De la Réutilisation de Données sous Licence dans le Web des Données

Patricia Serrano-Alvarado

De la Réutilisation de Données sous Licence dans le Web des Données

Patricia Serrano Alvarado

Maître de Conférences HDR
Nantes Université, LS2N, UMR 6004, F-44000 Nantes, France
Faculté des Sciences et des Techniques
Mél : Patricia.Serrano-Alvarado@univ-nantes.fr

Les travaux de recherche résumés dans cette communication ont été réalisés dans le contexte des travaux de la thèse en Informatique de Benjamin Moreau.

Résumé

L'exploitation de grandes masses de données (big data), actuellement mise en avant pour faire progresser la connaissance et améliorer les services, suppose de développer des outils de combinaisons entre des ressources de différentes natures et de différentes origines. Les applications Web facilitent la combinaison de données ou plus généralement l'intégration de ressources de types différents (images, vidéos, services Web, documents, code informatique, fichiers source de données, etc.) pour en créer de nouvelles. Lorsque l'on souhaite autoriser la réutilisation de données, choisir la licence appropriée pour une ressource qui en réutilise d'autres n'est pas simple. La licence doit être conforme aux restrictions imposées par les licences protégeant les ressources réutilisées. Le risque est soit de choisir une licence trop restrictive, rendant la nouvelle ressource difficilement réutilisable à son tour, soit de choisir une licence trop peu restrictive qui n'est pas conforme aux licences impliquées. Trouver le bon compromis entre conformité aux exigences d'autres licences et facilité de réutilisation ultérieure est un processus qui peut être difficile. Un traitement automatique des licences faciliterait cette tâche. La présente étude présente nos travaux sur la compatibilité de licences. Notre contribution est un modèle, nommé CaLi, qui permet d'ordonner des licences lisibles par machine selon leur compatibilité en utilisant des relations de restriction entre les licences pour définir leur compatibilité et leur conformité. Validé expérimentalement, CaLi a montré son efficacité et sa facilité d'utilisation dans différents cas d'utilisation. Ce travail est une étape vers la facilitation et l'encouragement de la publication et de la réutilisation de ressources sous licence dans le Web de données.

Mots-clés : licences, compatibilité de licences, Web des données, Web sémantique.

1. Introduction

Nos activités digitales sont nombreuses et quotidiennes. Elles concernent par exemple la messagerie électronique et instantanée, les bibliothèques digitales, les banques, les réseaux sociaux, les services publiques, les recherches scientifiques, les activités industrielles et bien d'autres activités. Ces activités génèrent des masses de données à l'origine du terme « big data ». Les défis du big data sont nombreux. Il faut pouvoir stocker des grandes masses de données, les analyser, les sécuriser, les comprendre, les intégrer et les réutiliser de manière efficace, tout en préservant la confidentialité des personnes ainsi que les exigences légales et réglementaires sur la gestion des données.

Le Web est un espace global qui permet l'interconnexion et la navigation de données de diverses natures. Typiquement nous y trouvons des pages Web reliées par des hyperliens mais également d'autres « ressources »[1] comme des images, des vidéos, du code informatique, des applications, des fichiers contenant des données semi-structurées comme des fichiers csv, excel, json, xml, etc. Le Web peut être vu comme un exemple de big data décentralisé à l'échelle de la planète. Son exploitation ouvre d'innombrables perspectives qui seront possibles uniquement si on développe les outils permettant la combinaison et la réutilisation intelligente et efficace des ressources du Web. Le « Web sémantique », considéré comme une évolution du Web, est apparu il y a une vingtaine d'années. Il se construit au-dessus du Web avec la prise en compte de métadonnées sémantiques sur les ressources. Les métadonnées sont de descriptions sémantiques qui permettent de représenter formellement les ressources. Par exemple, une image est enrichie sémantiquement lorsqu'on lui associe une date ou un lieu. Ainsi, le Web sémantique aide à la compréhension du Web et facilite l'automatisation de processus. La connaissance apportée par le Web sémantique peut faciliter de manière importante l'intégration et la combinaison dynamiques des ressources du Web.

En droit européen, tout contenu publié est automatiquement couvert par le droit d'auteur[2]. Cela est également valable dans le Web. Afin de faciliter la réutilisation et le contrôle des ressources du Web, les producteurs de ressources peuvent en plus leur associer une licence. Les licences permettent d'établir un contrat conférant un droit de (ré)utilisation (ou d'exploitation) à un tier. Typiquement, une licence décrit de manière précise les conditions de réutilisation des ressources, c'est-à-dire, les actions qu'un tier peut faire, ne doit pas faire ou est contraint de faire en cas de réutilisation. Creative Commons est une organisation ayant pour objectif de faciliter la diffusion et le partage d'information. Elle propose un ensemble de licences[3] fréquemment utilisées sur le Web. Mais d'autres licences existent comme les licences pour du code source informatique[4]. De plus, de nombreux gestionnaires de données créent de licences personnalisées.

Les applications Web facilitent la combinaison de ressources pour en créer de nouvelles. Cependant, lorsque l'on souhaite autoriser la réutilisation de données, choisir la licence appropriée pour une ressource qui en réutilise d'autres n'est pas simple. La licence doit être conforme aux restrictions imposées par les licences protégeant les ressources réutilisées. Le risque est soit de choisir une licence trop restrictive, rendant la nouvelle ressource difficilement réutilisable à son tour, soit de choisir une licence trop peu restrictive et non conforme aux licences impliquées. Trouver le bon compromis entre conformité aux exigences d'autres licences et facilité de réutilisation ultérieure est un processus difficile. Un traitement automatique des licences faciliterait cette tâche. Ainsi, dans ce chapitre, nous résumons nos travaux qui répondent à la question suivante : *comment automatiser le choix d'une licence pour une ressource lorsque celle-ci combine, en tout ou partie, d'autres ressources qui sont déjà sous licence* ?

Le chapitre est organisé comme suit. La section 2 introduit le contexte de ce travail, le Web sémantique et le Web des données. La section 3 est le cœur de ce chapitre, elle motive notre problématique et présente le modèle nommé CaLi qui permet de définir des classifications de licences selon leur compatibilité. La section 4 illustre l'utilité de CaLi avec plusieurs applications. La section 6 conclu ce chapitre.

[1] Nous utilisons le terme ressource comme une généralisation de donnée, fichier, code source informatique, application, etc.

[2] Pour un aperçu rapide, voir cette FAQ sur le droit d'auteur https://euipo.europa.eu/ohimportal/fr/web/observatory/faq-lu

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

[4] Voir la liste des licences compilées dans le projet SPDX https://spdx.org/licenses/

2. Le Web sémantique et le Web des données

Le Web s'est développé à l'échelle planétaire grâce aux recommandations proposées par le W3C (*World Wide Web Consortium*)[5]. Cet organisme à but non lucratif promeut depuis 1994 la compatibilité des technologies du Web. Une recommandation du W3C est un document technique qui a suivi plusieurs étapes de maturation et que les industriels peuvent utiliser pour développer les technologies du Web. Les recommandations du W3C sont considérées comme des standards techniques du Web. Ainsi, à l'origine du Web, le W3C a proposé, par exemple, HTML (*HyperText Markup Language*), le langage de balisage recommandé pour représenter les pages Web.

Le « Web des données » utilise les standards et recommandations du W3C pour le Web sémantique afin de faciliter l'intégration de jeux de données structurées sur le Web. L'objectif du Web des données est de construire un réseau global interconnecté d'information (pas uniquement de pages Web). La sémantique fournit des significations bien définies qui peuvent être utilisées automatiquement pour mieux comprendre les ressources du Web. Par exemple, une image peut être enrichie avec des métadonnées décrivant le sujet, l'appareil utilisé pour capter l'image, la date à laquelle l'image a été prise, le lieu, le contexte, etc. D'autres informations sémantiques importantes peuvent être annotées, comme les règles de confidentialité sur l'image, la licence sous laquelle elle peut être réutilisée, etc.

RDF (*Resource Description Framework*) [6] est la recommandation du W3C pour décrire la sémantique des ressources. Une description en RDF est un « fait » sous la forme d'un triplet composé d'un sujet, d'un prédicat et d'un objet. Par exemple le triplet suivant permet de décrire le fait que le propriétaire de la Tour Eiffel est la ville de Paris.

`Eiffel_Tower` *owner* `Paris .`

Dans ce triplet, le sujet est `Eiffel_Tower,` le prédicat est *owner* et l'objet est `Paris`. Afin d'avoir des triplets RDF appropriés, il est nécessaire d'utiliser des identifiants uniques, des URI. Pour être utile, au minimum le sujet doit être identifié avec précision grâce à un URI, qui permet de ne pas le confondre avec un autre sujet possible (ex. : la reproduction de la Tour Eiffel à Las Vegas) [7].

Dans le Web sémantique, les ontologies sont nécessaires pour une description uniforme des ressources. En informatique et science de l'information, une ontologie est un ensemble structuré de termes et de concepts permettant de décrire un domaine de connaissances. Les concepts peuvent être reliés par des relations sémantiques (ex. : un test médical doit être associé à une date) ou de subsomption (par exemple une image médicale est aussi un test médical). Les ontologies sont conçues de manière consensuelle par des experts du domaine décrit. OWL (*Web Ontology Language*)[8] et RDFS (*RDF Schema*)[9] sont recommandés par le W3C pour définir des

5 https://fr.wikipedia.org/wiki/World_Wide_Web_Consortium

6 http://www.w3.org/RDF

7 Notre triplet exemple avec des URI : https://dbpedia.org/page/Eiffel_Tower https://dbpedia.org/ontology/owner https://dbpedia.org/page/Paris

8 https://www.w3.org/TR/owl-features/

9 https://www.w3.org/TR/rdf-mt/#rdfs_interp

ontologies. Parmi les plus utilisées on peut citer Dublin Core [10] pour la description de documents, FOAF[11] pour la description de personnes et leurs relations ou Gene Ontology[12] pour la description de gènes. L'usage des ontologies pour décrire les ressources au sein d'une communauté facilite la mise en commun des ressources et leur l'intégration.

Le code RDF ci-dessous montre la description sémantique d'une image médicale (syntaxe Turtle[13]). L'ontologie utilisée est principalement Schema.org[14]. Les faits en RDF décrivent l'image. Le sujet décrit ici (en gras) est une radiographie du thorax. Les prédicats (en italique) décrivent le type de l'image, le dispositif utilisé, etc. Les objets sont les valeurs correspondantes qui peuvent être du texte (ex. : "Scanner ULTRA 3000") ou un concept d'une ontologie (ex. : schema:Radiography). Ainsi, le code RDF permet à un automate de lire et exploiter les métadonnées attachées à la ressource.

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix schema: <schema.org/>
@prefix dcterms: <http://purl.org/dc/terms/>
@prefix cclic: <https://creativecommons.org/licenses/>

:Chest_Xray_PA_3-8-2010.png  rdf:type  schema:ImagingTest ;
                        schema:usesDevice  "Scanner ULTRA 3000" ;
                        schema:imagingTechnique schema:Radiography ;
                        schema:description "Radiographie du thorax" ;
                        dcterms:license cclic:by-nc-nd/4.0 .
```

Fragment de code 1. Code RDF qui décrit une image.

La hiérarchie des concepts dans les ontologies permet une description fine des ressources. Un fragment des concepts utilisés dans la description de notre image se trouve sur la figure 1[15]. Dans l'ontologie schema.org, `Thing` est la classe la plus générale. Nous pouvons voir que le type attribué à notre image, la classe `ImagingTesting`, est une sous-classe de `MedicalTest`, qui est une sous-classe de `MedicalEntity` qui est sous-classe de `Thing`. `ImageTesting` hérite des propriétés de ses superclasses. Dans notre exemple ce sont les propriétés `usesDevices` et `description`. Ainsi, même si directement notre image n'est pas définie comme un test médical, grâce aux « techniques de raisonnement automatique » sur l'ontologie, un automate peut déduire cette connaissance. On peut de ce fait comprendre l'importance et les conséquences des descriptions ontologiques dans la compréhension des ressources du Web et l'automatisation de leurs traitements.

Ainsi, le Web des données permet aux données d'être traitées directement ou indirectement par des machines. Il permet de relier les données entre elles-mêmes dans un réseau global interconnecté d'information et rend ces liens exploitables par des machines. Il offre d'énormes

10 https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

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

12 http://geneontology.org/docs/go-enrichment-analysis/

13 https://www.w3.org/TR/turtle/

14 https://schema.org

15 Cette description peut être trouvée dans la documentation de schema.org : https://schema.org/ImagingTest

perspectives car il peut aider les utilisateurs à créer de nouvelles et inimaginables connaissances en combinant les informations sémantiques des ressources.
Le lecteur intéressé par le Web des données peut se référer à (Gandon, et al., 2012) et (Bizer, et al., 2012).

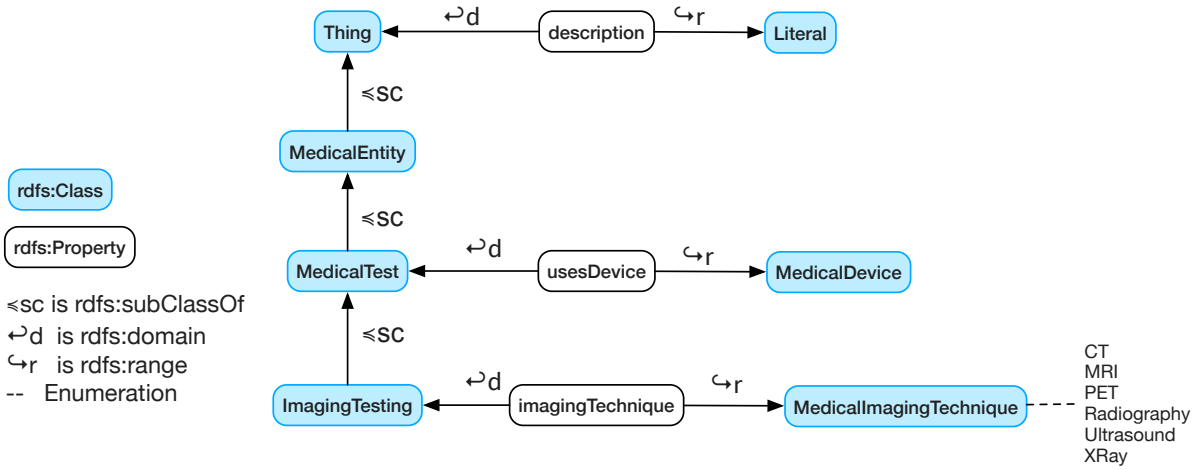

Figure 1. Fragment de l'ontologie qui permet de décrire des images médicales.

3. Sur la compatibilité de licences d'un point de vue informatique lors de la combinaison de ressources

Nous considérons la définition de compatibilité de licences suivante. Une licence est compatible avec une autre, si une ressource protégée par la première peut l'être aussi par la seconde sans enfreindre les conditions d'utilisation de la première. En général, nous avons remarqué que lorsqu'une licence est compatible avec une autre, la seconde est plus restrictive que la première.

Nous considérons qu'une licence est plus restrictive qu'une autre si la première autorise au plus les mêmes permissions (elle peut en autoriser moins mais pas plus) et impose au moins les mêmes obligations et interdictions (elle peut en imposer plus). Comme exemple, nous pouvons analyser trois licences Creative Commons synthétisées dans le tableau 1.

Licence	Permissions	Obligations	Interdictions
CC BY	Distribution, Reproduction, CommercialUse, DerivativeWorks	Notice, Attribution	
CC BY-NC	Distribution, Reproduction, DerivativeWorks	Notice, Attribution	CommercialUse
CC BY-NC-ND	Distribution, Reproduction	Notice, Attribution	CommercialUse, DerivativeWorks

Tableau 1. Description de 3 licences Creative Commons.

CC BY-NC-ND est plus restrictive que CC BY-NC qui est plus restrictive que CC BY. CC BY-NC-ND a moins de permissions, les mêmes obligations et plus d'interdictions. De ce fait, CC BY-NC-ND est conforme aux deux autres licences et elle peut protéger une ressource qui combine des ressources protégées par CC BY et CC BY-NC sans enfreindre les règles de ces licences. Une description des actions utilisées dans ce chapitre sont décrites dans le tableau 2.

Ici un actif est assimilé à une ressource. Ces actions et plein d'autres sont définies dans le vocabulaire ODRL[16].

Action	Description
Annotate	Pour ajouter des notes/commentaires explicatifs à l'Actif sans modifier l'Actif d'aucune autre manière.
Anonymize	Anonymiser tout ou partie de l'Actif.
Attribution	Le crédit doit être accordé au détenteur du droit d'auteur et/ou à l'auteur
CommercialUse	Exercice des droits à des fins commerciales.
DerivativeWorks	Pour créer un nouvel Actif dérivé à partir de cet Actif et pour éditer ou modifier le dérivé.
Distribution	Pour fournir l'Actif à des tiers.
Modify	Pour modifier le contenu existant de l'actif. Un nouvel actif n'est pas créé par cette action.
Notice	Les avis de droit d'auteur et de licence doivent être conservés intacts.
Read	Pour obtenir des données de l'actif.
Reproduction	Faire plusieurs copies.
ShareAlike	Les œuvres dérivées doivent être concédées sous les mêmes conditions ou des conditions compatibles que l'œuvre originale

Tableau 2. Description de quelques actions du vocabulaire ODRL.

D'un point de vue informatique, notre question de recherche était donc, « soit une licence quelconque, comment la positionner automatiquement dans un ensemble de licences selon leur compatibilité ? ». Le défi était de généraliser la relation d'ordre entre les licences tout en prenant en compte l'influence de la sémantique des actions des licences. Autrement dit, comment définir un programme informatique capable de gérer la compatibilité entre licences. L'objectif étant de savoir quelle licence peut protéger une ressource qui réutilise en tout ou partie d'autres ressources sous licence.

Dans la suite, la section 3.1 présente les travaux existants concernant les licences lisibles par une machine ainsi que les travaux actuels présentant une compatibilité de licences. La section 3.2 présente notre contribution appelé CaLi qui permet de définir des classifications de licences par compatibilité.

3.1. Travaux existants sur la compatibilité des licences

Les licences lisibles par une machine

Pour savoir si un ensemble de ressources peut être protégé par une licence, il est nécessaire de connaître la compatibilité des licences impliquées. La compatibilité automatique des licences nécessite des licences lisibles par une machine. La lisibilité par la machine est acquise grâce à l'annotation des licences par les outils tirés du Web des données. Les langages d'expression de licence tels que CC REL[17], ODRL[18] ou L4LOD[19] permettent une description fine des licences en RDF. De travaux tels que (Havur, et al., 2018) (Rodríguez Doncel, et al., 2014) proposent

[16] https://www.w3.org/TR/odrl-vocab/#actionConcepts

[17] https://creativecommons.org/ns

[18] https://www.w3.org/TR/odrl-model/

[19] https://ns.inria.fr/l4lod/

un ensemble de 60 licences bien connues décrites en CC REL et ODRL[20]. Creative Commons propose également ses licences en RDF[21].

Ainsi, il existe des licences lisibles par une machine et cohérentes décrites en RDF. Le fragment de code 2 montre la licence MIT en RDF[22]. Cette description permet à un automate d'identifier les actions qui sont obligées et permises. Dans MIT, Notice est une obligation (*Duty*). CommercialUse ou DerivativeWorks sont des autorisations (*Permission*). La première autorisation indique que la licence permet la réutilisation commerciale de la donnée. La deuxième autorisation permet la création d'œuvres composites en utilisant la donnée. Cette description permet aussi de renseigner l'URI de la page avec des termes en langage naturel de la licence (licensingTerms) ainsi que l'URI de la ressource qui est protégée par cette licence (target). Ces deux URI permettent d'identifier avec précision les ressources : à savoir aussi bien la licence que la ressource couverte par la licence.

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix cc: <http://creativecommons.org/ns#> .
@prefix l4lod: <http://ns.inria.fr/l4lod/v2/> .
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix odrs: <http://schema.theodi.org/odrs#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

:mit   rdf:type odrl:Policy ;
       rdfs:label "MIT" ;
       l4lod:licensingTerms <https://opensource.org/licenses/MIT> ;
       odrl:Duty
         [odrl:action
             cc:Notice ;];
       odrl:Permission
        [odrl:action
             cc:CommericalUse,
             cc:DerivativeWorks,
             cc:Distribution,
             cc:Reproduction,
             odrl:modify ;];
       odrl:target <http://cali.priloo.univ-nantes.fr/api/licenses> .
```

Fragment de code 2. Code RDF qui décrit la licence MIT.

Travaux existants sur la compatibilité entre licences

Établir la compatibilité entre deux licences n'est pas simple. Toutes les licences ne sont pas compatibles entre elles. De plus, une fois la compatibilité détectée, quelle licence doit protéger une ressource combinant plusieurs ressources ?
Creative Commons propose un tableau de compatibilité de 8 licences[23]. Les licences sont ordonnées par restrictivité, c'est-à-dire, en fonction des interdictions et obligations. La licence la plus restrictive doit être celle utilisée pour la ressource combinée car elle a toutes les

[20] https://old.datahub.io/dataset/rdflicense

[21] https://github.com/creativecommons/cc.licenserdf/tree/master/cc/licenserdf/licenses

[22] Ce code est disponible à ce URL http://cali.priloo.univ-nantes.fr/api/ld/licenses/659277524967313360415291774650613425561331568383952 76

[23] https://wiki.creativecommons.org/wiki/Wiki/cc_license_compatibility

interdictions et toutes les obligations des licences impliquées. Il y a des licences qui ne sont pas compatibles comme CC-ND et CC-NC-ND. Cela est dû au fait que la contrainte ND (Non Derivative), permet la distribution de la donnée mais pas sa modification : la donnée doit être distribuée tel que conçue originalement.

Dans le contexte du logiciel ouvert et gratuit (*Free Open Source Software*, FOSS), un graphe de compatibilité des licences les plus utilisées est proposé (Kapitsaki, et al., 2017). L'objectif est de détecter les violations de licences dans les progiciels[24]. Les auteurs considèrent qu'une licence est compatible avec une autre si le graphe contient un chemin de la première vers la seconde. Un logiciel combiné peut être protégé par la seconde licence, éventuellement avec des compléments de la première. Cependant, comme un tel graphe est construit à partir d'une interprétation manuelle de chaque licence, sa généralisation et son automatisation ne sont pas possibles[25].

L'inconvénient de ces approches est qu'il n'est pas possible d'ajouter automatiquement une nouvelle licence aux graphes de compatibilité des licences. C'est pourquoi, dans nos travaux, nous avons proposé CaLi, un modèle basé sur un ordonnancement de type treillis capable d'ordonner automatiquement et partiellement les licences.

En mathématiques, un ensemble partiellement ordonné formalise et généralise la notion intuitive d'ordre ou d'arrangement entre les éléments d'un ensemble. Un ensemble partiellement ordonné est un ensemble muni d'une relation d'ordre qui indique que pour certains couples d'éléments, l'un se place avant l'autre. Tous les éléments ne sont pas forcément comparables, contrairement au cas d'un ensemble muni d'un ordre total. Un ordre (ou ordre partiel) est une relation binaire sur un ensemble qui est réflexive, antisymétrique et transitive. Elle se note $\leq$[26].

Certaines conditions sur un ensemble partiellement ordonné conduisent à la définition d'un treillis. En mathématiques, un treillis est une structure algébrique d'un ensemble partiellement ordonné dans lequel chaque paire d'éléments admet une borne supérieure et une borne inférieure[27].

Dans le domaine du contrôle d'accès, (Denning, 1976) a proposé un modèle en treillis de flux d'information sécurisé. Ce modèle classifie des classes de sécurité avec les ressources associées. Une classe de sécurité est compatible avec une autre si le treillis contient un chemin de la première à la seconde. Ainsi, ce chemin représente le flux autorisé de ressources. Par exemple, une ressource protégée par la première classe peut circuler vers une ressource protégée par la seconde classe sans violer la première. Le treillis peut être généré automatiquement par une combinaison par paires de toutes les classes de sécurité.

Inspirés par le modèle de Denning, nous avons proposé le modèle CaLi pour ordonner des licences selon leur compatibilité. Ce modèle est indépendant de tout langage de description de licence, contexte d'application et type de ressource, de sorte qu'il peut être utilisé dans une grande variété de domaines.

[24] A noter qu'un progiciel est un logiciel commercialisé composé de nombreux logiciels.

[25] SPDX propose un jeu de données contenant des licences lisibles par machine. L'objectif est de décrire les statuts des licences en RDF. Malheureusement ces descriptions ne contiennent pas les actions des licences, donc un processus de compatibilité automatique ne peut pas s'y appliquer https://github.com/spdx/license-list-data/tree/master/rdfturtle.

[26] https://fr.wikipedia.org/wiki/Ensemble_partiellement_ordonné

[27] https://fr.wikipedia.org/wiki/Treillis_(ensemble_ordonné)

3.2. La compatibilité entre licences avec le modèle CaLi

Le modèle CaLi (de l'anglais *ClAssification of LIcenses*) (Moreau, 2020) (Moreau, et al., 2019)[28] est basé sur un ensemble partiellement ordonné sous forme de treillis permettant d'ordonner des licences. Il utilise une relation d'ordre basée sur la restrictivité entre les licences ainsi que des contraintes pour définir la compatibilité entre elles.

Le modèle CaLi

Dans une licence, les actions peuvent être réparties entre des *statuts* : permissions, obligations, interdictions, etc. Ainsi, le status permet de spécifier si une action est permise, obligée ou interdite. Une relation de compatibilité entre deux licences peut être établie uniquement si elles ont les mêmes actions. Les actions des licences peuvent être réparties différemment dans les statuts. Pour décider si une licence est moins restrictive qu'une autre, il est nécessaire de savoir si une action dans un statut est considérée comme moins restrictive que la même action dans un autre statut.

Dans le fragment de code 3, les licences CC BY et CC BY-NC décrites en RDF contiennent les mêmes actions. La seule différence est que l'action CommercialUse se trouve comme une permission dans CC BY et comme une interdiction dans CC BY-NC. Intuitivement, nous pouvons dire que CC BY-NC est la plus restrictive parmi les deux. Fort de cette intuition, que nous avons constaté sur les classifications de licences dont on a parlé précédemment, nous avons proposé d'ordonner les statuts par restrictivité sous la forme de treillis.

Treillis de restrictivité des statuts. Ainsi, un treillis de restrictivité de statuts défini (i) un ensemble des statuts (ex. : *permission, obligation, interdiction, recommandation, undefined*) d'une action dans une licence et (ii) la relation de restrictivité S entre ces statuts. L'ordre de restrictivité des trois statuts considérés dans l'exemple suivant est *Permission* ≤ *Duty* ≤ *Prohibition*. Autrement dit, nous considérons que les permissions sont moins restrictives que les obligations, lesquelles sont moins restrictives que les interdictions. Ce qui nous fait établir que CC BY est moins restrictive que CC BY-NC, c'est-à-dire, CC BY ≤ CC BY-NC.

```
…
:cc-by rdf:type odrl:Policy ;
rdfs:label "CC BY";
…
odrl:permission
     [odrl:action
          cc:Distribution,
          cc:Reproduction,
          cc:CommercialUse,
          cc:DerivativeWorks;];
odrl:duty
     [odrl:action
          cc:Notice,
          cc:Attribution;].
```

```
…
:cc-by-nc rdf:type odrl:Policy ;
rdfs:label "CC BY-NC";
…
odrl:permission
     [odrl:action
          cc:Distribution,
          cc:Reproduction,
          cc:DerivativeWorks;];
odrl:duty
     [odrl:action
          cc:Notice,
          cc:Attribution;];
odrl:prohibition
     [odrl:action
          cc:CommercialUse].
```

Fragment de code 3. Code RDF qui décrit deux licences Creative Commons : CC BY et CC BY-NC.

[28] Un résumé en français de cette publication a été publié (Moreau, et al., 2020)

Les statuts peuvent varier en fonction du contexte. Voici deux exemples :

- Dans un système de fichiers, les actions peuvent être soit autorisées soit interdites. Donc il n'y a que deux statuts et ils sont ordonnés dans le treillis de restrictivité suivant, où les permissions sont moins restrictives que les interdictions : *Permission* ≤ *Prohibition*.
- Inspirés par le langage permettant de décrire sémantiquement des licences ODRL, nous proposons le treillis de restrictivité de statuts *Undefined* ≤ *Permission* ≤ *Duty* ≤ *Prohibition*. Dans ODRL, les actions peuvent être autorisées, obligatoires, interdites ou non spécifiées (c'est-à-dire indéfinies). Le status *undefined* (en français indéfini) nous permet de positionner automatiquement deux licences même si certaines actions ne sont mentionnées au sein de l'une des licences Dans ce treillis, le statut indéfini est le moins restrictif et le statuts *Prohibition* (en français interdiction) est le plus restrictif.

Dans notre modèle, A est l'ensemble des actions qui peuvent exister dans une licence. Par exemple, les actions utilisées dans les licences Creative Commons sont au nombre de sept (voir le tableau 2 pour leur description) :

1. Distribution (diffusion, affichage public),
2. Reproduction (faire plusieurs copies),
3. DerivativeWorks (distribution d'œuvres dérivées),
4. CommercialUse (exercer des droits à des fins commerciales),
5. Notice (les droits d'auteur doivent être conservés intacts),
6. Attribution (le crédit doit être accordé à l'auteur),
7. ShareAlike (les œuvres dérivées doivent être protégées par des licences avec les mêmes conditions).

Le langage ODRL définit par contre soixante-douze actions[29]. Il comprend les actions utilisées par les licences de Creative Commons plus d'autres actions comme « *Annotate* » ou « *Anonymize* » (cf. tableau 2).

Treillis de restrictivité de licences. Dans CaLi, nous considérons l'ensemble de toutes les licences possibles avec (i) un treillis de restrictivité de statuts et (ii) un ensemble des actions pouvant exister dans une licence. Cet ensemble de licences peut-être ensuite ordonné (avec un ordre partiel) dans un *treillis de restrictivité de licences* : nous considérons que l_i est moins restrictif que l_j, noté $l_i \leq l_j$, si pour toutes les actions du modèle, $a \in A$, le statut de l'action a dans l_i est moins restrictif que celui de la même action dans l_j. Cet ordre est partiel car lorsque on ne peut pas déterminer que toutes les actions d'une licence sont moins restrictives que celles d'une autre licence, alors l'ordre de restrictivité entre ces deux licences ne peut pas être établi. Autrement dit, les licences ne sont pas comparables.

Nous avons remarqué que si deux licences ont une relation de restrictivité, il est possible qu'elles aient également une relation de compatibilité. Pour identifier la compatibilité entre les licences, dans CaLi nous affinons la relation de restrictivité avec des contraintes. Le but est de prendre en compte la sémantique des actions. De plus, les contraintes nous permettent de distinguer les licences valides des licences non valides. Nous considérons qu'une licence est non-valide si cette licence ne peut protéger aucune ressource.

[29] https://www.w3.org/TR/odrl-vocab/#actionConcepts

CaLi utilise deux types de contraintes, les contraintes de compatibilité et les contraintes de licences.

- **Contraintes de licence C_L**. Une contrainte de licence détermine si une licence est valide ou non. Comme exemple prenons l'action CommercialUse. Une licence est valide si elle interdit ou permet l'usage commercial. Cependant, une licence qui oblige l'usage commercial n'a pas tellement de sens et elle peut être considérée comme non valide. Dans un autre exemple, considérons l'action ShareAlike qui détermine qu'une ressource doit être protégée par la même licence ou une licence équivalente. On peut considérer que si une licence interdit ShareAlike, celle-ci est non valide. Ces deux exemples sont considérés également par Creative Commons car parmi leurs licences ces cas n'existent pas. D'une part, obliger à utiliser une donnée à des fins commerciales va à l'encontre du partage et de la philosophie des licences ouvertes sur le Web. D'autre part, interdire le partage de la donnée sous une licence équivalente s'oppose à l'économie générale des licences. Par ailleurs, une telle interdiction soulève des difficultés pratiques. En effet, cela signifie-t-il donc que la donnée doit être couverte par une licence plus restrictive ? ou bien qu'elle ne doit pas faire l'objet d'un partage ? L'imprécision d'une telle clause restrictive tout comme son décalage avec la « raison d'être » de telles licences, conduisent à qualifier la licence de non valide.

  Voici ces deux contraintes de licence sous forme de fonctions logiques ω_L :

$$\omega_{\mathcal{L}_1}(l_i) = \begin{cases} Faux & \text{si } l_i(cc : \textit{CommercialUse}) = \text{Duty}; \\ Vrai & \text{sinon.} \end{cases}$$

$$\omega_{\mathcal{L}_2}(l_i) = \begin{cases} Faux & \text{si } l_i(cc : \textit{ShareAlike}) = \text{Prohibition}; \\ Vrai & \text{sinon.} \end{cases}$$

- **Contraintes de compatibilité $C_{\rightarrow}$**. Une contrainte de compatibilité s'applique à deux licences qui ont une relation de restrictivité. Considérons deux exemples. Tout d'abord une licence qui interdit l'action de modifier la donnée. Dans l'esprit de la clause DerivativeWork, nous considérons que la distribution de la ressource modifiée, sous n'importe quelle licence, est interdite. En effet, si la modification de la donnée est interdite, la circulation de cette modification l'est tout autant. Ainsi, la contrainte de compatibilité peut établir qu'une relation de restrictivité $l_i \leq l_j$ (la première est moins restrictive que la seconde) se transforme en compatibilité à condition que la première n'interdit pas l'action de modifier. Dans un deuxième exemple, considérons l'action ShareAlike qui exige que les œuvres dérivées soient protégées par des licences avec les mêmes conditions (pas plus restrictives ni plus permissives). Donc une contrainte peut établir qu'une relation de restrictivité $l_i \leq l_j$ se transforme en compatibilité à condition que la première n'ait pas l'action de ShareAlike comme obligation.

  Voici ces deux contraintes de compatibilité sous forme de fonctions logiques $\omega_{\rightarrow}$:

$$\omega_{\rightarrow_1}(l_i, l_j) = \begin{cases} Faux & \text{si } l_i(cc : \textit{ShareAlike}) = \text{Duty}; \\ Vrai & \text{sinon.} \end{cases}$$

$$\omega_{\rightarrow_2}(l_i, l_j) = \begin{cases} Faux & \text{si } l_i(cc : \textit{DerivativeWorks}) = \text{Prohibition}; \\ Vrai & \text{sinon.} \end{cases}$$

Le modèle CaLi. Ainsi, un modèle CaLi est un quadruplet $\langle A, LS, C_L, C_{\rightarrow}\rangle$[30] capable d'ordonner partiellement des licences, tel que :

1. A est un ensemble d'actions (ex. : Read, Modify, Distribution, etc.) ;
2. LS est un treillis de restrictivité des statuts définissant (i) l'ensemble des statuts possibles (e.g., permission, obligation, interdiction, etc.) d'une action dans une licence et (ii) la relation de restrictivité entre ces statuts ;
3. $C_{\rightarrow}$ est un ensemble de contraintes sur la compatibilité permettant d'identifier les relations de restrictivité entre deux licences qui sont aussi des relations de compatibilité ;
4. Enfin, C_L est un ensemble de contraintes sur les licences permettant d'identifier les licences qui ne sont pas valides.

Un ordre CaLi pour les licences Creative Commons

Nous avons validé notre modèle de plusieurs manières. Dans un premier temps, nous avons défini un ordre CaLi pour les licences de Creative Commons nommé CC_CaLi. L'objectif était de constater que notre approche produit la même compatibilité entre les licences que celle déterminée par Creative Commons[31]. En suivant la logique de Creative Commons, nous avons utilisé les sept actions du langage CC Rel et le treillis de restrictivité de statuts : *Permission* ≤ *Duty* ≤ *Prohibition* (en français Permission ≤ Obligation ≤ Interdiction). Les contraintes de cet ordre sont celles introduites précédemment (ω_{L1}, ω_{L2}, $\omega_{\rightarrow 1}$, et $\omega_{\rightarrow 2}$). Mathématiquement, le nombre de licences qu'il est possible d'obtenir en distribuant les actions sur les statuts dans un ordre CaLi est exponentiel, à savoir, $|LS|^{|A|}$. Cela correspond au nombre de statuts à l'exposant qui correspond au nombre d'actions. Donc, pour l'ordre Creative Commons cela fait 3^7. Ce qui donne 2187 licences possibles. Cependant, le nombre de licences valides est de 972. Ceci est dû aux contraintes. En effet, il est possible d'avoir 5 actions dans n'importe quel statut (parmi les trois considérés) et 2 actions (CommercialUse et ShareAlike) dans uniquement 2 statuts, ce qui fait $3^5 * 2^2 = 972$ licences.

Ainsi, CC_CaLi est un modèle CaLi tel que :

1. A est l'ensemble de 7 actions {Distribution, Reproduction, DerivativeWorks, CommercialUse, Notice, Attribution, ShareAlike} ;
2. LS est le treillis de restrictivité des statuts : *Permission* ≤ *Duty* ≤ *Prohibition* ;
3. C_L est un ensemble de contraintes sur les actions et les status issue du vocabulaire CC : $C_L = \{\omega_{L1}, \omega_{L2}\}$ (cf. définition de contraintes de licence).
4. $C_{\rightarrow}$ est les ensembles de contraintes sur la compatibilité entre licences issue du vocabulaire CC : $C_{\rightarrow} = \{\omega_{\rightarrow 1}$, et $\omega_{\rightarrow 2}\}$ (cf. définition des contraintes de compatibilité).

La figure 2 montre deux sous-graphes de CC_CaLi. La partie gauche (a) de la figure montre uniquement les sept licences officielles de Creative Commons tandis que la partie droite (b) montre en plus d'autres licences valides. Grâce à $\omega_{\rightarrow 1}$, la relation de restrictivité entre CC BY-SA et CC BY-NC-SA n'est pas identifiée comme une relation de compatibilité. Et grâce à $\omega_{\rightarrow 2}$, la relation de restrictivité entre CC BY-ND et CC BY-NC-ND n'est pas identifiée comme une

[30] A pour actions, LS pour *lattice of status* (treillis de statuts), C_L pour contraintes sur les licences et $C_{\rightarrow}$ pour contraintes sur la relation de restrictivité.

[31] https://commons.wikimedia.org/wiki/File:CC_License_Compatibility_Chart.png

relation de compatibilité non plus. La relation de compatibilité du sous-graphe de gauche est conforme au graphe de compatibilité qu'il est possible de générer à partir du tableau de compatibilité des licences Creative Commons.

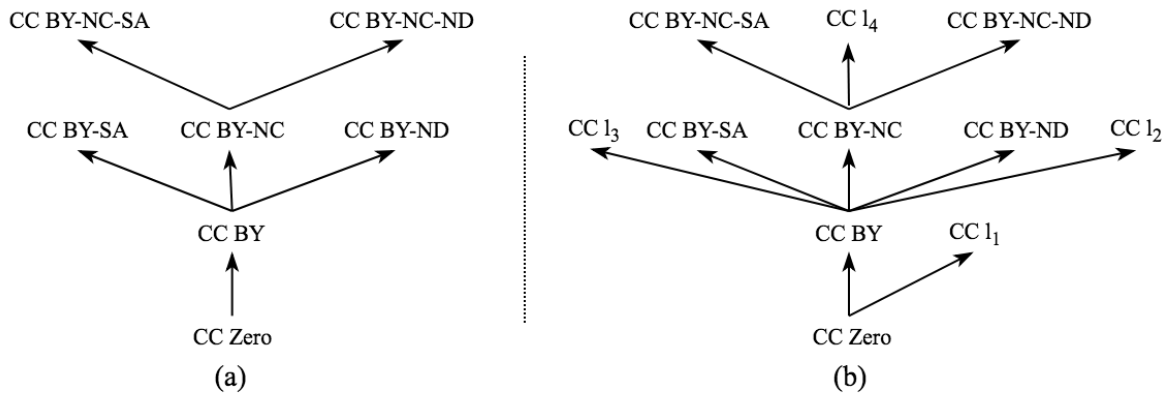

Figure 2. Deux sous-graphes de l'ordre CaLi pour les licences de Creative Commons : (a) montre les sept licences officielles de Creative Commons, (b) montre en plus d'autres licences valides.

4. Illustration de l'utilité du modèle CaLi

Cette section présente trois cas d'utilisation du modèle CaLi. La section 4.1 présente un outil pédagogique pour guider à la conception de classifications CaLi. La section 4.2 présente un moteur de recherche de jeux de données capable de trouver les ressources en base à la compatibilité de licences. Enfin, la section 4.3 introduit une stratégie utilisable dans un moteur de requêtes fédérées SPARQL qui souhaite préserver les licences.

4.1. Outil Web pour concevoir un graphe de compatibilité de licences avec CaLi

Nous avons conçu un outil Web a but pédagogique afin d'aider à comprendre le modèle CaLi (Moreau, et al., 2021). Le but est de permettre aux utilisateurs d'utiliser le modèle CaLi pour concevoir différents graphes de compatibilité de licences selon leur perception de la notion de restrictivité et en y ajoutant les contraintes adaptées au contexte sur lequel les licences seront utilisées. L'outil Web est disponible ici https://saas.ls2n.fr/cali/.

L'outil permet de définir des modèles CaLi où :

1. A est par défaut l'ensemble de 72 actions du langage ODRL.
2. Le treillis de restrictivité de statuts LS peut-être personalisé. Six statuts existent dans notre outil : indéfini (*Undefined*), permission (*permission*), dispensés (*Dispensation*), recommandation (*Recommendation*), obligation (*Duty*), et interdiction (*Prohibition*). L'utilisateur peut définir l'ordre de restrictivité souhaité. La seule contrainte est que le statuts « indéfini » sera toujours présent, cela afin d'y attribuer automatiquement les actions qui ne seront pas distribuées sur les autres statuts dans les licences. De plus, « indéfini » sera toujours le statut le moins restrictif.
3. L'utilisateur peut définir les contraintes pour la validité des licences C_L. Une licence sera valide uniquement si toutes les contraintes sont respectées. Chaque contrainte sera définie comme un ensemble de conditions. Une condition détermine si une action doit exister ou non dans un statut. Par exemple, la condition (CommercialUse $\notin$ Duty) signifie qu'une

licence valide ne doit pas avoir l'action CommercialUse en tant qu'obligation. La condition (ShareAlike ∉ Prohibition) signifie qu'une licence valide ne doit pas interdire l'action ShareAlike. Ces deux contraintes sont proposées par notre outil.
Les contraintes conjonctives doivent être définies en plusieurs contraintes. Par exemple, la contrainte définissant qu'une licence pour être valide ne doit pas à la fois permettre la modification et interdire la lecture comporte deux conditions : Read ∉ Prohibition et Modify ∉ Permission. Pour vérifier les deux conditions, elles doivent être définies dans deux contraintes, une condition par contrainte.

4. L'utilisateur peut également définir les contraintes de compatibilité entre licences $C_{\rightarrow}$. Ce type de contraintes concerne deux licences avec une relation de restrictivité. Par exemple, considérons l'action ShareAlike qui exige que la distribution de ressources dérivées se fasse uniquement sous les mêmes conditions. La contrainte de compatibilité (ShareAlike ∉ Duty) ∉ l_i signifie qu'une licence est compatible avec une autre si l'action ShareAlike n'est pas une obligation dans la première. Dans un autre exemple, la contrainte de compatibilité (DerivativeWorks ∉ Prohibition) ∉ l_i signifie qu'une licence est compatible avec une autre si la première n'interdit pas la distribution d'une ressource dérivée. Ces deux contraintes sont proposées par notre outil.

Une fois le modèle de compatibilité de licences établi, l'utilisateur peut créer des licences. Pour cela, il peut distribuer les actions d'ODRL sur les statuts. Les contraintes seront vérifiées automatiquement. L'outil propose les sept licences de Creative Commons, à savoir, CC Zéro, CC BY, CC BY-NC, CC BY-ND, CC BY-SA, CC BY-NC-ND et CC BY-NC-SA.

Enfin, l'utilisateur peut demander à l'outil Web de créer l'ordre de compatibilité des licences. L'outil génère un graphe de compatibilité qui représente l'ordre de compatibilité des licences. Le graphe peut alors être téléchargé en format RDF (syntaxe Turtle). Ce graphe peut être utilisé dans un outil ce qui permettrai de vérifier la compatibilité entre les licences du graphe (elles seraient les licences prise en compte par l'outil). L'outil introduit par la suite en est un exemple.

4.2. Outil Web de recherche de jeux de données basée sur les licences

Nous avons également conçu un outil Web de démonstration (Moreau, et al., 2019) qui permet la recherche de jeux de données en base à la compatibilité de licences. Il est disponible à cette adresse : http://cali.priloo.univ-nantes.fr/ld/graph.

Le graphe de compatibilité de licences sur lequel se fonde notre outil Web est celui de la figure 3. Le graphe reprend quinze licences ouvertes (nœuds bleus sur le graphe). Les flèches du graphe indiquent les relations de compatibilité. Le graphe permet de voir par exemple que la licence intitulée MIT est compatible avec toutes les licences se trouvant au-dessus d'elle (et qui sont plus restrictives). Pour rappel, la notion de compatibilité de notre modèle est transitive (ex. : si MIT est compatible avec CC BY, et CC BY est compatible avec CC BY-NC, alors transitivement MIT est compatible avec CC BY-NC), réflexive (toute licence est compatible avec elle-même) et asymétrique (ex. : MIT est compatible avec CC BY mais cette dernière n'est pas compatible avec MIT). Si deux licences sont compatibles l'une avec l'autre et vice-versa alors nous considérons qu'elles sont équivalentes. De ce fait, elles sont situées dans le même nœud du graphe de compatibilité.

Dans notre modèle, la notion de conformité se dérive de la notion de compatibilité. Si la licence MIT est compatible avec CC BY, alors CC BY est conforme à MIT. La relation de conformité est également transitive, réflexive et asymétrique. Ainsi, la licence MIT est conforme avec elle-même et avec toutes les licences se trouvant au-dessous d'elle (et qui sont moins restrictives), à savoir ODC-PDDL, Public-domain et CC-Ze.

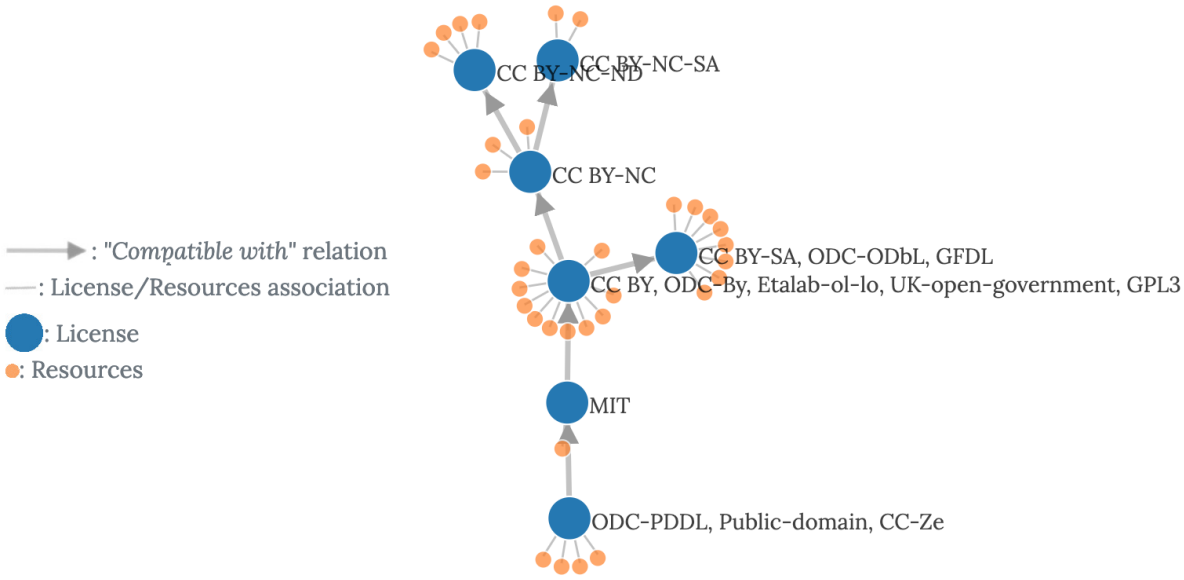

Figure 3; Graphe de compatibilité de licences et les jeux de données associés.

Pour illustrer l'utilité de ce graphe de compatibilité nous avons associé les licences à quarante jeux de données (points orange sur le graphe). Le graphe permet alors de visualiser à la fois la compatibilité et la conformité entre les licences mais aussi les usages pouvant être faits des jeux de données en fonction des licences qui leurs sont attachées. De manière générale, deux jeux de données sous licence peuvent être combinés s'il existe une licence conforme aux deux licences impliquées. Autrement dit, deux jeux de données peuvent être combinés s'il existe une licence avec laquelle les deux licences impliquées sont compatibles. Le jeu de données combiné sera protégé par la licence la plus restrictive. Ainsi, l'outil permet de répondre aux questions suivantes.

- *Trouver les jeux de données dont leur licence leur permet d'être protégés par une licence donnée.* Autrement dit, les jeux de données dont leurs licences sont compatibles avec une licence donnée. Si la licence donnée est CC BY-NC, la réponse sera composée de tous les jeux de données protégés par les licences se trouvant au-dessous de CC BY-NC dans le graphe (ex. : CC By et MIT mais pas CC BY-SA). Cela veut dire que les jeux de données protégés par ces licences peuvent être combinés et la licence qui peut protéger la combinaison est CC BY-NC.
- *Trouver les jeux de données dont leurs licences peuvent protéger ceux qui sont attachés à une licence donnée.* Autrement dit, les jeux de données dont leurs licences sont conformes à une licence donnée. Si la licence donnée est CC BY-NC, la réponse sera composée de tous les jeux de données protégés par les licences se trouvant au-dessus de CC BY-NC dans le graphe (ex. : CC BY-NC-ND et CC BY-NC-SA). Cela veut dire que les jeux de données protégés par CC BY-NC peuvent être combinés avec ceux attachés à CC BY-NC-SA (respectivement CC BY-NC-ND) et la licence qui peut protéger la combinaison est CC BY-NC-SA (respectivement CC BY-NC-ND). Il faut remarquer que la

combinaison des jeux de données attachés aux deux licences CC BY-NC-ND et CC BY_NC_SA ne peut pas être sous licence (car il n'y a pas de licence conforme à ces deux licences), donc ces jeux de données ne doivent pas être combinés.

Nous pouvons voir, grâce au graphe, que par exemple la licence intitulée Public-domain est parmi les moins restrictives. Elle est dès lors compatible avec toutes les licences du graphe de compatibilité. Le graphe souligne également que la licence nommée CC BY-SA est compatible uniquement avec des licences équivalentes à savoir ODC-ODbL et GFDL.

L'image de la figure 4 montre l'interface Web de notre outil de recherche. Les champs du formulaire de recherche sont remplis et indiquent que l'utilisateur recherche des « images » qui sont protégées par des licences « compatibles avec » la licence nommée « ODC-By ». La réponse à cette recherche contiendra des ressources protégées par ODC-By mais également des ressources protégées par des licences équivalentes à ODC-By et des licences moins restrictives et compatibles à ODC-By comme la licence MIT ou CC-Ze.

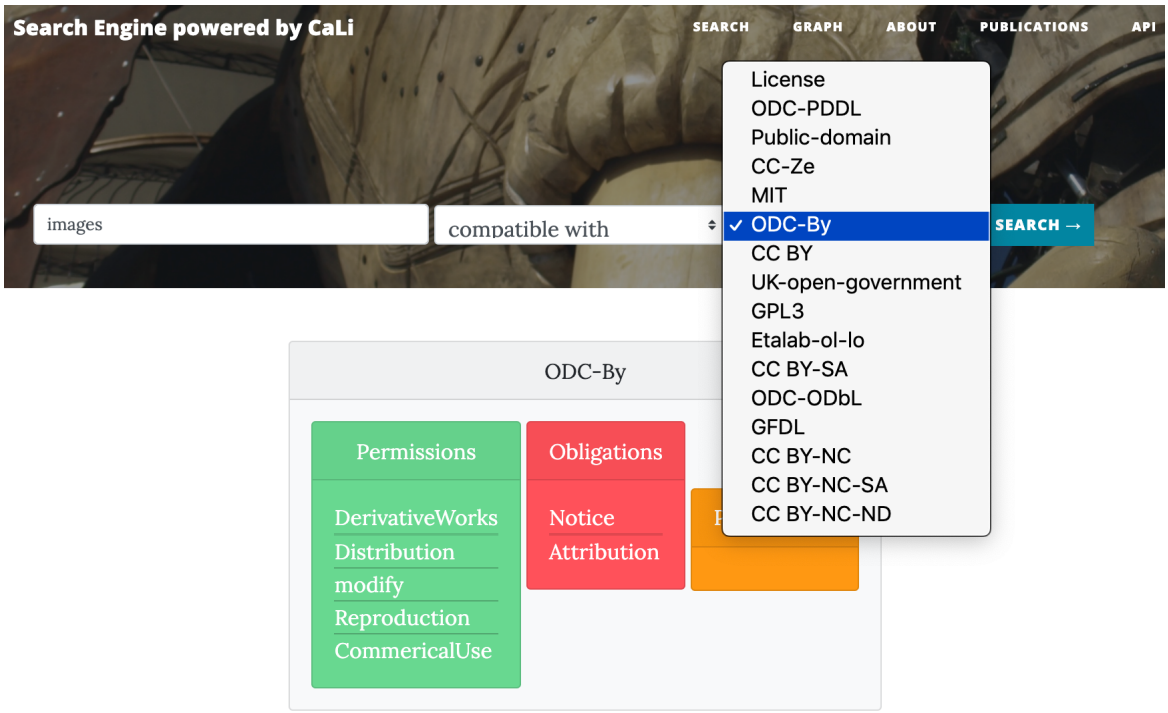

Figure 4. Exemple de recherche de jeux de données de l'outil de recherche Web basé sur la compatibilité de licences.

Notre dernier cas d'utilisation a pour but d'empêcher la combinaison de ressources lorsque les licences impliquées ne permettent pas d'obtenir une combinaison qui peut être sous licence.

4.3. Vérification de licences lors de l'exécution de « requêtes fédérées » sur le Web des Données

SPARQL[32] (*SPARQL Protocol and RDF Query Language*), est le langage de requêtes recommandé par le W3C pour l'interrogation de données en RDF. Une « requête fédérée » est une requête qui peut récupérer des informations à partir de plusieurs sources de données RDF réparties sur le Web des données. Les « moteurs de requêtes fédérées » implémentent les requêtes fédérées

[32] https://www.w3.org/TR/sparql11-query/

et peuvent être utilisés dans la conception d'outils qui ont besoin de combiner simultanément des sources de données du Web des données.

Lorsque plusieurs jeux de données sous licence participent à l'évaluation d'une requête fédérée, le résultat de la requête doit être protégé par une licence conforme aux licences des données impliquées. Si une telle licence n'existe pas, le résultat de la requête ne peut pas être protégé par une licence et, par conséquent, ne doit pas être réutilisé ni publié. Notez qu'avoir le droit d'interroger plusieurs ensembles de données individuellement ne signifie pas avoir le droit d'exécuter une requête fédérée qui croise ces ensembles de données afin de réutiliser le résultat de la requête. Ainsi, dans cette section nous montrons l'utilité d'une classification CaLi avec la vérification de licences au sein d'un moteur de requêtes fédérées sur de données RDF.

La stratégie de traitement de requêtes fédérées que nous proposons se nomme FLiQue (Moreau & Serrano Alvarado, 2021). FLiQue permet de détecter et de prévenir les conflits de licences. L'objectif est de permettre aux moteurs de requêtes fédérées de produire des résultats de requêtes réutilisables car protégés par des licences.

Une solution à l'incompatibilité des licences consiste à négocier avec les fournisseurs de données afin de modifier les termes d'une licence en conflit. Cependant, la négociation prend du temps et n'est pas toujours possible. Une deuxième solution consiste à écarter les jeux de données protégés par des licences en conflit. Cependant, cette solution peut conduire à un résultat vide. Pour faire face à ce problème, nous utilisons des techniques d'assouplissement de contraintes des requêtes SPARQL basées sur des ontologies.

Les techniques d'assouplissement de requêtes sont utilisées lorsqu'une requête ne rend pas de résultat ou le nombre de résultats est insuffisant vis-à-vis d'une application donnée. Par exemple, en reprenant l'ontologie de notre exemple sur les images médicales (cf. figure 1), supposons que la requête qui demande des ressources de type ImagingTesting ne retourne aucun résultat. Alors, à défaut de trouver des ressources de ce type, la requête pourrait être assouplie afin de demander des ressources de la super-classe de ImagingTesting, à savoir MedicalTest. Ce processus est répété autant de fois que nécessaire jusqu'à obtenir le nombre de ressources souhaité.

L'assouplissement de requêtes devient complexe lorsque le nombre de contraintes d'une requête grandit. Le nombre de possibilités d'assouplissement augmente de manière exponentielle en fonction du nombre de contraintes qu'il est possible d'assouplir ainsi que de la taille et la richesse de l'ontologie. En informatique, connaître toutes les possibilités de requêtes assouplies est un problème difficile. Donc, parmi les nombreuses possibilités qui peuvent s'offrir à l'assouplissement, lesquelles choisir en priorité ? Diverses techniques sont possibles afin de limiter le temps d'exécution des requêtes assouplies et la diminution de la qualité des résultats.

La requête SPARQL du fragment de code 4, cherche de ressources sous 4 contraintes : (1) que les ressources soient de type étudiant (prédicat *`type`* et objet `Student`), (2) que les étudiants soient inscrits à au moins un cours (prédicat *`enrolledIn`*), (3) que le cours soit de l'Université de Nantes (prédicat *`heldAt`*, et object `UniversityOfNantes`) et enfin (4) que le cours soit donné par l'enseignant Ben (sujet `Ben` et prédicat *`teaches`*).

```
SELECT * WHERE {
       ?student    rdf:type        ex:Student           . #1@{D3} CC BY-NC
       ?student    ex:enrolledIn   ?course              . #2@{D3} CC BY-NC
       ?course     ex:heldAt       ex:UniversityOfNantes . #3@{D1} CC BY
       ex:Ben      ex:teaches      ?course              . #4@{D2} CC BY-SA
}
```

Fragment de code 4. Requête en SPARQL.

Nous allons considérer que cette requête peut être évaluée sur les jeux de données sous licence de la figure 5. Avec les annotations à la fin de chaque contrainte de la requête, nous spécifions dans quelle source de données chaque condition peut être satisfaite. On considère que la source de données D3 contient des étudiants et leurs inscriptions (elle satisfait `#1` et `#2`). La source D1 contient des cours de l'Université de Nantes (elle satisfait `#3`) et la source D2 contient des enseignants (elle satisfait `#4`). D'après leurs licences, les trois sources de données considérées ne peuvent pas se combiner afin de produire un résultat sous licence. En effet, la licence de la source D2 étant CC BY-SA elle n'est pas compatible avec une autre licence (pour rappel, ShareAlike exige de publier la ressource sous les mêmes termes).

Afin de trouver une solution à l'évaluation de la requête, FLiQue analyse séparément les jeux de données en les partitionnant en sous-fédérations avec des licences compatibles. Supposons ici deux fédérations, F1 et F2. F1 composée de D1 et D2, et F2 composée de D1 et D3 (voir les ovales en pointillés sur la figure 5). Si au moins une fédération satisfait toutes les contraintes de la requête alors FLiQue exécute la requête et envoie le résultat avec la licence capable de le protéger. Cependant, si aucune fédération ne peut satisfaire toutes les contraintes de la requête, alors FLiQue démarre le processus d'assouplissement de contraintes.

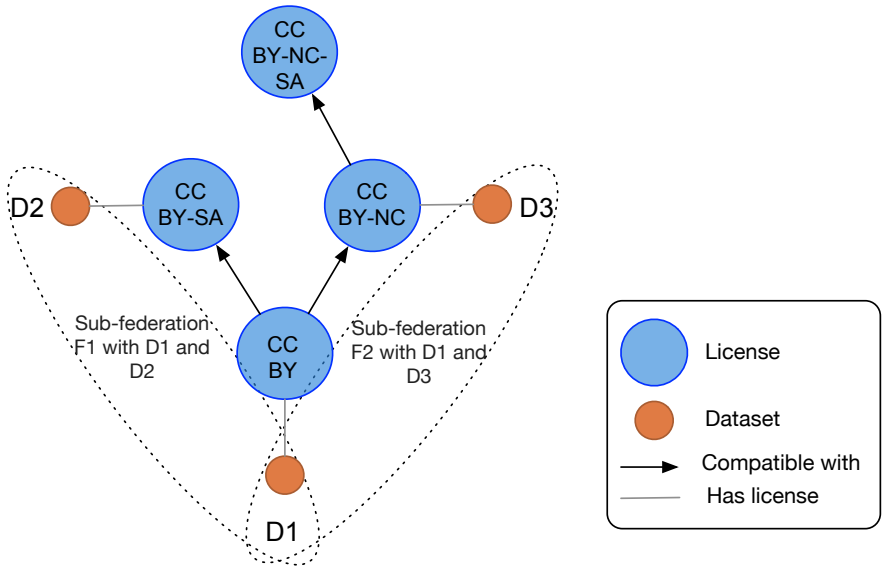

Figure 5.Exemple de sources de données avec les licences qui le protègent.

Le principe d'assouplissement utilisé dans FLiQue est simple. Considérez l'ontologie de la figure 6 qui complète notre exemple. En se basant sur l'ontologie, les concepts dans la requête initiale peuvent être remplacés par leurs concepts parents, à savoir, une classe par sa super-classe (ex. : le type `Student` par `Person`) ou une propriété par sa super-propriété (ex. :

teaches par *attends*). De plus, il est possible de remplacer un littéral par une variable (ex. : l'enseignant `Ben` par n'importe quel enseignant `?x`).

Le nombre de possibilités d'assouplissement est exponentiel. Il dépend du nombre de contraintes de la requête et de la richesse de l'ontologie. Pour trouver efficacement les requêtes fédérées assouplies les plus pertinentes, nous organisons les requêtes de la plus spécifique à la plus générale de la requête originale du point de vue ontologique. De plus, pour éviter de nous intéresser à de requêtes qui ne retournent pas de résultats supplémentaires, nous utilisons des statistiques sur les jeux de données, par exemple, le nombre de ressources par classe et le nombre de triplets par propriété.

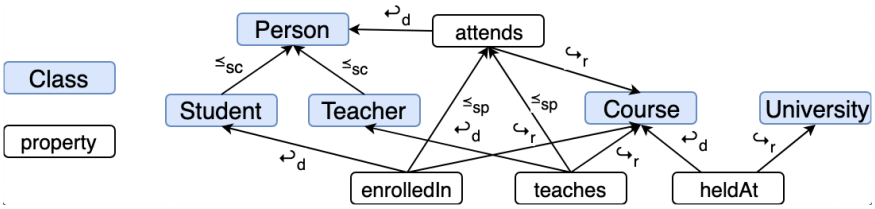

Figure 6. Exemple d'ontologie.

Nous avons conçu un prototype qui a démontré la faisabilité de FLiQue. L'évaluation expérimentale que nous avons réalisée montre que cette approche garantit la conformité des licences avec un coût d'exécution limité.

À notre connaissance, FLiQue est le premier travail qui utilise l'assouplissement de requêtes dans un environnement distribué. FLiQue est une étape vers la facilitation et l'encouragement de la publication ainsi que la réutilisation de ressources sous licence dans le Web des données. FLiQue n'est pas une stratégie de contrôle d'accès aux données. Au lieu de cela, il permet aux utilisateurs de données bien intentionnés de respecter les licences des ensembles de données impliqués dans une requête fédérée. L'inconvénient de l'assouplissement de contraintes réside dans la dégradation de la qualité des résultats. Cet inconvénient, inhérent à toute technique d'assouplissement de contraintes, est un problème encore ouvert à ce jour.

5. Conclusion

Nous avons présenté nos travaux sur la comptabilité automatique entre licences. Notre modèle de compatibilité n'a pas l'intention de fournir un avis juridique mais il permet d'exclure les licences qui contreviendraient à une licence particulière.

Notre modèle utilise la notion de restrictivité comme base pour définir la compatibilité et la conformité entre les licences. Cette stratégie fonctionne la plupart du temps, comme nous l'avons montré dans les nombreux exemples. Mais elle présente certaines limites. En particulier, CaLi n'a pas vocation à définir la compatibilité de deux licences si elle n'est pas cohérente avec leur relation de restrictivité. À titre d'exemple, considérons deux versions de licences MPL. La version 2.0 assouplit certaines obligations par rapport à la version 1.1. Ainsi, MPL-2.0 est moins restrictive que MPL-1.1. Avec les contraintes de CaLi, il est seulement possible de dire que MPL-2.0 est compatible avec MPL-1.1. Mais dans les textes légaux, il est dit le contraire, c'est-à-dire que MPL-1.1 est compatible avec MPL-2.0. Ainsi, les particularités dans l'usage de la compatibilité des licences, la granularité de la sémantisation des licences et la compréhension

de certaines actions (comme ShareAlike) sont les principales raisons de la différence entre les classifications CaLi et d'autres classifications. C'est le cas, par exemple, de notre graphe de compatibilité consacré aux licences de la plateforme GitHub[33] et du graphe présenté dans (Kapitsaki, et al., 2017).

Une perspective de ce travail est de prendre en compte d'autres aspects des licences liés aux contextes d'utilisation comme la juridiction, les dates de réutilisation, etc. Une autre perspective est d'analyser comment deux ordres de compatibilité peuvent être comparés. Autrement dit, étant donné deux ordres de CaLi, s'il y a un alignement entre leurs vocabulaires et que leurs treillis de restrictivité des statuts sont homomorphes alors trouver une fonction pour passer d'un ordre de CaLi à un autre.

Remerciements.
Nous remercions Margo Bernelin (CR) et Sonia Desmoulin-Canselier (MC), du laboratoire de Droit et Changement Social, pour nos richissimes discutions sur les aspects juridiques des licences.

Nous remercions également le projet interdisciplinaire DataSanté (financé par la Région Pays de la Loire et Nantes Université) pour nous avoir offert la possibilité de rencontrer de chercheurs des plusieurs disciplines grâce aux nombreux séminaires et conférences organisés.

Nous remercions enfin le dispositif CIFRE (Conventions Industrielles de Formation par la Recherche), l'entreprise Opendatasoft et l'ANRT (Association Nationale de la Recherche et de la Technologie) qui ont financé les travaux de thèse de Benjamin Moreau.

Bibliographie

Bizer, C. et al., 2012. *Web sémantique : méthodes et outils pour le Web de données.* France: Pearson Éducation .

Denning, D. E., 1976. A lattice model of secure information flow. *Communications of the ACM,* Volume 19, no. 5, p. 236–243.

Gandon, F., Corby, O. & Faron-Zucker, C., 2012. *Le web sémantique: Comment lier les données et les schémas sur le web?.* s.l.:Dunod.

Havur, G. et al., 2018. *DALICC: A Framework for Publishing and Consuming Data Assets Legally..* s.l., International Conference on Semantic Systems (SEMANTICS), Poster&Demo sessions.

Kapitsaki, G., Kramer, F. & Tselikas, N., 2017. Automating the License Compatibility Process in Open Source Software With SPDX. *Journal of Systems and Software,* Volume 131.

Moreau, B., 2020. *Facilitating Reuse on the Web of Data. PhD Thesis.* Nantes (Pays de la Loire): Université de Nantes.

[33] http://cali.priloo.univ-nantes.fr/rep/graph

Moreau, B., Confais, B., Vintache, D. & Serrano Alvarado, P., 2021. *A Tool to Define Step by Step the Compatibility of Licenses as Partial Orders with CaLi,* Nantes, France: Nantes University. Research report.

Moreau, B. & Serrano Alvarado, P., 2021. Ensuring License Compliance in Linked Data with Query Relaxation. *Transactions on Large-Scale Data- and Knowledge-Centered Systems,* Volume Springer. Lecture Notes in Computer Science 12920, pp. 97-129.

Moreau, B., Serrano Alvarado, P., Perrin, M. & Demontils, E., 2020. *Modéliser la Compatibilité entre les Licences.* Angers, France, 31es Journées francophones d'Ingénierie des Connaissances (IC), pp. 149-154.

Moreau, B., Serrano Alvarado, P., Perrin, M. & Desmontils, E., 2019. *A License-Based Search Engine.* Portorož, Slovenia, The Semantic Web: 16th International Conference, ESWC 2019. Poster and demo sessions. Springer. Lecture Notes in Computer Science vol 11503, pp. 130-135.

Moreau, B., Serrano Alvarado, P., Perrin, M. & Desmontils, E., 2019. *Modelling the Compatibility of Licenses.* Portorož, Slovenia, The Semantic Web: 16th International Conference, ESWC 2019. Springer Lecture Notes in Computer Science, vol 11503., pp. 255-269.

Rodríguez Doncel, V., Gómez-Pérez, A. & Villata, S., 2014. *A Dataset of RDF Licenses.* s.l., Legal Knowledge and Information Systems Conference (ICLKIS).