Enrichissement d'un module ontologique : proposition d'une méthode pour le cas de l'agriculture

Fabien Amarger[1,2], Catherine Roussey[1], Jean-Pierre Chanet[1], Ollivier Haemmerlé[2], Nathalie Hernandez[2]

[1] IRSTEA
Équipe COPAIN, 24 Av. des Landais CS 200 85, 63178, Aubiére
prenom.nom@irstea.fr
[2] IRIT - UMR 5505
Université de Toulouse le Mirail, Département de Mathématiques-Informatique, 5 allées Antonio Machado, F-31058 Toulouse Cedex
prenom.nom@univ-tlse2.fr

Résumé :
Afin de contribué au Web de données pour l'agriculture, nous souhaitons construire une ontologie de ce domaine. Les classes de haut niveau d'un module ontologique ont déjà été définies ; la problématique maintenant est de réussir à l'enrichir. Nous souhaitons exploiter un point fort de ce domaine qui est l'existence de nombreuses sources d'information. Nous avons posé l'hypothèse que l'utilisation de plusieurs sources lors d'un processus d'extraction et de transformation permet une extraction simplifiée et plus efficace que l'utilisation d'une unique source. Par rapport à cette hypothèse, nous avons défini quatre étapes d'une méthode de transformation qui permettrait l'enrichissement de notre module ontologique.

Mots-clés : Module Ontologique, Enrichissement d'ontologie, Extraction et transformation, Désambiguïsation

1 Introduction

Les données disponibles sur le Web sont généralement de deux natures : (1) des données non structurées difficilement exploitables de manière automatique, comme un ensemble de pages HTML, ou (2) des données structurées destinées à une utilisation particulière, comme une base de données,

difficilement réutilisables par d'autres applications. Le Web de données (Berners-Lee, 2006) est une application du Web sémantique (Berners-Lee *et al.*, 2001) facilitant l'accès, le partage et l'alignement des données. Le W3C a proposé des standards de représentation des données et de leur schéma : les données sont représentées sous forme de triplets RDF, tandis que RDFS et OWL définissent les schémas de données associés. Lorsque ces schémas sont suffisamment complexes, ils portent le nom d'ontologies. La publication des données et de leurs schémas sur le Web facilite leur réutilisation dans diverses applications.

Il existe actuellement de très nombreuses données disponibles sur le Web qui pourraient être transformées en ontologies pour enrichir le Web de données. Malheureusement, ces données sont souvent représentées dans des langages moins expressifs et moins formels que les langages de représentation d'ontologies. C'est particulièrement le cas dans le domaine de l'agriculture. Il existe par exemple le thésaurus AGROVOC, la base de données e-phy ou encore le corpus de texte des Bulletins de Santé du Végétal (BSV), mais très peu d'ontologies sont disponibles pour ce domaine. Nous essayons, dans nos travaux, de combler ce manque, en nous fondant particulièrement sur la réutilisation des sources disponibles.

Nous souhaitons créer une ontologie permettant de décrire les données concernant l'observation des attaques de bio-agresseurs (insectes, champignons, maladies, etc.) sur les cultures, ainsi que les techniques de traitement de ces agresseurs. Cette ontologie permettra de publier les données disponibles ; elle permettra également d'annoter les nombreux documents mobilisables pour faire évoluer les pratiques de traitement des agresseurs. L'ensemble de ces données deviendra alors interrogeable par des requêtes exprimées en langage naturel en utilisant le système SWIP (Pradel *et al.*, 2012).

Nous allons nous intéresser à certaines hypothèses sur la réutilisation de sources non ontologiques afin d'enrichir un module ontologique déjà existant. Pour cela nous allons d'abord présenter le module à enrichir, suivi d'un état de l'art sur les méthodes de transformation et enfin nos hypothèses concernant l'enrichissement du module.

2 Module ontologique

La notion de module est présentée en détail dans la méthode NeOn (Suarez-Figueroa *et al.*, 2012). Ce que nous appelons ici un module ontologique est un sous ensemble d'une ontologie globale que nous cher-

chons à construire. Ce découpage de l'ontologie générale est choisi en fonction des différentes thématiques recherchées et des patrons de conception disponibles [1]. Un patron de conception est une modélisation générique permettant de répondre à certaines questions. Par exemple un patron de conception est disponible si nous souhaitons utiliser une taxonomie (LinnaeanTaxonomy). Les patrons de conception permettent de modéliser une ontologie de façon rapide et efficace puisqu'ils suivent une procédure de validation par des experts sur le site. Concernant notre exemple d'une ontologie sur les agresseurs des plantes, nous pouvons séparer plusieurs modules existants :

- la classification taxonomique des organismes vivants,
- la description des cultures en France en fonction de leur type d'usage,
- l'occupation culturale des sols,
- la rotation des cultures sur les parcelles agricoles,
- les observations faites sur les cultures,
- les produits phytosanitaires.

La description complète de tous ces modules est disponibles sur le site AgriOntology [2]. Ce wiki regroupe toutes les informations nécessaires à l'élaboration de l'ontologie finale.

Nous allons nous intéresser particulièrement au premier module. Ce module est le plus abouti pour l'instant. Il nous sert de cas d'étude dans un premier temps pour généraliser une méthodologie qui sera ensuite appliquée aux autres modules.

D'après la figure 1, ce module permet de spécifier la classification taxonomique des organismes vivants. Pour cela nous avons réutilisé deux patrons de conception définis par le projet Neon et plusieurs ontologies déjà existantes (SKOS, TaxonConcept, DarwinCore, Biological Taxonomy Vocabulary et TaxMeOn).

Dans la figure 1, en gras sont représentés les ajouts qui ont été faits aux ontologies réutilisées. La classe "irstea:LivingOrganism" (organisme vivant) représente un être vivant. Cette classe est reliée à un "skos:Concept" par la relation "neon:isClassifiedBy" (*estClassifiéPar*), un "skos:Concept" représentant une agglomération de termes désignant le même concept. Cette classe est spécialisée par l'intermédiaire de la classe "neon:Taxon" qui spécifie qu'une classe peut être un élément taxonomique, c'est à dire représentant la désignation d'un organisme vivant suivant une classification donnée. Cette classification est représentée par la classe "skos:ConceptScheme" et

1. http ://ontologydesignpatterns.org/
2. https ://sites.google.com/site/agriontology/home/irstea

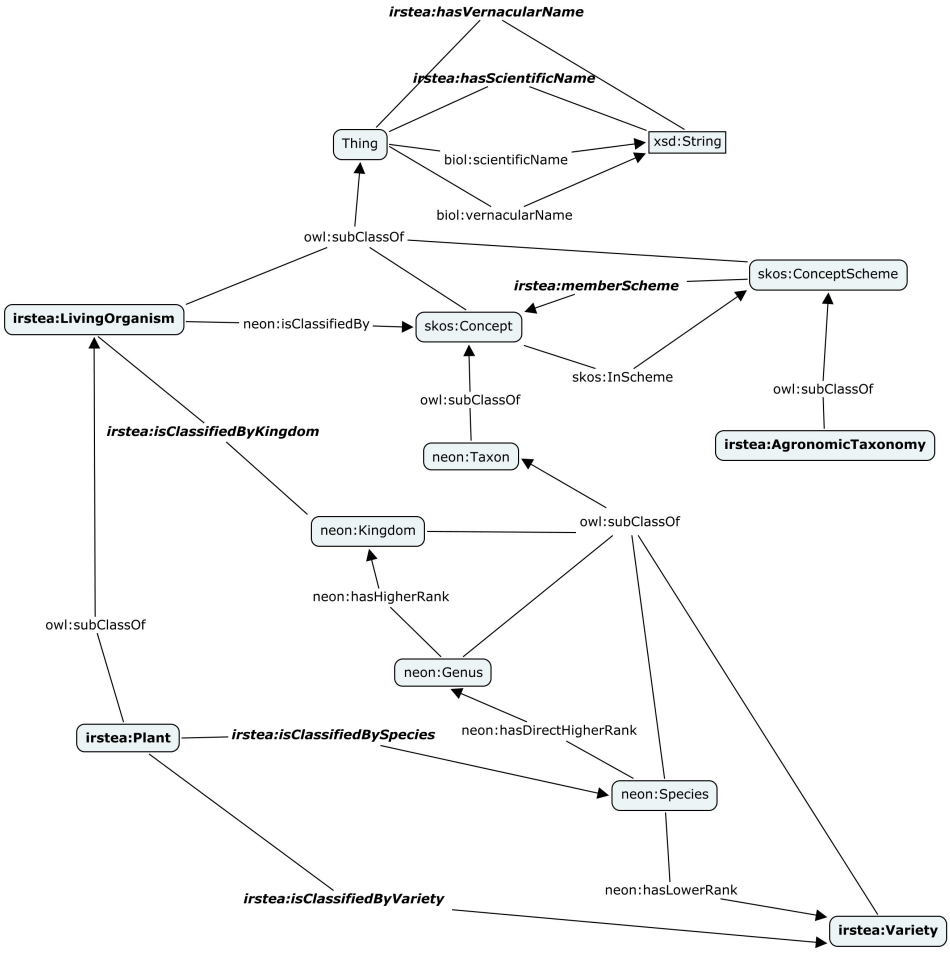

FIGURE 1 – Module AgronomicTaxon développé par l'IRSTEA

spécialisée par "irstea:AgronomicTaxonomy". La relation "skos:InScheme" ("dans le projet", ou ici "dans la classification") permet de faire le lien entre un taxon et une classification. Dans la figure 1, ces taxons sont répartis en quatre types hiérarchisés "neon:Kingdom" (règne), "neon:Genus" (genre), "neon:Species" (espèce), "irstea:Variety" (variété). Tous les éléments de l'ontologie utilisant ce module auront une relation "irstea:hasVernacularName" (*aPourNomCommun*) et "irstea:hasScientificName" (*aPourNomScientifique*) qui permettent de spécifier le nom scientifique et commun.

Nous ne disposons que des classes de haut niveau du module. Il faut donc réussir à le compléter pour qu'il corresponde à nos attentes. Cet enri-

chissement peut apporter de nouvelles classes, de nouvelles relations [3], de nouveaux attributs [4] ou encore des instances (de classes, de relations, ...), à condition que les éléments générés soient des spécialisations des éléments existants. Dans le cas contraire, ils sont considérés comme hors domaine.

Nous souhaitons parvenir à effectuer cet enrichissement en utilisant les différentes sources du domaine, telles que le thésaurus AGROVOC ou la base de données publique e-phy.

3 État de l'art sur la transformation de sources non ontologiques en ontologie

Nous nous sommes particulièrement intéressés à deux formats de sources : les thésaurus et les bases de données. Nous avons donc étudié les différentes méthodes de transformation relatives à ces deux formats.

Lors de nos précédents travaux (Amarger *et al.*, 2013), nous avons étudié en détail les méthodes de transformation de thésaurus en ontologies. Cette étude nous a permis de dégager plusieurs tendances.

Certaines méthodes de transformation automatisent au maximum le processus, telles que (Soergel *et al.*, 2004) ou encore (Chrisment *et al.*, 2008). Le premier définit un certain nombre de patrons de transformations permettant d'orienter le processus. Le second se concentre sur la désambiguïsation de relations en utilisant un traitement automatique d'un corpus de texte. Tous deux témoignent d'une difficulté particulière quant à l'extraction de relations hiérarchiques de subclassof. Les travaux de (Soergel *et al.*, 2004) nécessitent la définition d'un patron pour chaque type de transformation. Dans les travaux de (Chrisment *et al.*, 2008) les relations hiérarchiques ne sont pas traitées.

Les travaux de (Villazón-Terrazas *et al.*, 2010) utilisent une ressource externe (comme DBpedia par exemple) qui permet de faciliter la désambiguïsation.

Les travaux récents de transformation de thésaurus en ontologies tels que (Charlet *et al.*, 2012; Kless *et al.*, 2012) effectuent cette transformation manuellement. Ce retour aux transformations manuelles pourrait s'expliquer par les résultats mitigés des méthodes de transformation automatiques de la littérature.

Nous nous sommes ensuite orientés vers les méthodes de transformation de bases de données. Deux tendances sont identifiables.

3. nous appelons relations les object properties de OWL

4. nous appelons attributs les data properties et les annotation properties de OWL

La première est la définition de règles de transformations. Dans les travaux de (Sequeda *et al.*, 2012) sont définies formellement des règles permettant la transformation de bases de données en ontologies. Les mêmes types de règles sont présentes dans (Cerbah, 2008), mais une évolution a été apportée concernant les attributs catégorisants, qui permettent une extraction d'une hiérarchie subClassOf. La deuxième tendance est la transformation par l'intermédiaire de requêtes. Nous retrouvons notamment Triplify(Auer *et al.*, 2009) qui permet l'expression de requêtes SQL pour orienter la transformation. Cette tendance est aussi suivie par le W3C puisqu'un standard la concernant est en cours de développement, le R2RML (Das *et al.*, 2012). Plusieurs travaux de transformation de bases de données effectuent une transformation spécifique à la source. C'est notamment le cas de (Krivine *et al.*, 2009) qui utilisent des caractéristiques spécifiques de la base de données pour extraire une hiérarchie de subClassOf.

Tous les travaux cités précédemment ont encore certaines lacunes. Nous retrouvons des faiblesses concernant le filtrage des classes extraites. Les travaux effectuant ce filtrage le font manuellement, ou par l'intermédiaire de requêtes ciblées par exemple. Cela devient fastidieux si nous nous intéressons à la transformation de sources de grande taille, comme AGROVOC. Les travaux présentent souvent une méthode de validation des informations extraites, mais ne valident pas l'extraction des classes, considérées comme valides par défaut. Enfin, tous les travaux présentant une méthode automatique témoignent d'une grande difficulté à extraire des relations hiérarchiques, autant pour les thésaurus que pour les bases de données.

Tous les travaux présentés précédemment ne traitent qu'une source à la fois. Nous pouvons néanmoins citer les travaux de (Charlet *et al.*, 2012) qui proposent une utilisation de plusieurs sources, mais de manière incrémentale. Aucune des méthodologies ne prend en compte la possibilité de multiplier ces sources pour confronter les informations extraites.

4 Proposition

Notre objectif est donc de pouvoir proposer une méthodologie permettant de répondre aux problèmes précédemment cités en utilisant les avantages liés à l'existence de plusieurs sources du même domaine.

Pour cela, nous avons posé les deux hypothèses suivantes :

– "Lors de la création d'une ontologie, toutes les sources n'ont pas le même intérêt"

– "La présence de la même information dans plusieurs sources implique une augmentation de la confiance attribuée à cette information"

D'après la première hypothèse, nous pouvons imaginer la définition d'un score d'intérêt pour chaque source à exploiter dans le processus. Ce score d'intérêt serait le reflet de la pertinence de la source par rapport au domaine étudié. La deuxième hypothèse nous amène à imaginer un score de confiance pour chaque information extraite directement proportionnel aux scores d'intérêt des sources dans lesquelles cette information apparaît. Une confiance trop faible pourrait impliquer un abandon de l'information ou au moins une remise en question de celle-ci car il n'est pas possible de l'affirmer avec suffisamment de conviction. Nous pouvons imaginer une fourchette paramétrable pour le score de confiance dans laquelle l'information serait validée manuellement, en dessous elle serait rejetée, au dessus elle serait acceptée.

Nous observons dans ces hypothèses une distinction entre sources et ressources. Il est nécessaire de définir ces termes :

Source : Ce que nous appelons une source est un élément que nous cherchons à transformer, c'est à dire que les informations qui seront ajoutées au module ontologique proviendront de ces sources. Ce peut être par exemple le cas d'AGROVOC.

Ressource : Une ressource nous aide à transformer les sources ; nous ne cherchons pas à la transformer, mais nous allons l'utiliser pour nous aider sur certains aspects de la transformation de sources. Par exemple lors d'une étape de désambiguïsation, DBpedia peut être utilisé en tant que ressource.

À partir de ces hypothèses, nous avons défini quatre étapes permettant l'enrichissement du module ontologique :

4.1 Étape 1 : Filtrage sur le domaine

L'idée générale de ce filtrage automatique serait l'utilisation de ressources externes pour désambiguïser l'appartenance d'une classe au domaine étudié. Ce domaine serait borné par le module créé. L'extraction serait initiée par un alignement entre les feuilles du module et la ressource. Cet alignement peut se faire par une technique d'alignement terminologique ou bien, pour plus de fiabilité, manuellement. Le module ne disposant pas d'une grande quantité de classes, un alignement manuel est tout à fait envisageable. Suite à cette étape initiale, l'extraction des classes se

ferait par la même méthode simple [5]. Pour chaque classe extraite, une désambiguïsation par une ressource générique serait effectuée pour déterminer la liste montante des hyperonymes. Si un de ces hyperonymes est aussi un des éléments alignés avec le module, alors la classe extraite fait partie du domaine. Dans le cas contraire cette classe ne serait pas exportée vers le document final car elle ne ferait pas partie du domaine. De cette manière, nous obtiendrions une liste de classes potentielles faisant partie du domaine d'étude.

4.2 Étape 2 : Validation des classes

Après l'étape précédente, nous ne pouvons pas garantir l'existence propre d'une classe faisant partie de la liste. Il se peut qu'une des classes présente dans la liste précédente n'ait pas lieu d'être. Par exemple une classe extraite à partir d'une base de données qui ne représenterait pas une classe devant faire partie de l'ontologie mais une relation entre d'autres classes. Ce genre de problématique est dû à la nature des sources étudiées qui est différente de celles d'une ontologie. Pour valider ces classes, nous envisageons d'entrecroiser le traitement de plusieurs sources et ressources durant cette étape. Par exemple si une classe extraite à partir d'une première source est similaire à une classe extraite à partir d'une deuxième source, alors la confiance de l'existence de cette classe serait augmentée. De même, lors de l'étape précédente, si la classe est alignée avec un élément d'une ressource générique, alors sa confiance serait là-encore augmentée. Nous pouvons envisager l'utilisation de plusieurs autres ressources qui permettraient une validation supplémentaire. De plus cet alignement général permettrait une implémentation simplifiée de l'ontologie générée dans le web de données. Celui-ci préconisant les liens entre les différentes données disponible sur le web.

4.3 Étape 3 : Extraction de la hiérarchie (subClassOf)

Dans un thésaurus, des relations hiérarchiques [6] existent mais ne sont pas toujours correctes. En effet, le but premier d'un thésaurus étant une utilisation par des humains (majoritairement des documentalistes), certaines libertés d'utilisation apparaissent et ne permettent pas une extraction

5. Un concept du thésaurus (ou une table de la base de données) transformé en une classe de l'ontologie

6. les relations hiérarchiques dans un thésaurus s'intitulent narrower et broader

simple (Soergel *et al.*, 2004). Concernant les bases de données, le problème réside dans la formalisation des relations hiérarchiques. Dans les travaux de (Sequeda *et al.*, 2011) sont soulignées les différentes méthodes de représentation d'une hiérarchie (subClassOf). Celles-ci pouvant être confondues avec de simples relations, il devient particulièrement complexe d'extraire cette hiérarchie.

Dans les travaux de (Rector, 2003), il est précisé qu'il n'est pas conseillé d'exprimer explicitement les relations hiérarchiques (subClassOf) mais plutôt d'utiliser un raisonneur pour les déduire. Il est donc plus important de se concentrer sur la définition des axiomes présents dans l'ontologie que sur la hiérarchie (subClassOf) entre les classes.

C'est pour cela que nous avons défini un certain nombre d'axiomes dans notre module qui nous permettront d'inférer les relations hiérarchiques entre les classes extraites. La problématique d'extraction de la hiérarchie (subClassOf) est donc reportée sur l'extraction des relations autres que hiérarchiques.

4.4 Étape 4 : Désambiguïsation des relations

D'après notre étude, la découverte de l'existence d'une relation entre deux classes est simple. Dans un thésaurus, toute relation entre deux regroupements de termes (skos:Concept) permet de déduire l'existence d'une relation. Pour l'extraction à partir d'une base de données, (Sequeda *et al.*, 2011) propose une règle permettant de déduire l'existence d'une relation entre deux classes potentielles si il existe une clef étrangère de l'une d'elle vers l'autre, ou si une table binaire spécifiant une relation n-n existe.

Le véritable enjeu consiste à réussir à désambiguïser ces relations. L'application des axiomes cités dans le paragraphe précédent deviendrait alors possible, ainsi que la déduction des relations hiérarchiques (subClassOf).

Les méthodes les plus efficaces d'extraction de relations se fondent sur une étude de corpus. Les seuls travaux d'extraction automatique de relations à partir d'un thésaurus (Chrisment *et al.*, 2008) utilisent eux-aussi un corpus. Nous envisageons donc un traitement automatique du corpus des bulletins de santé du végétal pour nous permettre de désambiguïser ces relations. Cette étape est encore à l'étude et n'est pas détaillée dans ce papier.

Nous envisageons néanmoins d'effectuer une validation au même titre que celle proposée dans le paragraphe précédent. Celle-ci viendrait valider les désambiguïsations effectuées par l'intermédiaire d'un alignement des relations. Plus la relation est présente dans plusieurs sources et ressources

et plus la confiance dans cette relation serait grande.

4.5 Exemple

Pour illustrer nos propos nous allons étudier l'exemple d'une extraction pour l'enrichissement du module. Partons de la sous-partie d'AGROVOC suivante :
$"TriticumDurum" --BroaderTerm-->"Triticum"$

La première étape est l'extraction des classes et la détermination du fait qu'elles sont dans le domaine ou non. L'extraction de "Triticum" (blé) et "Triticum Durum" (blé dur) nécessite une désambiguïsation par ressource externe. Nous utilisons dans cet exemple Freebase comme ressource externe en précisant que le domaine est "biology", la relation d'hyperonymie est "Higher classification" et la relation de type est "Rank". Après un alignement manuel du module à Freebase, nous pouvons déduire que "Triticum" est un "Genus" (genre) et que "Triticum Durum" est un "Species" (espèce). Elles font donc partie du domaine.

Concernant la validation des classes extraites, l'alignement est déjà fait avec la ressource qui a été utilisée pour la désambiguïsation. Un alignement est effectué avec d'autres ressources, comme DBpedia, qui permettrait d'augmenter la confiance pour la création de ces classes. En faisant une recherche par mot-clef sur DBpedia, nous arrivons à récupérer des URI.

La hiérarchie que nous souhaitons représenter ici n'est pas directement ordonnée par la relation "subClassOf" mais par "hasDirectHigherRank" (*aPourRangSupérieurDirect*). La problématique revient au même. La relation "Broader Term" ne sera pas traduite automatiquement par une hiérarchie mais seulement par une relation potentielle inconnue (nous connaissons l'existence de la relation sans en connaître son type).

Imaginons que l'étude du corpus des bulletins de santé du végétal nous permette de désambiguïser la relation en précisant que "Triticum Durum" a pour rang supérieur (hasHigherRank) "Triticum". Nous n'avons donc pas directement la relation "hasDirectHigherRank". Néanmoins, un axiome présent dans le module stipule que : "si une classe A est subClassOf de Species et une classe B est subClassOf de Genus, et que la classe A a la relation hasHigherRank vers la classe B, alors la classe A a la relation hasDirectHigherRank vers la classe B". Nous obtenons donc notre relation "hasDirectHigherRank" comme souhaité. Le résultat de l'extraction est visible dans la figure 2.

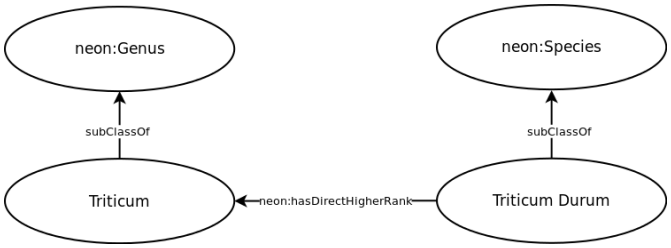

FIGURE 2 – Exemple de résultat d'extraction

5 Conclusion

Nous avons listé dans cet article les pistes de recherche qui répondent à la problématique de l'enrichissement d'ontologie à partir de sources non ontologiques dans le cas de l'agriculture. Enfin, nous avons défini deux hypothèses qui permettent un meilleur enrichissement et quatre étapes pour le réaliser. Ces hypothèses reposent sur notre conviction qu'une transformation utilisant plusieurs sources est plus efficace que si elle n'en utilise qu'une.

Ces pistes permettent de dresser le panorama de nos futurs travaux concernant l'enrichissement du module.

Références

AMARGER F., ROUSSEY C., CHANET J.-P., HAEMMERLÉ O. & HERNANDEZ N. (2013). État de l'art : Extraction d'information à partir de thésaurus pour générer une ontologie. *INFORSID*.

AUER S., DIETZOLD S., LEHMANN J., HELLMANN S. & AUMUELLER D. (2009). Triplify : light-weight linked data publication from relational databases. In *Proceedings of the 18th international conference on World wide web*, WWW '09, p. 621–630, New York, NY, USA : ACM.

BERNERS-LEE T. (2006). Linked data - design issues. http ://www.w3.org/DesignIssues/LinkedData.html.

BERNERS-LEE T., HENDLER J. & LASSILA O. (2001). The semantic web. *Scientific american*, **284**, 28–37.

CERBAH F. (2008). Learning highly structured semantic repositories from relational databases. In *The Semantic Web : Research and Applications*, p. 777–781. Springer.

CHARLET J., DECLERCK G., DHOMBRES F., GAYET P., MIROUX P., VANDENBUSSCHE P. *et al.* (2012). Construire une ontologie médicale pour la recherche d'information : problématiques terminologiques et de modélisation. *Actes des 23es journées francophones d'Ingénierie des connaissances*, p. 33–48.

CHRISMENT C., HAEMMERLÉ O., HERNANDEZ N. & MOTHE J. (2008). Méthodologie de transformation d'un thesaurus en une ontologie de domaine. *Revue d'Intelligence Artificielle*, **22**, 7–37.

DAS S., SUNDARA S. & CYGANIAK R. (2012). R2RML : RDB to RDF mapping language. http ://www.citeulike.org/group/14833/article/11522782.

KLESS D., JANSEN L., LINDENTHAL J. & WIEBENSOHN J. (2012). A method for re-engineering a thesaurus into an ontology. volume Formal Ontology in Information Systems, p. 133 : Ios PressInc.

KRIVINE S., NOBÉCOURT J., SOUALMIA L., CERBAH F. & DUCLOS C. (2009). Construction automatique d'ontologie à partir de bases de données relationnelles : application au médicament dans le domaine de la pharmacovigilance. *Actes des 20es journées francophones d'Ingénierie des connaissances*.

PRADEL C., HAEMMERLÉ O. & HERNANDEZ N. (2012). A semantic web interface using patterns : the SWIP system. *Graph Structures for Knowledge Representation and Reasoning*, p. 172–187.

RECTOR A. L. (2003). Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In *Proceedings of the 2nd international conference on Knowledge capture*, p. 121–128.

SEQUEDA J., ARENAS M. & MIRANKER D. (2012). On directly mapping relational databases to rdf and owl (extended version). *Computing Research Repository*.

SEQUEDA J. F., TIRMIZI S. H., CORCHO O. & MIRANKER D. P. (2011). Survey of directly mapping SQL databases to the semantic web. *The Knowledge Engineering Review*, **26**, 445–486.

SOERGEL D., LAUSER B., LIANG A., FISSEHA F., KEIZER J. & KATZ S. (2004). Reengineering thesauri for new applications : The AGROVOC example. *Journal of Digital Information*, **4**.

SUAREZ-FIGUEROA M.-C., GOMEZ-PEREZ A., MOTTA E. & GANGEMI A. (2012). *Ontology Engineering in a Networked World*.

VILLAZÓN-TERRAZAS B. C., SUÁREZ-FIGUEROA M. & GÓMEZ-PÉREZ A. (2010). A pattern-based method for re-engineering non-ontological resources into ontologies. *International Journal on Semantic Web and Information Systems*, **6**, 27–63.