase est une succession de chaînes de caractères fixes et d'éléments

5. On appelle projection d'un graphe requête dans une base de connaissances l'opération visant à apparier le graphe requête aux graphes faits afin de chercher des réponses à la requête. Il s'agit en termes d'algorithmique de graphes d'une recherche d'homomorphismes.

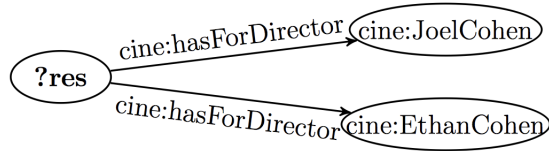


FIGURE 5.6 – Graphe requête pour l'exemple 4

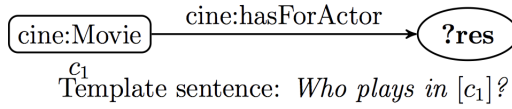


FIGURE 5.7 – Patron issu de la requête 5.3.

faisant référence à un élément qualifiant du graphe. Ces éléments référençant un sommet sont définis pour être potentiellement remplacés par un terme désignant la ressource qui se substituera à l'élément qualifiant lors du processus d'instantiation du patron. Par exemple, l'instantiation du patron 5.7 pour la requête 1 mènera à la génération de la phrase descriptive : “*Who plays in The Artist ?*”. Dans notre approche, nous considérons des constructions plus complexes de phrases descriptives.

De la même façon, nous pouvons construire les patrons 5.8 et 5.9, respectivement inspirés des graphes requêtes 5.4 et 5.5. Ces patrons sont assez similaires, l'un cherchant le réalisateur d'un film et l'autre un film dirigé par un réalisateur. Le patron 5.10 est le résultat de la fusion de ces deux patrons ; ce graphe est donc formé de deux sommets qualifiants c_1 et c_2 , l'un faisant référence à la classe `cine:Movie` et l'autre à la classe `cine:Director`. Comme les éléments qualifiants correspondent à deux classes différentes, la phrase descriptive n'est plus exprimée sous la forme d'une question, afin de pouvoir anticiper le cas où l'un ou l'autre des sommets est le focus de la question. L'instantiation du patron pour la requête 2 sera le graphe requête de la figure 5.11 et la phrase descriptive sera : “*The Artist was directed by some director.*”. La sous-chaîne correspondant au focus est mise en évidence syntaxiquement.

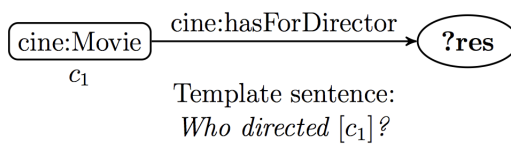


FIGURE 5.8 – Patron proposé pour représenter la requête 5.4

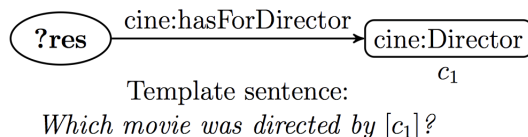


FIGURE 5.9 – Patron proposé pour représenter la requête 5.5

Pour permettre la généralisation du graphe requête 5.6, le patron 5.10 doit être instancié en répétant l'unique triplet qui le compose avec deux instances différentes de la classe `cine:Director`. Nous avons donc défini la possibilité d'indiquer qu'un patron est “répétable” en offrant la possibilité de fixer sa cardinalité. La figure 5.12 montre un exemple de patron qui peut être répété n fois et son instantiation pour la requête 4 générera le graphe présenté en

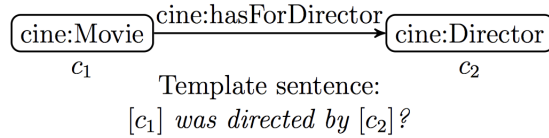


FIGURE 5.10 – Patron correspondant à la fusion des patrons 5.8 and 5.9



FIGURE 5.11 – Graphe requête issu de l’instantiation du patron 5.10 pour la requête de l’exemple 2.

figure 5.6 avec la phrase descriptive “**A movie** was directed by Joel Coen and Ethan Coen.”

De la même façon, il est possible de préciser qu’un sous-graphe du patron, que nous appelons sous-patron, peut avoir une cardinalité minimale. Dans le cas où la cardinalité minimale est 0, ce sous-patron est optionnel. Le patron présenté dans la figure 5.13 est une évolution du patron 5.12, contenant un triplet optionnel pour représenter les acteurs qui jouent dans un film.

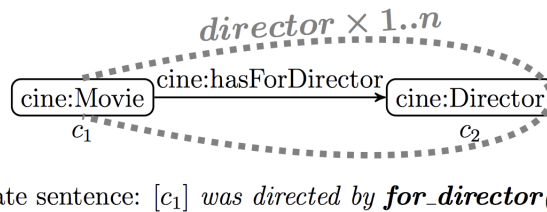


FIGURE 5.12 – Exemple de patron complexe, contenant un sous-patron répétable.

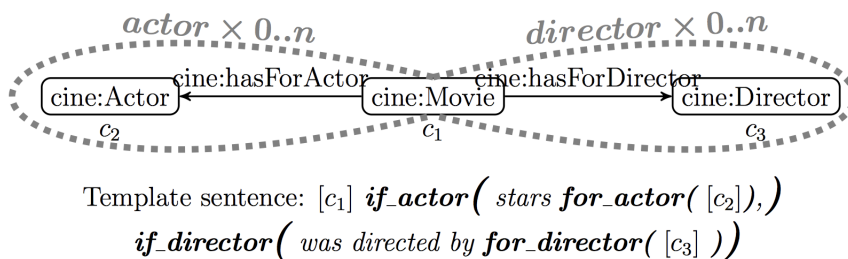


FIGURE 5.13 – Exemple de patron complexe, contenant des sous patrons optionnels et répétables.

Nous ne présentons pas la définition formelle des patrons de requêtes ni l’algorithme de construction de la requête finale à partir de la requête utilisateur, le tout étant présenté de manière exhaustive dans Pradel et al. (2013).

5.2.4.2 Génération de la requête formelle

La traduction en la requête pivot en une requête graphe se fait par l’intermédiaire des patrons. La figure 5.14 illustre toutes les étapes de ce processus.

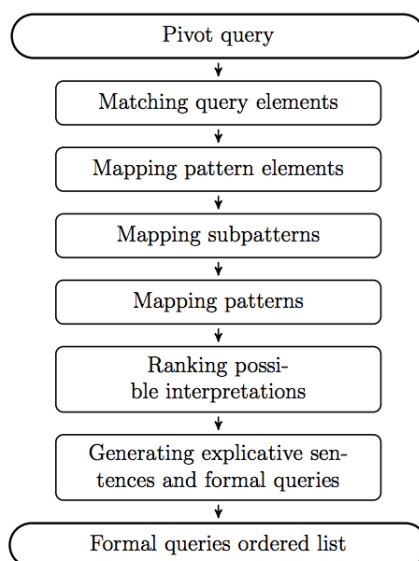


FIGURE 5.14 – Étapes de l’interprétation de la requête pivot.

Dans un premier temps, des correspondances sont établies entre les éléments de la requête utilisateur et les éléments de la base de connaissances (instances, classes, propriétés et types de données).

Les étapes suivantes consistent à associer chaque patron de requêtes à la requête pivot à traiter, de façon conforme aux correspondances préalablement établies. Cette opération est réalisée de façon ascendante : tout d’abord sont générées des associations entre éléments de patron et éléments de requête, puis entre sous-patron et requête pivot, en commençant par les sous patrons basiques (ceux qui ne contiennent aucun autre sous-patron), et en élargissant progressivement aux sous-patrons englobants, jusqu’à obtenir les associations du patron lui-même. Chaque correspondance de patron obtenue est une interprétation potentielle de la requête pivot.

Dans le but de présenter à l’utilisateur les interprétations les plus pertinentes en priorité, une score de pertinence est calculée pour chaque association de patron. Ce score reflète la fiabilité supposée de l’association de patron concernée, et donc de l’interprétation qu’elle incarne. Les associations sont ensuite ordonnées selon cette note avant d’être soumises à l’utilisateur.

Enfin, la génération pour chaque association de patron de la phrase descriptive (la phrase présentée à l’utilisateur représentant le sens la requête) et de la requête SPARQL (la requête exécutée sur la base de connaissances si la phrase descriptive associée est sélectionnée par l’utilisateur) se fait facilement, grâce au modèle de phrase descriptive et au graphe de chaque patron.

Les détails de ces étapes sont décrits dans [Pradel et al. \(2013\)](#).

5.2.5 Implementation tout en SPARQL

Un prototype de notre approche a été mis en place afin d’évaluer son efficacité. Il a été implémenté sous forme de services Web et utilise TreeTagger [Schmid \(1994\)](#) pour le POS tagging et MaltParser [Nivre et al. \(2006\)](#) pour l’analyse des dépendances dans la requête utilisateur. Le système réalise la deuxième étape principale du processus (traduction de la requête pivot en requête formelle) en exploitant un serveur SPARQL fondé sur le moteur de requêtes ARQ⁶. L’interprétation des requêtes pivots est réalisée en exploitant les capacités du

6. <http://openjena.org/ARQ/>

moteur SPARQL. En effet, une des fonctionnalités principales de SPARQL consiste à appairer des graphes, ce qui est précisément le but de cette étape d'interprétation. Cette tâche est donc entièrement réalisée par le biais d'exécutions de requêtes SPARQL. Les résultats de chacune des étapes sont systématiquement enregistrés dans la base de connaissances, à l'aide d'updates SPARQL, ce qui les rend disponibles pour les étapes suivantes.

Notre implémentation s'appuie sur deux ontologies : l'une pour représenter les patrons, l'autre pour représenter les graphes requêtes. Elles sont présentées dans [Pradel et al. \(2014\)](#).

L'architecture de notre système reposant sur SPARQL présente plusieurs intérêts. L'un des plus évidents est la facilité d'utilisation d'un système de cache. Le résultat de chaque étape de traitement étant validé dans l'entrepôt RDF, il est facile de réexploiter les données précédemment générées. Par exemple, le système Swip réalise qu'une requête pivot entrante a déjà été traitée lorsque son URI est déjà présente dans l'entrepôt et que le résultat de son interprétation peut être directement retourné. En outre, la correspondance d'un mot-clé donné n'est effectuée qu'une seule fois pour toutes les requêtes contenant ce même mot-clé.

De plus, avec cette architecture, il existe un asynchronisme total et naturel entre les côtés client et serveur : une fois l'URI de requête traitée, elle est renvoyée au client qui peut ensuite mettre à jour les résultats de l'interprétation de manière progressive et indépendante du serveur, en interrogeant directement l'entrepôt RDF. Cette approche garantit également une homogénéité et une cohérence totales puisque les données d'entrée, les résultats intermédiaires et les sorties sont stockés et manipulés au moyen de technologies standard. Il convient également de noter que, bien que la première partie de l'approche ait été mise en œuvre de manière plus "traditionnelle" à l'aide de services Web, elle pourrait être modifiée en exploitant l'initiative *NLP2RDF* [Hellmann et al.](#) visant à fournir les résultats des outils populaires de TAL dans un format commun, appelé *NIF* (NLP Interchange Format) et intégré dans le cadre du Web sémantique.

Un inconvénient contrebalance ces avantages : le manque de contrôle sur le traitement des requêtes SPARQL et donc sur les performances globales du système. Le serveur SPARQL est utilisé comme une boîte noire et son efficacité influence directement celle du processus d'interprétation. L'expérience a montré que le serveur ARQ n'était pas adapté aux grandes requêtes (plus de vingt triplets dans plusieurs sous-requêtes) et que le fractionnement de ces requêtes complexes en un ensemble équivalent de requêtes successives était beaucoup plus efficace.

5.2.6 Aperçu de l'évaluation

Le système Swip a été évalué dans le cadre de la tâche 1 du défi QALD-3⁷. La méthode d'évaluation a été définie par les organisateurs du défi. Elle consiste à calculer, pour chaque requête test, la précision, le rappel et la F-mesure de la traduction SPARQL renvoyée par le système, par rapport à une requête (gold standard) faite à la main. Nous avons participé aux deux sous-tâches proposées par les organisateurs du défi, l'une ciblant DBpedia⁸ et l'autre ciblant un export RDF de Musicbrainz⁹ fondé sur la *music ontology* [Raimond et al. \(2007\)](#). Bien que cette tâche proposât des requêtes en langue naturelle dans plusieurs langues, nous n'avons pris en compte que les questions en anglais. La qualité des résultats varie en fonction de la base de connaissances cible. Une analyse détaillée et critique de ces résultats est présentée dans [\(Pradel et al., 2013\)](#). Nous résumons maintenant cette analyse.

Patrons Nous avons défini des patrons de requêtes à partir des jeux de données fournis lors des campagnes QALD [Cabrio et al. \(2013\)](#) en 2011 et 2013 : QALD-1¹⁰ et QALD-

7. <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=task1&q=3>

8. <http://dbpedia.org>

9. <http://musicbrainz.org/>

10. <http://www.sc.cit-ec.uni-bielefeld.de/qald-1>

3¹¹. Nous avons considéré la tâche portant sur l'interrogation de Musicbrainz¹², base de connaissances du domaine de la musique et comportant plus de 50 millions de triplets. Nous avons construit des patrons de requêtes à partir de 100 requêtes d'entraînement fournies lors de la campagne de 2013. Nous avons identifié cinq patrons de requêtes. Parmi les 50 questions de test, seulement deux correspondaient à des graphes requêtes qui ne pouvaient pas être générés à partir de l'un de nos 5 patrons, si l'on ne tient pas compte des requêtes classées hors champ (out of scope) par les organisateurs – auxquelles la base de connaissances ne pouvait de toute façon pas répondre. Ce taux de couverture élevé montre que notre hypothèse de base est fondée pour ce qui concerne les besoins en information portant sur des bases de connaissances relatives à un domaine circonscrit.

Aperçu des résultats Le jeu de données de test sur Musicbrainz était composé de 50 questions en langue naturelle. Nous en avons traité 33. 24 ont été interprétées correctement, 2 ont reçu une réponse partielle et les autres ont échoué. La précision moyenne, le rappel et la F-mesure, calculés par les organisateurs du défi, sont tous égaux à 0,51. Il est difficile d'évaluer ces performances, aucun autre participant au défi ne s'étant attaqué au jeu de données Musicbrainz.

Cependant, ces résultats sont plutôt bons comparés à ceux obtenus par les participants du jeu de données DBpedia. En effet, parmi les autres participants, le système *squall2sparql* affichait de loin les meilleures performances, mais ce système n'accepte pas les phrases complètes en langue naturelle comme entrée Ferré (2012). La deuxième meilleure F-mesure d'un système participant est de 0,36.

Bien entendu, la pertinence d'une telle comparaison directe est très discutable, car les deux bases de graphes présentent des différences significatives. Notre système a d'autre part obtenu des résultats médiocres sur DBpedia.

Les résultats obtenus par Swip sur l'ensemble de données DBpedia, composé de 100 questions tests, ne sont pas aussi bons que ceux de Musicbrainz. Nous expliquons ces résultats par le fait que les requêtes sur cette base de données sont beaucoup plus variées que les requêtes que l'on peut poser sur une base de données couvrant un domaine particulier. Les patrons que nous avons réalisés sur le jeu d'apprentissage ne couvraient que 19% des requêtes du jeu de test. Par ailleurs, les triplets contenus dans DBpedia ne font pas forcément référence aux entités définies dans l'ontologie DBpedia ce qui rend la prise en compte de patrons définis à partir de l'ontologie moins intéressante.

Prise en compte des agrégats Afin d'améliorer les performances de Swip, nous avons ajouté des fonctionnalités permettant de détecter et de prendre en compte dans la traduction SPARQL des expressions simples impliquant des agrégats, les questions impliquant des fonctions d'agrégat représentant une proportion significative des requêtes non prises en charge par notre système. Ces fonctionnalités, décrites dans (Amarger et al., 2013a), consistent essentiellement à identifier des modèles textuels traduits à l'aide d'agrégats dans le jeu de données d'apprentissage et, le cas échéant, de l'élément de requête qui est l'objet de la fonction d'agrégat. Ces informations sont ensuite ajoutées à la requête pivot dont la définition a été légèrement mise à jour pour l'occasion. Par exemple, la question 29 de l'ensemble de données de test QALD concernant Musicbrainz "*Quels groupes ont enregistré plus de 50 albums ?*" est traduite en "band": "record" = "album (50+)". La deuxième étape principale du processus, l'appariement des patrons à la requête pivot, reste la même jusqu'à la génération de la requête SPARQL proprement dite, où des fonctions d'agrégat sont ajoutées en fonction du contenu de la requête pivot.

11. <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=task1&q=3>

12. <http://musicbrainz.org/>

Nouveaux résultats Nous avons réévalué les traductions des requêtes portant sur Musicbrainz (grâce à l’outil d’évaluation¹³ mis à disposition par les organisateurs), en répondant cette fois à toutes les requêtes, y compris les requêtes hors champ, afin de connaître les résultats que nous aurions obtenus en prenant en compte ces caractéristiques de la méthode d’évaluation. Nous avons ainsi pu valider les quatre requêtes hors champ du jeu de test. Nous avons également corrigé au préalable une coquille dans un patron qui avait empêché l’interprétation correcte de la requête 11, cet impair relevant d’une maladresse de notre part et non d’une faiblesse de l’approche. Après ces modifications, nous avons obtenu des valeurs moyennes de précision, de rappel et de F-mesure de 0,61.

5.3 Interroger plusieurs bases de connaissances

De plus en plus de bases de connaissances dont les domaines se recouvrent sont aujourd’hui publiées sur le LOD. Tirer profit de l’ensemble des bases de connaissances qui peuvent répondre à un même besoin en information est primordial pour satisfaire les utilisateurs finals.

Afin de permettre l’interrogation simultanée de plusieurs bases de connaissances portant sur le même domaine mais fondées sur des ontologies différentes, il est nécessaire d’établir des correspondances entre ces bases de connaissances. Cependant, réécrire des requêtes SPARQL manuellement tenant compte de ces correspondances entre vocabulaires est une tâche fastidieuse.

L’approche que nous proposons consiste à considérer que des correspondances entre les bases de connaissances existent – elles ont été générées manuellement ou automatiquement – et qu’en les exploitant, nous pouvons traduire automatiquement la requête SPARQL. Ce travail faisant suite aux travaux présentés dans la section précédente, nous restreignons les requêtes SPARQL que nous considérons à celles qui peuvent être générées par le système Swip.

L’originalité de notre approche est de considérer des alignements complexes entre les bases de connaissances. Ce type de correspondances est présenté dans le chapitre 2. L’intérêt des correspondances simples est largement reconnu dans le domaine de l’alignement d’ontologies [Euzenat and Shvaiko \(2013\)](#). Cependant, la possibilité d’établir des correspondances plus expressives est nécessaire afin d’aligner un plus grand ensemble d’entités entre bases de connaissances [Thiéblin et al. \(2019\)](#) et donc de traduire de façon plus complète une requête.

Contrairement à l’orientation prise dans les travaux présentés dans la section 3.2, la finalité des travaux présentés dans cette section est d’exploiter la connaissance représentée dans plusieurs bases de connaissances sans avoir à les transformer ou les fusionner. Notre objectif est ici de préserver l’aspect décentralisé du Web, qui s’applique également au LOD, et de permettre à l’utilisateur de tirer profit de l’ensemble des informations disponibles tout en tenant compte du fait que différents points de vue sur un domaine peuvent coexister.

Ce travail a été présenté initialement dans [Thiéblin et al. \(2016\)](#) et [Thiéblin et al. \(2017b\)](#), puis étendu dans [Thiéblin et al. \(2018\)](#).

5.3.1 Motivations

Les ontologies d’un même domaine suivent souvent des choix de modélisation différents. Nous illustrons dans cette section cette hétérogénéité à partir de trois bases de connaissances décrivant des classifications de taxons. Nous considérons les bases de connaissances Agronomic

13. <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=evaltool&q=3>

Taxon¹⁴, DBpedia¹⁵, et TAXREF-LD¹⁶.

Pour chacune des bases de connaissances considérées, nous représentons à partir d'une adaptation du langage de visualisation VOWL Lohmann et al. (2016), le vocabulaire défini pour représenter la notion de *triticum durum taxon*¹⁷.

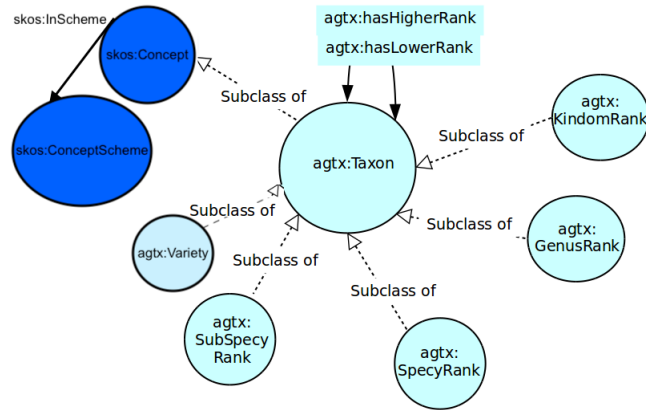


FIGURE 5.15 – Fragment de l'ontologie Agronomic Taxon

Agronomic Taxon Comme nous l'avons décrit dans le chapitre 4 et illustré sur la figure 5.15, Agronomic Taxon représente un taxon à partir d'une classe centrale `agtx : Taxon`, elle-même sous-classe de `skos : Concept`. `agtx : Taxon` est spécialisé en différentes sous-classes représentant chacun des rangs utilisés pour décrire un taxon. Les propriétés `agtx : hasHigherRank` et `agtx : hasLowerRank` décrivent la hiérarchie entre les différents rangs taxonomiques. Pour chaque `agtx : Taxon` les noms vernaculaires et/ou scientifiques peuvent être représentés (même s'ils n'apparaissent pas dans la figure).

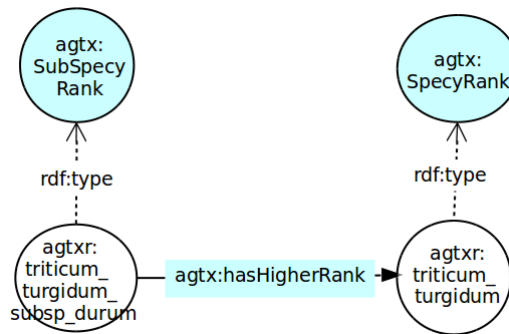


FIGURE 5.16 – Triticum durum taxon représenté dans Agronomic Taxon

La figure 5.16 illustre le fait que, dans Agronomic Taxon, le taxon *triticum_turgidum_subsp_durum* est une sous-espèce qui a pour rang taxonomique supérieur l'espèce *triticum_turgidum_durum*.

14. AgronomicTaxon ontology PREFIX agt : <http://ontology.irstea.fr/agronomictaxon/core/>
 AgronomicTaxon instances PREFIX agtxr : <http://ontology.irstea.fr/agronomicTaxon/resource/goldStandard/>

15. DBpedia ontology PREFIX dbo : <http://dbpedia.org/ontology/>
 DBpedia instances PREFIX dbr : <http://dbpedia.org/resource/>

16. TAXREF-LD properties PREFIX txrfp <http://taxref.mnhn.fr/lod/property/>
 TAXREF-LD taxonomic ranks PREFIX txrfrk : <http://taxref.mnhn.fr/lod/taxrank/>
 TAXREF-LD instances PREFIX txr : <http://taxref.mnhn.fr/lod/>

17. Dans certaines sources, *triticum durum* peut aussi être désigné par *triticum turgidum subsp durum*

DBpedia DBpedia¹⁸ Auer et al. (2007) couvre différents domaines, dont celui qui nous concerne. DBpedia contient de nombreuses instances de taxons, mais le vocabulaire utilisé pour les décrire n'est que partiellement documenté, ce qui rend difficile sa compréhension. Comme cela est illustré dans la figure 5.17, chaque taxon est une instance d'espèce donc de `dbo:Species`, peu importe le rang taxonomique représenté. Ce choix de modélisation est un exemple d'approximation typique de ce que l'on peut trouver dans l'ontologie DBpedia. Des propriétés différentes sont définies pour représenter les différents rangs (dont l'espèce). La figure 5.18 illustre la description du taxon *triticum durum* dans DBpedia et montre qu'il est décrit comme ayant pour genre le taxon *Triticum*.

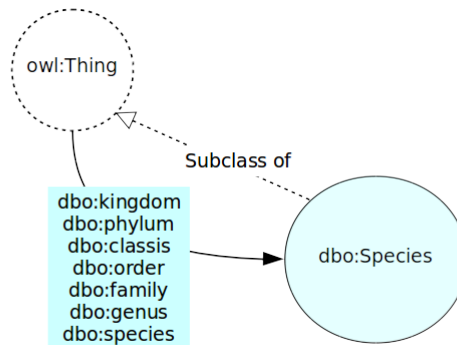


FIGURE 5.17 – Fragment de l'ontologie de DBpedia

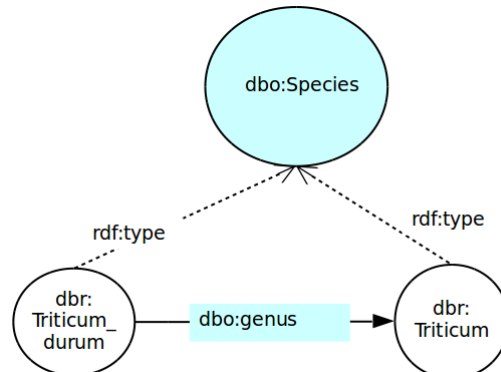


FIGURE 5.18 – *Triticum durum* taxon dans DBpedia

TAXREF-LD est le resultat de la publication de la classification française TAXREF sur le LOD à partir des travaux présentés dans Michel et al. (2017).

Contrairement à Agronomic Taxon et DBpedia qui se concentrent sur la représentation d'un taxon et de ses rangs taxonomiques, TAXREF-LD permet également de modéliser le fait qu'un taxon caractérise un ensemble de spécimens inventoriés dans le monde. Dans cette optique-là, comme cela est illustré dans la figure 5.19, un taxon est à la fois représenté par une classe (pour regrouper les spécimens) et une instance pour pouvoir décrire ses rangs taxonomiques. La classe `txrfrk:TaxonomicRank` permet de typer les taxons qui ont un rang taxonomique. Elle est spécialisée par différentes classes représentant les types de rangs (non représentés sur la figure). La propriété `txrfp:hasRank` lie un taxon à son rang taxonomique. Les propriétés `txrfp:hasReferenceName` et `txrpk:hasSynonym` lient la classe regroupant les

18. <http://dbpedia.org/> prefixes : *dbo* ontology, *dbr* instances

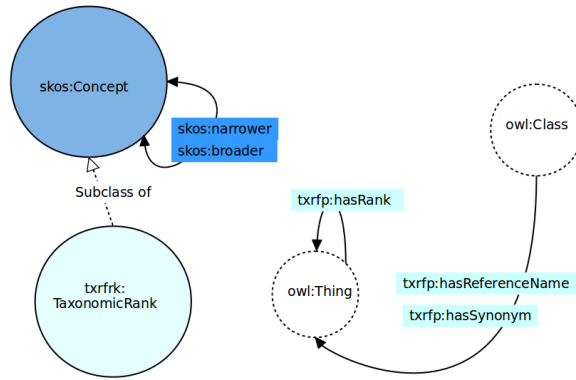


FIGURE 5.19 – Fragment de l'ontologie TAXREF-LD

AgronomicTaxon
<pre> SELECT ?specy WHERE { ?specy rdf:type agtx :Taxon. ?taxon agtx :prefScientificName ?label ?taxon agtx :hasLowerRank+ ?specy. FILTER (regex(?label, "triticum\$", "i")). } </pre>

TABLEAU 5.1 – Requête exprimée sur Agronomic Taxon pour retrouver les sous-taxons de *Triticum*.

spécimens à un taxon. La figure 5.20 présente la description du taxon *triticum durum*, nommé *triticum turgidum subsp durum*, représenté à la fois comme une classe et une instance. Son rang taxonomique est spécifié par la propriété txrfrp : hasTaxonomicRank. Dans ce cas, son rang est txrfrk : Subspecies. La hiérarchie taxonomique est également précisée par la propriété skos : broader entre les instances de skos : Concept.

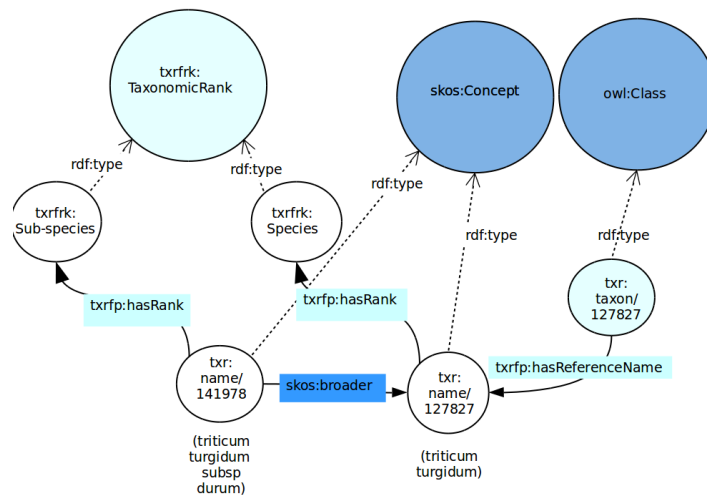


FIGURE 5.20 – Letriticum durum taxon représenté dans TAXREF-LD

Correspondances utiles pour rechercher les de taxons spécialisant un taxon donné

Pour illustrer notre approche, considérons que nous cherchons à retrouver tous les taxons de rang taxonomique plus spécifique que le taxon *Triticum*. La requête présentée dans le tableau 5.1 permet de répondre à ce besoin en information sur Agronomic Taxon.

Bien qu'Agronomic Taxon et TaxREF-LD réutilisent le vocabulaire SKOS, la notion de rang taxonomique est représentée de façon différente dans les trois sources.

On comprend aisément que des correspondances simples ne sont pas assez expressives pour pouvoir représenter les correspondances entre toutes les entités mentionnées dans la

requête.

Différents types de correspondances sont utiles pour transformer notre requête :

- des correspondances exprimant des relations d'équivalence :
 - (1) $\forall x, agtx : Taxon(x) \equiv dbo : Species(x)$ est une correspondance simple qui indique que toute instance x typée avec la classe *Taxon* de l'ontologie Agronomic Taxon est également typée par la classe *Species* de l'ontologie DBpedia.
 - (2) $\forall x, agtx : Taxon(x) \equiv \exists y, txrfp : hasRank(x,y) \wedge txrfp : TaxonomicRank(y)$ est une correspondance complexe qui exprime que toute instance x typée par la classe *Taxon* dans l'ontologie Agronomic Taxon est liée par la propriété *hasRank* de TAXREF-LD à une instance y elle même typée avec la classe *TaxonomicRank*.
- des correspondances reposant sur la relation de subsumption :
 - (3) $\forall x,y, agtx : hasLowerRank(x,y) \geq dbo : Species(x) \wedge dbo : Species(y) \wedge (dbo : genus(x,y) \vee dbo : family(x,y) \vee dbo : order(x,y) \vee dbo : classis(x,y) \vee dbo : phylum(x,y) \vee dbo : kingdom(x,y))$ est une correspondance complexe qui exprime que SI deux instances x et y sont liées par la relation *hasHigherRank* d'Agronomic Taxon ALORS x et y sont typées par la classe *Species* de DBpedia et x et y sont liées par la propriété *genus* ou *family* ou *order* ou *classis* ou *phylum* ou *kingdom* de DBpedia.
- des correspondances reposant sur la relation de subsumption :
 - (4) $\forall x,y, agtx : hasLowerRank(x,y) \leq \exists a,b, txrfp : hasRank(x,a) \wedge txrfp : hasRank(y,b) \wedge txrfk : TaxonomicRank(a) \wedge txrfk : TaxonomicRank(b) \wedge skos : broader+(y,x)$ ¹⁹ est une correspondance complexe qui exprime que si deux instances x et y sont liées par au moins une relation *skos : broader* et qu'il existe deux instances a et b tel que a est le rang taxonomique de x et que b est le rang taxonomique de y alors x et y sont liées par la relation *hasLowerRank* d'Agronomic Taxon.
 - (5) $\forall x,y, agtx : prefScientificName(x,y) \leq rdfs : label(x,y)$ est une correspondance simple exprimant la spécialisation des deux propriétés.
 - (6) $\forall x,y, agtx : prefScientificName(x,y) \leq skos : prefLabel(x,y)$

Un requête SPARQL est intrinsèquement dépendante des ontologies utilisées pour décrire les données. Pour fédérer les informations contenues dans différentes bases de connaissances, les requêtes doivent être adaptées à chacune d'elles. La tâche consistant à réécrire une requête écrite pour une ontologie o_1 en une requête adaptée à l'ontologie o_2 depend des liens établis entre o_1 et o_2 . Si les ontologies partagent une partie de leur vocabulaire (certaines classes et propriétés de SKOS dans nos exemples) pour ces entités, aucune transformation n'est à apporter dans la requête. Si les ontologies sont liées par des correspondances simples ou complexes, la requête doit être réécrite.

Une approche naïve pour la réécriture de requêtes SPARQL consiste à remplacer les IRI des entités de la requête initiale par les entités équivalentes dans l'alignement en utilisant des correspondances simples. Cette approche est intégrée à l'API d'alignement [David et al. \(2011\)](#). Cependant, cette approche ne prend en compte que les relations d'équivalence (la spécialisation ou la généralisation ne sont pas considérées). Une autre approche plus sophistiquée est présentée dans [Makris et al. \(2010\)](#) et [Makris et al. \(2012\)](#). Elle applique un ensemble de règles prédéfinies reposant sur des correspondances complexes impliquant des expressions pour les classes, les propriétés d'objet, les propriétés de types, et les instances. Cependant, comme dans [Zheng et al. \(2012\)](#) et [Correndo et al. \(2010\)](#), les correspondances complexes considérées ne sont pas exprimées dans un langage propre aux systèmes. [Correndo and Shadbolt \(2011\)](#) étendent leur approche initiale en permettant l'expression des correspondances via un sous-ensemble des constructeurs définis dans EDOAL²⁰, langage qui tient lieu de

19. le symbole $+$ représente la transitivité de la relation

20. <http://alignapi.gforge.inria.fr/edoal.html>

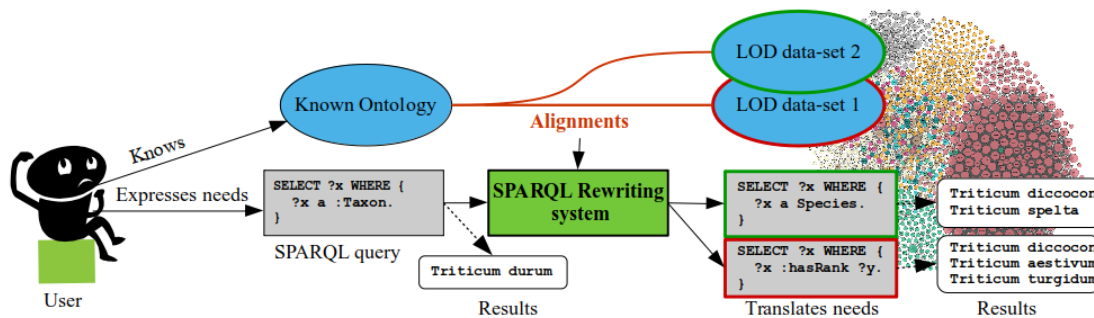


FIGURE 5.21 – Workflow global de l’approche

référence pour exprimer les correspondances entre ontologies Euzenat and Shvaiko (2013), Thiéblin et al. (2019). L’approche que nous avons proposée dans Thiéblin et al. (2016) repose sur des correspondances exprimées en EDOAL et prend en compte un ensemble plus important d’expressions telles que la restriction de propriété d’objets à partir de cardinalité (appelée dans EDOAL *Class by Attribute Occurrence expression*).

Certaines approches ont également été proposées dans le but d’interroger simultanément plusieurs bases de connaissances sur le LOD en aidant l’utilisateur à adapter l’expression de son besoin en information. L’approche proposée dans Torre-Bastida et al. (2013), par exemple, repose sur la prise en compte de correspondances explicitement décrites dans les bases de connaissances (par les prédicats `owl:sameAs`, `owl:equivalentClass`, ou `owl:equivalentProperty` par exemple) pour transformer automatiquement les requêtes. La même approche est suivie dans le projet SemaGrow Lokers et al. (2014) dans lequel des alignements simples entre les différentes bases de connaissances sont utilisés pour traduire les requêtes. La performance de ces méthodes dépend de la façon dont les bases de connaissances sont explicitement liées. En général, seules des correspondances entre instances (via le prédicat `owl:sameAs`) sont exprimées, ce qui limite la reformulation Zaveri et al. (2016). L’approche présentée dans Benedetti et al. (2015) aide les utilisateurs à exprimer leur besoin par le biais d’une interface graphique qui présente un résumé de l’ontologie via ses concepts et propriétés principales. Plus intuitif, SimplePARQL Djebali and Raimbault (2015) propose une façon de formuler une requête en utilisant des termes plutôt que les URI des entités de l’ontologie. Cette approche impose que les entités de l’ontologie soient exhaustivement décrites avec les termes qui permettent de les désigner en langue naturelle. En comparaison, notre approche cherche à limiter l’intervention humaine au moment de l’interrogation en exploitant des correspondances complexes préalablement exprimées.

5.3.2 Illustration de l’approche de réécriture

Nous avons proposé dans Thiéblin et al. (2016) une approche permettant la réécriture de requêtes à partir de correspondances complexes. La figure 5.21 donne une vue globale de l’approche. Nous en présentons ici les principes.

Notre approche repose sur l’hypothèse selon laquelle des correspondances complexes existent. Ces correspondances peuvent être générées à partir d’approches de la littérature que nous avons classifiées dans Thiéblin et al. (2019). Dans les travaux présentés dans Thiéblin et al. (2018) et Thiéblin et al. (2017a), elles ont été générées manuellement et représentées en EDOAL.

Les requêtes initiales considérées par l’approche sont du type $Q_{o_1} = SELECT\ DISTINCT? (Var + | '*')WHERE \{T_{Q_{o_1}}\}$, où Var correspond à l’ensemble des variables utilisées comme attributs de projection et $T_{Q_{o_1}}$ un ensemble de

Agronomic Taxon	DBpedia	TAXREF-LD
<pre>SELECT ?specy WHERE { ?specy rdf :type agtx :Taxon. ?taxon agtx :prefScientific- Name ?label. ?taxon agtx :hasLower- Rank+ ?specy. filter (regex(?label, "triticum\$", "i")). }</pre>	<pre>SELECT ?specy WHERE { ?specy rdf :type dbo :Species. ?taxon rdfs :label ?label. ?specy rdf :type dbo :Spe- cies. ?taxon rdf :type dbo :Species. { ?specy dbo :kingdom ?taxon. } UNION { ?specy dbo :phy- lum ?taxon. } UNION { ?specy dbo :classis ?taxon. } UNION { ?specy dbo :order ?taxon. } UNION { ?specy dbo :fa- mily ?taxon. } UNION { ?specy dbo :genus ?taxon. }</pre>	<pre>SELECT ?specy WHERE { ?specy txrfrp :hasRank ?va- riable_temp0. ?variable_temp0 rdf :type txrfrk :TaxonomicRank. ?taxon skos :prefLabel ?label. ?specy skos :broa- der+ ?taxon. ?taxon txrfrp :has- Rank ?variable_temp1. ?va- riable_temp1 rdf :type txrfrk :Taxo- nomicRank. ?specy txrfrp :has- Rank ?variable_temp2. ?va- riable_temp2 rdf :type txrfrk :Taxo- nomicRank. filter (regex(?label, "triticum\$", "i")). }</pre>

TABLEAU 5.2 – Requête initiale exprimée sur Agronomic Taxon pour retrouver les sous-taxons de *Triticum* et les requêtes pour DBpedia et TAXREF-LD obtenues par réécriture automatique.

triplets exprimés pour une base de connaissances o_1 . Conçue pour les requêtes générées par Swip, l’approche ne considère que les requêtes dont les triplets appartenant à $T_{Q_{o_1}}$ ont pour sujet une variable.

L’algorithme détaillé dans Thiéblin et al. (2016) produit, à partir de $T_{Q_{o_1}}$ et de l’alignement complexe $A_{o_1 \rightarrow o_2}$, $T_{Q_{o_2}}$ qui contiendra les triplets exprimés à partir d’entités de o_2 .

L’algorithme prend en compte les différents types d’expressions qui peuvent être exprimées dans les correspondances complexes mais se limite aux correspondances établies pour des expressions simples (c’est à dire exprimées à partir de l’identifiant) de o_1 .

L’algorithme traduit itérativement chaque triplet en mettant en place un processus de transformation dépendant du fait que la correspondance porte sur une classe ou sur une propriété.

Pour la requête présentée dans le tableau 5.1, le premier triplet (`?specy rdf : type agtx : Taxon`) est réécrit en utilisant la correspondance (1) pour DBpedia et (2) pour TAXREF-LD. Le second triplet est généré à partir des correspondances (5) et (6) alors que le dernier l’est à partir des correspondances (3) et (4). Les clauses FILTER et SELECT sont identiques car elles n’impliquaient aucun élément mis en correspondance. Le tableau 5.2 présente le résultat du processus de réécriture.

5.3.3 Évaluations

Le système de réécriture de requêtes et l’ensemble des correspondances complexes considérées utilisés pour évaluer l’approche sont disponibles en ligne²¹.

Nous avons évalué notre approche sur la transformation de requêtes correspondant à 5 besoins en information d’experts de l’agronomie. Nous avons considéré 4 bases de connaissances correspondant aux bases de connaissances préalablement présentées, ainsi qu’AGROVOC²².

21. https://framagit.org/IRIT_UT2J/sparql-translator-complex-alignment

22. <http://aims.fao.org/aos/agrovoc>

L'ensemble des requêtes (initiales et générées) sont disponibles en ligne²³. Ces évaluations ont montré que des informations complémentaires étaient contenues dans les sources et que l'approche permettait de facilement les retrouver. Pour pallier les différents niveaux de granularité des modélisations, des correspondances définies à partir de la relation de subsumption ont été considérées. Prendre en compte les correspondances établies pour des entités plus spécifiques que celles de la requête initiale permet d'obtenir des résultats, même si ceux-ci ne couvrent qu'une partie du besoin. Prendre en compte des entités plus générales permettait également de retrouver des résultats pertinents, car la transformation des autres triplets de la requête avait pour effet de contraindre la description de ces entités.

Plusieurs limites de notre approche ont été identifiées. Tout d'abord, les requêtes obtenues automatiquement ne sont pas optimisées, ce qui peut poser problème pour l'interrogation de grandes bases de connaissances. Une interprétation globale de la requête en lieu et place d'une interprétation triplet par triplet permettrait de contourner cette limite. Une autre limite vient du fait que certaines requêtes attendent comme réponse des classes dans un certain cas et des instances dans d'autres, alors que les correspondances complexes que nous considérons sont établies entre entités de même nature (classe avec classe, propriété d'objet avec propriété d'objet). Ce type de correspondance n'a *a priori* pas de sens d'un point de vue logique et traduire la requête dans sa globalité permettrait, là encore, de prendre en compte cet aspect.

5.4 Conclusion

Dans cette section, nous avons présenté l'approche que nous avons conçue pour permettre aux utilisateurs d'interroger des bases de connaissances représentées en termes de graphes étiquetés. Cette approche est implémentée dans le système Swip et se caractérise principalement par l'utilisation de patrons de requêtes permettant l'interprétation de la requête utilisateur en langue naturelle et sa traduction en une requête graphe formelle.

Nous avons justifié l'utilisation de patrons de requêtes et détaillé notre processus d'interprétation en deux étapes, cette décomposition facilitant le multilinguisme. Nous nous sommes également concentrés sur notre implémentation qui exploite les fonctionnalités de SPARQL en proposant une ontologie pour représenter les requêtes et une ontologie pour représenter les patrons. Nous avons donné un aperçu des résultats de notre participation au défi QALD. Forts des résultats encourageants obtenus, nous avons étendu l'approche en suivant différentes directions. Nous avons proposé :

- une approche prenant en compte les agrégats lorsque ceux ci sont exprimés au niveau des attributs de projection de la requête ;
- une approche permettant de reformuler automatiquement la requête générée pour que celle-ci s'adapte à de nouvelles bases de connaissances à partir de correspondances complexes.

Nous avons également commencé à travailler sur des extensions à ces travaux dans le cadre de deux collaborations. Avec Mari Carmen Suárez-Figueroa de l'Ontology Engineering Group de l'Université Polytechnique de Madrid²⁴, nous avons utilisé Swip dans une activité de vérification de la conformité de l'ontologie aux exigences établies lors de la conception de celle-ci [Suarez De Figueroa Baonza et al. \(2012\)](#). Pour cela nous avons (a) transformé les exigences fonctionnelles exprimées par des questions de compétences en langue naturelle en requêtes SPARQL et (b) demandé à un expert de vérifier si les requêtes générées retournaient les réponses attendues. Pour mettre en œuvre Swip, nous avons proposé une première approche permettant de générer semi-automatiquement les patrons de requêtes en analysant les entités de l'ontologie mentionnées fréquemment dans les questions de compétences et en proposant des patrons à partir de la connaissance représentée pour ces entités dans l'ontologie.

23. https://framagit.org/IRIT_UT2J/sparql-translator-complex-alignment/tree/master/mtsr2017/

24. <http://www.oeg-upm.net/>

Dans le cadre de notre collaboration avec l'équipe SARA du LAAS, nous avons également utilisé Swip pour interroger une base de connaissances²⁵ décrivant les données collectées par les capteurs déployées dans le bâtiment ADREAM²⁶ Lannes et al. (2017). Ce bâtiment est un "laboratoire vivant", équipé de plus de 6500 capteurs, qui sert de support à une recherche pluridisciplinaire entre gestion de l'énergie (en particulier solaire), calcul haute performance, objets connectés, drones et robots mobiles. Une première évaluation nous a permis de mettre en évidence que des traitements plus fins devaient être mis en œuvre pour adapter Swip aux requêtes ayant une dimension spatio-temporelle. Swip n'est par exemple pas adapté au traitement des requêtes du type "*Quelle est la consommation journalière moyenne en électricité le mois dernier pour les objets déployés près de panneaux solaires ?*". La prise en compte de l'ambiguïté des termes "*le mois dernier*" (s'agit-il des 30 jours qui précèdent la requête ou bien du mois précédent ?) et de "*près de*" nécessiterait d'enrichir l'approche.

Nous avons prévu d'étendre notre travail dans plusieurs directions :

- expérimenter la facilité d'adaptation à différentes langues d'utilisateurs ; nous n'avons jusqu'ici considéré que le français et l'anglais et souhaitons participer à des tâches portant sur le multilinguisme comme celles proposées dans le cadre des différentes campagnes QALD²⁷ ;
- expérimenter des méthodes pour automatiser ou faciliter la conception de patrons de requêtes. Nous voulons d'abord déterminer automatiquement les structures de patrons en analysant des exemples de graphes requêtes, puis confronter les méthodes développées à une approche fondée sur l'apprentissage de requêtes en langue naturelle ;
- explorer de nouvelles pistes pour que l'approche évolue et se concentre davantage sur les données elles-mêmes plutôt que sur l'ontologie, afin d'obtenir de meilleurs résultats sur les jeux de données provenant du Web de données liées, tels que DBpedia. Les tâches des campagnes d'évaluation comme QALD se concentrent maintenant sur ce type de données.

25. <https://syndream.laas.fr:8082/>

26. <https://www.laas.fr/public/fr/le-projet-adream>

27. <http://qald.aksw.org/>

 Le Web Sémantique pour les objets connectés

Sommaire

6.1	Introduction	71
6.2	Le Web Sémantique pour l’IoT : notre vision	72
6.2.1	Motivations	72
6.2.2	Les nœuds	73
6.2.3	Les fonctions	74
6.2.3.1	Les fonctions centrées contenu	74
6.2.3.2	Les fonctions centrées communication	76
6.2.3.3	Bilan	79
6.2.4	Adaptation des technologies du Web Sémantique aux contraintes de l’IoT	79
6.3	Distribuer les traitements sur des nœuds Fog	80
6.3.1	Motivation	80
6.3.1.1	Scénario illustratif dans une usine connectée	80
6.3.1.2	Des règles pour représenter un besoin applicatif	82
6.3.2	Vue générale de l’approche	83
6.3.2.1	Hypothèses faites sur l’architecture	83
6.3.2.2	Principes de l’approche EDR	84
6.3.2.3	Circulation des messages dans le réseau	85
6.3.3	Les règles EDR et leur propagation	86
6.3.3.1	Structure modulaire des règles	86
6.3.3.2	Description des 4 modules d’une règle EDR	88
6.3.3.3	Activation dynamique des modules d’une règle EDR	88
6.3.4	EDRT : une stratégie de propagation pour EDR	89
6.3.4.1	Une stratégie centrée sur les types de propriétés de l’environnement	89
6.3.4.2	Description des nœuds pour EDRT	91
6.3.4.3	Règles dans EDRT	92
6.3.5	Évaluations	93
6.4	Conclusion	93

6.1 Introduction

L’Internet s’est récemment transformé en un milieu dans lequel nous sommes globalement connectés au travers d’objets déployés dans notre environnement. Comme nous l’avons

présenté dans le chapitre 2, le SWoT a pour objectif de favoriser l’interopérabilité sémantique dont ce nouvel Internet a besoin. L’utilisation de vocabulaires déréférencables permet d’associer un contexte aux messages échangés entre objets, ce contexte étant représenté dans un format intelligible pour ces derniers [Corcho and García-Castro \(2010\)](#), [Murdock \(2016\)](#). L’objectif est de faciliter la communication machine à machine, en gommant l’hétérogénéité des données recueillies, des objets, des plates-formes et des applications déployées.

L’interopérabilité est justement une des finalités majeures du Web Sémantique. Les travaux du domaine se consacrent à faciliter l’exploitation des données et des informations au contenu hétérogène stockées sur des serveurs Web, dans l’objectif de proposer des applications à destination des humains. Le domaine du SWoT propose une nouvelle perspective à ces travaux en les plaçant dans le contexte d’échanges entre machines déployées dans des architectures aux composants plus divers que ceux des architectures clients-serveur classiques du Web. Ce domaine replace l’usage des ontologies dans un contexte initialement envisagé par Tim Berners-Lee dans l’article fondateur du Web Sémantique [Berners-Lee et al. \(2001\)](#) : la communication entre agents logiciels, qui peuvent être vus aujourd’hui comme des objets connectés.

Une dizaine d’années après l’apparition du SWoT [Scioscia and Ruta \(2009\)](#), [Pfisterer et al. \(2011\)](#), il est important d’évaluer les apports concrets du Web Sémantique pour l’IoT. Nous proposons pour cela de tracer les grandes lignes d’un panorama actuel des travaux dans lesquels les ontologies sont utilisées pour favoriser l’interopérabilité au sein d’un réseau d’objets. Nous mettons l’accent sur le type d’objets du réseau sur lesquels les traitements “sémantiques” sont actuellement mis en œuvre.

Forts de ces analyses, nous avons proposé une approche générique permettant la distribution dynamique de règles de déduction dans un réseau IoT. Cette approche exploite aussi bien les objets aux capacités de calcul et de stockage importantes (tels que les serveurs considérés dans la programmation “dans le nuage” ou *Cloud computing*) que les objets aux capacités plus restreintes (telles que les passerelles utilisées dans la programmation dite “dans le brouillard” ou *Fog computing*). L’approche est générique dans le sens où elle peut être spécialisée pour soutenir différentes stratégies de distribution liées aux exigences des applications ciblées.

L’ensemble de ces travaux ont été menés en collaboration avec l’équipe Sara du LAAS, dans le cadre du co-encadrement de la thèse de Nicolas Seydoux [Seydoux \(2018\)](#). Ces travaux ont donné lieu à plusieurs publications : [Seydoux et al. \(2018b\)](#), [Seydoux et al. \(2018c\)](#), [Seydoux et al. \(2019a\)](#) et [Seydoux et al. \(2017a\)](#).

6.2 Le Web Sémantique pour l’IoT : notre vision

6.2.1 Motivations

Afin de clarifier ce qui s’apparente à un foisonnement de travaux sur le sujet, nous avons souhaité brosser un panorama du SWoT tel qu’il se présente après une dizaine d’années d’activité. Nous avons analysé 64 articles publiés entre 2009 et 2018. Ces articles ont été publiés dans des conférences ou des journaux accessibles depuis les bibliothèques digitales IEEEExplore, ScienceDirect et ACM Digital. Ils ont été retrouvés à partir des mots-clés “SWoT”, “Semantic Web”, et “Internet of Things” et conservés s’ils décrivaient l’architecture sur laquelle les approches ont été déployées.

Les approches proposées dans le cadre du SWoT ont pour but de s’adapter aux caractéristiques propres de l’IoT. Grossièrement parlant, l’IoT est un réseau reposant sur des communications machine à machine dans un contexte où les machines elles-mêmes et les applications pour lesquelles elles sont utilisées sont très hétérogènes.

Nous avons cherché à établir une typologie des machines (les nœuds) et des communications entre ces machines (qui ont lieu au travers de ce nous appelons des fonctions). Nous présentons cette typologie dans les sous-sections suivantes.

6.2.2 Les nœuds

Nous appelons *nœuds* les objets actifs déployés dans une architecture IoT. Cette notion, présente dans les architectures décrites dans [Ben-Alaya et al. \(2015\)](#) ou [Perera et al. \(2014a\)](#), est une notion abstraite faisant référence à la fois aux objets physiques (capteurs, smartphones, serveurs, passerelles, etc.) et aux objets virtuels (définis dans [Nitti et al. \(2016\)](#) ou [Reinfurt et al. \(2016\)](#)) correspondant aux services que les objets proposent. Cette notion évite d’avoir à considérer deux points de vue différents : un service peut être vu comme un point d’entrée sur un objet et un objet peut être une implémentation physique d’un service. Pour nous, un nœud est une entité active qui a une adresse sur le réseau. Une entité active peut envoyer et/ou recevoir des requêtes, ce qui correspond à la fois aux capacités des services et des objets connectés. Cette notion n’inclut cependant pas les ressources informationnelles que représentent les descriptions d’objets sur le WoT et sur le SWoT. Ces descriptions ainsi que les entités d’ontologies à partir desquelles elles sont explicitées sont uniquement consultables par leur IRI.

Nous distinguons trois types de nœuds que nous présentons dans les paragraphes suivants.

Les nœuds puissants ou nœuds Cloud nommés *weakly constrained nodes* dans [Zanella et al. \(2014\)](#), *Infrastructure Nodes* dans [Ben-Alaya et al. \(2015\)](#), ou *Cloud* dans [Liu et al. \(2015\)](#), [Szilagyi and Wira \(2016\)](#). Dans cette catégorie, nous regroupons les serveurs Cloud locaux ou distants ainsi que les ordinateurs portables ou machines aux capacités embarquées importantes, comme par exemple les robots. Ces nœuds ont des capacités de traitement, de communication et de stockage importantes. Dans le cadre du SWoT, ces nœuds sont les serveurs en charge des opérations sur de grandes quantités de données, comme cela est présenté dans [Le-Phuoc et al. \(2016\)](#) ou [Su et al. \(2018\)](#).

Les nœuds intermédiaires ou nœuds Fog très souvent nommés *passerelles* ([Compton et al. \(2009\)](#), [Ben-Alaya et al. \(2015\)](#), [Desai et al. \(2015\)](#)), établissent des “ponts” entre les nœuds plus puissants et les nœuds aux capacités contraintes. Ces nœuds sont généralement destinés à la transmission de messages ou à l’interfaçage entre protocoles. Les approches du SWoT [Barnaghi et al. \(2009\)](#) offrent de nouvelles fonctionnalités à ces nœuds en les considérant comme des nœuds capables de transformer le contenu d’un message initialement exprimé dans une représentation brute en un message sémantiquement plus riche. Dans l’architecture proposée dans [Desai et al. \(2015\)](#), la passerelle est à la fois consacrée à l’interopérabilité technique (communication avec les nœuds contraints) et à l’interopérabilité sémantique (description sémantique du contenu du message) entre les différents nœuds. Dans la même logique, pour [Nikoli et al. \(2011\)](#) et [Zanella et al. \(2014\)](#), les nœuds intermédiaires permettent la communication au sein de réseaux d’objets sans fil, mais ils sont également en charge de récupérer, décrire sémantiquement et redistribuer les données collectées par les capteurs et les services exploitant la captation.

Les nœuds contraints ont typiquement très peu de ressources, tant en termes de batterie que de capacités de traitement, de communication ou de stockage. Ils sont présents dans tout réseau IoT et sont directement connectés au monde physique. Ces nœuds contraints étaient essentiellement des capteurs dans les premiers travaux sur le SWoT tels que ceux présentés dans [Compton et al. \(2009\)](#), mais ont évolué aujourd’hui vers des types d’objets plus variés tels que les actionneurs [Seydoux et al. \(2016a\)](#) ou des objets composites comme dans [Ben-Alaya et al. \(2015\)](#).

La distinction entre ces trois types de nœuds est récurrente dans les architectures proposées pour le SWoT [Xu and Helal \(2016\)](#), [Su et al. \(2018\)](#), [Ben-Alaya et al. \(2015\)](#), [Zanella et al. \(2014\)](#), [Liu et al. \(2015\)](#). La figure 6.1 propose une représentation du modèle d’architecture trois-tiers que nous avons observée. Il est important de noter que les réseaux IoT sont déployés

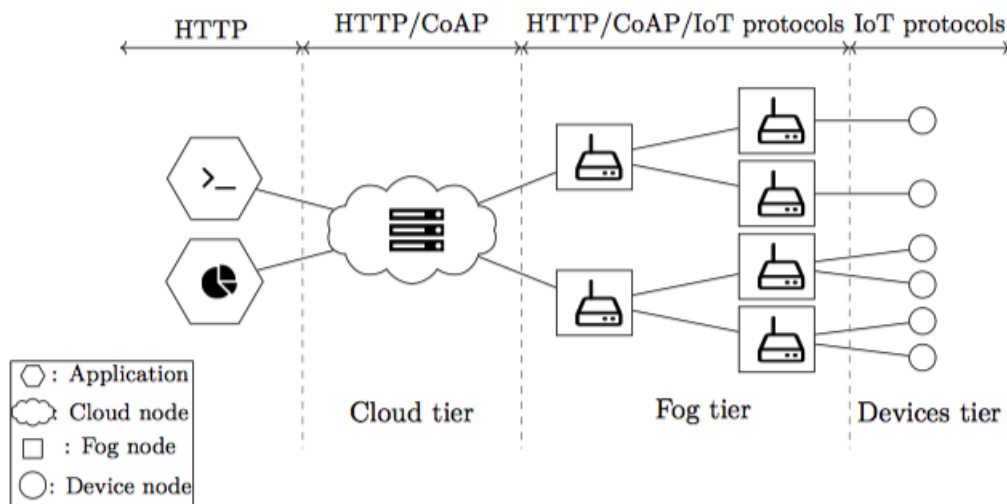


FIGURE 6.1 – Patron des architectures SWoT trois-tiers

pour mettre en place des applications qui exploitent les données recueillies par des capteurs et définissent le comportement des nœuds en capacité d’agir sur l’environnement dans lequel les objets sont déployés. Dans ce modèle, les applications accèdent à des services proposés par les nœuds Cloud à travers des protocoles du Web et les nœuds contraints sont connectés à ces nœuds Cloud via des nœuds Fog intermédiaires. Les nœuds Fog communiquent avec les nœuds contraints via des protocoles spécifiques à l’IoT.

6.2.3 Les fonctions

Lorsque nous avons analysé la façon dont les technologies du Web Sémantique sont intégrées dans l’IoT, différents types de fonctions indépendantes des applications visées pour le réseau ont émergé. Elles sont à considérer comme des traitements unitaires qui peuvent être combinés les uns aux autres, comme c’est le cas dans [Kibria \(2015\)](#), [Poslad et al. \(2015\)](#), [Perera and Vasilakos \(2016\)](#) ou [Le-Phuoc et al. \(2016\)](#) qui présentent des plates-formes complètes. Deux types de fonctions se dégagent :

- **les fonctions explicitant le contenu des messages**, destinées à modifier le contenu des messages pour le rendre pertinent pour une application donnée. Ces fonctions reprennent les prérogatives du Web Sémantique en considérant que le “contenu” des messages peut être vu comme une donnée, une information ou une connaissance (Cf. chapitre 2).
- **les fonctions destinées à la communication entre nœuds**, qui sont propres au domaine de l’IoT. Ces fonctions ont spécifiquement été mises en place pour permettre l’interopérabilité au niveau des nœuds et sont consacrées à la description de leurs caractéristiques.

La section suivante décrit les fonctions relevant de ces deux types et indique sur quel type de nœud la fonction est réalisée.

6.2.3.1 Les fonctions centrées contenu

Ces fonctions ont pour objectif de modifier les messages, de manière à passer de données brutes (essentiellement des mesures fournies par des capteurs) à des informations de plus ou moins haut niveau. Un parallèle peut être fait avec les fonctions habituelles du Web Sémantique comme, par exemple, l’annotation sémantique de textes se fondant sur des ontologies. Dans le SWoT, les ontologies sont utilisées pour décrire ou transformer des données. Pour

l'ensemble de ces fonctions, des ontologies spécifiques au(x) domaine(s) visé(s) par les applications exploitant le réseau d'objets sont utilisées. Le LOV4IoT¹ recense une partie de ces ontologies, catégorisées par domaines.

Nous présentons maintenant les quatre fonctions que nous avons identifiées.

Enrichissement Cette fonction a pour but de décrire, en se fondant sur des ontologies, les données brutes collectées par les capteurs. L'objectif est d'explicitier leur contexte d'acquisition (type du matériel utilisé, position géographique du capteur, date, valeur, unité de mesure, etc.). On peut donc par exemple transformer une donnée fournie par un capteur de température en l'information selon laquelle le capteur #357 de type ds18b20 et de marque dallas semiconductor placé dans la cuisine a mesuré à 16h11 une température de 22,4°C. Répondant directement à la problématique de l'interopérabilité, cette fonction est la première qui a été mise en œuvre sur le SWoT, comme par exemple dans les travaux décrits dans [Sheth et al. \(2008\)](#). Si certaines approches comme [Compton et al. \(2009\)](#) ou [Aïssaoui et al. \(2016\)](#) implémentent cette fonction d'enrichissement au niveau des nœuds Cloud, des approches récentes proposent de la mettre en œuvre au niveau des nœuds Fog. [Desai et al. \(2015\)](#), [Mathew et al. \(2014\)](#) et [Al-Osta et al. \(2017\)](#) proposent par exemple de déployer sur le réseau des "passerelles sémantiques" vues comme un service d'annotation capable de recevoir les données brutes collectées par des nœuds contraints et de les enrichir à partir d'ontologies telles que SSN.

L'enrichissement n'est possible que si le nœud qui réalise cette fonction a de la connaissance sur les objets qui lui communiquent les données. Ceci est généralement mis en œuvre en fournissant aux services d'enrichissement des correspondances entre les schémas de données envoyés par les capteurs et les ontologies utilisées pour les décrire [Ferdinand et al. \(2004\)](#).

Comme cela est justifié dans [Mathew et al. \(2014\)](#), décentraliser la fonction d'enrichissement permet d'éviter d'avoir à maintenir sur le nœud Cloud une représentation exhaustive et à jour des objets appartenant aux réseaux sous-jacents. L'ajout ou la suppression d'objets n'ont pas besoin d'être gérés sur ce nœud central, mais plus localement au niveau du nœud intermédiaire auquel les capteurs sont directement connectés, ce qui est un atout face à la mobilité des nœuds dans l'IoT. Ces nœuds intermédiaires sont également plus proches des objets contraints, ce qui permet, comme l'analysent [Khan et al. \(2015\)](#), de gérer plus facilement l'hétérogénéité des objets.

Abaissement C'est la fonction inverse de l'enrichissement, qui transforme une description sémantisée reposant sur une ontologie en un message dans un format brut, adapté au nœud contraint avec lequel la passerelle communique. Cette fonction n'est que peu mise en œuvre dans les approches du SWoT [Seydoux et al. \(2016c\)](#). Ceci peut s'expliquer par l'intérêt relativement récent pour les réseaux composés d'actionneurs. Ce type de fonction présente pourtant un intérêt pour ces objets : on peut par exemple envisager la génération d'un message contenant la valeur de la température à destination d'un radiateur connecté à partir de l'information selon laquelle une pièce n'est pas confortable, afin qu'il modifie sa température.

Abstraction La fonction d'abstraction prend en entrée une information telle que les informations produites par la fonction Enrichissement. Elle consiste à analyser l'information contenue dans un message et à exploiter des représentations de connaissances (ontologies ou règles) pour pouvoir générer une information de plus haut niveau. On peut par exemple conclure d'une température en hiver de 17°C dans un pièce à vivre que cette température est inconfortable, et ajouter cette information de haut niveau à l'information initiale. La valeur de confort étant dépendante des préférences de l'utilisateur, la déduction à faire ne peut pas être représentée de façon générale dans l'ontologie.

1. <https://lov4iot.appspot.com/?p=ontologies>

L'abstraction est notamment définie dans [Henson et al. \(2012\)](#) comme “la représentation d'un environnement dérivé d'une observation d'un capteur”.

De nombreuses approches exploitent les nœuds Cloud pour mettre en place la fonction d'abstraction, comme par exemple [Khan et al. \(2015\)](#), [Ploennigs et al. \(2017\)](#), [Pease et al. \(2017\)](#) ou [Pease et al. \(2017\)](#). Ces approches justifient le déploiement de cette fonction sur ce type de nœuds par la quantité d'information à exploiter et le coût des raisonnements à mettre en œuvre. D'autres approches comme [Sarkar et al. \(2015\)](#), [Maarala et al. \(2017\)](#) et [Su et al. \(2018\)](#) décentralisent la fonction d'abstraction en la mettant en place directement sur des nœuds intermédiaires. La volonté est alors d'abstraire l'information localement au niveau du nœud, à partir des connaissances restreintes que ce nœud a sur l'environnement.

Dans l'ensemble des travaux relatifs à cette fonction, l'abstraction est réalisée dans le but de fournir une information pertinente aux applications exploitant le réseau IoT. Ceci est obtenu en exploitant la connaissance représentée dans des ontologies, mais aussi en utilisant des règles métiers en lien avec l'application visée. Les approches utilisant de telles règles dans le cadre de l'IoT sont présentées dans [Sezer et al. \(2018\)](#). Ce type de règles est également considéré dans les travaux du Web Sémantique. Les langages proposés pour représenter les ontologies reposant sur l'hypothèse du monde ouvert, les assertions représentées dans une base de connaissances sont considérées comme potentiellement incomplètes. Or dans certains cas, il peut être nécessaire de tirer des conclusions de la , même si elle n'est que partiellement décrite. Différents travaux et langages ont été proposés dans le cadre du Web Sémantique pour mettre en œuvre la complémentarité de ces deux types de représentations [Eiter et al. \(2008\)](#), [Kifer and Boley \(2013\)](#), [Horrocks et al. \(2004\)](#), [Motik and Rosati \(2010\)](#). Nous expliquons l'engouement pour les règles dans l'IoT essentiellement par le fait que l'ensemble des connaissances utiles pour les applications ne peuvent pas être définies *a priori* dans des ontologies. Les règles permettent assez simplement d'injecter de nouvelles connaissances postérieurement à la définition des ontologies utilisées.

Agrégation L'agrégation est une fonction par le biais de laquelle les contenus de plusieurs messages sont utilisés pour produire une nouvelle information. Dans le cas de radiateurs connectés déployés dans une maison, par exemple, la moyenne de leur consommation énergétique journalière pour chauffer la maison ne peut se calculer que par l'agrégation de la consommation de chacun des appareils. Cette fonction est essentiellement réalisée sur les nœuds Cloud [Poslad et al. \(2015\)](#) [Le-Phuoc et al. \(2016\)](#), [Kharlamov et al. \(2016\)](#), [Ullah et al. \(2017\)](#), [Fensel et al. \(2017\)](#) car dans les applications visées, elle nécessite d'avoir une vue globale sur un ensemble de données stockées.

6.2.3.2 Les fonctions centrées communication

Les 7 fonctions que nous présentons maintenant concernent la représentation des nœuds et les utilisations qui peuvent être faites de ces représentations. Ces fonctions exploitent les ontologies spécifiques à l'IoT, comme celle qui est présentée dans la section 3.3. Ces représentations des nœuds composant un réseau peuvent être centralisées sur un nœud Cloud qui est alors le seul à avoir une vision globale du réseau. Elles peuvent aussi être délocalisées sur les différents nœuds du réseau. Dans ce cas-là, elles sont utilisées pour donner aux nœuds une représentation de leurs nœuds voisins.

Abstraction du nœud L'abstraction consiste à représenter un nœud par une description reposant sur une ou plusieurs ontologies [Vlacheas et al. \(2013\)](#). Dans [Mrissa et al. \(2015\)](#), ces représentations sont appelées des avatars. Dans le cas d'un réseau de nœuds hétérogènes, l'abstraction permet d'unifier la représentation des nœuds et de leurs caractéristiques indépendamment de leur type, de leur constructeur et des domaines d'application pour lesquels ils sont conçus [Nitti et al. \(2016\)](#).

Les premiers travaux portant sur cette fonction [Compton et al. \(2009\)](#) ont mené à la définition de l'ontologie SSN ². Même si cette fonction est généralement réalisée manuellement, certains travaux visent à l'automatiser. Dans [Pfisterer et al. \(2011\)](#), un ensemble de nœuds sont décrits dans une base de connaissances sur un nœud Cloud et ce nœud Cloud est en charge de décrire de nouveaux nœuds en analysant la similarité entre les messages que ces nouveaux nœuds envoient par rapport à ceux qui sont émis par les nœuds d'ores et déjà décrits.

De nombreuses approches mettent en place cette fonction sur des nœuds Fog, comme [Khan et al. \(2015\)](#), [Ploennigs et al. \(2017\)](#), [Dey et al. \(2017\)](#) ou [Kaed et al. \(2018b\)](#). Comme pour la fonction d'enrichissement, la décentralisation de la fonction d'abstraction permet de gérer localement l'hétérogénéité et la dynamique des objets qui se connectent au réseau.

Composition La composition est la fonction qui associe des nœuds dans le but de créer un nœud virtuel composite offrant de nouveaux services sur le réseau. La composition repose sur l'abstraction préalable des nœuds, que ce soient des nœuds physiques [Vlacheas et al. \(2013\)](#), [Foteinos et al. \(2013\)](#) ou des services [Han et al. \(2014\)](#). Dans le cadre du SWoT, la composition sur les nœuds Cloud est faite par des règles [Vlacheas et al. \(2013\)](#), par des plans d'exécution [Foteinos et al. \(2013\)](#) ou à partir de patrons [Kibria \(2015\)](#).

Bien que peu d'approches détaillent la fonction de composition sur les nœuds Fog, les travaux présentés dans [Sarkar et al. \(2015\)](#) et [Kelaidonis et al. \(2016\)](#) la mentionnent. Le faible nombre d'approches déployant cette fonction sur ces nœuds intermédiaire s'explique par le fait qu'elle ne peut être mise en œuvre que quand le nœud a accès à de nombreuses abstractions, ce qui se prête mieux à une représentation centralisée des nœuds du réseau.

Configuration La configuration est la fonction qui consiste à configurer physiquement un nœud en fonction de sa représentation. Cette fonction peut être utilisée soit pour changer la configuration d'un nœud, soit pour envoyer une commande à un actionneur qui changera son état. On peut par exemple imaginer un radiateur connecté dont la température est configurée à 21°C dans sa propre représentation. Si sa représentation est modifiée pour fixer sa température à 19°C, la fonction configuration consistera à le reconfigurer physiquement. Différentes approches réalisent ces fonctions sur des nœuds Cloud. Dans [He et al. \(2012\)](#), les observations faites par des capteurs et la connaissance représentée dans une ontologie sont utilisées pour déclencher des actionneurs. [Han et al. \(2014\)](#) proposent une fonction de composition et utilisent la description du nœud composite obtenu pour implémenter le service. Dans [Aïssaoui et al. \(2016\)](#), les exigences de l'utilisateur du système et une description de l'environnement sont pris en compte pour déduire la configuration que doivent suivre des services.

La possibilité de réduire le délai nécessaire à la mise en place de la configuration est une raison qui a motivé les travaux qui déploient cette fonction sur les nœuds Fog. [Kiljander et al. \(2014\)](#) [Kaed et al. \(2018b\)](#) et [Lee and Nair \(2016\)](#) proposent des architectures dans lesquelles des nœuds intermédiaires consomment les messages émis par les capteurs et contrôlent les actionneurs en modifiant leur représentation. Ces approches étant étroitement liées aux applications déployées sur le réseau, elles reposent en plus des ontologies soit sur un ensemble de règles prédéfinies, soit sur des règles décrivant les adaptations à mettre en œuvre.

Découverte La découverte est la fonction par laquelle les descriptions des nœuds voisins d'un nœud sont récupérées. De la même façon que pour la composition, la découverte dépend de la fonction d'abstraction qui produit la description du nœud. Les descriptions des nœuds permettent en effet de retrouver un nœud par le biais de requêtage [Ara et al. \(2014\)](#), [Kiljander et al. \(2014\)](#) et [Lee and Nair \(2016\)](#). Dans [Kiljander et al. \(2014\)](#), les nœuds Fog peuvent découvrir des nœuds en interrogeant une base de connaissances centrale contenant l'ensemble des descriptions. Les nœuds Fog ne connaissent *a priori* que l'adresse de cette

2. <http://purl1.oclc.org/NET/ssnx/ssn>

base de connaissances. Dans [Lee and Nair \(2016\)](#), la base de connaissances est stockée sur les nœuds Fog qui sont en charge d’assurer la découverte des nœuds auxquels ils sont connectés. Une telle gestion est justifiée par le fait qu’elle permet de prendre en compte, au niveau de ces nœuds, des politiques d’accès en fonction des applications, ce qui peut être utile, notamment pour des questions de vie privée. [Ruta et al. \(2016\)](#) propose un mécanisme de découverte différent, qui repose sur l’architecture “Physical Web” de Google³ et qui consiste à analyser le contenu des messages et à en extraire une représentation sémantique pour décider si le message doit être transmis à un nœud ou non.

Exposition L’exposition est une fonction complémentaire de la fonction découverte. Elle consiste, pour un nœud, à communiquer aux autres nœuds sa représentation.

Les auteurs de [Ben-Alaya et al. \(2015\)](#) proposent d’étendre le mécanisme d’exposition proposé par le standard oneM2M pour intégrer des descriptions sémantiques. Un nœud peut alors s’authentifier et se décrire auprès d’un nœud dont il connaît l’adresse. Un autre aspect de l’exposition est le *proxying* : après avoir découvert des nœuds, un nœud peut les exposer, jouant alors le rôle de “parrain”, comme cela est fait dans [Nikoli et al. \(2011\)](#).

[Christophe et al. \(2011\)](#) et [Mayer and Basler \(2013\)](#) envisagent de mettre en place les fonctions d’exposition directement au niveau des nœuds Fog. Ces descriptions peuvent ensuite être interrogées à partir de points d’accès. L’exposition n’est alors que partielle, car l’adresse du point d’accès (interface REST) doit être connue par les autres nœuds (par exemple dans un annuaire).

Sélection La sélection est une fonction dans laquelle un nœud décide quel nœud il choisit pour exécuter une tâche. Cette fonction a un intérêt quand plusieurs nœuds présentent les caractéristiques permettant de réaliser cette tâche. Cette fonction repose sur différents types de critères intégrés lors du requêtage des représentations des nœuds. Ces critères sont soit définis pour l’application visée comme dans [Vlacheas et al. \(2013\)](#), [Perera et al. \(2014b\)](#), [Han et al. \(2014\)](#) ou [Fredj et al. \(2013\)](#), soit précisés par l’utilisateur [Ara et al. \(2014\)](#) [Hussein et al. \(2016\)](#). Des approches récentes intègrent la sélection au niveau des nœuds Fog [Dey et al. \(2017\)](#). La sélection de nœuds est généralement dirigée par des politiques de gestion générale sur le réseau (comme l’économie d’énergie ou la collection de certains types d’information) ce qui explique que la sélection soit plutôt faite sur des nœuds Cloud.

Dissémination La dissémination est la fonction qui consiste à définir la façon dont la transmission d’un message d’un nœud à un autre est décidée. Ceci peut l’être soit parce que l’émetteur a manifesté son intérêt pour un certain type de messages, soit parce que l’application exploitant les échanges impose une certaine politique. C’est le cas, par exemple, lorsque des politiques de routage sont mises en œuvre. La fonction de dissémination peut être vue comme une fonction servant à la communication tout en pouvant considérer le contenu du message échangé.

Les approches déployées sur des nœuds Cloud se concentrent sur la dissémination du contenu utile pour les applications. Dans [Le-Phuoc et al. \(2012\)](#) et [Le-Phuoc et al. \(2016\)](#), des requêtes exprimées sur des flux de données sont utilisées pour identifier les informations qui seront disséminées aux applications. [Poslad et al. \(2015\)](#) définissent un mécanisme de propagation au niveau de nœuds Cloud en identifiant à partir de la connaissance représentée dans une ontologie et des requêtes les informations “urgentes” qui doivent être transmises prioritairement aux applications qui utilisent le réseau IoT.

Lorsqu’elle est mise en place au niveau des nœuds Fog, la fonction de dissémination s’attache à définir la route que doivent prendre les messages au niveau des nœuds intermédiaires du réseau. [Ashraf et al. \(2010\)](#) propose une approche distribuée qui détermine, au niveau de chaque nœud, le chemin optimal que doit suivre un message pour atteindre son destinataire.

3. <http://google.github.io/physical-web/>

Chaque nœud a une représentation de l'ensemble de ses nœuds voisins et évalue la distance "sémantique" entre la représentation qu'il a de lui-même et celles qu'il a des nœuds voisins pour choisir ceux qui sont sémantiquement les plus proches. Ces représentations reposent sur une ontologie qui représente la consommation énergétique engendrée par la transmission d'un message. Pour pouvoir être manipulée par les nœuds, l'ontologie qui est utilisée pour décrire ces représentations est sérialisée dans un format propre à l'approche. Dans l'approche proposée dans [Fredj et al. \(2013\)](#), qui repose sur une topologie de réseaux hiérarchiques, les descriptions des nœuds sont regroupées en catégories au niveau du nœud Fog. Chaque nœud Fog transmet ensuite une représentation de cette catégorie à son nœud supérieur. Les nœuds supérieurs utilisent alors une mesure de distance sémantique sur cette représentation pour décider de la propagation d'un message à leurs nœuds fils.

6.2.3.3 Bilan

L'analyse des fonctions de l'IoT reposant sur les technologies et standards du Web Sémantique nous permet de constater que, pour la majorité d'entre elles, des approches ont été proposées à la fois pour les nœuds Cloud et pour les nœuds Fog.

Ces fonctions reposent sur des descriptions d'objets et du contenu des messages définis à partir d'ontologies, mais aussi sur des règles exprimant les besoins spécifiques des applications.

La tâche complexe d'analyse des données produites par un réseau IoT est très souvent mise en œuvre dans les approches portant sur le SWoT ; [Fensel et al. \(2017\)](#), [Pease et al. \(2017\)](#), [Howell et al. \(2017\)](#), [Wang et al. \(2018\)](#) en sont des exemples. Cette tâche déployée sur des nœuds Cloud repose sur une combinaison de fonctions comme l'enrichissement, l'abstraction de contenu, l'agrégation et la dissémination, dont certaines peuvent être déployées sur des nœuds Fog. Cette tâche implique à la fois la collecte de contenu par un système autonome et la mise en place d'un système d'aide à la décision à destination d'humains. Elle est mise en œuvre sur des nœuds Cloud car, pour les approches proposées, elle nécessite d'avoir une vue globale sur l'ensemble des données collectées mais aussi parce qu'elle requiert des capacités de calcul et de stockage afin de mettre en place les technologies du Web Sémantique que les nœuds intermédiaires ne sont toujours pas en mesure de fournir. Pour pallier cette limite, de nouvelles approches utiles pour l'IoT sont développées dans le cadre du Web Sémantique. Un panorama d'entre elles est présenté dans la sous-section suivante.

6.2.4 Adaptation des technologies du Web Sémantique aux contraintes de l'IoT

Une des caractéristiques principales des nœuds intermédiaires est qu'ils sont limités en termes de batterie, de capacités de traitements, de mémoire et de modes de communication avec les autres nœuds du réseau. Limiter la charge de traitements et la nature ainsi que la quantité des messages échangés est une façon de limiter la consommation d'énergie du nœud. Des travaux récents proposent des solutions allant dans ce sens.

Pour montrer comment les technologies du Web sémantique peuvent être adaptées à l'IoT, une comparaison entre les différents formats de sérialisation de RDF est proposée dans [Su et al. \(2015\)](#). Les auteurs comparent pour différents formats la taille des messages contenant une même information et le temps de traitement nécessaire pour les générer et les échanger. Ils concluent sur l'importance de la concision du format mais aussi sur l'impact du protocole utilisé pour générer le message.

La réduction du coût de la communication peut également être réalisée en utilisant des protocoles adaptés. [Loseto et al. \(2016\)](#) proposent par exemple une extension des spécifications du W3C faites pour les plates-formes de données liées appelées LDP⁴ (pour Linked Data Platform). Alors que nativement, les primitives de LDP sont liées à HTTP⁵, les auteurs de

4. <https://www.w3.org/TR/ldp/>

5. <https://tools.ietf.org/html/rfc2616>

cet article les implémentent au travers du protocole CoAP⁶, protocole Web consacré aux objets contraints qui limite la taille de l'en-tête et du corps des messages.

Pour que les technologies du Web Sémantique soient exploitées sur des nœuds contraints, il faut aussi que ces nœuds soient en capacité de stocker les descriptions RDF et de raisonner dessus. Les approches présentées dans Bazoobandi et al. (2015) et Charpenay et al. (2018), décorrélées du cadre de l'IoT, répondent à cette contrainte et permettent de stocker des descriptions RDF dans un format de sérialisation compressé. De la même façon, Van Woensel and Abidi (2018) proposent un algorithme permettant de déployer du raisonnement dans des environnements contraints en termes de mémoire, adapté notamment au traitement de flux de données. Bien qu'étant une préoccupation récente des chercheurs du Web Sémantique, l'intérêt porté par la communauté laisse entrevoir de nouvelles solutions pour déployer la pile de technologies du Web Sémantique sur les objets contraints de l'IoT.

6.3 Distribuer les traitements sur des nœuds Fog

Dans les travaux présentés dans cette section, nous considérons qu'un réseau IoT est déployé pour collecter des données sur un environnement et que ces données peuvent être abstraites pour fournir des informations à des applications reposant sur ce réseau.

Bien qu'ils soient utilisés pour mettre en œuvre différentes fonctions, les nœuds Fog ne sont que très rarement utilisés pour disséminer de l'information à des applications. Ceci est de notre point de vue regrettable car diffuser de l'information dès que celle-ci est disponible permet de favoriser la réactivité des applications vis-à-vis de la réalité observée dans l'environnement.

Cependant, dans nos travaux, le Fog computing n'a pas vocation à remplacer le Cloud computing : nous le considérons comme une approche complémentaire. En effet, bien que les technologies évoluent, les capacités contraintes des nœuds Fog empêchent d'y déployer tous les traitements. D'autre part, pour certains traitements qui nécessitent une vue globale sur les données, les nœuds Cloud centraux restent les plus adaptés. Nous souhaitons pourtant tirer profit du passage à l'échelle permis par les approches déployées sur des nœuds Fog, comme cela est souligné dans Maarala et al. (2017). Décentraliser les traitements permet également de réduire la charge du nœud Cloud par rapport à la situation dans laquelle il est seul à les réaliser Su et al. (2018).

Bien qu'il soit possible de définir la connaissance permettant de décrire les objets et les données qu'ils produisent lors du déploiement du réseau, il est peu réaliste d'envisager que la connaissance utile pour les traitements des différentes applications qui reposeront sur le réseau soit définie *a priori*.

Ces connaissances doivent donc être intégrées dans le réseau au moment où leur exploitation est nécessaire. Comme nous l'avons vu en analysant les travaux sur le SWoT, les règles sont une façon de représenter de la connaissance liée à des besoins applicatifs. Notre proposition repose donc sur l'injection de règles dans le réseau lorsqu'une application en a besoin. Nous proposons ensuite de déployer dynamiquement ces règles de nœud en nœud en fonction d'une stratégie définie pour chacune d'entre elles. Après avoir présenté nos motivations, nous décrivons EDR (pour Emergent Distributed Reasoning), l'approche générique que nous proposons. Nous introduisons ensuite EDRT, raffinement d'EDR mettant en œuvre une stratégie de propagation liée aux propriétés de l'environnement.

6.3.1 Motivation

6.3.1.1 Scénario illustratif dans une usine connectée

Considérons, pour illustrer nos motivations, la topologie d'un réseau déployé dans une usine connectée schématisée sur la figure 6.2. L'usine est composée de deux étages, chacun se

6. <http://coap.technology/>

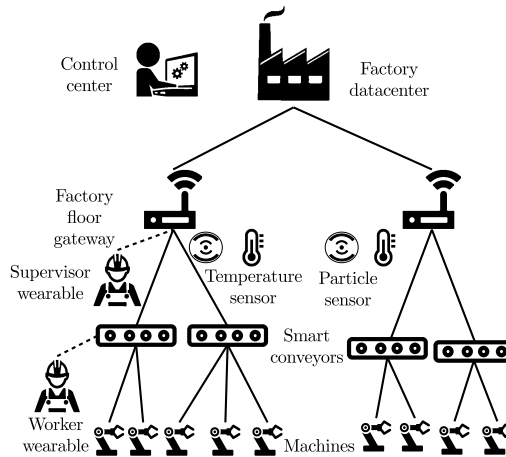


FIGURE 6.2 – Scénario dans une usine connectée

consacrant à la production d’un type de produits. Les étages sont modulaires : le déploiement des objets connectés peut être modifié en fonction de nouvelles décisions de production.

Chaque étage est équipé de tapis roulants transportant les produits d’une machine à l’autre pour la transformation. Les nœuds du système sont organisés de manière hiérarchique : les machines sont connectées à des tapis roulants connectés à la passerelle d’étage qui collecte et transmet les données au centre de données de l’usine. L’usine est équipée de capteurs pour assurer la sécurité des travailleurs : chaque étage est équipé de capteurs de présence, de luminosité, de particules et de température, et les travailleurs sont équipés de dispositifs portables communiquant avec les tapis roulants proches. Les observations des différents capteurs sont utilisées afin d’identifier les situations potentiellement dangereuses, puis d’avertir le centre de contrôle, où des actions peuvent être entreprises à distance.

Les situations dangereuses sont décrites avec des règles de déduction, fondées sur la description sémantique des observations et de l’environnement. Par exemple, *“la présence d’un travailleur à proximité d’un tapis roulant en fonctionnement dans un environnement à faible luminosité constitue un risque pour la sécurité individuelle”* est une règle potentielle. Certaines règles sont également destinées à l’assurance qualité : les capteurs disponibles dans l’usine, tels que les capteurs de température ou les capteurs intégrés aux machines et aux tapis roulants, permettent le contrôle continu de la qualité de la production. Certaines opérations sont sensibles à la température ; une règle d’assurance qualité peut être la suivante : *“la détection d’une température supérieure à un certain seuil pendant le fonctionnement des machines constitue une rupture dans la chaîne du froid”*.

La sécurité et la qualité étant des applications sensibles au temps, le traitement des règles doit être aussi rapide que possible. De plus, la mobilité de certains capteurs (portés par les ouvriers, par exemple) combinée à la modularité des étages de l’usine font que la topologie du réseau est dynamique, évolue dans le temps.

Lorsque les règles sont centralisées au niveau de la base de connaissances du nœud Cloud (datacenter sur la figure), elles ne sont appliquées qu’une fois que les données collectées par les différents capteurs ont été transmises à ce nœud central dans la topologie. De plus, elles seront appliquées sur une base de connaissances volumineuse car contenant toutes les observations faites par l’ensemble des capteurs du réseau.

Une alternative consiste à mettre en place pour chaque nœud Fog (tels que les passerelles ou les tapis roulants de notre topologie) une base de connaissances contenant les observations issues de l’enrichissement des données produites par les capteurs, puis à propager les règles directement sur les bases de connaissances de ces nœuds. Les déductions seront alors produites dès que les observations seront générées.

Une stratégie naïve de propagation des règles consiste à stocker l’ensemble des règles précédemment citées dans la base de connaissances de chaque nœud. Dans le cas de cette

stratégie, certains nœuds n'auront pas dans leur base de connaissances les observations nécessaires pour produire des déductions. Par exemple, la passerelle de l'étage 2 qui n'est pas connectée à des équipements d'ouvriers ne pourra pas appliquer la règle destinée à la sécurité individuelle. Avec la stratégie de propagation naïve, la passerelle sera surchargée par des traitements qu'elle n'est pas en mesure d'assurer. L'objectif de la stratégie EDRT décrite dans la section 6.3.4 est d'éviter ce cas de figure et de ne propager une règle que si le nœud a dans sa base de connaissances les types d'observations nécessaires pour l'appliquer. Un autre exemple concerne le cas de la règle portant sur la qualité de la production. L'appliquer sur la base de connaissances du nœud passerelle de l'étage 2 peut ne pas avoir d'intérêt si le produit confectionné est un meuble, même si des observations sur la température et sur l'état de fonctionnement des machines sont contenue dans la base de connaissances du nœud. La stratégie de propagation naïve entraînera là encore une surcharge inutile pour le nœud. Ce type d'exemples met en évidence le fait que la stratégie de propagation d'une règle peut dépendre de différentes caractéristiques du nœud, en l'occurrence ici du produit dont il participe à la confection.

L'objectif d'EDR est de définir un algorithme pour la propagation de règles pouvant s'adapter à différentes stratégies. EDR produit la déduction au plus tôt tout en limitant les traitements inutiles et la quantité de messages échangés entre nœuds.

6.3.1.2 Des règles pour représenter un besoin applicatif

Complémentaires à la connaissance représentée dans une ontologie, les règles peuvent être vues comme une façon de représenter de la connaissance liée à un besoin applicatif spécifique [Boley et al. \(2007\)](#). Des règles de déduction classiques composées d'une partie prémisses et d'une partie conclusion sont utilisées : si la partie prémisses (aussi appelée corps de la règle) est vérifiée, alors le moteur déduit la partie conclusion (aussi appelée tête de la règle).

Dans l'IoT, les règles sont essentiellement utilisées pour expliciter les déductions à faire pour les applications qui reposent sur le réseau d'objets. Elles ont été considérées sur le SWoT pour tirer des conclusions à partir des observations réalisées par les capteurs dès les premiers travaux [Sheth et al. \(2008\)](#). Depuis, un grand nombre de travaux listés dans [Sezer et al. \(2018\)](#) exploitent ce type de représentations.

Dans le but de faciliter la réutilisation, les principes du Linked Rules ont été définis dans [Khandelwal et al. \(2011\)](#). Ces principes reprennent les principes du LOD : les règles sont identifiées par des URI dérivable, exprimées dans des standards du W3C et peuvent être liées les unes aux autres. Inspiré par ces travaux, *the Sensor-based Linked Open Rules* (S-LOR) [Gyrard et al. \(2017\)](#) est consacré à la réutilisabilité de règles à partir d'observations de capteurs.

Différents formalismes ont été proposés pour exprimer ce genre de règles pour le Web Sémantique tels que SWRL ⁷, SPIN ⁸ ou plus récemment l'extension SHACL-AF ⁹ de SHACL ¹⁰. SHACL, récent standard du W3C, vise à représenter des contraintes appelées des *formes* (ou *shapes* en anglais) sur un graphe RDF. Les règles SHACL, spécifiées dans une extension du standard, permettent d'exprimer que lorsqu'une forme est détectée, de nouveaux triplets doivent être générés. En d'autres termes, dans une règle SHACL, la forme joue le rôle de prémisses, et les triplets à générer lorsque la prémisses est vérifiée jouent le rôle de conclusion.

Dans la plupart des approches, comme [Gyrard et al. \(2017\)](#) ou [Xu et al. \(2017\)](#), les règles sont utilisées sur une base de connaissances associée au nœud Cloud. Or le coût de l'exécution des règles croît rapidement lorsque la taille de la base de connaissances augmente [Maarala et al. \(2017\)](#). La résilience du système est également affectée car les traitements reposent uniquement sur un nœud central. Une façon de contourner ces limites est de déployer les

7. <https://www.w3.org/Submission/SWRL/>

8. <https://www.w3.org/Submission/spin-modeling/>

9. <https://www.w3.org/TR/shacl-af/>

10. <https://www.w3.org/TR/shacl/>

raisonnements sur les nœuds intermédiaires du réseau. Comme cela est décrit dans [Patel et al. \(2017\)](#), ce type d’approches limite la latence dans l’exécution des traitements, favorise la prise en compte de la dynamique du réseau ainsi que le passage à l’échelle. Les capacités contraintes de ces nœuds doivent cependant être prises en compte.

Les approches qui déploient des règles sur les nœuds intermédiaires le font dans des configurations particulières : soit elles optimisent le placement en ne considérant qu’un seul nœud intermédiaire [Kaed et al. \(2018a\)](#), soit elles distribuent les règles *a priori*, sans considérer les évolutions potentielles de la topologie du réseau ni l’ajout de nouvelles règles [Maarala et al. \(2017\)](#), [Su et al. \(2018\)](#), [Kaed et al. \(2018b\)](#), [Su et al. \(2018\)](#).

Dans l’approche EDR que nous avons développée dans le cadre de nos travaux et que nous présentons maintenant, nous cherchons à distribuer les règles dynamiquement, de nœud en nœud, en fonction de l’évolution du réseau.

6.3.2 Vue générale de l’approche

EDR est une approche visant à propager des règles permettant l’abstraction d’informations à partir de données collectées par des capteurs. Les informations générées ont vocation à être utilisées par des applications exploitant le réseau d’objets. L’objectif d’EDR est de transmettre à ces applications les informations dès que le système les produit.

L’approche EDR est générique dans le sens où la stratégie de propagation des règles – qui va déterminer quels nœuds se chargent d’appliquer quelles règles – n’est pas définie *a priori*. Les conditions à satisfaire pour la propagation d’une règle sont décrites au niveau de la règle elle-même.

Dans l’optique d’assurer la décentralisation de l’approche, l’algorithme sur lequel EDR repose est exécuté en parallèle sur chacun des nœuds de la topologie capable d’effectuer les traitements nécessaires et d’accéder à une base de connaissances qui lui est propre. Cet algorithme a pour rôle de définir le comportement du nœud dès qu’il reçoit un message d’un de ses voisins. Le nœud doit alors stocker dans sa base de connaissances l’information contenue dans le message. Un raisonneur met à jour le contenu de la base de connaissances. Le comportement du nœud est défini en fonction des déductions effectuées. L’information obtenue après déduction pourra par exemple amener le nœud à disséminer le contenu du message reçu à ses voisins.

6.3.2.1 Hypothèses faites sur l’architecture

Topologie du réseau EDR repose sur l’hypothèse selon laquelle l’architecture du réseau d’objets est hiérarchique : les nœuds sont organisés suivant une structure arborescente et ne communiquent qu’avec leurs voisins immédiats. Leurs voisins sont soit leur unique parent, soit leurs nœuds fils. Cette hypothèse est faite car ce type de topologies est fréquent dans les réseaux IoT, comme cela est souligné dans [Rodriguez et al. \(2010\)](#), [Zanella et al. \(2014\)](#), [Ben-Alaya et al. \(2015\)](#), [Szilagyi and Wira \(2016\)](#) ou [Su et al. \(2018\)](#). Suivant cette hypothèse, nous considérons qu’il n’existe qu’un chemin entre un nœud et l’un de ses ancêtres, ce qui simplifie la dissémination des messages dans notre approche, tout en la rendant applicable dans la majorité des architectures.

Nous considérons également que les applications exploitant les objets connectés ne sont pas déployées sur un nœud de la topologie mais ont la possibilité de communiquer avec le nœud Cloud à la racine de la topologie, comme cela est illustré sur la figure 6.1. Nous considérons également que chaque nœud Fog a la possibilité de communiquer avec ces applications.

Règles pour représenter le besoin applicatif Nous faisons également l’hypothèse selon laquelle les informations utiles pour chaque application peuvent être exprimées sous forme de règles représentant l’information à déduire d’une observation faite sur l’environnement dans lequel les objets sont déployés. L’application transmet la règle au nœud Cloud, nœud

racine de la topologie, qui est alors en charge de la propager aux nœuds de la topologie. À chaque exécution de la règle, l'information générée est directement envoyée par le nœud qui l'a produite à l'application qui a fourni la règle.

6.3.2.2 Principes de l'approche EDR

Fonctionnement global EDR repose sur l'échange de *messages* entre voisins immédiats, les échanges étant guidés par le contenu de la base de connaissance du nœud. Le nœud a dans sa base de connaissances locale une représentation de la topologie du réseau, limitée aux nœuds auxquels il est directement connecté. Ce choix a été fait pour limiter la quantité de messages échangés sur le réseau et pour permettre une prise en compte plus rapide de l'évolution de la topologie (en termes de suppression et d'ajout de nœuds). Les mises à jour de sa base de connaissances, qui permettent au nœud d'avoir une vision consistante de la réalité du réseau, sont dépendantes des échanges de messages entre nœuds.

Un nœud propage une règle à un de ses nœuds voisins s'il considère que le nœud a la capacité de traiter la règle. Cette décision est prise à partir de la stratégie de déploiement qui est définie dans la règle et de la représentation que le nœud a de son voisin dans sa base de connaissances.

Pour qu'un nœud ait une représentation de ce que ses voisins sont en capacité de faire, les nœuds voisins immédiats s'échangent des descriptions de leurs caractéristiques (leur localisation, l'entité en charge de leur administration, le type de données qu'ils observent, le type de données dont ils ont besoin pour faire des déductions, etc.). Pour assurer la dynamique du placement des règles, les nœuds envoient de nouvelles représentations de leurs caractéristiques chaque fois que celles-ci évoluent.

Lorsqu'un nœud applique une règle et produit une nouvelle information, il l'envoie uniquement à ses voisins immédiats qu'il sait avoir un intérêt pour ce type d'informations.

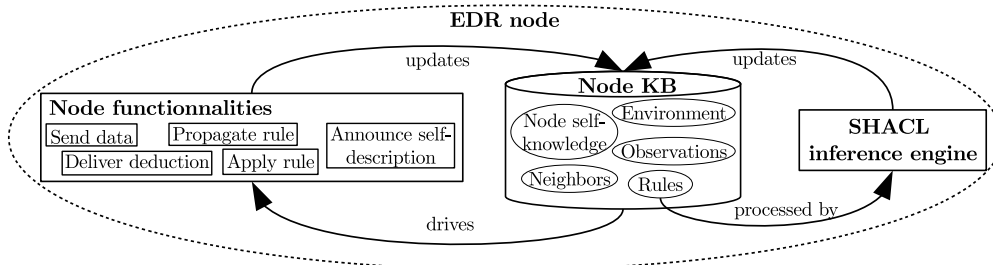


FIGURE 6.3 – Rôle de la base de connaissances dans le comportement d'un nœud d'EDR

Une représentation fonctionnelle d'un nœud dans EDR est donnée dans la figure 6.3.

Base de connaissances locale d'un nœud Chaque nœud a une base de connaissances locale où toute la connaissance nécessaire à l'exécution de l'algorithme est stockée. Cette connaissance est utilisée pour mettre en place les fonctionnalités des nœuds. Les mises à jour de la base de connaissances sont réalisées par le moteur d'inférences à l'aide de règles.

La base de connaissances contient :

- la connaissance qu'un nœud a sur ses propres caractéristiques ;
- la connaissance qu'il a sur ses voisins immédiats ;
- la connaissance qu'il a sur l'organisation statique de l'environnement dans lequel il est déployé (tel que par exemple le bâtiment auquel appartient la pièce dans laquelle le nœud est situé) ;
- les dernières observations décrivant les caractéristiques dynamiques de son environnement ;
- les règles qu'il a reçues d'autres nœuds.

Comportement d'un nœud La base de connaissances d'un nœud est utilisée pour contrôler le comportement du nœud qui repose sur des fonctionnalités simples :

- envoyer une information sur l'environnement (typiquement une observation) à un nœud ;
- envoyer la description d'une de ses caractéristiques à un nœud ;
- envoyer une règle à un nœud ;
- envoyer une déduction obtenue après application d'une règle à une application ou à un nœud ;
- appliquer une règle sur sa base de connaissances.

Ontologie EDR Du contenu de la base de connaissances du nœud dépend son comportement. Dans l'objectif de rendre EDR générique, les descriptions contenues dans la base de connaissances reposent sur une ontologie¹¹ définie spécifiquement pour l'approche. L'ontologie permet de :

- décrire un nœud, que ce soit pour que le nœud lui même connaisse ses propres caractéristiques ou pour que le nœud sache quelles caractéristiques annoncer à ses nœuds voisins ;
- décrire une règle, ce qui permet notamment à un nœud de savoir quel nœud ou application a introduit la règle ; cela permet lui également de savoir si la règle est activée sur sa base de connaissances et s'il l'a transférée à ses nœuds voisins pour que ceux-ci la propagent ;
- décrire une déduction faite suite à l'exécution d'une règle de façon à représenter de quelle règle elle est issue pour ensuite l'envoyer au nœud qui est à l'origine de son introduction dans le réseau.

6.3.2.3 Circulation des messages dans le réseau

Le fonctionnement de l'approche EDR repose sur l'échange de messages entre les nœuds du réseau. Un nœud ne communique qu'avec ses voisins immédiats (son parent ou ses enfants) à l'exception des applications auxquelles les déductions réalisées à partir des règles sont transmises. Afin de pouvoir déployer les règles dans l'ensemble de la topologie, un mécanisme de proxy (ou parrainage) est mis en place.

L'ontologie EDR permet de décrire les propriétés qui sont effectivement celles du nœud et celles qui sont des propriétés de ses parents ou enfants dont il a la charge¹².

Les messages échangés peuvent être de trois types :

- des notifications de caractéristiques de nœuds, provenant d'un nœud voisin ;
- des observations résultant de l'enrichissement de données brutes collectées par des capteurs ou de l'abstraction de telles informations suite à l'application d'une ou plusieurs règles EDR ;
- des règles EDR.

Propagation des caractéristiques d'un nœud L'algorithme d'EDR dépend des descriptions que les nœuds s'échangent au sujet de leurs caractéristiques. Chacun des nœuds sur lesquels l'algorithme est déployé a une description des nœuds de son voisinage et propage dans le réseau leurs caractéristiques comme étant les siennes. Le nœud n'a pas une représentation de la topologie générale du réseau mais simplement des caractéristiques de l'ensemble de ses descendants et de l'ensemble de ses ancêtres.

Plus concrètement, lorsqu'un nœud est notifié des caractéristiques de son parent, il va représenter dans sa base de connaissances le fait que ce sont effectivement les caractéristiques

11. <https://w3id.org/laas-iot/edr>

12. *edr:ParentProxiedProperty*, et *edr:ChildrenProxiedProperty*

de son parent, puis il va diffuser à ses enfants ces caractéristiques comme étant les siennes. Par exemple, si un nœud veut être notifié dès qu'un certain type d'observations est produit, il va en notifier ses enfants. Chaque enfant va stocker cette information dans sa base de connaissances et va à son tour notifier ses propres enfants récursivement. Inversement, lorsqu'un nœud recevra une observation, il la stockera dans sa base de connaissances et la transmettra à son parent si, dans sa base de connaissances, son parent est décrit comme intéressé par ce type d'observation. Pour décorréliser EDR d'une stratégie de propagation, une partie du vocabulaire d'EDR représente les caractéristiques qui doivent être annoncées en distinguant les types de voisins concernés.

Caractéristiques à annoncer à un nœud parent

Comme nous l'avons vu dans le chapitre 4, différentes ontologies ont été proposées pour décrire les nœuds d'un réseau IoT. Dans le vocabulaire d'EDR, nous avons défini une classe¹³ qui permet d'enrichir la description d'un nœud par la ou les caractéristiques qui doivent être annoncées à son nœud parent. Cette information sera utilisée par le nœud pour construire le message annonçant sa description à son nœud parent.

Caractéristiques à annoncer aux nœuds enfants

L'annonce des caractéristiques d'un nœud à ses nœuds enfants se fait de la même façon. Nous avons cependant défini deux sous-classes disjointes permettant de préciser si la caractéristique doit être envoyée à tous les enfants ou seulement à un sous-ensemble des nœuds enfants, car le nœud peut choisir de cibler les caractéristiques qu'il partage¹⁴.

L'intérêt qu'un nœud a pour un type d'observations parce qu'il en a besoin pour appliquer une règle est aussi représenté grâce au vocabulaire et fait partie de ses caractéristiques¹⁵.

Propagation des observations Lorsqu'un nœud reçoit un message correspondant à une observation, il stocke le message dans sa base de connaissances. L'observation sera alors évaluée par toutes les règles activées dans la base de connaissances. De plus, le nœud transférera le message à ses nœuds voisins qui ont été décrits dans sa base de connaissances comme intéressés par le type d'observation qu'il vient de recevoir. Il est important de noter qu'une observation peut :

- soit être générée par un nœud à partir de la fonction d'enrichissement appliquée sur les données collectées par des capteurs auxquels il est connecté ;
- soit être obtenue par l'application d'une règle.

Le processus mis en place pour qu'un nœud propage une règle est décrit dans la section suivante.

6.3.3 Les règles EDR et leur propagation

La généralité d'EDR repose sur la description qui est faite des règles de déduction. La stratégie de propagation n'est pas définie dans l'algorithme mais directement au niveau de la règle, ce qui permet de finement décrire et paramétrer sa propagation.

6.3.3.1 Structure modulaire des règles

Une règle EDR est composée de 4 parties distinctes, chacune étant une règle SHACL. Nous appelons ces 4 composants de règles des *modules*, représentés dans la figure 6.4. Les 4 modules d'une règle EDR, que nous détaillons dans la section suivante, sont :

- le module **cœur** qui contient la règle de déduction proprement dite, qui consiste en une abstraction sémantique d'une observation, ;

13. *edr:ParentAnnouncedProperty*

14. *edr:ChildrenAnnouncedProperty* *edr:AllChildrenAnnouncedProperty*

15. *edr:isInterestedIn*

- le module **activation** qui définit le fait que la règle doit être activée ou pas dans la base de connaissances du nœud ; ce module analyse si le nœud est en capacité d’appliquer la règle ;
- le module **propagation** qui définit quelle doit être la stratégie choisie pour propager la règle ; ce module analyse si les voisins du nœud sont en capacité d’appliquer la règle ;
- le module **résultat** qui décrit à quel nœud les déductions faites doivent être transmises ; ce module analyse si parmi les voisins du nœud, certains sont intéressés par la déduction produite.

Pour représenter une règle EDR comme une ressource à part entière, nous définissons la notion d’enveloppe de règles qui permet, via le vocabulaire de l’ontologie EDR, de décrire les différents modules d’une règle EDR ¹⁶. Un exemple de règle est disponible en ligne ¹⁷.

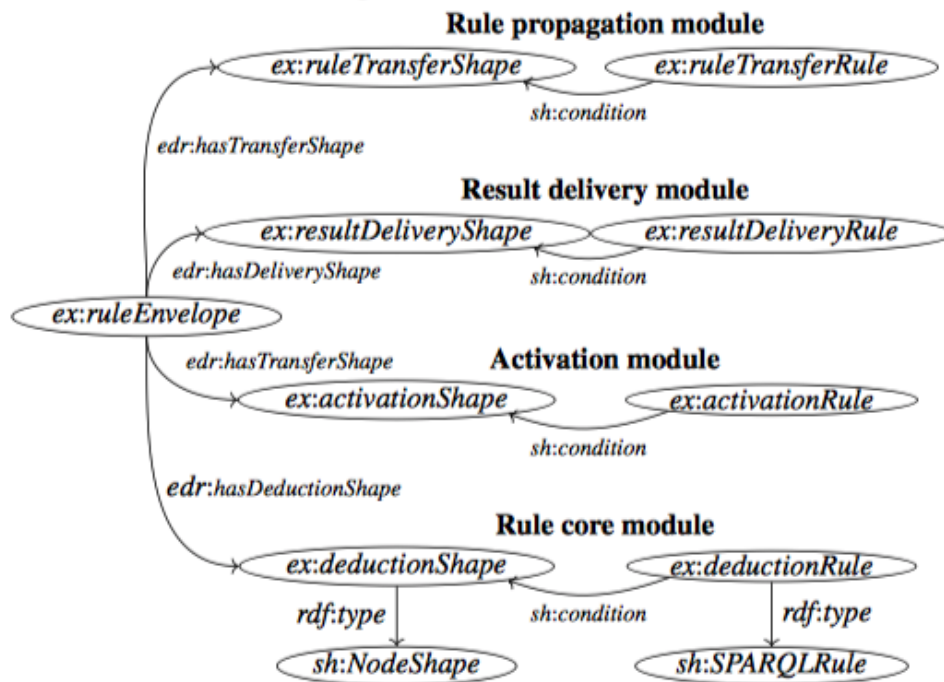


FIGURE 6.4 – Modules constituant une règle EDR

Dans le fonctionnement du moteur SHACL, une règle est par défaut évaluée sur une base de connaissances dès qu’elle y est stockée, puis chaque fois que le contenu de la base de connaissances est modifié.

Lorsqu’un nœud reçoit une règle, il la stocke dans sa base de connaissances. Les règles SHACL composant les différents modules enrichissent la description de la règle par des informations indiquant :

- si la règle doit être activée dans la BC du nœud ;
- à quel(s) nœud(s) voisins la règle doit être propagée ;
- à quel(s) nœud(s) voisins les résultats de chaque exécution de la règle doivent être transmis.

Le comportement du nœud est ensuite défini en fonction de ces informations.

Pour garantir la décidabilité, seules des règles DL-safe sont considérées et l’approche EDR n’est adaptée que pour prendre en compte des règles entre lesquelles il ne peut pas y avoir de cycle, également appelées règles stratifiées.

16. *edr:hasTransferShape*, *edr:hasApplyShape*, *edr:hasDeliveryShape* et *edr:hasDeductionShape*

17. <https://w3id.org/laas-iot/edr/iiot/r1.ttl>

6.3.3.2 Description des 4 modules d'une règle EDR

Module cœur C'est la partie opérationnelle de la règle, qui décrit l'abstraction à générer à destination d'une application (ou éventuellement de plusieurs). Il repose sur une représentation logique de la déduction à produire et permet d'abstraire une information d'une observation. De façon à éviter d'évaluer inutilement la règle SHACL contenue dans le module cœur chaque fois qu'une information est intégrée dans la base de connaissances d'un nœud, le module activation permet d'enrichir la description de la règle EDR avec un triplet indiquant si le nœud est en mesure d'appliquer le module cœur. Le module cœur est donc évalué sur la base de connaissances seulement si la description de la règle dans la base de connaissances du nœud indique que cela a un intérêt ¹⁸.

Module activation Ce module détecte le fait que le nœud soit en capacité d'appliquer la règle ou non. Pour cela, il exprime par le biais d'une forme SHACL les caractéristiques que le nœud doit remplir pour pouvoir appliquer le module cœur et enrichit la description de la règle EDR dans la base de connaissances du nœud en fonction. Les caractéristiques du nœud que ce module considère peuvent être aussi variées que nécessaires : elles peuvent correspondre par exemple à ses capacités physiques (batterie, capacité de stockage), aux données qu'il produit ou aux entités en charge de la gestion du nœud. Si ces conditions sont respectées, la règle SHACL du module décrit dans la base de connaissances du nœud le fait que la règle est active, ainsi que le type d'information que le nœud est en capacité de produire grâce à cette règle, et le type d'observation dont il a besoin pour l'appliquer. Le nœud propage ensuite ses nouvelles caractéristiques à ses nœuds voisins.

Module propagation Il décrit les conditions qui doivent être remplies pour que le nœud transfère une règle à un voisin et enrichit la description de la règle avec les nœuds à qui la règle doit être transmise si les conditions sont respectées ¹⁹. Ce module repose sur la représentation des caractéristiques de ses voisins dont un nœud dispose. De la même façon que pour le module activation, les prémisses décidant du déclenchement de cette règle SHACL peuvent dépendre de toutes les caractéristiques exprimées dans la description des nœuds voisins (données produites, capacités, entité gestionnaire, ...).

Module résultat Il décrit les nœuds à qui les déductions faites à partir du module cœur doivent être envoyées. Ce module permet de rajouter à la description de la règle EDR dans la base de connaissances du nœud la ou les applications qui sont à l'origine de l'injection de la règle dans le réseau ²⁰. Il permet également d'identifier les nœuds voisins qui peuvent être intéressés par la déduction produite par la règle. Pour cela, le module définit les conditions à remplir par les nœuds voisins pour que les résultats leur soient envoyés et enrichit la description de la règle EDR en fonction ²¹. Ces conditions peuvent être encore une fois exprimées à partir des différentes caractéristiques dont le nœud dispose dans la description de ses voisins.

6.3.3.3 Activation dynamique des modules d'une règle EDR

Les différents modules d'une règle EDR n'ont pas besoin d'être évalués sur la base de connaissances d'un nœud chaque fois que celle-ci évolue. En effet, ils dépendent d'informations spécifiques reçues par le nœud et stockées dans sa base de connaissances.

Pour réduire les coûts de traitement, les modules autres que le module cœur sont décrits comme activés ou non en fonction des messages que le nœud reçoit. Le standard SHACL permet de le faire via son vocabulaire ²².

18. `<r,edr:isRuleActive,true>`

19. `<r,edr:transferableTo,n'>`

20. `<o,edr:consumesResult,r>`

21. `<n',edr:consumesResult,r>`

22. spécifications SHACL Advanced Features <https://www.w3.org/TR/shacl-af/#deactivated>

La figure 6.5 décrit l’algorithme d’EDR à partir du contenu des messages qu’un nœud reçoit.

Quand une règle est reçue par un nœud (étape (1) sur la figure), ses 4 modules sont par défaut activés. La base de connaissances du nœud est mise à jour en décrivant :

- si le nœud est en mesure d’évaluer la règle (module activation) ;
- à qui le nœud doit propager les résultats des déductions si la règle est activée (module transfert) ;
- si le nœud doit propager la règle à ses nœuds voisins (module propagation).

En fonction du contenu de la base de connaissances obtenue, le nœud propagera ou non la règle (étape (3) sur la figure). Les modules activation, transfert et propagation sont ensuite désactivés (étape (4)). Par la suite, le nœud appliquera le module cœur lorsque la base de connaissances est mise à jour, si le module activation a décrit la règle comme activée.

Quand un nœud reçoit un message correspondant à une évolution de la description de ses voisins (telle que la déconnexion d’un nœud, la connexion d’un nouveau nœud, ou un changement dans les caractéristiques d’un nœud), la propagation de toutes les règles EDR doit de nouveau être évaluée. Les modules de toutes les règles sont activés (étape (2) sur la figure) et les informations contenues dans le message sont intégrées par le nœud à sa base de connaissances. Le nœud transfère les évolutions de la topologie à ses voisins. En fonction des informations intégrées à la base de connaissances, le nœud adapte son comportement et décide ou non d’activer certaines règles et/ou de les propager (étape (3)). Les modules autres que le module cœur sont enfin désactivés (étape (4)).

Quand le nœud reçoit une observation, il l’intègre à sa base de connaissances, sur laquelle seuls les modules cœur des différentes règles EDR décrites comme actives (c’est à dire capables d’être traitées par le nœud) sont évalués. Si cette étape mène à la création de nouvelles informations, celles-ci sont décrites comme étant issues de l’application d’une règle²³ (étape (5)) et sont propagées par le nœud aux nœuds voisins et aux applications qui sont indiquées dans la description de la règle EDR comme devant être destinataires de ce type d’informations.

6.3.4 EDRT : une stratégie de propagation pour EDR

L’approche EDR a été implémentée pour une stratégie visant à propager les règles sur les nœuds si les nœuds produisent les types d’observations utiles pour appliquer les règles. Cette stratégie est nommée EDRT (Emergent Distributed Reasoning with Types). Elle est décrite en détail dans [Seydoux et al. \(2018a\)](#). Nous en donnons ici les principes.

6.3.4.1 Une stratégie centrée sur les types de propriétés de l’environnement

EDRT implémente une stratégie de propagation des règles dirigée par les types de propriétés observées dans l’environnement où les objets sont déployés. Ces propriétés sont soit des propriétés sur l’environnement que les capteurs observent, comme la luminosité ou la température, soit des propriétés de plus haut niveau déduites par des règles à partir des observations réalisées par les capteurs, comme par exemple la propriété de confort pour une pièce. Pour EDRT, nous considérons que la description d’un nœud intègre dans ses caractéristiques les types de propriétés observées par le nœud.

De façon à transmettre le plus rapidement possible aux applications concernées les informations sur les propriétés observées, EDRT implémente une stratégie suivant laquelle les règles doivent être placées dans la topologie au plus près des capteurs qui observent les propriétés. Cette stratégie vise à limiter le nombre de nœuds par lesquels les observations sont échangées et à déployer les règles sur des bases de connaissances dont la taille sera limitée, comme cela est préconisé dans [Maarala et al. \(2017\)](#).

23. pour pouvoir tracer la provenance de déduction, l’information générée par une règle est décrite comme provenant de la règle²⁴

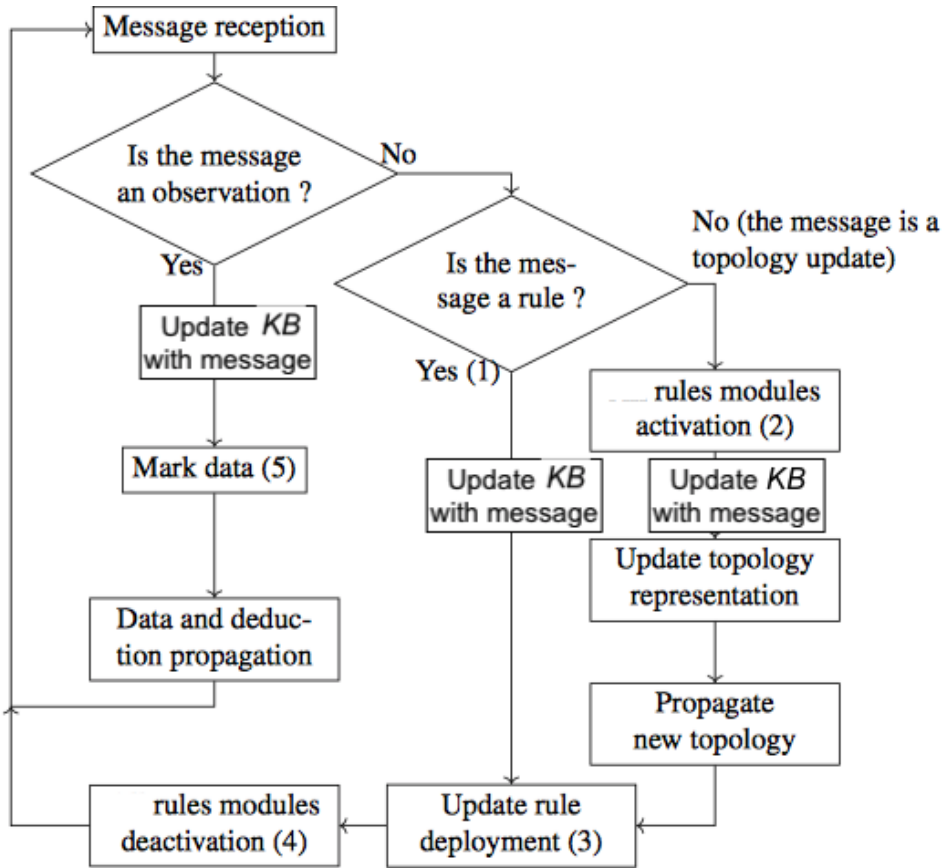


FIGURE 6.5 – Vue générale de l’algorithme d’EDR

Soit $R_{Visibility}$ une règle définie pour illustrer notre approche. Cette règle permet de générer une observation de faible luminosité autour d’une machine si une personne est présente près de cette machine, que la machine est en activité, et que la luminosité près de cette machine est inférieure à 300 lumens.

Pour que cette règle puisse être appliquée par un nœud, la stratégie EDRT considère que le nœud doit observer les propriétés de *Presence*, *Luminosity* et *Activity* qui sont les propriétés apparaissant en prémisse de la règle. Un nœud qui appliquera la règle observera la propriété de *LowMachineVisibility* qui est la propriété apparaissant en conclusion de la règle.

La figure 6.6 illustre la propagation de $R_{Visibility}$ par EDRT sur un exemple de topologie. Cette topologie intègre 5 nœuds Fogs qui sont en charge de la fonction d’enrichissement pour les données transmises par les capteurs auxquels ils sont directement connecté. Le nœud $M111$ observe alors directement les propriétés de *Presence*, *Luminosity* et *Activity*, il en est de même pour le nœud $M112$. De la même façon, le nœud $M121$ observe les propriétés de *Presence* et *Activity*, le nœud $M122$ les propriétés de *Luminosity* et *Activity* et le nœud $C120$ la propriété de *Temperature*.

La règle $R_{Visibility}$ est introduite par le centre de contrôle au niveau du centre de données, nœud Cloud de la topologie. Elle est ensuite propagée aux nœuds Fog de la topologie. La stratégie définie pour cette règle la propagera au plus près des nœuds produisant directement les types de propriétés nécessaires à l’application de la règle. Dans l’exemple, ce sont les nœuds $M111$, $M112$ et $C120$ qui appliqueront la règle avec cette stratégie. $M111$ et $M112$ l’appliquent parce qu’ils observent directement les types de données nécessaires. $C120$ l’applique par la combinaison des propriétés observées par ses deux fils.

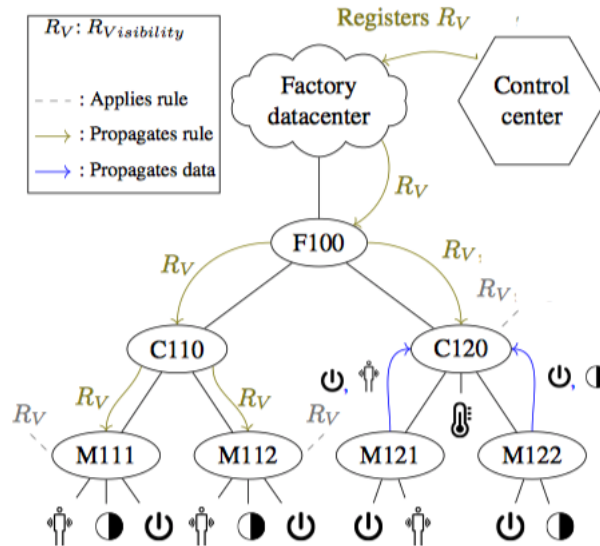


FIGURE 6.6 – Exemple de règles déployées suivant EDRT

6.3.4.2 Description des nœuds pour EDRT

Les informations utiles pour la mise en place d’une stratégie de propagation sont identifiées dans la description d’un nœud en spécialisant le vocabulaire spécifique à EDR.

Dans EDRT, la description d’un nœud inclut les propriétés qu’il observe. Ces propriétés sont décrites à partir du vocabulaire d’EDR afin de représenter dans la base de connaissances du nœud si ces caractéristiques sont :

- directement celles du nœud (c’est à dire des propriétés observées par des capteurs dont le nœud a la charge de décrire les données ou des propriétés observées par l’application d’une règle) ;
- celles de ses descendants.

Dans l’exemple présenté dans la figure 6.6, le nœud $M121$ annonce au nœud $C120$ qu’il observe la *Presence* et l’*Activity*. Le nœud $C110$ stocke ces informations dans sa base de connaissances et propage l’information en indiquant à son nœud père $F100$ qu’il observe à son tour la *Presence* et l’*Activity*. Le nœud $M122$ annonce au nœud $C120$ qu’il observe la *Luminosity* et l’*Activity*. Le nœud $C110$ stocke ces informations dans sa base de connaissances et propage l’information en indiquant à son nœud père $F100$ qu’il observe une nouvelle propriété qui est la *Luminosity* (il lui a déjà annoncé qu’il observait la *Presence* et l’*Activity*).

Dans cet exemple, nous considérons qu’après application de la stratégie de propagation des règles, le nœud $C120$ applique la règle R_V et observe alors la propriété *visibility*. Au moment où la règle est activée sur le nœud, il annonce à son parent $F100$ l’observation de cette propriété comme une de ses nouvelles caractéristiques.

Dans EDR, la description d’un nœud inclut également le type d’observations dont un nœud a besoin pour appliquer une règle. Dans le cadre d’EDRT, un nœud a besoin d’observations relatives à un type de propriétés pour appliquer une règle. Ces propriétés sont représentées en spécialisant le vocabulaire d’EDR comme des caractéristiques que le nœud doit transmettre à ses enfants et pour lesquelles le nœud doit jouer le rôle de proxy. L’idée sous-jacente est qu’un nœud fils ne transmettra des observations à son père que s’il sait que le nœud père est intéressé par le type d’observations produites, la volonté étant de limiter les messages échangés sur le réseau.

Si nous reprenons l’exemple de la figure 6.6, lorsque la règle R_V sera rendue active sur le nœud $c120$, celui-ci notifiera ses enfants qu’il consomme des données de type *Presence*, *Luminosity*, *Activity*. $M121$ et $M122$ lui transmettront ensuite ce type d’observations dès qu’ils auront sémantisé les données que les capteurs auxquels ils sont reliés produisent.

6.3.4.3 Règles dans EDRT

Pour l’instant, les règles sont définies manuellement. La représentation SHACL de la règle $R_{visibility}$ est disponible en ligne²⁵.

Module cœur Le module cœur décrit la déduction à produire à partir d’une observation.

Pour la règle $R_{visibility}$, le module cœur est exprimé à partir d’une règle SHACL qui rajoute à la base de connaissances du nœud les triplets exprimant que $LowMachineVisibility(?m)$ si la forme SHACL exprimant $Location(?l) \wedge Presence(?l, ?o_1) \wedge ?o_1 = True \wedge Luminescence(?l, ?o_2) \wedge ?o_2 < 300lm \wedge Machine(?m) \wedge Activity(?m, ?o_3) \wedge ?o_3 = True \wedge LocatedIn(?m, ?l)$ est retrouvée.

Module propagation L’objectif d’EDRT étant de propager les règles aux nœuds les plus profonds dans le réseau capables d’appliquer la règle, le transfert d’une règle d’un nœud à l’un de ses nœuds fils ne se fait que si le nœud fils observe toutes les propriétés nécessaires à l’application du cœur de la règle. Pour cela, le module transfert repose sur :

- une prémisse exprimée par une forme SHACL qui identifie si les types des propriétés nécessaires à l’application font partie des caractéristiques que le nœud connaît de son fils ;
- une conclusion qui, si la forme est retrouvée, enrichit la description de la règle dans la base de connaissances du nœud pour indiquer à quel nœud elle doit être transférée.

Pour notre exemple, la règle $R_{visibility}$ sera transférée à un nœud fils si ce nœud fils observe les propriétés de $Presence$, de $Luminescence$ et d’ $Activity$. Dans la topologie illustrée dans la figure 6.6, au niveau de la base de connaissances du nœud $F100$, le module transfert décrira la règle comme à transférer au nœud $C120$ et au nœud $C110$.

Module activation Pour activer une règle, le nœud doit avoir dans ses caractéristiques l’observation des types de propriétés nécessaires à l’application de la règle. De plus, aucun de ses enfants ne doit observer l’ensemble des propriétés requises pour l’appliquer (sinon la règle sera transmise au nœud fils).

Lorsqu’une règle est activée sur un nœud, celui-ci informe ses fils qu’il est intéressé par les types de propriétés nécessaires à l’application de la règle.

Le module activation repose donc sur :

- une forme permettant de détecter le fait que le nœud a dans ses caractéristiques propres les propriétés nécessaires à l’activation de la règle ;
- une conclusion enrichissant, si la forme est retrouvée, la description de la règle en indiquant qu’elle est active et la description des caractéristiques du nœud en indiquant que le nœud est intéressé par les types d’observations nécessaires à l’application de la règle.

Dans la topologie illustrée dans la figure 6.6, le module activation décrira la règle $R_{visibility}$ comme activée dans les bases de connaissances des nœuds $M11$ et $C120$ et les caractéristiques de ces nœuds seront mises à jour pour indiquer qu’ils sont intéressés par les observations de $Presence$, de $Luminescence$ et d’ $Activity$. Ces descriptions seront ensuite transmises à leurs éventuels nœuds fils.

Module résultat Dans EDRT, les déductions produites à la suite de l’application d’une règle sont transmises aux applications à l’origine des règles mais aussi aux nœuds qui ont annoncé être intéressés par les types de propriétés produits par la règle. Pour cela, le module résultat, repose sur :

25. <https://w3id.org/laas-iot/edr/iiot/visibility.ttl>

- une forme qui identifie les nœuds voisins intéressés par les types de résultats produits par la règle ;
- une conclusion mettant à jour dans la base de connaissances du nœud la description de la règle à partir des nœuds auxquels les résultats doivent être transmis si la forme est retrouvée.

Dans notre exemple, il n'est pas représenté qu'un nœud est intéressé par la propriété de confort. Le module résultat ne mettra donc pas à jour la base de connaissances des nœuds.

6.3.5 Évaluations

EDRT a été implémenté²⁶ et a été évalué en simulant différentes topologies composées de 30 à 90 nœuds Fog et de capteurs organisés suivant différentes topologies arborescentes. Ces simulations ont été déployées sur différents types d'environnements (serveur, ordinateur portable, Raspberry Pi 2, Raspberry Pi 3) dans l'objectif d'évaluer leur déploiement dans des environnements contraints.

EDRT a été comparé sur ces topologies à plusieurs stratégies de propagation de règles permettant d'évaluer le délai mis pour transmettre les observations, pour produire les déductions et pour transmettre les abstractions aux applications.

Le détail des différentes configurations est donné dans [Seydoux et al. \(2019b\)](#).

Les conclusions principales sont que la distribution des règles sur les nœuds Fog diminue le délai de transmission des abstractions aux applications par rapport à une stratégie où toutes les règles sont appliquées au niveau du nœud central. L'augmentation du nombre de nœuds dans la topologie n'influence que partiellement le délai de transmission pour EDRT, contrairement à ce qui est observé pour la stratégie dans laquelle les règles sont activées sur le nœud central. Cependant, ceci n'est valable que dans une certaine limite liée à la capacité de stockage, de communication et de traitements de l'environnement considéré. Pour pallier cela, les caractéristiques techniques du nœud devraient être prises en compte pour définir si le nœud est effectivement en capacité de traiter une règle au moment où elle lui est transmise.

6.4 Conclusion

Nous avons brossé un panorama des différents travaux portant sur le SWoT, qui montre que des solutions ont été proposées pour mettre en œuvre les différentes fonctions nécessaires à la gestion de l'interopérabilité sémantique pour la communication machine à machine dans l'IoT. Ces fonctions considèrent aussi bien le contenu des messages que leur transmission. Elles permettent également d'unifier les représentations qui sont faites des nœuds en vue de gommer l'hétérogénéité des constructeurs qui les proposent et des plates-formes qui les gèrent. La majorité des fonctions peuvent être déployées en suivant une approche dans le nuage aussi bien qu'une approche dans le brouillard.

Dans ce contexte, nous avons proposé l'approche générique EDR pour la distribution dynamique de règles dans une architecture IoT. Dans les approches existantes considérant le raisonnement fondé sur des règles pour le SWoT, le calcul est souvent effectué uniquement sur des nœuds Cloud, ce qui peut potentiellement conduire à un goulot d'étranglement centralisé et créer une surcharge de communication sur le réseau. Afin de résoudre ces problèmes, des approches décentralisées sont proposées dans la littérature, tirant parti du paradigme de l'informatique dans le brouillard. Dans de tels cas, les calculs sont diffusés entre les nœuds Fog afin de se rapprocher des capteurs produisant les données. Toutefois, ces approches de raisonnement distribué ne traitent pas de l'emplacement des règles : il est statique, calculé au moment de la conception, et tous les nœuds exécutent le même ensemble de règles.

Avec EDR, les contributions décrites dans ce mémoire permettent de remédier à ces lacunes en tirant parti de la complémentarité entre le Cloud et le Fog afin d'associer des nœuds

26. <https://framagit.org/nseydoux/edr>

puissants distants et stables et des ressources locales, limitées et disponibles de manière opportuniste. EDR est une approche générique de raisonnement fondée sur des règles réparties de manière dynamique, reposant sur des règles SHACL modulaires. L'exécution par les nœuds Fog des fonctionnalités d'EDR est contrôlée via un vocabulaire dédié décrivant les connaissances dans les bases de connaissances de chaque nœud. Ce vocabulaire est utilisé par les modules de règles pour mettre en œuvre des stratégies de déploiement permettant la propagation de règles entre voisins à travers le niveau Fog du modèle Cloud-Fog-Device. Les stratégies de déploiement visent à optimiser le positionnement des règles pour des critères personnalisables, tels que le temps de réponse (considéré dans les travaux présentés) ou la consommation d'énergie, en fonction des connaissances stockées dans les bases de connaissances de chaque nœud.

Ces connaissances incluent une description de ses voisins, l'état actuel de l'environnement fondé sur les observations de capteurs et des connaissances de base. Dans l'ensemble, EDR déploie de manière purement décentralisée les règles. Afin de renforcer sa généricité, l'approche EDR elle-même est rendue agnostique aux stratégies de déploiement individuelles. Elle doit être affinée en injectant des règles incorporant leur propre stratégie de déploiement, sélectionnées en fonction des exigences au niveau de l'application.

Pour montrer l'intérêt de notre contribution, nous avons proposé EDRT, un raffinement d'EDR mettant en œuvre une stratégie de déploiement orientée vers la réduction des délais pour la transmission des déductions aux applications. EDRT vise à déployer des règles sur les nœuds Fog au plus près des capteurs, tout en évitant les calculs inutiles. Les règles sont ainsi propagées vers les capteurs produisant le type de données qu'ils consomment, aussi profondément que possible dans la topologie.

Par conséquent, décharger l'infrastructure Cloud en appliquant les règles dans le Fog, tout en considérant le nœud Cloud à la fois comme une ressource de calcul et comme un point d'entrée des règles dans le réseau, permet des déploiements évolutifs pour le SWoT.

Cependant, ne pas tenir compte des ressources disponibles dans le brouillard a montré des limites, et dans les travaux futurs, nous avons l'intention de développer des stratégies de distribution capables d'équilibrer la charge entre les nœuds Cloud et Fog en fonction des capacités des nœuds.

Les travaux futurs incluront une évaluation détaillée des performances de ce mécanisme d'adaptation. Nous souhaitons également étendre nos travaux pour faciliter l'expression des différents modules composant les règles. La généricité obtenue permet la mise en œuvre de plusieurs politiques, mais elle nécessite de la part de l'équipe de développement une expertise complète du SWoT afin de créer ces différents modules. Nous envisageons pour cela de mettre en place une interface visant à générer les modules à partir de contraintes exprimées en langage naturel.

7.1 Des ontologies pour décrire

Les ontologies sont au cœur des travaux présentés dans ce manuscrit. Nous avons montré qu'elles permettaient de décrire différents types de ressources. Ces ressources peuvent correspondre à des notions d'intérêt pour une communauté spécialisée. Un domaine d'application a été au cœur d'une partie importante de nos travaux : le domaine de l'agriculture, considéré dans la section 3.2 et dans la section 5.3. Les ressources peuvent aussi être des entités du monde réel dont les descriptions peuvent être utiles pour permettre à un utilisateur final de retrouver des informations ; c'est le cas pour ce qui concerne les œuvres cinématographiques ou musicales considérées dans le cadre de l'évaluation du système de questions-réponses présenté dans la section 5.2 ou sur les événements étudiés dans le cadre des travaux présentés dans la section 4.3. Comme nous l'avons décrit dans la section 4.2, la description de textes a également un intérêt pour les systèmes de recherche d'information documentaire. Nos travaux plus récents présentés dans le chapitre 6 montrent également que décrire à l'aide d'ontologies les observations faites par des capteurs d'une part, et les caractéristiques des objets d'un réseau IoT d'autre part, facilite l'interopérabilité au sein des systèmes qui les exploitent.

Dans le cadre de nos travaux, les ontologies sont également utilisées pour représenter la connaissance nécessaire à la mise en œuvre des tâches que les systèmes qui les intègrent proposent. Ces approches se placent dans la continuité de mon travail de thèse [Hernandez \(2005\)](#). J'avais alors montré que, pour des tâches de recherche d'information, l'utilisation conjointe d'une ontologie représentant le domaine couvert par les documents et d'une ontologie représentant les caractéristiques des documents utiles pour la tâche permettaient d'améliorer la pertinence des documents retournés. Dans les travaux présentés dans le chapitre 5, les ontologies sont également utilisées pour faciliter la mise en œuvre du processus de transformation d'une requête exprimée en langage naturel en une requête SPARQL. Les ontologies proposées permettent de décrire les requêtes et les patrons de requêtes que le système de questions/réponses manipule. Ces descriptions sont stockées et manipulées lors des différents traitements. La base de connaissances qu'elles constituent est alors utilisée comme un système de cache, ce qui évite la répétition de traitements déjà réalisés. L'utilisation de ces ontologies rend également l'approche plus facilement réutilisable. Une interface a été mise en place pour faciliter la définition de patrons à partir des entités définies dans l'ontologie. La publication sur le Web de la base de patrons permettrait également de les partager et de les compléter.

Un ontologie est également utilisée pour décrire les informations définissant le comportement d'un nœud dans l'approche EDR destinée à la distribution de règles de déduction dans un réseau d'objets et décrite dans la section 6.3. L'ontologie permet de décrire les entités

de la base de connaissances du nœud qui doivent être utilisées pour générer le contenu des messages émis par le nœud ainsi que pour définir les destinataires des messages.

Les ontologies ont donc un intérêt double : décrire les ressources mais aussi décrire les informations utiles aux traitements mis en œuvre par le système dans lequel elles sont intégrées.

7.2 Des patrons ou formes pour façonner

Une autre caractéristique commune à nos différents travaux est de considérer que les connaissances représentées dans une base de connaissances gagnent à être manipulées à partir de patrons ou de formes prédéfinies. Dans la section 3.2, nous avons présenté une approche permettant de guider la réutilisation de descriptions contenues dans une base de connaissances à partir d’une amorce de “module” qui représente la “forme” que doivent prendre les descriptions pour être utilisées dans un nouveau système. De façon similaire, les patrons de requêtes considérés dans le système de questions-réponses Swip et présentés dans la section 5.2.4.1 permettent d’identifier des formes de la base de connaissances qui correspondent à de réels besoins en information des utilisateurs. Cette notion de patrons est également considérée pour extraire des descriptions d’images à partir de textes. Dans nos travaux décrits dans Jaouachi et al. et non présentés dans ce manuscrit, le parti pris a été de ne pas décrire une image de façon exhaustive, mais de guider son annotation à partir d’éléments susceptibles d’intéresser des utilisateurs finals car représentés dans des patrons de requêtes. La notion de forme se retrouve également dans l’approche que nous avons proposée pour faire évoluer une RTO et les descriptions de textes qui en dépendent, présentée dans la section 4.2. Des formes d’annotation sont définies pour évaluer la qualité des descriptions. Ces formes sont utilisées pour identifier un besoin de changement de la RTO et/ou des descriptions.

Le récent engouement pour les formalismes permettant de représenter des formes (ou “shapes”) montre que ce type de représentations suscite également l’intérêt des chercheurs du Web Sémantique. Deux principaux langages, SHACL¹ et ShEx² ont en effet été proposés pour décrire et représenter la ou les formes que les graphes RDF doivent prendre. Ces langages prennent le contrepied des standards tels que RDFS ou OWL qui, reposant sur l’hypothèse du monde ouvert, n’ont pas pour objectif de poser des contraintes sur la connaissance à représenter. Au-delà de l’intérêt qu’elles présentent pour la validation de graphes RDF Labra Gayo et al. (2017), ces formes ont aussi pour objectif de faciliter la spécification des graphes attendus par des interfaces de communication (services Web par exemple) ainsi que la génération de formulaires pour la manipulation de données par les utilisateurs Prud’hommeaux et al. (2014). De notre point de vue, les formes ne doivent pas se limiter à une façon de contraindre les descriptions RDF de ressources ou à un moyen de transformer des descriptions d’un schéma en un autre (comme XSLT³ le permet pour les documents XML). Elles doivent être exploitées en tirant profit des ontologies qui sont utilisées pour décrire les données qu’elles “façonnent” afin d’identifier les connaissances utiles pour des applications. Nous voyons dans ces formes une façon de représenter une vue “sémantique” sur les données. Bien que la forme contraigne les descriptions qui sont considérées sur ces données, la sémantique de ces descriptions reste disponible au niveau de l’ontologie. Décrire, partager et réutiliser les formes identifiées dans les bases de connaissances du LOD permettrait de donner un nouveau potentiel à ces bases de connaissances en mettant en évidence la connaissance utile pour des applications à l’image de ce que nous faisons dans nos travaux.

1. <https://www.w3.org/TR/shacl/>

2. <http://shex.io/shex-semantics/>

3. <https://www.w3.org/TR/xslt/all/>

7.3 Réutiliser des ontologies ?

De récents travaux montrent que les ontologies sont en réalité très peu réutilisées sur le LOD [Fernández-López et al. \(2019\)](#). Dans la suite de cette section, nous cherchons des éléments de réponse à ce constat et traçons des perspectives que nous souhaitons aborder à l’avenir.

7.3.1 Place de la “sémantique”

En 2001, Tim Berners-Lee posait les fondations du Web Sémantique, dont l’objectif principal était d’ajouter une couche de sémantique au Web. Près d’une vingtaine d’années plus tard, cette “sémantique” prend différentes formes. L’adoption du modèle RDF et ses différents formats de sérialisation est une réalité tangible, attestée par le nombre de bases de connaissances (1,239 bases de connaissances disponibles en mars 2019⁴) et de triplets (près de 150 milliards de triplets en 2016⁵) disponibles sur le LOD, mais également par la quantité de pages Web contenant des descriptions dans les formats de sérialisation adaptés aux pages que sont RDFa⁶, microdata⁷, microformat⁸ [Bizer et al. \(2013\)](#). RDF joue donc le rôle de modèle uniforme permettant de lier les ressources à la couche de sémantique, facilitant ainsi l’interopérabilité syntaxique. La sémantique des descriptions RDF repose quant à elle sur les entités des ontologies que ces descriptions réfèrent. Accessible à partir d’une URI dérivable (si les ontologies suivent les principes du LOD), la représentation de ces entités n’est cependant pas toujours facilement interprétable par les humains. Lorsque cette représentation repose sur une forte composante axiomatique, l’interprétation nécessite la maîtrise des constructeurs des familles de Logiques de Description utilisées. Or les acteurs du Web (que ce soient les développeurs Web ou les personnes en charge de publier des données) n’ont souvent pas les compétences requises. [Zaveri et al. \(2016\)](#) listent un ensemble d’erreurs dans l’usage des formalismes menant à l’inconsistance des bases de connaissances disponibles sur le LOD.

Par ailleurs, lorsque l’ontologie est volumineuse et qu’une entité est définie à l’aide d’autres entités de l’ontologie, appréhender la sémantique choisie par les concepteurs n’est pas toujours aisé. La documentation disponible ainsi que les différentes annotations au niveau de l’entité elle-même sont des éléments indispensables pour nous aider dans cette tâche. Des exemples de descriptions réalisées à partir de l’ontologie tels que ceux qui sont fournis dans la documentation de SSN-SOSA⁹ sont très utiles. Dans la réalité, cette documentation est très peu disponible et l’accès à la représentation de l’ontologie via son IRI n’est possible que pour un sous-ensemble des ontologies publiées [Fernández-López et al. \(2019\)](#).

Parallèlement à ce constat, une analyse des vocabulaires utilisés sur le LOD en 2014 [Schmachtenberg et al. \(2014\)](#) montre que certains vocabulaires tels que FOAF¹⁰ et DC-terms¹¹ peuvent être qualifiés de “populaires”, car ils sont utilisés dans une grande proportion de bases de connaissances du LOD, quels que soient les domaines couverts. Notons que l’interprétation de la sémantique de ces vocabulaires est facilitée par la simplicité des notions qu’ils définissent.

L’engouement pour les vocabulaires reposant sur de la sémantique “informelle” se retrouve également dans l’utilisation massive du vocabulaire *schema.org* pour décrire le contenu des pages Web. Même si la raison principale de son utilisation repose sur l’influence qu’ont aujourd’hui les acteurs du Web qui le promeuvent, la réutilisation de ce vocabulaire repose aussi

4. <https://lod\protect\discretionary{\char\hyphenchar\font}{-}{cloud.net>

5. <http://lodstats.aksw.org/> et [Ermilov et al. \(2016\)](#)

6. <https://www.w3.org/TR/html\protect\discretionary{\char\hyphenchar\font}{-}{rdfa/>

7. <https://www.w3.org/TR/microdata/>

8. http://microformats.org/wiki/Main_Page-fr ou JSON-LD <https://www.w3.org/TR/json-ld/>

9. <https://www.w3.org/TR/vocab-ssn/>

10. <http://xmlns.com/foaf/spec/>

11. <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/2012-06-14/?v=terms>

sur la facilité qu’ont les développeurs à appréhender sa sémantique [Mika \(2015\)](#). Appauvrir la sémantique des ontologies est également suivi par le W3C pour les ontologies qu’il promeut. L’ontologie SOSA-SSN a par exemple été proposée comme une évolution de l’ontologie SSN dans l’objectif d’élargir le public qui pourra la réutiliser. Dans le module cœur de SOSA SSN ¹², la signature des propriétés n’est plus définie à partir des constructeurs dédiés du langage RDFS, mais à partir des propriétés à la sémantique “informelle” `domainIncludes` ¹³ et `rangeIncludes` ¹⁴ qui sont des propriétés d’annotation définies dans `schema.org`.

La volonté d’utiliser des ontologies pour décrire des données et la difficulté pour certains acteurs du Web à appréhender leur sémantique a mené à un Web des données liées fortement hétérogène.

Nous partageons le point de vue défendu dans [Fernández-López et al. \(2019\)](#) selon lequel pour être réutilisée, une ontologie doit être soutenue par une communauté qui veille à maintenir à jour une documentation portant aussi bien sur les exigences auxquelles l’ontologie répond que sur l’explicitation de la sémantique des différentes entités. Cette explicitation peut se faire en ajoutant des propriétés d’annotation à ces entités mais également en fournissant des exemples de données décrites à partir de l’ontologie. La communauté doit également être réactive par rapport aux demandes d’information et de mises à jour adressées par les acteurs du Web.

7.3.2 Faciliter la réutilisation de vocabulaires

Dans le cadre d’un Contrat de Recherche Externalisé mené en collaboration avec la société Orange et le LAAS, nous abordons depuis quelques mois la réutilisation de vocabulaires par des non-experts du Web Sémantique. Orange propose la plate-forme Thing’in ¹⁵ qui représente les environnements cyber-physiques comme des systèmes, et permet de cataloguer et de rechercher les objets qui les composent. La volonté d’Orange est de référencer l’ensemble des objets disponibles, peu importent la plate-forme et l’application qui les pilotent, de façon à rendre ces objets possiblement réutilisables dans d’autres applications. Une condition nécessaire pour que Thing’in satisfasse cet objectif est que des descriptions des objets soient intégrées à la plate-forme. Cependant, ces descriptions sont généralement disponibles dans des formats hétérogènes (tableaux de valeurs, objets JSON), issus pour la plupart de la documentation destinée aux utilisateurs de ces objets ou des plates-formes qui gèrent les objets (plates-formes IoT ou open data). La plate-forme Thing’in reposant sur des graphes de propriétés tout en étant compatible avec les principes du LOD, elle offre la possibilité de décrire les objets à partir du vocabulaire RDF et d’ontologies qu’elle inventorie. Le parti pris d’Orange, que nous avons encouragé, est de ne pas développer pour chaque jeu de données une nouvelle ontologie mais de réutiliser des ontologies qui ont déjà été utilisées dans la plate-forme, quitte à ne couvrir que partiellement les descriptions dites “brutes” des objets. La motivation sous-jacente est de faciliter le requêtage des descriptions tout en limitant les alignements entre ontologies à produire. Le processus d’injection de description dans Thing’in consiste actuellement à analyser les descriptions disponibles sur un objet, consulter la centaine d’ontologies disponibles dans la plate-forme et produire manuellement cette description. La plate-forme étant à destination des partenaires d’Orange, la génération manuelle de telles descriptions n’est pas suffisante. Comme nous l’avons vu dans le chapitre 4, des langages ont été proposés pour transformer les données dites brutes en descriptions reposant sur des ontologies. Cependant, leur utilisation est complexe et elle implique d’avoir identifié l’élément de la description brute de l’objet et l’entité de l’ontologie à mettre en correspondance. Nous travaillons actuellement sur une approche semi-automatique visant à solliciter l’utilisateur d’une part pour expliciter la sémantique des descriptions brutes et d’autre part pour valider

12. <https://www.w3.org/TR/vocab-ssn/#Developments>

13. <https://schema.org/domainIncludes>

14. <https://schema.org/rangeIncludes>

15. <https://hellofuture.orange.com/fr/thingin-la-plateforme-du-graphe-des-objets/>

les descriptions RDF générées. Bien que développée pour Thing’in, notre approche se voudra générique dans la mesure où elle accède à un entrepôt d’ontologies via une API REST et injecte la description produite via le même type de services. Au-delà des cas d’utilisations identifiés par Orange, nous avons également pour ambition de la mettre en place pour enrichir le LOD de façon à sémantiser des descriptions brutes et prendre en compte des ontologies du LOV pour aider à la publication de données.

Nous voyons dans ce projet l’opportunité d’aborder différentes problématiques qui s’ouvrent comme perspectives à nos travaux :

- quels sont les critères à respecter pour que la description sémantique d’un objet permette sa réutilisation dans différents contextes non identifiés au moment de la génération de cette description ?
- comment expliciter la sémantique des descriptions brutes dans le but de faciliter leur mise en correspondance avec des entités d’ontologies ?
- comment proposer automatiquement des descriptions d’objets à partir d’entités d’ontologies dont la sémantique est souvent informelle ou dépendante de plusieurs entités ?
- comment évaluer la qualité des descriptions générées alors qu’*a priori* elles ne couvriront qu’une partie des descriptions brutes ?
- comment “verbaliser” à destination de l’utilisateur la description générée et les entités des ontologies sur lesquelles elle repose pour qu’il puisse valider sa pertinence ?

7.4 Les ontologies pour gérer l’interopérabilité ?

Les ontologies ont pour but de favoriser l’interopérabilité. Cependant, comme nous l’avons précédemment évoqué, de nombreuses ontologies sont aujourd’hui disponibles pour différents domaines [Fernández-López et al. \(2019\)](#). L’intérêt grandissant pour les techniques d’alignement laisse penser que ce n’est pas au niveau des ontologies mais au niveau de leur mise en correspondance que l’interopérabilité est concrètement mise en œuvre.

7.4.1 L’interopérabilité au niveau de l’alignement

Comme nous l’avons illustré dans la section 5.3 et analysé dans [Thiéblin et al. \(2019\)](#), les alignements complexes sont considérés pour exprimer de façon plus complète les correspondances qui peuvent être identifiées entre différentes ontologies d’un même domaine. L’étude des approches existantes montre que, contrairement à l’intuition que l’on peut avoir, mettre en correspondance de façon plus expressive des ontologies ne repose pas forcément sur des techniques plus sophistiquées que celles employées pour générer des alignements simples. La plupart des approches considèrent les ontologies comme des graphes, des arbres ou des ensembles de descriptions indépendamment de leur expressivité. La classification que nous avons proposée montre que les approches peuvent être caractérisées par le type de correspondances générées ou la structure utilisée pour guider le processus de détection des correspondances. Certaines approches sont mono-stratégie alors que d’autres en combinent plusieurs. Alors que certaines approches reposent sur l’existence de correspondances simples, d’autres exploitent des ressources externes telles que les interfaces de requêtage sur le Web, des règles ou encore des questions de compétences. Une partie des approches exploitent les instances des bases de connaissances mais sont, dans ce cas-là, dépendantes de la qualité de leurs contenus. Comme pour les alignements simples, les approches d’alignement complexe se consacrent à identifier des correspondances d’équivalence. Certaines correspondances sont par ailleurs cohérentes dans le contexte où elles sont produites alors qu’elles ne sont pas équivalentes du point de vue logique. Dans cette perspective, des approches comme celle présentée dans [Bouquet et al. \(2003\)](#) visent à représenter le contexte dans lequel l’alignement est valide. De notre point de vue, dans les mêmes perspectives que celles présentées pour la réutilisation d’ontologies, les

alignements doivent être documentés, partagés dans le cadre de communautés puis maintenus pour être concrètement utilisables dans une perspective d'interopérabilité. Cependant, au vu de la maîtrise des formalismes sur lesquels elles reposent, mettre en place ce mode de partage est un défi de taille qui attend la communauté du Web Sémantique qui se consacre à l'alignement. Dans nos travaux, nous proposons de relever ce défi au niveau des domaines pour lesquels les ontologies sont construites. Une perspective de nos travaux est donc d'aborder la question de l'alignement dans le cadre de l'IOT.

7.4.2 Un défi qui attend l'IoT

La problématique de l'alignement est en effet cruciale dans le domaine de l'IoT. L'interopérabilité sémantique a pour l'instant été abordée en intégrant des ontologies pour mettre en place les différentes fonctions dont l'IoT a besoin comme nous l'avons présenté dans la section 6.2.3. Nous avons également mentionné dans la section 3.3 du chapitre le fait que de nombreuses ontologies sont aujourd'hui disponibles dans le domaine. Les différents acteurs du domaine, qui ont par ailleurs du mal à se mettre d'accord au niveau des normes et des standards, promeuvent désormais chacun leur propre ontologie. Aussi, les bases de connaissances en lien avec l'IoT (consommation en énergie des bâtiments intelligents, production de fermes expérimentales...) publiées sur le Web et décrites avec ces différentes ontologies se multiplient. Aligner leurs contenus permettrait aux chercheurs de nombreuses disciplines d'utiliser ces données pour enrichir leurs analyses. De la même façon, aligner les ontologies utilisées au sein même des plates-formes IoT permettrait à des objets intégrés à différents systèmes et reposant sur différentes ontologies d'interagir. Dans cette perspective, nous souhaitons mettre en place une approche qui permettrait de répertorier et de documenter les alignements entre ontologies pour l'IoT. À l'image de ce que nous avons proposé pour IoT-O et de ce qui est proposé pour SSN SOSA, des alignements entre ontologies commencent à être définis. Or ces alignements ne sont que très peu documentés et il est difficile d'en avoir une vue globale. Nous souhaitons, à l'instar de ce qui est fait sur le LOV4IoT, développer une plate-forme qui regrouperait les alignements existants en explicitant le contexte dans lequel ils ont été faits (intention des concepteurs, sémantique de la relation exprimée au niveau de la correspondance, exemples d'utilisation des correspondances). Expliciter ce contexte impose de définir des formalismes pour les représenter mais aussi de mettre en place des outils qui permettront aux utilisateurs de les visualiser.

7.5 Re-décentraliser le Web Sémantique

Le Web et le Web Sémantique ont été pensés pour être décentralisés. Pour mettre en œuvre le Web, Tim Berners-Lee a proposé une architecture visant à répartir des informations sur différents serveurs tout en proposant des technologies et des solutions libres pour mettre en œuvre ces serveurs et les navigateurs déployés sur des machines clientes. Le Web devait être un espace où aucune autorité ne pourrait décider qui peut s'exprimer et ce qui peut être dit. Ce nouveau mode de partage de l'information a révolutionné l'accès à l'information et la communication pour tout un chacun. Cependant, une dérive que l'on peut attribuer à l'absence d'autorité a mené au développement de grandes plates-formes gérées par des acteurs qui, aujourd'hui, dominent le Web. Le paradoxe est que ces plates-formes centralisent les données qu'elles collectent et manipulent [Berners-Lee \(2017\)](#). Parallèlement, l'architecture clients-serveur sur laquelle repose le Web a été largement utilisée pour déployer des traitements sur les serveurs, laissant souvent de côté les ressources des machines clientes. En plus de n'exploiter qu'une partie des machines de l'architecture, cette pratique peut présenter des limites quand les serveurs stockent une grande masse de données et supportent de nombreux traitements. Dans ce contexte, les serveurs peuvent être vus comme des goulots d'étranglement. Une tendance actuelle des travaux du Web Sémantique consiste à décentraliser le stockage des données et leur requête [Aebeloe et al. \(2019\)](#). Des travaux comme

ceux présentés dans [Verborgh et al. \(2016\)](#) ou [Minier et al. \(2018\)](#) visent à déplacer une partie du traitement nécessaire au requêtage de grandes bases de connaissances au niveau du navigateur. De notre point de vue, déployer des traitements sur les différentes machines du Web est nécessaire pour supporter la quantité de ressources qui y sont aujourd’hui publiées. Nous souhaitons poursuivre les travaux que nous avons amorcés sur le SWoT en définissant de nouvelles stratégies de distribution de règles prenant en compte les capacités et l’état des machines en temps réel. Nous souhaitons également exploiter ce genre d’approches pour mettre en place des traitements à partir de stratégies qui ont pour objectif de limiter la consommation d’énergie.

7.6 Quelle place pour les utilisateurs finals dans l’IoT ?

7.6.1 Agir sur le réseau d’objets à partir de la langue naturelle

L’IoT est un réseau de machines dans lequel l’humain n’a pas réellement sa place. La description par les technologies du Web Sémantique de ses objets et des données qu’ils produisent permet de favoriser l’interopérabilité entre machines. Cela pourrait également permettre d’intégrer l’humain à l’IoT. Pour vérifier cette hypothèse, nous envisageons de poursuivre nos travaux [Lannes et al. \(2017\)](#) visant à intégrer à un réseau d’objets connectés un système de questions/réponses. Nous souhaitons étendre le système pour l’adapter aux spécificités des requêtes exprimées à destination des bases de connaissances de l’IoT. Il s’agira alors de prendre en compte les informations spatio-temporelles pouvant être exprimées dans une requête, mais aussi d’adapter Swip à la prise en compte de requêtes sur des flux de données. Les données issues de capteurs sont en effet généralement disponibles en temps réel sous forme de flux. Les approches présentées dans [Le-Phuoc et al. \(2016\)](#) ou [Perera et al. \(2014b\)](#) gagneraient à être rendues plus accessibles à l’utilisateur par l’expression de requêtes en langue naturelle.

Nous souhaitons également proposer une approche permettant d’exprimer des commandes à destination des objets en capacité d’agir sur l’environnement (aussi appelés actionneurs) présents dans un réseau IoT. [Mayer et al. \(2014\)](#) insiste en effet sur le fait que des interfaces pour faciliter les interactions entre utilisateurs et objets de l’IoT sont nécessaires. Nous avons montré dans [Seydoux et al. \(2016b\)](#) qu’il était possible de piloter ce type d’objets à partir de politiques de gestion définies dans une base de connaissances. Dans de futurs travaux, nous souhaitons permettre à l’utilisateur de définir ces politiques de gestion et les commandes qu’il souhaite mettre en œuvre sur le réseau en langue naturelle. Les commandes étant liées aux fonctionnalités déployées sur les objets, notre approche par patrons paraît adaptée.

7.6.2 Les données personnelles

Le développement de l’IoT suscite auprès du grand public une méfiance grandissante vis-à-vis de l’exploitation des données personnelles. Les objets n’interagissent pas seulement avec l’environnement mais aussi avec les individus.

De fait, l’Internet des objets peut être très intrusif vis-à-vis des personnes évoluant dans des environnements connectés. Dans le déluge de données qu’il gère, une grande partie d’entre elles constituent des données à caractère personnel, c’est-à-dire des données permettant d’identifier la personne, directement ou indirectement. Certaines sont même susceptibles d’être des données sensibles, se rapportant par exemple à la santé, aux opinions ou à l’orientation sexuelle de chacun. Dans les architectures technologiques existantes, il est difficile d’appréhender et de contrôler les données selon leur nature, alors que le régime juridique applicable peut se révéler particulièrement exigeant pour le responsable de traitement, c’est-à-dire la personne définissant la finalité du traitement et fournissant les moyens techniques de le mener. La protection des données personnelles est assurée dans l’Union Européenne par le Règlement Général de Protection des Données (RGPD) qui est entré en vigueur le 25 mai 2018. Il octroie des droits aux personnes concernées par les données : droit à l’information,

droits d'accès, de rectification, d'opposition, d'effacement. Le responsable de traitement doit assumer de nouvelles obligations pour prouver sa conformité au règlement, telle que la tenue du registre des traitements. Par ailleurs, de nouveaux concepts apparaissent, tels la *privacy by design* et la *privacy by default*, selon lesquels la protection des données personnelles et de la vie privée des individus doit être prise en compte dès la conception d'un service. Les avancées récentes dans le domaine du SWoT ont montré leur efficacité afin de gommer les hétérogénéités inhérentes à l'IoT. Cependant, ces approches ne prennent généralement pas en compte dans leurs descriptions les aspects législatifs en lien avec les données personnelles. Nous souhaitons poursuivre nos recherches en exploitant les technologies du Web Sémantique pour faciliter la maîtrise des données personnelles collectées sur n'importe quel type d'utilisateur de l'IoT. Nous souhaitons donc :

- proposer une infrastructure à même d'héberger une gestion des données et des services en accord avec la réglementation et les aspirations des individus ;
- proposer des vocabulaires (ou ontologies) et un mécanisme d'enrichissement sémantique de la donnée personnelle permettant de décrire au mieux sa nature en vue de la protéger ;
- proposer des mécanismes distribués d'utilisations et de traitements au "juste" endroit avec le niveau de visibilité et de fiabilité adéquate en s'appuyant sur une architecture de type fog computing.

L'extension des travaux préliminaires ayant mené à la proposition de l'algorithme de distribution du raisonnement EDR est une piste envisagée. Nous souhaitons en effet mettre en place des stratégies de propagation prenant en compte les droits associées aux données collectées.

Les travaux de [Kirrane et al. \(2018\)](#) et [Delanaux et al. \(2018\)](#) ont montré que les technologies du Web Sémantique étaient adaptées pour mettre en place des politiques d'anonymisation des données avant de les publier sur le LOD. L'exploitation de ces technologies permet également d'envisager une façon de restaurer la confiance du public en exploitant la connaissance représentée pour expliciter sous une forme intelligible par un utilisateur final la nature des données collectées ainsi que les traitements qui en sont faits. Il s'agira alors de proposer des approches pour "verbaliser" les descriptions sémantiques des données collectées ou des traitements sur ces données. Par ailleurs, il peut également être envisagé de prévenir des usages illégitimes en permettant à l'utilisateur de définir des politiques de gestion de ces données qui seront intégrées au réseau. Comme nous l'avons évoqué, ce type de politiques doit alors pouvoir être exprimé par l'utilisateur et intégré au réseau.

Solid¹⁶, pour Social Linked Data, est une récente initiative de Tim Berners-Lee destinée à inciter les utilisateurs finals et les organismes du Web à dissocier le stockage et la gestion des données les concernant des applications qui les utilisent. En reprenant la démarche suivie lors de l'invention du Web, Solid repose sur un ensemble de standards du W3C et sur des implémentations permettant de rendre concrète cette initiative. Brièvement, les données personnelles sont stockées sur des POD (Personal Online Datastore) dans lesquels les détenteurs des données définissent les droits octroyés aux différentes applications. Les perspectives que nous souhaitons développer dans la suite de nos travaux seront suivies d'implémentations pouvant contribuer à Solid. En effet, il nous semble indispensable de soutenir cette initiative qui permettrait au Web de retrouver sa philosophie originelle. Sans un soutien fort des développeurs et des chercheurs pour Solid, l'accès au Web par les utilisateurs finals présentera à l'avenir de plus en plus de risques pour les individus, car les acteurs majeurs du domaine ont montré jusqu'alors qu'ils avaient comme principale préoccupation la mercantilisation de nos données personnelles.

16. <https://solid.inrupt.com/>

Bibliographie

- Framework of the web of things. Technical report, International Telecommunication Union, 2012a. *Cité page 13*
- Overview of the web of things (y.2060). Technical report, International Telecommunication Union, 2012b. *Cité page 12*
- Internet Health Report. Technical report, Mozilla Foundation, 2018. URL <https://internethealthreport.org/>. *Cité page 12*
- Rafik Abbas. *Filtrage et agrégation d'informations vitales relatives à des entités. (Filtering and aggregating vital information related to entities)*. PhD thesis, Paul Sabatier University, Toulouse, France, 2015. URL <https://tel.archives-ouvertes.fr/tel-01266560>. *Cité page 37*
- Rafik Abbas, Karen Pinel-Sauvagnat, Nathalie Hernandez, and Mohand Boughanem. IRIT at trec knowledge base acceleration 2013 : Cumulative citation recommendation task. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA., 2013. *Cité page 42*
- Rafik Abbas, Nathalie Hernandez, Karen Pinel-Sauvagnat, and Mohand Boughanem. Détection d'informations vitales pour la mise à jour de bases de connaissances. In *IC 2015 : 26es Journées francophones d'Ingénierie des Connaissances (Proceedings of the 26th French Knowledge Engineering Conference)*, Rennes, France, July 1-3, 2015., 2015a. *Cité page 37*
- Rafik Abbas, Nathalie Hernandez, Karen Pinel-Sauvagnat, and Mohand Boughanem. Accelerating the update of knowledge base instances by detecting vital information from a document stream. In *International Conference on Web Intelligence and Intelligent Agent Technology (IEEE/WIC/ACM)*, pages 173–176, 2015b. *4 citations pages 5, 9, 37, et 46*
- Rafik Abbas, Karen Pinel-Sauvagnat, Nathalie Hernandez, and Mohand Boughanem. Leveraging temporal expressions to filter vital documents related to an entity. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC)*, pages 1093–1098, 2015c. *Cité page 5*
- Rafik Abbas, Karen Pinel-Sauvagnat, Nathalie Hernandez, and Mohand Boughanem. Leveraging temporal expressions to filter vital documents related to an entity. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, April 13-17, 2015*, pages 1093–1098, 2015d. URL <http://dx.doi.org/10.1145/2695664.2695910>. *Cité page 42*
- Christian Aebeloe, Gabriela Montoya, and Katja Hose. A decentralized architecture for sharing and querying semantic data. In *European Semantic Web Conference*, pages 3–18. Springer, 2019. *Cité page 100*
- François Aïssaoui, Guillaume Garzone, and Nicolas Seydoux. Providing interoperability for autonomic control of connected devices. In *InterIoT*, pages 1–7, 2016. *2 citations pages 75 et 77*
- Mahmud Al-Osta, Bali Ahmed, and Gherbi Abdelouahed. A Lightweight Semantic Web-based Approach for Data Annotation on IoT Gateways. *Procedia Computer Science*, 113 :186–193, 2017. ISSN 18770509. doi : 10.1016/j.procs.2017.08.339. URL <http://dx.doi.org/10.1016/j.procs.2017.08.339>. *Cité page 75*

- Faisal Alkhateeb, Jean-François Baget, and Jérôme Euzenat. Extending sparql with regular expression patterns (for querying rdf). *Journal of Web Semantics*, 7(2) :57–73, 2009. *Cité page 51*
- Fabien Amarger. *Vers un système intelligent de capitalisation de connaissances pour l'agriculture durable : construction d'ontologies agricoles par transformation de sources existantes*. Thèse de doctorat, Université de Toulouse-le-Mirail, Toulouse, France, décembre 2015. URL https://www.irit.fr/publis/MELODI/Amarger_These_2015.pdf. (Soutenance le 19/02/2016). *4 citations pages 17, 21, 23, et 26*
- Fabien Amarger, Catherine Roussey, Ollivier Haemmerle, Nathalie Hernandez, and Romain Guillaume. Muska : un système de fusion d'ontologies fondé sur le consensus et l'estimation de la confiance. *Revue d'Intelligence Artificielle (RIA)*, 32 :313–344. *Cité page 24*
- Fabien Amarger, Ollivier Haemmerlé, Nathalie Hernandez, and Camille Pradel. Taking SPARQL 1.1 extensions into account in the SWIP system. In *Conceptual Structures for STEM Research and Education*, page 75–89, Bombay, Inde, 2013a. Springer. URL http://link.springer.com/chapter/10.1007/978-3-642-35786-2_7. *3 citations pages 48, 55, et 61*
- Fabien Amarger, Catherine Roussey, Jean-Pierre Chanet, Ollivier Haemmerlé, and Nathalie Hernandez. Etat de l'art : Extraction de connaissances à partir de thesaurus pour générer une ontologie. In *Actes du XXXIème Congrès INFORSID, Paris, France, 29-31 Mai 2013.*, pages 29–44, 2013b. URL http://inforsid.fr/actes/2013/2013_1a_1%20Amarger.pdf. *2 citations pages 19 et 21*
- Fabien Amarger, Jean-Pierre Chanet, Ollivier Haemmerlé, Nathalie Hernandez, and Catherine Roussey. SKOS sources transformations for ontology engineering : Agronomical taxonomy use case. In *Metadata and Semantics Research - 8th Research Conference, MTSR 2014, Karlsruhe, Germany, November 27-29, 2014. Proceedings*, pages 314–328, 2014. doi : 10.1007/978-3-319-13674-5_29. URL https://doi.org/10.1007/978-3-319-13674-5_29. *Cité page 17*
- Fabien Amarger, Jean-Pierre Chanet, Ollivier Haemmerlé, Nathalie Hernandez, and Catherine Roussey. Construction d'une ontologie par transformation de systèmes d'organisation des connaissances et évaluation de la confiance. *Ingénierie des Systèmes d'Information*, 20(3) :37–61, 2015. doi : 10.3166/isi.20.3.37-61. URL <https://doi.org/10.3166/isi.20.3.37-61>. *Cité page 17*
- Fabien Amarger, Jean-Pierre Chanet, Ollivier Haemmerlé, Nathalie Hernandez, and Catherine Roussey. Knowledge engineering method based on consensual knowledge and trust computation : The MUSCKA system. In *Graph-Based Representation and Reasoning - 22nd International Conference on Conceptual Structures (ICCS) Proceedings*, pages 177–190, 2016a. *3 citations pages 4, 11, et 17*
- Fabien Amarger, Jean-Pierre Chanet, Ollivier Haemmerlé, Nathalie Hernandez, and Catherine Roussey. Dealing with incompatibilities during a knowledge bases fusion process. In *Graph-Based Representation and Reasoning - 22nd International Conference on Conceptual Structures (ICCS) Proceedings*, pages 252–260, 2016b. *2 citations pages 4 et 17*
- Safina Showkat Ara, Zia Ush Shamszaman, and Ilyoung Chong. Web-of-objects based user-centric semantic service composition methodology in the internet of things. *International Journal of Distributed Sensor Networks*, 2014, 2014. ISSN 15501477. doi : 10.1155/2014/482873. *2 citations pages 77 et 78*
- Muhammad Ikram Ashraf, Leonardo Goratti, Jussi Haapola, and Carlos Pomalaza-Raez. Distributed semantic algorithm for power constrained publish/subscribe routing. In *2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 91–96. IEEE, dec 2010. ISBN 978-1-4244-7174-4. doi : 10.1109/ISSNIP.2010.5706761. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5706761>. *Cité page 78*
- Kevin Ashton. That 'internet of things' thing. *RFID journal*, 22(7) :97–114, 2009. *2 citations pages 12 et 17*
- Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Virgi Pavlu, and Tetsuya Sakai. Trec 2013 temporal summarization. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, USA., 2013. *Cité page 42*

- Nikolaos Athanasis, Vassilis Christophides, and Dimitris Kotzinos. Generating on the fly queries for the semantic web : The ics-forth graphical rql interface (grql). In *International Semantic Web Conference*, pages 486–501, 2004. *Cité page 51*
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia : A Nucleus for a Web of Open Data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web : The 6th International Semantic Web Conference ISWC and the 2nd Asian Semantic Web Conference ASWC*, volume 4825 of *LNCS*, pages 722–735, Busan, Korea, November 2007. Springer Berlin Heidelberg. ISBN 978-3-540-76297-3 978-3-540-76298-0. doi : 10.1007/978-3-540-76298-0_52. URL http://link.springer.com/10.1007/978-3-540-76298-0_52. *Cité page 64*
- Isabelle Augenstein, Sebastian Padó, and Sebastian Rudolph. Lodifier : Generating linked data from unstructured text. In *Proceedings of the 9th International Conference on The Semantic Web : Research and Applications, ESWC'12*, pages 210–224, Berlin, Heidelberg, 2012. ISBN 978-3-642-30283-1. doi : 10.1007/978-3-642-30284-8_21. URL http://dx.doi.org/10.1007/978-3-642-30284-8_21. *Cité page 41*
- Nathalie Aussenac-Gilles and Fabien Gandon. From the knowledge acquisition bottleneck to the knowledge acquisition overflow : A brief french history of knowledge acquisition. *Int. J. Hum.-Comput. Stud.*, 71(2) :157–165, 2013. doi : 10.1016/j.ijhcs.2012.10.009. URL <http://dx.doi.org/10.1016/j.ijhcs.2012.10.009>. *Cité page 15*
- Sasikanth Avancha, Chintan Patel, and Anupam Joshi. Ontology-driven adaptive sensor networks. *Proceedings of MOBIQUITOUS 2004 - 1st Annual International Conference on Mobile and Ubiquitous Systems : Networking and Services*, (Section 2) :194–202, 2004. doi : 10.1109/MOBIQ.2004.1331726. *2 citations pages 17 et 28*
- Payam Barnaghi, Stefan Meissner, Mirko Presser, and Klaus Moessner. Sense and sens' ability : Semantic data modelling for sensor networks, 2009. *Cité page 73*
- Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. Semantics for the Internet of Things : early progress and back to the future. *International Journal on Semantic Web and Information Systems*, 8(1) :1–21, 2012. ISSN 1552-6283. doi : 10.4018/jswis.2012010101. *Cité page 12*
- Hamid R. Bazoobandi, Steven De Rooij, Jacopo Urbani, Annette Ten Teije, Frank Van Harmelen, and Henri Bal. A Compact In-Memory Dictionary for RDF Data. In *ESWC*, volume 9088, pages 205–220, 2015. ISBN 9783319188171. doi : 10.1007/978-3-319-18818-8. *Cité page 80*
- Mahdi Ben-Alaya, Samir Medjiah, Thierry Monteil, and Khalil Drira. Toward semantic interoperability in oneM2M architecture. *IEEE Communications Magazine*, 53(12) :35–41, 2015. ISSN 0163-6804. doi : 10.1109/MCOM.2015.7355582. *3 citations pages 73, 78, et 83*
- Michael Bendersky and W. Bruce Croft. Analysis of long queries in a large scale search log. In *Proceedings of the 2009 workshop on Web Search Click Data*, page 8–14, 2009. URL <http://dl.acm.org/citation.cfm?id=1507511>. *Cité page 49*
- Fabio Benedetti, Sonia Bergamaschi, and Laura Po. Lodex : A tool for visual querying linked open data. In Serena Villata, Jeff Z. Pan, and Mauro Dragoni, editors, *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference*, volume 1486, Bethlehem, USA, october 2015. ceur-ws.org. URL <http://ceur-ws.org/Vol-1486/>. *Cité page 67*
- Berners-Lee, Tim, Hendler, James, and Lassila Ora. The semantic web. *Scientific american*, 284(5) : 28–37, 2001. *2 citations pages 3 et 72*
- Tim Berners-Lee. Three challenges for the web, according to its inventor. *World Wide Web Foundation. Recuperado de https://webfoundation.org/2017/03/web-turns-28-letter*, 2017. *Cité page 100*
- Timothy J Berners-Lee. Information management : A proposal. Technical report, 1989. *Cité page 47*

- A. Bernstein, E. Kaufmann, and C. Kaiser. Querying the semantic web with ginseng : A guided input natural language search engine. In *In 15th Workshop on Information Technologies and Systems Las Vegas NV*, number December, pages 45–50. MV-Wissenschaft, Münster, 2005. *Cité page 50*
- Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3) :1–22, MarMar 2009. ISSN 1552-6283. doi : 10.4018/jswis.2009081901. URL <http://dx.doi.org/10.4018/jswis.2009081901>. *Cité page 8*
- Christian Bizer, Kai Eckert, Robert Meusel, Hannes Mühleisen, Michael Schuhmacher, and Johanna Völker. Deployment of rdfa, microdata, and microformats on the web—a quantitative analysis. In *International Semantic Web Conference*, pages 17–32. Springer, 2013. *Cité page 97*
- Harold Boley, Michael Kifer, Paula-Lavinia Pătrânjan, and Axel Polleres. Rule interchange on the web. In *Reasoning Web International Summer School*, pages 269–309. Springer, 2007. *Cité page 82*
- Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16, New York, New York, USA, 2012. ACM Press. ISBN 978-1-4503-1519-7. doi : 10.1145/2342509.2342513. *Cité page 14*
- E Paslaru Bontas, Malgorzata Mochol, and Robert Tolksdorf. Case studies on ontology reuse. In *Proceedings of the IKNOW05 International Conference on Knowledge Management*, volume 74, page 345, 2005. *Cité page 16*
- Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-OWL : contextualizing ontologies. In Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, volume 2870 of *Lecture Notes in Computer Science*, pages 164–179. Springer, 2003. doi : 10.1007/978-3-540-39718-2_11. *Cité page 99*
- Coen Bron and Joep Kerbosch. Algorithm 457 : finding all cliques of an undirected graph. *Communications of the ACM*, pages 575–577, 1973. bibtex : bron_algorithm_1973. *Cité page 26*
- Patrice Buche, Juliette Dibie-Barthélemy, Liliana Ibanescu, and Abir Saïd. Modélisation d’une ressource termino-ontologique de domaine pour l’annotation sémantique de tableaux. In *Extraction et gestion des connaissances (EGC’2011), Actes, 25 au 29 janvier 2011, Brest, France*, pages 581–586, 2011. URL <http://editions-rnti.fr/?inprocid=1001021>. *Cité page 10*
- E. Cabrio, J. Cojan, A.P. Aprosio, B. Magnini, A. Lavelli, and F. Gandon. Qakis : an open domain qa system based on relational patterns. In *International Semantic Web Conference (Posters & Demos)*, 2012. *Cité page 52*
- Elena Cabrio, Philipp Cimiano, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Christina Unger, and Sebastian Walter. QALD-3 : multilingual question answering over linked data. Valencia, Spain, September 2013. URL <http://www.clef-initiative.eu/documents/71612/57dbe6e4-fe94-4cf9-963c-10f1dbbe3ef9>. *Cité page 60*
- Jean Charlet, Sylvie Szulman, Nathalie Aussenac-Gilles, Adeline Nazarenko, Nathalie Hernandez, Nadia Nadah, Éric Sardet, Jean Delahousse, Henry Valéry Téguiak, and Audrey Baneyx. Dafoe : une plateforme pour construire des ontologies à partir de textes et de thésaurus. In *Extraction et gestion des connaissances (EGC’2010), Actes, 26 au 29 janvier 2010, Hammamet, Tunisie*, pages 631–632, 2010. URL <http://editions-rnti.fr/?inprocid=1001386>. *Cité page 10*
- Victor Charpenay, K Sebastian, and Harald Kosch. Towards a Binary Object Notation for RDF. In *Extended Semantic Web Conference*, page 15, 2018. *Cité page 80*
- M Chein and M-L Mugnier. Conceptual graphs : fundamental notions. *Revue d’intelligence artificielle*, 6(4) :365–406, 1992. *Cité page 48*
- Alexandros Chortaras and Giorgos Stamou. Mapping diverse data to RDF in practice. In Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web - ISWC 2018 - 17th*

- International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, pages 441–457. Springer, 2018. doi : 10.1007/978-3-030-00671-6_26. *Cité page 19*
- Bizer Christian, Heath Tom, and Berners-Lee Tim. Linked data : The story so far. In *Semantic services, interoperability and web applications : emerging concepts*, pages 205–227. IGI Global, 2011. *Cité page 4*
- Benoit Christophe, Mathieu Boussard, Monique Lu, Alain Pastor, and Vincent Toubiana. The web of things vision : Things as a service and interaction patterns. *Bell Labs Technical Journal*, 16(1) : 55–61, jun 2011. ISSN 10897089. doi : 10.1002/bltj.20485. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6770362>. *Cité page 78*
- Philipp Cimiano. *Ontology learning and population from text : algorithms, evaluation and applications*, volume 27. Springer Science & Business Media, 2006. *Cité page 4*
- Fabio Ciravegna, Alexiei Dingli, Daniela Petrelli, and Yorick Wilks. User-system cooperation in document annotation based on information extraction. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 122–137. Springer, 2002. *Cité page 37*
- Aaron Clemmer and Stephen Davies. Smeagol : a "specific-to-general" semantic web query interface paradigm for novices. In *Proceedings of the 22nd international conference on Database and expert systems applications - Volume Part I, DEXA'11*, pages 288–302, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23087-5. URL <http://dl.acm.org/citation.cfm?id=2035368.2035396>. *Cité page 51*
- CoGui. A conceptual graph editor. Web site, 2009. <http://www.lirmm.fr/cogui/>. *Cité page 51*
- Catherine Comparot, Ollivier Haemmerlé, and Nathalie Hernandez. An easy way of expressing conceptual graph queries from keywords and query patterns. In *ICCS*, pages 84–96, 2010. *Cité page 48*
- Michael Compton, Cory Henson, Laurent Lefort, Holger Neuhaus, and Amit Sheth. A survey of the semantic specification of sensors. In *International Conference on Semantic Sensor Networks*, volume 522, pages 17–32, 2009. doi : 10.1.1.352.9902. URL <http://corescholar.libraries.wright.edu/cgi/viewcontent.cgi?article=2040&context=knoesis>. *3 citations pages 73, 75, et 77*
- Oscar Corcho and Raúl García-Castro. Five challenges for the Semantic Sensor Web. *Semantic Web*, 1(1) :121–125, 2010. ISSN 15700844. doi : 10.3233/SW-2010-0005. *Cité page 72*
- Gianluca Correndo and Nigel Shadbolt. Translating expressive ontology mappings into rewriting rules to implement query rewriting. In Pavel Shvaiko, Jérôme Euzenat, Tom Heath, Christoph Quix, Ming Mao, and Isabel Cruz, editors, *Proceedings of the 6th International Workshop on Ontology Matching (OM-2011) in conjunction with the International Semantic Web Conference (ISWC2011)*, volume 814, Bonn, Germany, October 2011. ceur-ws.org. URL <http://www.om2011.ontologymatching.org/>. *Cité page 66*
- Gianluca Correndo, Manuel Salvadores, Ian Millard, Hugh Glaser, and Nigel Shadbolt. Sparql query rewriting for implementing data integration over linked data. In *Proceedings of the 2010 EDBT/ICDT Workshops*, pages 4 :1–4 :11, Lausanne, Switzerland, March 2010. ACM. ISBN 978-1-60558-990-9. doi : 10.1145/1754239.1754244. URL <http://doi.acm.org/10.1145/1754239.1754244>. *Cité page 66*
- Danica Damljanovic, Milan Agatonovic, Hamish Cunningham, and Kalina Bontcheva. Improving habitability of natural language interfaces for querying ontologies with feedback and clarification dialogues. *Web Semantics : Science, Services and Agents on the World Wide Web*, 19(2), 2013. *Cité page 51*
- Souripriya Das, Seema Sundara, and Richard Cyganiak. R2rml : RDB to RDF Mapping Language. W3c recommandation, W3C, 2012. *Cité page 19*
- Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn. The Alignment API 4.0. *Semantic Web*, 2(1) :3–10, 2011. *Cité page 66*

- Jeremy Debattista, Sören Auer, and Christoph Lange. Luzzu—a framework for linked data quality assessment. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 124–131. IEEE, 2016. *Cité page 34*
- Jeremy Debattista, Christoph Lange, Sören Auer, and Dominic Cortis. Evaluating the quality of the lod cloud : An empirical investigation. *Semantic Web*, (Preprint) :1–43, 2018. *Cité page 34*
- S.M. Dekleva. Is natural language querying practical? *ACM SIGMIS Database*, 25(2) :24–36, 1994. *Cité page 50*
- Maria del Carmen Suarez de Figueroa Baonza. *NeOn methodology for building ontology networks : specification, sheduling and reuse*. PhD thesis, 2010. *Cité page 16*
- Rémy Delanaux, Angela Bonifati, Marie-Christine Rousset, and Romuald Thion. Query-based linked data anonymization. In *International Semantic Web Conference*, pages 530–546. Springer, 2018. *Cité page 102*
- Pratikumar Desai, Amit Sheth, and Pramod Anantharam. Semantic Gateway as a Service architecture for IoT Interoperability. In *Kno.e.sis Publications*, 2015. *3 citations pages 12, 73, et 75*
- S.E. Desert. Westlaw is natural v. boolean searching : A performance study. *Law Libr. J.*, 85 :713, 1993. *Cité page 50*
- Sounak Dey, Abhijan Bhattacharyya, and Arijit Mukherjee. Semantic data exchange between collaborative robots in fog environment : Can CoAP be a choice? *GIoTS 2017 - Global Internet of Things Summit, Proceedings*, 2017. doi : 10.1109/GIOTS.2017.8016232. *2 citations pages 77 et 78*
- Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. RML : A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In Christian Bizer, Tom Heath, Sören Auer, and Tim Berners-Lee, editors, *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*, volume 1184 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014. *Cité page 19*
- Sonia Djebali and Thomas Raimbault. SimplePARQL : a new approach using keywords over SPARQL to query the web of data. In *Proceedings of the 11th International Conference on Semantic Systems - SEMANTICS '15*, pages 188–191, Vienna, Austria, September 2015. ACM Press. ISBN 978-1-4503-3462-4. doi : 10.1145/2814864.2814890. URL <http://dl.acm.org/citation.cfm?doid=2814864.2814890>. *Cité page 67*
- Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmman, Shaohua Sun, and Wei Zhang. Knowledge vault : A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014a. *Cité page 20*
- Xin Luna Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Kevin Murphy, Shaohua Sun, and Wei Zhang. From data fusion to knowledge fusion. *Proceedings of the VLDB Endowment*, 7(10) : 881–892, 2014b. *Cité page 20*
- Doug Downey, Susan Dumais, Dan Liebling, and Eric Horvitz. Understanding the relationship between searchers’ queries and information goals. In *Proceedings of the 17th ACM conference on Information and knowledge management*, page 449–458, 2008. URL <http://dl.acm.org/citation.cfm?id=1458143>. *Cité page 49*
- Gideon Dror, Yoelle Maarek, Avihai Mejer, and Idan Szpektor. From query to question in one click : suggesting synthetic questions to searchers. In *Proceedings of the 22nd international conference on World Wide Web*, page 391–402, 2013. URL <http://dl.acm.org/citation.cfm?id=2488423>. *Cité page 49*
- Thomas Eiter, Giovambattista Ianni, Thomas Krennwallner, and Axel Polleres. Rules and ontologies for the semantic web. In *Reasoning web*, pages 1–53. Springer, 2008. *Cité page 76*
- Shady Elbassuoni, Maya Ramanath, Ralf Schenkel, and Gerhard Weikum. Searching rdf graphs with sparql and keywords. *IEEE Data Eng. Bull.*, 33(1) :16–24, 2010. *Cité page 51*

- Khadija Elbedweihy, Stuart N. Wrigley, and Fabio Ciravegna. Evaluating semantic search query approaches with expert and casual users. In *Proceedings of the 11th International Conference on The Semantic Web - Volume Part II, ISWC'12*, pages 274–286. Springer-Verlag, 2012. *Cité page 52*
- David Eppstein and Darren Strash. Listing all maximal cliques in large sparse real-world graphs. *Experimental Algorithms*, pages 364–375, 2011. bibtex : eppstein_listing_2011. *Cité page 26*
- Ivan Ermilov, Jens Lehmann, Michael Martin, and Sören Auer. Lodstats : The data web census dataset. In *International Semantic Web Conference*, pages 38–46. Springer, 2016. *Cité page 97*
- Jérôme Euzenat, Axel Polleres, and François Scharffe. Processing Ontology Alignments with SPARQL. In *2008 International Conference on Complex, Intelligent and Software Intensive Systems*, pages 913–917. IEEE, 2008. ISBN 978-0-7695-3109-0. doi : 10.1109/CISIS.2008.126. URL <http://ieeexplore.ieee.org/document/4606787/>. *Cité page 33*
- Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching, Second edition*. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38720-3. *5 citations pages 11, 20, 23, 62, et 67*
- Peter Exner and Pierre Nugues. Entity extraction : From unstructured text to dbpedia rdf triples. WoLE'12, 2012. *Cité page 41*
- Anna Fensel, Dana Kathrin Tomic, and Andreas Koller. Contributing to appliances' energy efficiency with Internet of Things, smart data and user engagement. *Future Generation Computer Systems*, 76 :329–338, 2017. ISSN 0167739X. doi : 10.1016/j.future.2016.11.026. URL <http://dx.doi.org/10.1016/j.future.2016.11.026>. *2 citations pages 76 et 79*
- Matthias Ferdinand, Christian Zirpins, and David Trastour. Lifting XML Schema to OWL. *Web Engineering*, (3140) :354–358, 2004. ISSN 16113349. doi : 10.1007/978-3-540-27834-4_44. *Cité page 75*
- Mariano Fernández-López, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. Ontology development by reuse. In *Ontology Engineering in a Networked World*, pages 147–170. Springer, 2012. *Cité page 16*
- Mariano Fernández-López, María Poveda-Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. Why are ontologies not reused across the same domain? *Journal of Web Semantics*, 57 :100492, 2019. ISSN 1570-8268. doi : <https://doi.org/10.1016/j.websem.2018.12.010>. URL <http://www.sciencedirect.com/science/article/pii/S1570826818300726>. *4 citations pages 16, 97, 98, et 99*
- S. Ferré and A. Hermann. Semantic search : reconciling expressive querying and exploratory search. *The Semantic Web—ISWC 2011*, pages 177–192, 2011. *Cité page 51*
- S. Ferré and A. Hermann. Reconciling faceted search and query languages for the semantic web. *International Journal of Metadata, Semantics and Ontologies*, 7(1) :37–54, 2012. *Cité page 51*
- S. Ferré, A. Hermann, M. Ducassé, et al. Combining faceted search and query languages for the semantic web. In *Semantic Search over the Web (SSW)-Advanced Information Systems Engineering Workshops-CAiSE Int. Workshops*, volume 83, pages 554–563, 2011. *Cité page 51*
- Sébastien Ferré. Squall : A controlled natural language for querying and updating rdf graphs. In *Controlled Natural Language*, pages 11–25. Springer, 2012. *2 citations pages 51 et 61*
- Vassilis Foteinos, Dimitris Kelaidonis, George Poullos, Panagiotis Vlacheas, Vera Stavroulaki, and Panagiotis Demestichas. Cognitive management for the internet of things : A framework for enabling autonomous applications. *IEEE Vehicular Technology Magazine*, 8(4) :90–99, 2013. ISSN 15566072. doi : 10.1109/MVT.2013.2281657. *2 citations pages 12 et 77*
- J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Re, and I. Soboroff. Building an Entity-Centric stream filtering test collection for TREC 2012. In *Proceedings of the Text REtrieval Conference (TREC)*, 2012. URL http://trec.nist.gov/act_part/conference/notebook.papers/KBA.OVERVIEW.pdf. *Cité page 42*

- John R. Frank, Steven J. Bauer, Max Kleiman-Weiner, Daniel A. Roberts, Nilesh Tripuraneni, Ce Zhang, and Christopher Re. Evaluating stream filtering for entity profile updates for trec 2013. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA., 2013. *Cité page 42*
- Sameh Ben Fredj, Mathieu Boussard, Daniel Kofman, and Ludovic Noirie. A Scalable IoT Service Search Based on Clustering and Aggregation. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 403–410. IEEE, aug 2013. ISBN 978-0-7695-5046-6. doi : 10.1109/GreenCom-iThings-CPSCoM.2013.86. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6682100>. *2 citations pages 78 et 79*
- Aldo Gangemi. Ontology design patterns for semantic web content. In *International semantic web conference*, pages 262–276. Springer, 2005. *Cité page 16*
- Raúl García-Castro, Armin Haller, and Nandana Mihindukulasooriya. On the usage of the ssn ontology. W3c note, W3C, 2019. URL <https://w3c.github.io/sdw/ssn-usage/>. *Cité page 35*
- Daniel Gerber, Sebastian Hellmann, Lorenz Bühmann, Tommaso Soru, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Real-time rdf extraction from unstructured data streams. In *The Semantic Web-ISWC 2013*, pages 135–150. Springer, 2013. *Cité page 46*
- M. Grabisch and M. Roubens. Application of the choquet integral in multicriteria decision making. *Fuzzy Measures and Integrals-Theory and Applications*, pages 348–374, 2000. *Cité page 25*
- Dominique Guinard, Vlad Trifa, Thomas Pham, and Olivier Liechti. Towards physical mashups in the web of things. *INSS2009 - 6th International Conference on Networked Sensing Systems*, pages 196–199, 2009. ISSN 1383469X. doi : 10.1109/INSS.2009.5409925. *Cité page 13*
- A. Guzmàn-Arenas and A.-D. Cuevas. Knowledge accumulation through automatic merging of ontologies. *Expert Systems with Applications*, 37(3) :1991–2005, 2010. *Cité page 20*
- Amelie Gyrard, Martin Serrano, and Ghislain A. Atemezing. Semantic web methodologies, best practices and ontology engineering applied to Internet of Things. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 412–417. IEEE, 2015. ISBN 978-1-5090-0366-2. doi : 10.1109/WF-IoT.2015.7389090. *2 citations pages 27 et 28*
- Amelie Gyrard, Martin Serrano, Joao Bosco Jares, Soumya Kanti Datta, and Muhammad Intizar Ali. Sensor-based Linked Open Rules (S-LOR) : An Automated Rule Discovery Approach for IoT Applications and its use in Smart Cities. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1153–1159. International World Wide Web Conferences Steering Committee, 2017. ISBN 978-1-4503-4914-7. doi : 10.1145/3041021.3054716. *Cité page 82*
- Armin Haller, Krzysztof Janowicz, Simon J. D. Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. The modular SSN ontology : A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web*, 10(1) :9–32, 2019. doi : 10.3233/SW-180320. URL <https://doi.org/10.3233/SW-180320>. *Cité page 34*
- Son N. Han, Gyu Myoung Lee, and Noel Crespi. Semantic Context-Aware Service Composition for Building Automation System. *IEEE Transactions on Industrial Informatics*, 10(1) :752–761, feb 2014. doi : 10.1109/TII.2013.2252356. URL <http://ieeexplore.ieee.org/document/6478809/>. *2 citations pages 77 et 78*
- Michael Hartung, James F. Terwilliger, and Erhard Rahm. Recent advances in schema and ontology evolution. In Zohra Bellahsene, Angela Bonifati, and Erhard Rahm, editors, *Schema Matching and Mapping*, Data-Centric Systems and Applications, pages 149–190. Springer, 2011. doi : 10.1007/978-3-642-16518-4_6. *Cité page 38*
- Jing He, Yanchun Zhang, Guangyan Huang, and Jinli Cao. A smart web service based on the context of things. *ACM Transactions on Internet Technology*, 11(3) :1–23, 2012. ISSN 15335399. doi : 10.1145/2078316.2078321. URL <http://dl.acm.org/citation.cfm?doid=2078316.2078321>. *Cité page 77*

- Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating nlp using linked data. *Cité page 60*
- Cory Henson, Amit Sheth, and Krishnaprasad Thirunarayan. Semantic perception : Converting sensory observations to abstractions. *IEEE Internet Computing*, 16(2) :26–34, 2012. ISSN 10897801. doi : 10.1109/MIC.2012.20. *Cité page 76*
- Nathalie Hernandez. *Ontologies de domaine pour la modélisation du contexte en Recherche d’information. (Domain ontologies for context modeling in information retrieval)*. PhD thesis, Paul Sabatier University, Toulouse, France, 2005. URL <https://tel.archives-ouvertes.fr/tel-00371323>. *Cité page 95*
- Matthias Hert, Gerald Reif, and Harald C Gall. A comparison of RDB-to-RDF mapping languages. In Chiara Ghidini, Axel-Cyrille Ngonga Ngomo, Stefanie N. Lindstaedt, and Tassilo Pellegrini, editors, *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, ACM International Conference Proceeding Series, pages 25–32. ACM, 2011. doi : 10.1145/2063518.2063522. *Cité page 19*
- Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6) :895–920, 2017. *Cité page 52*
- Laura Hollink, Peter Mika, and Roi Blanco. Web usage mining with semantic analysis. In *Proceedings of the 22nd international conference on World Wide Web*, page 561–570, 2013. URL <http://dl.acm.org/citation.cfm?id=2488438>. *Cité page 49*
- Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosf, Mike Dean, et al. Swrl : A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21(79) : 1–31, 2004. *Cité page 76*
- Shaun Howell, Yacine Rezgui, and Thomas Beach. Integrating building and urban semantics to empower smart water solutions. *Automation in Construction*, 81 :434–448, 2017. ISSN 09265805. doi : 10.1016/j.autcon.2017.02.004. URL <http://dx.doi.org/10.1016/j.autcon.2017.02.004>. *Cité page 79*
- Dina Hussein, Son N. Han, Gyu Myoung Lee, Noel Crespi, and Emmanuel Bertin. Towards a dynamic discovery of smart services in the social internet of things. *Computers & Electrical Engineering*, 2016. ISSN 00457906. doi : 10.1016/j.compeleceng.2016.12.008. *3 citations pages 17, 28, et 78*
- Krzysztof Janowicz and Michael Compton. The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In *Proceedings of the 9th International Semantic Web Conference, 3rd International Workshop on Semantic Sensor Networks*, pages 7–11, 2010. *Cité page 31*
- B.J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval : a study of user queries on the web. In *ACM SIGIR Forum*, volume 32, pages 5–17. ACM, 1998. *Cité page 49*
- B.J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs : a study and analysis of user queries on the web. *Information processing & management*, 36(2) :207–227, 2000. *Cité page 49*
- Rim Teyeb Jaouachi, Mouna Torjmen Khemakhem, Nathalie Hernandez, Ollivier Haemmerlé, and Maher Ben Jemaa. Semantic annotation of images extracted from the web using RDF patterns and a domain ontology. In *Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS)*. *Cité page 96*
- Raja Jurdak, Cristina Videira Lopes, and Pierre Baldi. A framework for modeling sensor networks, 2004. *2 citations pages 17 et 28*
- Charbel El Kaed, Imran Khan, Andre Van Den Berg, Hicham Hossayni, and Christophe Saint-Marcel. SRE : semantic rules engine for the industrial internet-of-things gateways. *IEEE Trans. Industrial Informatics*, 14(2) :715–724, 2018a. doi : 10.1109/TII.2017.2769001. *Cité page 83*

- Charbel El Kaed, Imran Khan, Andre Van Den Berg, Hicham Hossayni, and Christophe Saint-Marcel. SRE : Semantic Rules Engine For the Industrial Internet- Of-Things Gateways. *IEEE Transactions on Industrial Informatics*, 14(2) :715–724, 2018b. ISSN 15513203. doi : 10.1109/TII.2017.2769001. *2 citations pages 77 et 83*
- Megan Katsumi and Michael Grüninger. What is ontology reuse? In *FOIS*, pages 9–22, 2016. *Cité page 16*
- E. Kaufmann and A. Bernstein. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Web Semantics : Science, Services and Agents on the World Wide Web*, 8(4) :377–393, 2010. *2 citations pages 50 et 51*
- E. Kaufmann, A. Bernstein, and R. Zumstein. Querix : A natural language interface to query ontologies based on clarification dialogs. In *5th International Semantic Web Conference ISWC 2006*, number November, pages 5–6. Citeseer, 2006. *Cité page 50*
- E. Kaufmann, A. Bernstein, and L. Fischer. Nlp-reduce : A” naïve” but domain-independent natural language interface for querying ontologies. In *4th European Semantic Web Conference ESWC 2007*, pages 1–2, 2007. *Cité page 50*
- Daniel Kayser. La représentation des connaissances. 1997. URL <http://hal.archives-ouvertes.fr/hal-00091603/>. *Cité page 7*
- Dimitrios Kelaidonis, Angelos Rouskas, Vera Stavroulaki, Panagiotis Demestichas, and Panagiotis Vlachas. A federated Edge Cloud-IoT architecture. *EUCNC 2016 - European Conference on Networks and Communications*, pages 230–234, 2016. doi : 10.1109/EuCNC.2016.7561038. *Cité page 77*
- Imran Khan, Rifat Jafrin, Fatima Zahra Errounda, Roch Glitho, Noël Crespi, Monique Morrow, and Paul Polakos. A Data Annotation Architecture for Semantic Applications in Virtualized Wireless Sensor Networks. 2015. URL <https://arxiv.org/ftp/arxiv/papers/1501/1501.07139.pdf>. *3 citations pages 75, 76, et 77*
- Ankesh Khandelwal, Ian Jacobi, and Lalana Kagal. Linked rules : Principles for rule reuse on the web. *Lecture Notes in Computer Science*, 6902 LNCS :108–123, 2011. ISSN 03029743. doi : 10.1007/978-3-642-23580-1_9. *Cité page 82*
- Evgeny Kharlamov, Yannis Kotidis, Theofilos Mailis, Christian Neuenstadt, Charalampos Nikolaou, Özgür Özcep, Christoforos Svingos, Dmitriy Zheleznyakov, Steffen Lamparter, Ian Horrocks, Yannis Ioannidis, and Ralf Möller. Towards Analytics Aware Ontology Based Access to Static and Streaming Data (Extended Version). In *ISWC*, pages 1–22, 2016. ISBN 9783319465470. doi : 10.1007/978-3-319-46547-0. URL <http://arxiv.org/abs/1607.05351>. *Cité page 76*
- Muhammad Golam Kibria. Knowledge based open IoT service provisioning through cooperation between physical web and WoO. In *2015 Seventh International Conference on Ubiquitous and Future Networks*, pages 395–400. IEEE, 2015. ISBN 978-1-4799-8993-5. doi : 10.1109/ICUFN.2015.7182573. *4 citations pages 17, 28, 74, et 77*
- Michael Kifer and Harold Boley. Rif overview. *W3C working draft, W3C,(October 2009)*. <http://www.w3.org/TR/rif-overview>, 2013. *Cité page 76*
- Jussi Kiljander, Alfredo D’elia, Francesco Morandi, Pasi Hyttinen, Janne Takalo-Mattila, Arto Ylisaukko-Oja, Juha-Pekka Soinen, and Tullio Salmon Cinotti. Semantic Interoperability Architecture for Pervasive Computing and Internet of Things. *IEEE Access*, 2 :856–873, 2014. ISSN 2169-3536. doi : 10.1109/ACCESS.2014.2347992. *Cité page 77*
- Sabrina Kirrane, Serena Villata, and Mathieu d’Aquin. Privacy, security and policies : A review of problems and solutions with semantic web technologies. *Semantic Web*, 9(2) :153–161, 2018. *Cité page 102*
- Michel C. A. Klein and Dieter Fensel. Ontology versioning on the semantic web. In Isabel F. Cruz, Stefan Decker, Jérôme Euzenat, and Deborah L. McGuinness, editors, *Proceedings of SWWS’01, The first Semantic Web Working Symposium, Stanford University, California, USA, July 30 - August 1, 2001*, pages 75–91, 2001. *Cité page 37*

- Michel C. A. Klein, Dieter Fensel, Atanas Kiryakov, and Damyan Ognyanov. Ontology versioning and change detection on the web. In Asunción Gómez-Pérez and V. Richard Benjamins, editors, *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Sigüenza, Spain, October 1-4, 2002, Proceedings*, volume 2473 of *Lecture Notes in Computer Science*, pages 197–212. Springer, 2002. doi : 10.1007/3-540-45810-7_20. *Cité page 37*
- Jose Emilio Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, and Dimitris Kontokostas. *Validating RDF Data*, volume 7 of *Synthesis Lectures on the Semantic Web : Theory and Technology*. Morgan & Claypool Publishers LLC, sep 2017. doi : 10.2200/s00786ed1v01y201707wbe016. URL <https://doi.org/10.2200/s00786ed1v01y201707wbe016>. *Cité page 96*
- Mathilde Lannes, Fabien Amarger, **Nicolas Seyoux**, and Nathalie Hernandez. Du langage naturel à la connaissance il n'y a qu'un pas : SWIP. In *IC*, pages 229–232, Caen, France, 2017. URL <https://hal.archives-ouvertes.fr/hal-01570295>. *2 citations pages 70 et 101*
- Danh Le-Phuoc, Hoan Quoc Nguyen-Mau, Josiane Xavier Parreira, and Manfred Hauswirth. A middleware framework for scalable management of linked streams. *Web Semantics : Science, Services and Agents on the World Wide Web*, 16 :42–51, 2012. ISSN 15708268. doi : 10.1016/j.websem.2012.06.003. *Cité page 78*
- Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Hung Ngo Quoc, Tuan Tran Nhat, and Manfred Hauswirth. The Graph of Things : A step towards the Live Knowledge Graph of connected things. *Web Semantics : Science, Services and Agents on the World Wide Web*, 2016. ISSN 15708268. doi : 10.1016/j.websem.2016.02.003. *5 citations pages 73, 74, 76, 78, et 101*
- Yann Hang Lee and Shankar Nair. A Smart Gateway Framework for IOT Services. *Proceedings - 2016 IEEE International Conference on Internet of Things ; IEEE Green Computing and Communications ; IEEE Cyber, Physical, and Social Computing ; IEEE Smart Data, iThings-GreenCom-CPSCoM-Smart Data 2016*, pages 107–114, 2016. doi : 10.1109/iThings-GreenCom-CPSCoM-SmartData.2016.44. *2 citations pages 77 et 78*
- Laurent Lefort and Cory Henson. Semantic sensor network xg final report. Technical Report June 2011, 2011. *Cité page 34*
- Maxime Lefrançois, Antoine Zimmermann, and Noorani Bakerally. A sparql extension for generating rdf from heterogeneous formats. In *European Semantic Web Conference*, pages 35–50. Springer, 2017. *Cité page 19*
- J. Lehmann and L. Bühmann. Autosparql : let users query your knowledge base. In *Proceedings of the 8th extended semantic web conference on The semantic web : research and applications-Volume Part I*, pages 63–79. Springer-Verlag, 2011. *Cité page 52*
- Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014. *Cité page 42*
- Yuanguai Lei, Victoria S. Uren, and Enrico Motta. Semsearch : A search engine for the semantic web. In *EKAW*, pages 238–245, 2006. *Cité page 52*
- Juan Li, Nazia Zaman, and Honghui Li. A Decentralized Locality-Preserving Context-Aware Service Discovery Framework for Internet of Things. In *2015 IEEE International Conference on Services Computing*, pages 317–323. IEEE, jun 2015. ISBN 978-1-4673-7281-7. doi : 10.1109/SCC.2015.51. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7207369>. *2 citations pages 17 et 28*
- Jiaqiang Liu, Yong Li, Min Chen, Wenxia Dong, and Depeng Jin. Software-defined internet of things for smart urban sensing. *IEEE Communications Magazine*, 53(9) :55–63, 2015. ISSN 0163-6804. doi : 10.1109/MCOM.2015.7263373. *Cité page 73*
- Qian Liu, Yue Liu, Dayong Wu, and Cheng Xueqi. Ictnet at temporal summarization track trec 2013. In *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA., 2013. *Cité page 42*

- Steffen Lohmann, Stefan Negru, Florian Haag, and Thomas Ertl. Visualizing ontologies with VOWL. *Semantic Web*, 7(4) :399–419, 2016. *Cité page 63*
- Rob Lokers, Stasinios Konstantopoulos, Armando Stellato, Rob Knapen, and Sander Janssen. Designing innovative linked open data and semantic technologies in agro-environmental modelling. In *7th International Congress on Environmental Modelling and Software (iEMSs)*, pages 392–398, San Diego, California, USA, June 2014. URL <https://scholarsarchive.byu.edu/iemssconference/2014/Stream-A/45/>. *Cité page 67*
- Vanessa Lopez, Victoria Uren, Marta Sabou, and Enrico Motta. Is question answering fit for the semantic web? : a survey. *Semantic Web*, 2(2) :125–155, 2011. *Cité page 52*
- Giuseppe Loseto, Saverio Ieva, Filippo Gramegna, Michele Ruta, Floriano Scioscia, Eugenio Di Sciascio, and Bari I. Linked Data (in low-resource) Platforms : a mapping for Constrained Application Protocol. In *ISWC*, Kobe, 2016. *Cité page 79*
- Altti Ilari Maarala, Xiang Su, and Jukka Rieki. Semantic Reasoning for Context-aware Internet of Things Applications. *IEEE Internet of Things Journal*, 2017. *5 citations pages 76, 80, 82, 83, et 89*
- Konstantinos Makris, Nektarios Gioldasis, Nikos Bikakis, and Stavros Christodoulakis. Ontology Mapping and SPARQL Rewriting for Querying Federated RDF Data Sources. In Robert Meersman, Tharam Dillon, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems, OTM 2010*, volume 6427 of *LNCS*, pages 1108–1117, Hersonssisos, Crete, Greece, October 2010. Springer. ISBN 978-3-642-16948-9 978-3-642-16949-6. doi : 10.1007/978-3-642-16949-6_32. URL http://link.springer.com/10.1007/978-3-642-16949-6_32. *Cité page 66*
- Konstantinos Makris, Nikos Bikakis, Nektarios Gioldasis, and Stavros Christodoulakis. SPARQLRW : transparent query access over mapped RDF data sources. In *Proceedings of the 15th International Conference on Extending Database Technology - EDBT'12*, pages 610–613, Berlin, Germany, March 2012. ACM Press. doi : 10.1145/2247596.2247678. URL <http://dl.acm.org/citation.cfm?doid=2247596.2247678>. *Cité page 66*
- Sujith Samuel Mathew, Yacine Atif, Quan Z. Sheng, and Zakaria Maamar. Building sustainable parking lots with the Web of Things. *Personal and Ubiquitous Computing*, 18(4) :895–907, 2014. ISSN 16174909. doi : 10.1007/s00779-013-0694-7. *Cité page 75*
- Simon Mayer and Gianin Basler. Semantic metadata to support device interaction in smart environments. *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication - UbiComp '13 Adjunct*, pages 1505–1514, 2013. doi : 10.1145/2494091.2497584. URL <http://dl.acm.org/citation.cfm?doid=2494091.2497584>. *Cité page 78*
- Simon Mayer, Andreas Tschofen, Anind K Dey, and Friedemann Mattern. User interfaces for smart things—a generative approach with semantic interaction descriptions. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(2) :12, 2014. *Cité page 101*
- John P McCrae, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. The ontolex-lemon model : development and applications. In *Proceedings of eLex 2017 conference*, pages 19–21, 2017. *Cité page 10*
- Richard McCreddie, Craig Macdonald, and Iadh Ounis. Incremental update summarization : Adaptive sentence selection based on prevalence and novelty. In *Proceedings of the 23rd International Conference on Conference on Information and Knowledge Management*, pages 301–310, New York, USA, 2014. ISBN 978-1-4503-2598-1. doi : 10.1145/2661829.2661951. URL <http://doi.acm.org/10.1145/2661829.2661951>. *Cité page 42*
- Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *National Institute of Standards and Technology, Information Technology Laboratory*, 145 :7, 2011. ISSN 1472-0213. doi : 10.1136/emj.2010.096966. *Cité page 13*

- Franck Michel, Loïc Djimenou, Catherine Faron-Zucker, and Johan Montagnat. Translation of relational and non-relational databases into RDF with xr2rml. In Valérie Monfort, Karl-Heinz Krempels, Tim A. Majchrzak, and Ziga Turk, editors, *WEBIST 2015 - Proceedings of the 11th International Conference on Web Information Systems and Technologies, Lisbon, Portugal, 20-22 May, 2015*, pages 443–454. SciTePress, 2015. doi : 10.5220/0005448304430454. *Cité page 19*
- Franck Michel, Olivier Gargominy, Sandrine Tercerie, and Catherine Faron-Zucker. A Model to Represent Nomenclatural and Taxonomic Information as Linked Data. Application to the French Taxonomic Register, TAXREF. In Alsayed Algergawy, Naouel Karam, Friederike Klan, and Clément Jonquet, editors, *Proceedings of the 2nd International Workshop on Semantics for Biodiversity (S4BioDiv 2017) co-located with 16th International Semantic Web Conference (ISWC 2017)*, volume 1933, Vienna, Austria, October 2017. CEUR-WS.org. URL <http://ceur-ws.org/Vol-1933/paper-3.pdf>. *Cité page 64*
- Peter Mika. On schema. org and why it matters for the web. *IEEE Internet Computing*, 19(4) :52–55, 2015. *Cité page 98*
- Thomas Minier, Hala Skaf-Molli, Pascal Molli, and Maria-Esther Vidal. Intelligent clients for replicated triple pattern fragments. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 400–414, 2018. *Cité page 101*
- Boris Motik and Riccardo Rosati. Reconciling description logics and rules. *Journal of the ACM (JACM)*, 57(5) :30, 2010. *Cité page 76*
- M Mrissa, L Medini, J.-P. Jamont, N Le Sommer, and J Laplace. An Avatar Architecture for the Web of Things. *Internet Computing, IEEE*, 19(2) :30–38, 2015. ISSN 1089-7801. doi : 10.1109/MIC.2015.19. *Cité page 76*
- Paul Murdock and al. Semantic Interoperability for the Web of Things, 2016. *Cité page 28*
- Paul & al. Murdock. White paper : Semantic interoperability for the web of things, 2016. URL <http://dx.doi.org/10.13140/RG.2.2.25758.13122>. *Cité page 72*
- Lina Nachabe, Marc Girod-Genet, and Bachar El Hassan. Unified Data Model for Wireless Sensor Network. *IEEE Sensors Journal*, 15(7) :3657–3667, jul 2015. ISSN 1530-437X. doi : 10.1109/JSEN.2015.2393951. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7014284>. *2 citations pages 17 et 28*
- Siniša Nikoli, Valentin Penca, and Zora Konjovi. Semantic Web Based Architecture for Managing Hardware Heterogeneity in Wireless Sensor Network. In *International Journal of Computer Science and Applications*, volume 8, pages 38–58, 2011. ISBN 9781450301480. *2 citations pages 73 et 78*
- Michele Nitti, Virginia Pilloni, Giuseppe Colistra, and Luigi Atzori. The Virtual Object as a Major Element of the Internet of Things : A Survey. *IEEE Communications Surveys and Tutorials*, 18(2) : 1228–1240, 2016. ISSN 1553877X. doi : 10.1109/COMST.2015.2498304. *2 citations pages 73 et 76*
- J. Nivre, J. Hall, and J. Nilsson. Maltparser : A data-driven parser-generator for dependency parsing. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006), May 24-26, 2006, Genoa, Italy*, pages 2216–2219. European Language Resource Association, Paris, 2006. *Cité page 59*
- Natalya Fridman Noy and Mark A. Musen. PROMPTDIFF : A fixed-point algorithm for comparing ontology versions. In Rina Dechter, Michael J. Kearns, and Richard S. Sutton, editors, *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*, pages 744–750. AAAI Press / The MIT Press, 2002. *Cité page 37*
- Kévin Ottens, Nathalie Hernandez, Marie Pierre Gleizes, and Nathalie Aussenac-Gilles. A multi-agent system for dynamic ontologies. *Journal of Logic and Computation*, 19(5) :831–858, 2009. doi : 10.1093/logcom/exn050. URL <https://doi.org/10.1093/logcom/exn050>. *2 citations pages 4 et 46*

- Vicky Papavassiliou, Giorgos Flouris, Irini Fundulaki, Dimitris Kotzinos, and Vassilis Christophides. On detecting high-level changes in RDF/S kbs. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings*, volume 5823 of *Lecture Notes in Computer Science*, pages 473–488. Springer, 2009. doi : 10.1007/978-3-642-04930-9_30. 2 citations pages 37 et 38
- Pankesh Patel, Muhammad Intizar Ali, and Amit Sheth. On Using the Intelligent Edge for IoT Analytics. *IEEE Intelligent Systems*, 32(5) :64–69, 2017. ISSN 1541-1672. doi : 10.1109/MIS.2017.3711653. 2 citations pages 14 et 83
- Heiko Paulheim. Knowledge graph refinement : A survey of approaches and evaluation methods. *Semantic web*, 8(3) :489–508, 2017. Cité page 11
- Sarogini Grace Pease, Paul P. Conway, and Andrew A. West. Hybrid ToF and RSSI real-time semantic tracking with an adaptive industrial internet of things architecture. *Journal of Network and Computer Applications*, 99(August 2016) :98–109, 2017. ISSN 10958592. doi : 10.1016/j.jnca.2017.10.010. URL <http://dx.doi.org/10.1016/j.jnca.2017.10.010>. 4 citations pages 17, 28, 76, et 79
- Charith Perera and Athanasios V. Vasilakos. A knowledge-based resource discovery for Internet of Things. *Knowledge-Based Systems*, 109, 2016. ISSN 09507051. doi : 10.1016/j.knosys.2016.06.030. Cité page 74
- Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things : A survey. *IEEE Communications Surveys and Tutorials*, 16(1) :414–454, 2014a. ISSN 1553877X. doi : 10.1109/SURV.2013.042313.00197. Cité page 73
- Charith Perera, Arkady Zaslavsky, Chi Harold Liu, Michael Compton, Peter Christen, and Dimitrios Georgakopoulos. Sensor search techniques for sensing as a service architecture for the internet of things. *IEEE Sensors Journal*, 14(2) :406–420, 2014b. ISSN 1530437X. doi : 10.1109/JSEN.2013.2282292. 2 citations pages 78 et 101
- Jorge Perez, Marcelo Arenas, and Claudio Gutierrez. Semantics of SPARQL. Technical report, Technical Report TR/DCC-2006-17, Universidad de Chile, 2006. URL http://www.dcc.uchile.cl/cgutierrez/ftp/sparql_semantics.pdf. Cité page 47
- Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. nsparql : A navigational language for rdf. *Web Semantics : Science, Services and Agents on the World Wide Web*, 8(4) :255–270, 2010. Cité page 51
- Georgios Petasis, Vangelis Karkaletsis, Georgios Paliouras, Anastasia Krithara, and Elias Zavitsanos. Ontology population and enrichment : State of the art. In *Knowledge-driven multimedia information extraction and ontology evolution*, pages 134–166. Springer-Verlag, 2011. Cité page 41
- Dennis Pfisterer, Kay Romer, Daniel Bimschas, Oliver Kleine, Richard Mietz, Cuong Truong, Henning Hasemann, Alexander Kröller, Max Pagel, Manfred Hauswirth, Marcel Karnstedt, Myriam Leggieri, Alexandre Passant, and Ray Richardson. SPITFIRE : toward a semantic web of things. *IEEE Communications Magazine*, 49(11) :40–48, 2011. doi : 10.1109/MCOM.2011.6069708. 3 citations pages 13, 72, et 77
- Joern Ploennigs, Amadou Ba, and Michael Barry. Materializing the Promises of Cognitive IoT : How Cognitive Buildings are Shaping the Way. *IEEE Internet of Things Journal*, 4662(c) :1–8, 2017. ISSN 23274662. doi : 10.1109/JIOT.2017.2755376. 2 citations pages 76 et 77
- Stefan Poslad, Stuart E. Middleton, Fernando Chaves, Ran Tao, Ocal Necmioglu, and Ulrich Bugel. A Semantic IoT Early Warning System for Natural Environment Crisis Management. *IEEE Transactions on Emerging Topics in Computing*, 3(2) :246–257, 2015. ISSN 2168-6750. doi : 10.1109/TETC.2015.2432742. 3 citations pages 74, 76, et 78
- Rachel A Pottinger and Philip A Bernstein. Merging models based on given correspondences. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 862–873. VLDB Endowment, 2003. Cité page 20

- C. Pradel. Allowing end users to query graph-based knowledge bases. In *Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, pages 8–15, Galway, Ireland, October 2012. *Cité page 49*
- C. Pradel, O. Haemmerlé, and N. Hernandez. Expressing conceptual graph queries from patterns : how to take into account the relations. In *Proceedings of the 19th International Conference on Conceptual Structures, ICCS'11, Lecture Notes in Artificial Intelligence # 6828*, pages 234–247, Derby, GB, July 2011a. Springer. *2 citations pages 48 et 49*
- Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez. Expression de requêtes SPARQL à partir de patrons : prise en compte des relations. In *IC 2011 : 22es journées Ingénierie des Connaissances (Proceedings of the 22nd French Knowledge Engineering Conference), Chambéry, France, May 16-20, 2011.*, pages 771–787, 2011b. *Cité page 55*
- Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez. A semantic web interface using patterns : The swip system (regular paper). In Madalina Croitoru, Sebastian Rudolph, Nic Wilson, John Howse, and Olivier Corby, editors, *IJCAI-GKR Workshop, Barcelona, Spain, 16/07/2011-16/07/2011*, number 7205 in LNAI, pages 172–187, <http://www.springerlink.com>, mai 2012. Springer. *2 citations pages 48 et 53*
- Camille Pradel, Guillaume Peyet, Ollivier Haemmerlé, and Nathalie Hernandez. Swip at qald-3 : results, criticisms and lesson learned (working notes). In *CLEF 2013, Valencia, Spain, 23/09/2013-26/09/2013*, 2013. *4 citations pages 48, 58, 59, et 60*
- Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez. Swip : A natural language to SPARQL interface implemented with SPARQL. In *Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures (ICCS)*, pages 260–274, 2014. *3 citations pages 5, 9, et 60*
- Eric Prud'hommeaux, Jose Emilio Labra Gayo, and Harold Solbrig. Shape expressions : an rdf validation and transformation language. In *Proceedings of the 10th International Conference on Semantic Systems*, pages 32–40. ACM, 2014. *Cité page 96*
- Y Raimond, SA Abdallah, MB Sandler, and F Giasson. The music ontology. 2007. *Cité page 60*
- S. Raunich and E. Rahm. Target-driven merging of taxonomies with Atom. *Information Systems*, 42 : 1–14, 2014. *Cité page 20*
- Lukas Reinfurt, Uwe Breitenbücher, Michael Falkenthal, Frank Leymann, and Andreas Riegg. Internet of things patterns. In *Proceedings of the 21st European Conference on Pattern Languages of Programs - EuroPlop '16*, pages 1–21, New York, New York, USA, 2016. ACM Press. ISBN 9781450340748. doi : 10.1145/3011784.3011789. URL <http://dl.acm.org/citation.cfm?doid=3011784.3011789>. *Cité page 73*
- Axel Reymonet, Jérôme Thomas, and Nathalie Aussenac-Gilles. Modélisation de ressources termino-ontologiques en owl. In *Journées Francophones d'Ingénierie des Connaissances (IC 2007)*, pages 169–180. Cépaduès Editions, 2007. *3 citations pages iv, 9, et 10*
- Axel Reymonet, Jérôme Thomas, and Nathalie Aussenac-Gilles. Ontologies et recherche d'information : une application au diagnostic automobile. In *IC 2010 : 21es Journées Ingénierie des Connaissances 2010 (Proceedings of the 21st French Knowledge Engineering Conference), Nîmes, France, June 8-10, 2010.*, pages 283–294, 2010. *Cité page 38*
- Janessa Rivera and Rob van der Meulen. Gartner says the internet of things installed base will grow to 26 billion units by 2020. Technical report, Gartner cabinet, 2013. *Cité page 12*
- Ismael Bouassida Rodriguez, Jérôme Lacouture, and Khalil Drira. Semantic Driven Self-Adaptation of Communications Applied to ERCMS. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pages 1292–1299. IEEE, 2010. ISBN 978-1-4244-6695-5. doi : 10.1109/AINA.2010.158. *Cité page 83*
- Catherine Roussey, Jean-Pierre Chanet, Vincent Cellier, and Fabien Amarger. Agronomic taxon. In *Proceedings of second international Workshop on Open Data (WOD 2013)*. BNF Paris, 2010. *Cité page 18*

- Jennifer Rowley. The wisdom hierarchy : representations of the DIKW hierarchy. *Journal of Information Science*, 33(2) :163–180, 2007. ISSN 0165-5515. doi : 10.1177/0165551506070706. URL <http://alturl.com/7qike>. *Cité page 7*
- Alistair Russell and Paul R. Smart. Nitelight : A graphical editor for sparql queries. In *International Semantic Web Conference (Posters & Demos)*, 2008. *Cité page 51*
- David J. Russomanno, Cartik Kothari, and Omoju Thomas. Sensor ontologies : From shallow to deep models. In *Annual Southeastern Symposium on System Theory*, volume 37, pages 107–112, 2005. ISBN 0-7803-8808-9. doi : 10.1109/SSST.2005.1460887. *2 citations pages 17 et 28*
- Michele Ruta, Saverio Ieva, Giuseppe Loseto, and Eugenio Di Sciascio. From the Physical Web to the Physical Semantic Web : knowledge discovery in the Internet of Things. In *enth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2016. *Cité page 78*
- Luis Sánchez, Jorge Lanza, Juan Ramón Santana, Rachit Agarwal, Pierre-Guillaume Raverdy, Tarek Elsaleh, Yasmin Fathy, SeungMyeong Jeong, Aris Dadoukis, Thanasis Korakis, Stratos Keranidis, Philip O’Brien, Jerry Horgan, Antonio Sacchetti, Giuseppe Mastandrea, Alexandros G. Fragkiadakis, Pavlos Charalampidis, Nicolas Seydoux, Christelle Ecrepont, and Mengxuan Zhao. Federation of internet of things testbeds for the realization of a semantically-enabled multi-domain data marketplace. *Sensors*, 18(10) :3375, 2018. doi : 10.3390/s18103375. URL <https://doi.org/10.3390/s18103375>. *Cité page 33*
- Chayan Sarkar, Akshay Uttama Akshay, R. Venkatesha Prasad, Abdur Rahim, Ricardo Neisse, and Gianmarco Baldini. DIAT : A scalable distributed architecture for IoT. In *IEEE Internet of Things Journal*, volume 2, 2015. ISBN 2327-4662 VO - PP. doi : 10.1109/JIOT.2014.2387155. *2 citations pages 76 et 77*
- François Scharffe, Jérôme Euzenat, and Dieter Fensel. Towards design patterns for ontology alignment. In *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, page 2321, New York, New York, USA, mar 2008. ACM Press. ISBN 9781595937537. doi : 10.1145/1363686.1364236. *Cité page 16*
- Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the linked data best practices in different topical domains. In *International Semantic Web Conference*, pages 245–260. Springer, 2014. *Cité page 97*
- H. Schmid. Probabilistic part-of-speech tagging using decision trees. 1994. *Cité page 59*
- Floriano Scioscia and Michele Ruta. Building a Semantic Web of Things : Issues and perspectives in information compression. *ICSC 2009 - 2009 IEEE International Conference on Semantic Computing*, pages 589–594, 2009. doi : 10.1109/ICSC.2009.75. *2 citations pages 13 et 72*
- Nicolas Seydoux. *Towards interoperable IOT systems with a constraint-aware semantic web of things. (Vers des systèmes IOT interopérables avec un réseau d’objets sémantique tenant compte des contraintes)*. PhD thesis, INSA Toulouse, France, 2018. URL <https://tel.archives-ouvertes.fr/tel-02093952>. *3 citations pages 17, 33, et 72*
- Nicolas Seydoux, Mahdi Ben Alaya, Nathalie Hernandez, Thierry Monteil, and Ollivier Haemmerlé. Sémantique et Internet des objets : d’un état de l’art à une ontologie modulaire. In *Journées Francophones d’Ingénierie des Connaissances (IC)*, pages 105–116. HAL-INRIA, juillet 2015. *Cité page 17*
- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. IoT-O, a core-domain IoT ontology to represent connected devices networks. In *International Conference on Knowledge Engineering and Knowledge Management (EKAW 2016)*, pages 561–576. Springer, 2016a. URL <http://oatao.univ-toulouse.fr/18880/>. *5 citations pages 9, 11, 17, 33, et 73*
- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. IoT-O, a core-domain IoT ontology to represent connected devices networks. In *EKAW*, 2016b. *3 citations pages 4, 34, et 101*

- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Lowering knowledge : Making constrained devices semantically interoperable. In *International Semantic Web Conference (ISWC 2016)*, ISWC 2016 Posters & Demonstrations Track. CEUR-WS, 2016c. Poster. *Cité page 75*
- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Capturing the contributions of the semantic web to the IoT : a unifying vision (extended abstract). *Semantic Web technologies for the Internet of Things*, 2017a. *Cité page 72*
- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Capturing the contributions of the semantic web to the iot : A unifying vision. In *Proceedings of the Second Workshop on Semantic Web Technologies for the Internet of Things co-located with 16th International Semantic Web Conference (ISWC 2017)*, 2017b. *Cité page 5*
- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Towards Cooperative Semantic Computing : a Distributed Reasoning approach for Fog-enabled SWoT. In *International Conference on cooperative information systems*, 2018a. *Cité page 89*
- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Towards Cooperative Semantic Computing : a Distributed Reasoning approach for Fog-enabled SWoT. In *COOPIS*, 2018b. *Cité page 72*
- Nicolas Seydoux, Drira Khalil, Nathalie Hernandez, and Thierry Monteil. A Distributed Scalable Approach for Rule Processing : Computing in the Fog for the SWoT. In *Web intelligence*, Santiago, Chili, 2018c. *Cité page 72*
- Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Towards cooperative semantic computing : a distributed reasoning approach for fog-enabled swot. In *International Conference on Cooperative Information Systems (CoopIS 2018)*, pages 407–425. Springer-Verlag, 2019a. URL <https://www.springerprofessional.de/en/towards-cooperative-semantic-computing-a-distributed-reasoning-a/16207242>. *2 citations pages 5 et 72*
- Nicolas Seydoux, Khalil Drira¹, Nathalie Hernandez, and Thierry Monteil. EDR : A generic approach for the distribution of rule-based reasoning in a cloud-fog continuum. *Semantic Web journal*, to appear, 2019b. *Cité page 93*
- Omer Berat Sezer, Erdogan Dogdu, and Ahmet Murat Ozbayoglu. Context Aware Computing, Learning and Big Data in Internet of Things : A Survey. *IEEE Internet of Things Journal*, 5(1) :1–1, 2018. ISSN 2327-4662. doi : 10.1109/JIOT.2017.2773600. *2 citations pages 76 et 82*
- Amit Sheth, Cory Henson, and Satya S Sahoo. Semantic Sensor Web. In *IEEE Internet Computing*, volume 12, pages 78–83, 2008. ISBN 1089-7801 VO - 12. doi : 10.1109/MIC.2008.87. *2 citations pages 75 et 82*
- C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. In *ACM SIGIR Forum*, volume 33, pages 6–12. ACM, 1999. *Cité page 49*
- D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications : The AGROVOC example. *Journal of Digital Information*, 4(4), 2004. *Cité page 19*
- John F Sowa. Conceptual structures : information processing in mind and machine. 1983. *Cité page 48*
- Vassilis Spiliopoulos, Alexandros G Valarakos, and George A Vouros. CSR : discovering subsumption relations for the alignment of ontologies. In *European Semantic Web Conference*, pages 418–431. Springer, 2008. *Cité page 34*
- A. Spink, D. Wolfram, M.B.J. Jansen, and T. Saracevic. Searching the web : The public and their queries. *Journal of the American society for information science and technology*, 52(3) :226–234, 2001. *Cité page 49*
- Vlad Stirbu. Towards a RESTful plug and play experience in the Web of Things. *Proceedings - IEEE International Conference on Semantic Computing 2008, ICSC 2008*, pages 512–517, 2008. ISSN 00392499. doi : 10.1109/ICSC.2008.51. *Cité page 13*

- Ljiljana Stojanovic. Methods and tools for ontology evolution. 2004. *Cité page 40*
- Ljiljana Stojanovic, Alexander Maedche, Boris Motik, and Nenad Stojanovic. User-driven ontology evolution management. In Asunción Gómez-Pérez and V. Richard Benjamins, editors, *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Sigüenza, Spain, October 1-4, 2002, Proceedings*, volume 2473 of *Lecture Notes in Computer Science*, pages 285–300. Springer, 2002. doi : 10.1007/3-540-45810-7_27. *2 citations pages 37 et 38*
- Heiner Stuckenschmidt and Michel C. A. Klein. Integrity and change in modular ontologies. In Georg Gottlob and Toby Walsh, editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 900–908. Morgan Kaufmann, 2003. *Cité page 38*
- Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra. *Modular ontologies : concepts, theories and techniques for knowledge modularization*, volume 5445. Springer, 2009. *Cité page 16*
- Xiang Su, Jukka Riekk, Jukka K. Nurminen, Johanna Nieminen, and Markus Koskimies. Adding semantics to internet of things. *Concurrency and Computation : Practice and Experience*, 27(8) : 1844–1860, 2015. ISSN 15320634. doi : 10.1002/cpe.3203. *Cité page 79*
- Xiang Su, Pingjiang Li, Jukka Riekk, Xiaoli Liu, Jussi Kiljander, Juha-Pekka Soinen, Christian Prehofer, Huber Flores, and Yuhong Li. Distribution of Semantic Reasoning on the Edge of Internet of Things. In *IEEE UbiComp*, number November, page 79, 2018. ISBN 9781450348140. *4 citations pages 73, 76, 80, et 83*
- Maria Del Carmen Suarez De Figueroa Baonza, Camille Pradel, and Nathalie Hernandez. Verifying ontology requirements with SWIP (poster). In *International Conference on Knowledge Engineering and Knowledge Management (EKAW 2012)*, page (electronic medium), <http://www.springerlink.com>, 2012. Springer. *Cité page 69*
- Ioan Szilagyi and Patrice Wira. Ontologies and Semantic Web for the Internet of Things - a survey. In *IECON. IEEE*, 2016. URL <http://ieeexplore.ieee.org/document/7793744/>. *3 citations pages 12, 73, et 83*
- Élodie Thiéblin, Fabien Amarger, Ollivier Haemmerlé, Nathalie Hernandez, and Cassia Trojahn. Rewriting SELECT SPARQL queries from 1 : n complex correspondences. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, Oktie Hassanzadeh, and Ryutaro Ichise, editors, *Proceedings of the 11th International Workshop on Ontology Matching (OM 2016) co-located with the 15th International Semantic Web Conference (ISWC 2016)*, volume 1766, pages 49–60, Kobe, Japan, October 2016. CEUR-WS.org. *3 citations pages 62, 67, et 88*
- Élodie Thiéblin, Fabien Amarger, Nathalie Hernandez, Catherine Roussey, and Cassia Trojahn. Cross-Querying LOD Datasets Using Complex Alignments : An Application to Agronomic Taxa. In *International Conference on Metadata and Semantics Research (MTSR 2017)*, pages 25–37. Springer, 2017a. URL http://www.irit.fr/publis/MELODI/Thieblin_MTSR2017.pdf. *2 citations pages 5 et 67*
- Élodie Thiéblin, Fabien Amarger, Nathalie Hernandez, Catherine Roussey, and Cassia Trojahn Dos Santos. Cross-Querying LOD Datasets Using Complex Alignments : An Application to Agronomic Taxa. In Emmanouel Garoufallou, Sirje Virkus, Rania Siatra, and Damiana Koutsomihia, editors, *11th International Conference Metadata and Semantics Research, MTSR 2017*, volume 755 of *CCIS*, pages 25–37, Tallinn, Estonia, November 2017b. Springer. ISBN 978-3-319-70862-1 978-3-319-70863-8. doi : 10.1007/978-3-319-70863-8_3. URL http://link.springer.com/10.1007/978-3-319-70863-8_3. *Cité page 62*
- Élodie Thiéblin, Nathalie Hernandez, Catherine Roussey, and Cássia Trojahn. Cross-querying LOD data sets using complex alignments : an experiment using agronomictaxon, agrovoc, dbpedia and TAXREF-LD. *IJMSO*, 13(2) :104–119, 2018. doi : 10.1504/IJMSO.2018.098387. URL <https://doi.org/10.1504/IJMSO.2018.098387>. *2 citations pages 62 et 67*
- Élodie Thiéblin, Ollivier haemmerlé Nathalie Hernandez, and Cássia Trojahn. Survey on complex ontology matching. *Semantic Web journal*, to appear, 2019. *6 citations pages 20, 23, 33, 62, 67, et 99*

- C.W. Thompson, P. Pazandak, and H.R. Tennant. Talk to your semantic web. *Internet Computing, IEEE*, 9(6) :75–78, 2005. *Cité page 50*
- Anis Tissaoui. *Évolution cohérente des ressources termino-ontologiques et des annotations sémantiques*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 2013. (Soutenance le 05/12/2013). *Cité page 37*
- Anis Tissaoui, Nathalie Aussenac-Gilles, Nathalie Hernandez, and Philippe Laublet. Evonto - joint evolution of ontologies and semantic annotations. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD)*, pages 226–231, 2011a. *3 citations pages 4, 37, et 40*
- Anis Tissaoui, Nathalie Aussenac-Gilles, Philippe Laublet, and Nathalie Hernandez. EvOnto : évolution de ressource termino-ontologique pour l’annotation sémantique. In Rafik Bouaziz, Pierre Bourque, and Gilles Falquet, editors, *Journées Francophones sur les Ontologies (JFO)*, pages 5–17. ACM SIGAPPFR, 2011b. URL http://www.irit.fr/publis/IC3/JFO_2011_Tissaoui_final.pdf. *Cité page 4*
- Anis Tissaoui, Nathalie Aussenac-Gilles, Philippe Laublet, and Nathalie Hernandez. Evonto : un outil d’évolution de ressource termino-ontologique pour l’annotation sémantique. *Technique et Science Informatiques*, 32(7-8) :817–840, 2013. doi : 10.3166/tsi.32.817-840. URL <https://doi.org/10.3166/tsi.32.817-840>. *Cité page 37*
- Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, pages 28–42, 2006. *Cité page 26*
- Ana I. Torre-Bastida, Jesús Bermúdez, Arantza Illarramendi, Eduardo Mena, and Marta González. Query Rewriting for an Incremental Search in Heterogeneous Linked Data Sources. In Henrik Legind Larsen, Maria J. Martin-Bautista, María Amparo Vila, Troels Andreasen, and Henning Christiansen, editors, *10th International Conference Flexible Query Answering Systems (FQAS 2013)*, volume 8132 of *LNCS*, pages 13–24, Granada, Spain, September 2013. Springer. ISBN 978-3-642-40768-0 978-3-642-40769-7. doi : 10.1007/978-3-642-40769-7_2. URL http://link.springer.com/10.1007/978-3-642-40769-7_2. *Cité page 67*
- Thanh Tran, Haofen Wang, Sebastian Rudolph, and Philipp Cimiano. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In *ICDE*, pages 405–416, 2009. *Cité page 52*
- Farhan Ullah, Muhammad Asif Habib, Muhammad Farhan, Shehzad Khalid, Mehr Yahya Durrani, and Sohail Jabbar. Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare. *Sustainable Cities and Society*, 34(March) :90–96, 2017. ISSN 22106707. doi : 10.1016/j.scs.2017.06.010. URL <http://dx.doi.org/10.1016/j.scs.2017.06.010>. *Cité page 76*
- Mike Uschold, Mike Healy, Keith Williamson, Peter Clark, and Steven Woods. Ontology reuse and application. In *Formal ontology in information systems*, volume 179, page 192. IOS Press Amsterdam, 1998. *Cité page 16*
- William Van Woensel and Syed Sibte Raza Abidi. Optimizing Semantic Reasoning on Memory-Constrained Platforms Using the RETE Algorithm. In *ESWC*, volume 10843 *LNCS*, pages 682–696, 2018. ISBN 9783319934167. doi : 10.1007/978-3-319-93417-4_44. *Cité page 80*
- Pierre-Yves Vandenbussche, Ghislain A Atemezing, María Poveda-Villalón, and Bernard Vatant. Linked open vocabularies (lov) : a gateway to reusable semantic vocabularies on the web. *Semantic Web*, 8(3) :437–452, 2017. *Cité page 11*
- Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, Ben De Meester, Gerald Haesendonck, and Pieter Colpaert. Triple pattern fragments : a low-cost knowledge graph interface for the web. *Journal of Web Semantics*, 37 :184–206, 2016. *Cité page 101*
- Boris Villazon-Terrazas, Mari Carmen Suarez-Figueroa, and Asuncion Gomez-Perez. A Pattern-Based Method for Re-Engineering Non-Ontological Resources into Ontologies. *International Journal on Semantic Web and Information Systems*, 6(4) :27–63, 2010. *Cité page 19*

- Panagiotis Vlacheas, Raffaele Giaffreda, Vera Stavroulaki, Dimitris Kelaidonis, Vassilis Foteinos, George Poullos, Panagiotis Demestichas, Andrey Somov, Abdur Biswas, and Klaus Moessner. Enabling smart cities through a cognitive management framework for the internet of things. *IEEE Communications Magazine*, 51(6) :102–111, 2013. ISSN 0163-6804. doi : 10.1109/MCOM.2013.6525602. *3 citations pages 76, 77, et 78*
- Max Völkel, Carlos F Enguix, Sebastian Ryszard Kruk, Anna V Zhdanova, Robert Stevens, and York Sure. Semversion : A versioning system for rdf and ontologies. Project deliverable, Knowledge Web, 2005. *Cité page 37*
- Haofen Wang, Kang Zhang, Qiaoling Liu, Thanh Tran, and Yong Yu. Q2semantic : a lightweight keyword interface to semantic search. In *Proceedings of the 5th European semantic web conference on The semantic web : research and applications*, pages 584–598. Springer-Verlag, 2008. *Cité page 52*
- Jingang Wang, Dandan Song, Chin-Yew Lin, and Lejian Liao. BIT and MSRA at trec kba ccr track 2013. In *Notebook of the Text Retrieval Conference 2013 (TREC 2013)*, Gaithersburgh, MD, USA., 2013. *Cité page 42*
- Shiyong Wang, Jiafu Wan, Di Li, and Chengliang Liu. Knowledge reasoning with semantic data for real-time data processing in smart factory. *Sensors (Switzerland)*, 18(2) :1–10, 2018. ISSN 14248220. doi : 10.3390/s18020471. *Cité page 79*
- Wei Wang, Suparna De, Yuchao Zhou, Xin Huang, and Klaus Moessner. Distributed Sensor Data Computing in Smart City Applications. In *Proceedings of WoWMoM 2017*, number June, 2017. ISBN 9781538627235. doi : 10.1109/WoWMoM.2017.7974338. *Cité page 28*
- Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyashev. Ontology-based data access : A survey. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 5511–5519. ijcai.org, 2018. doi : 10.24963/ijcai.2018/777. *Cité page 19*
- Guangquan Xu, Yan Cao, Yuanyuan Ren, Xiaohong Li, and Zhiyong Feng. Network security situation awareness based on semantic ontology and user-defined rules for internet of things. *IEEE Access*, 5 :21046–21056, 2017. *Cité page 82*
- Tan Xu, Paul McNamee, and Douglas W. Qard. HLTCOE at TREC 2013 : Temporal summarization. In *Proceedings of the Text REtrieval Conference*, Gaithersburgh, 2013. *Cité page 43*
- Yi Xu and Abdelsalam Helal. Scalable Cloud-Sensor Architecture for the Internet of Things. *IEEE Internet of Things Journal*, 3(3) :285–298, 2016. ISSN 23274662. doi : 10.1109/JIOT.2015.2455555. URL http://wso2.com/wso2/{_}resources/wso2/{_}whitepaper/{_}a-reference-architecture-for-the-internet-of-things.pdf. *Cité page 73*
- Fouad Zablith, Grigoris Antoniou, Mathieu d’Aquin, Giorgos Flouris, Haridimos Kondylakis, Enrico Motta, Dimitris Plexousakis, and Marta Sabou. Ontology evolution : a process-centric survey. *The Knowledge Engineering Review*, 30(01) :45–75, 2015. *3 citations pages 36, 42, et 46*
- Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1) :22–32, 2014. ISSN 2327-4662. doi : 10.1109/JIOT.2014.2306328. *3 citations pages 12, 73, et 83*
- Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality assessment for linked data : A survey. *Semantic Web*, 7(1) :63–93, 2016. *3 citations pages 34, 67, et 97*
- Dimitris Zeginis, Yannis Tzitzikas, and Vassilis Christophides. On the foundations of computing deltas between RDF models. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 637–651. Springer, 2007. doi : 10.1007/978-3-540-76298-0_46. *Cité page 37*

- Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pages 334–342, New York, NY, USA, 2001. ISBN 1-58113-331-6. doi : 10.1145/383952.384019. URL <http://doi.acm.org/10.1145/383952.384019>. *Cité page 43*
- Chunyun Zhang, Weiyan Xu, Fanyu Meng, Hongyan Li, Tong Wu, and Lixin Xu. The information extracion systems of pris at temporal summarization track. In *Proceedings of the Text REtrieval Conference*, Gaithersburgh, 2013. *Cité page 43*
- Xiaoqing Zheng, Stuart E. Madnick, and Xitong Li. SPARQL Query Mediation over RDF Data Sources with Disparate Contexts. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *WWW Workshop on Linked Data on the Web (LDOW2012)*, volume 937, Lyon, France, April 2012. CEUR-WS.org. URL <http://events.linkedata.org/ldow2012/>. *Cité page 66*
- Qi Zhou, Chong Wang, Miao Xiong, Haofen Wang, and Yong Yu. Spark : Adapting keyword query to semantic search. In *ISWC/ASWC*, pages 694–707, 2007. *Cité page 52*
- Nansu Zong, Sejin Nam, Jae-Hong Eom, Jinhyun Ahn, Hyunwhan Joe, and Hong-Gee Kim. Aligning ontologies with subsumption and equivalence relations in linked data. *Knowledge-Based Systems*, 76 :30–41, 2015. *Cité page 34*