



Modéliser la Compatibilité Entre les Licences

Benjamin Moreau, Patricia Serrano-Alvarado, Matthieu Perrin, Emmanuel Desmontils

► To cite this version:

Benjamin Moreau, Patricia Serrano-Alvarado, Matthieu Perrin, Emmanuel Desmontils. Modéliser la Compatibilité Entre les Licences. 31es Journées francophones d'Ingénierie des Connaissances (IC), Sébastien Ferré, Jun 2020, Angers, France. hal-02877913

HAL Id: hal-02877913

<https://hal.science/hal-02877913>

Submitted on 22 Jun 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.



Distributed under a Creative Commons Attribution 4.0 International License

Modéliser la Compatibilité Entre les Licences

Benjamin Moreau^{1,2}, Patricia Serrano-Alvarado², Matthieu Perrin²,
Emmanuel Desmontils²

¹ Opendatasoft

{Name.Lastname}@opendatasoft.com

² UMR6004 – Université de Nantes, Nantes, France

{Name.Lastname}@opendatasoft.com

Résumé : Le web facilite la création de ressources par combinaison (code source, services web, données liées, etc.). Pour un producteur de données, choisir une licence adaptée qui protégera sa combinaison de ressources n'est pas facile. En effet, la licence doit être assez restrictive pour se conformer aux licences des ressources combinées, mais assez permissive pour ne pas limiter son usage. Trouver le bon compromis entre restrictivité et usage n'est pas évident. La possibilité d'ordonner automatiquement les licences selon leur compatibilité faciliterait ce choix. Nous proposons donc *CaLi*, un modèle capable d'ordonner partiellement les licences. Notre approche utilise la restrictivité entre les licences pour définir la compatibilité. Notre travail a pour objectif de faciliter et encourager la publication et la réutilisation de ressources protégées par des licences sur le Web.

Mots-clés : Licences, Combinaison, Compatibilité, Web des Données, RDF

1 Introduction

Afin de faciliter la réutilisation des ressources sur le Web, les producteurs de ressources doivent définir leur licence avant de les partager ou de les publier (Seneviratne *et al.*, 2009). Une licence indique précisément les conditions d'utilisation de la ressource, i.e., quelles actions sont autorisées, obligatoires, interdites.

Creative Commons ¹ (CC) est une organisation ayant pour objectif de faciliter la diffusion et le partage d'informations. Elle propose un ensemble de licences ² fréquemment utilisées sur le web. Cependant, un grand nombre d'autres licences sont mis à disposition et de nombreux producteurs de données créent leur propre licences.

Pour un producteur de données, choisir une licence adaptée qui protégera sa ressource n'est pas facile. En effet, les licences des ressources combinées doivent être compatibles avec la licence, mais aussi permettre la réutilisation de la ressource. Le risque est de choisir une licence trop restrictive qui empêchera la réutilisation de la ressource ou, au contraire, de choisir une licence trop permissive qui ne la protégera pas assez.

Nous pensons qu'un modèle capable d'ordonner les licences selon leur compatibilité faciliterait ce choix. Notre définition de la compatibilité est la suivante : *Une licence l_i est compatible avec une licence l_j si une ressource protégée par l_i peut être protégée par l_j sans violer les conditions d'utilisation de l_i .* Si une licence l_i est compatible avec une licence l_j alors, les ressources protégées par l_i sont réutilisables avec des ressources protégées par l_j . En général, quand une licence l_i est compatible avec une licence l_j , l_j est plus (ou autant) restrictive que l_i . Nous considérons qu'*une licence l_j est plus (ou autant) restrictive qu'une licence l_i si l_j autorise au plus les mêmes permissions et impose au moins les mêmes obligations et interdictions.*

Dans la majorité des cas, quand l_i est moins restrictive que l_j alors, l_i est compatible avec l_j . La Figure 1 représente trois licences Creative Commons en RDF décrites à l'aide

1. <https://creativecommons.org/>

2. <https://creativecommons.org/licenses/>

de l'ontologie ODRL³. Dans cet exemple, (a) est moins restrictive que (b), (b) est moins restrictive que (c) et, par transitivité, (a) est moins restrictive que (c). Nous remarquons que (a) est compatible avec (b) et (c), mais (b) n'est pas compatible avec (c). Dans cet exemple, la restrictivité entre (b) et (c) n'est pas accompagnée d'une relation de compatibilité. Ceci est dû à la sémantique de l'action *DerivativeWorks* qui interdit la distribution d'une modification de la ressource sous une autre licence⁴. En effet, selon la sémantique de leurs actions, la relation de restrictivité entre deux licences n'implique pas forcément une relation de compatibilité.

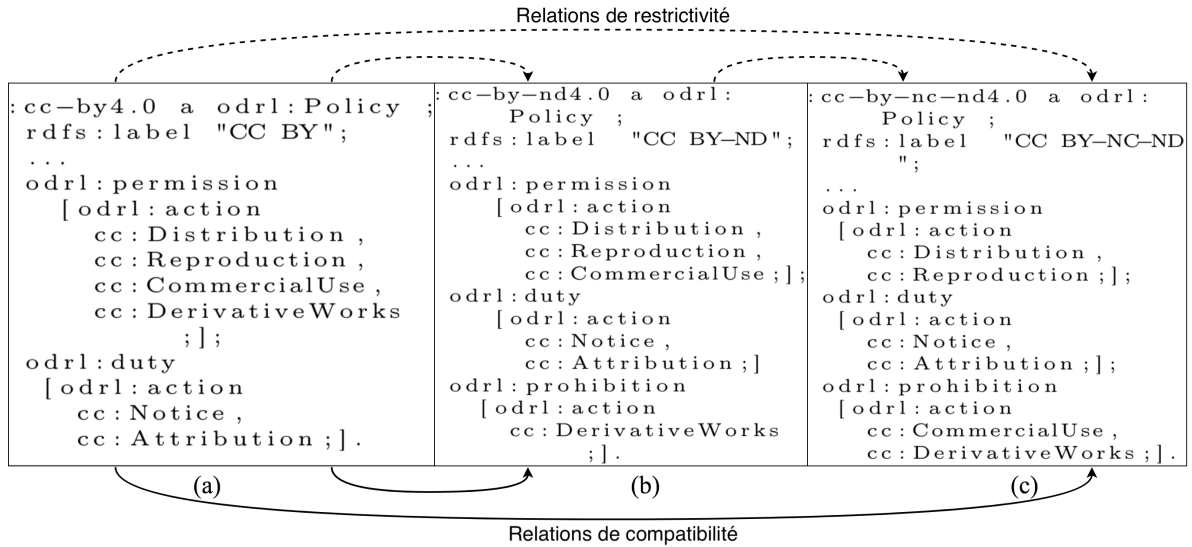


FIGURE 1: Trois licences Creative Commons décrites en RDF.

Notre question de recherche est : Soit une licence l_i , comment positionner automatiquement l_i dans un ensemble de licences selon leur compatibilité ? Notre défi est de généraliser la relation d'ordre entre les licences tout en prenant en compte l'influence de la sémantique des actions.

Nous proposons CaLi (C**l**Assification of L**ic**enses), un modèle basé sur un treillis permettant d'ordonner des licences. CaLi utilise la relation de restrictivité et des contraintes pour définir la compatibilité entre les licences.

L'article complet (Moreau *et al.*, 2019b) a été publié dans les actes de l'Extended Semantic Web Conférence (ESWC 2019). Il contient les algorithmes complets et une évaluation expérimentale des algorithmes implémentés. Un papier de démonstration a aussi été publié (Moreau *et al.*, 2019a) et illustre l'utilisation de notre modèle dans un moteur de recherche basé sur un ordre CaLi. Ce dernier est capable de trouver les ressources dont les licences sont compatibles avec une licence particulière.

Dans la suite, nous présentons brièvement comment ordonner des licences à l'aide du modèle CaLi, ainsi que nos expérimentations.

2 Ordonner des licences avec le modèle CaLi

Dans une licence, les actions (e.g., lecture, modification, etc.) peuvent être réparties entre ce que nous appelons des *statuts*, e.g., permissions, obligations, interdictions, etc. Dans l'exemple d'introduction, nous considérons que les permissions sont moins restrictives que les obligations lesquelles sont moins restrictives que les interdictions. L'ordre de restrictivité entre les statuts est subjectif et peut varier en fonction du contexte.

3. <https://www.w3.org/TR/odrl-model/>

4. https://wiki.creativecommons.org/wiki/Wiki/cc_license_compatibility

Nous avons remarqué que si deux licences ont une relation de restrictivité alors il est possible qu'elles aient aussi une relation de compatibilité. L'avantage de la relation de restrictivité entre les licences est qu'elle peut être obtenue automatiquement selon les statuts des actions. Inspiré des travaux sur le contrôle d'accès basé sur des treillis Davey & Priestley (2002), nous définissons une relation de restrictivité entre les licences.

Pour identifier la compatibilité entre les licences, nous affinons la relation de restrictivité avec des contraintes. L'objectif est de prendre en compte la sémantique des actions. Les contraintes distinguent aussi les licences valides des non-valides. Nous considérons qu'une licence est non-valide si aucune ressource ne peut être protégée par cette dernière., e.g., une licence qui, simultanément, autorise l'action *Derive*⁵ et interdit l'action *DerivativeWorks*⁶.

2.1 Modèle CaLi

Inspiré par les modèles de contrôle d'accès basés sur des treillis, le modèle CaLi est un tuple $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ capable d'ordonner partiellement des licences, tel que :

1. $\mathcal{A}$ est un ensemble d'actions (e.g., *lire*, *modifier*, *distribuer*, etc.);
2. $\mathcal{LS}$ est un treillis de restrictivité des statuts définissant (i) l'ensemble des statuts possibles (e.g., permission, obligation, interdiction, etc.) d'une action dans une licence et (ii) la relation de restrictivité $\leq_S$ entre ces statuts. La Figure 2 montre plusieurs exemples de treillis de restrictivité de statuts;
3. $C_{\rightarrow}$ est un ensemble de contraintes sur la compatibilité permettant d'identifier les relations de restrictivité entre deux licences qui sont aussi des relations de compatibilité;
4. Enfin, $C_{\mathcal{L}}$ est un ensemble de contraintes sur les licences permettant d'identifier les licences qui ne sont pas valides.

$\mathcal{L}_{\mathcal{A}, \mathcal{LS}}$ définit l'ensemble exhaustif des licences exprimables avec $\mathcal{A}$ et $\mathcal{LS}$. $(\mathcal{L}_{\mathcal{A}, \mathcal{LS}}, \leq_R)$ est le treillis de restrictivité des licences définissant la relation de restrictivité $\leq_R$ sur l'ensemble de licences $\mathcal{L}_{\mathcal{A}, \mathcal{LS}}$. L'ensemble de contraintes $C_{\mathcal{L}}$ identifient les licences non-valides. Si deux licences valides ont une relation de restrictivité alors, il est possible qu'elles aient une relation de compatibilité. L'ensemble de contraintes $C_{\rightarrow}$ identifient les relations de compatibilité parmi les relations de restrictivité.

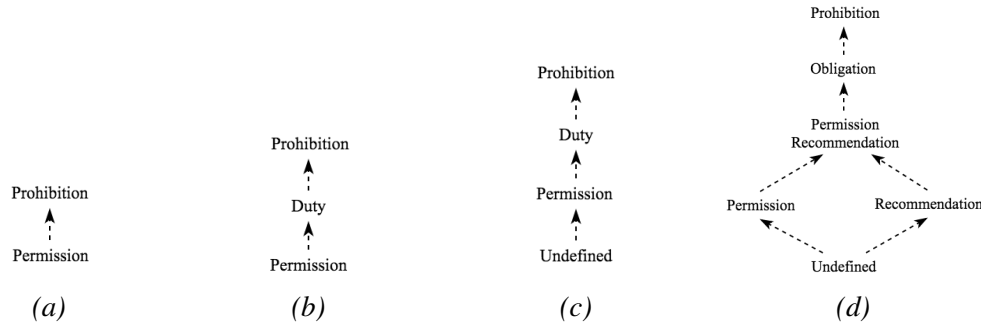


FIGURE 2: Exemples de treillis de restrictivité de statuts ($\mathcal{LS}$). Les flèches pointillées représentent la restrictivité. 2a est inspiré par les systèmes de fichiers où les actions peuvent être soit autorisées, soit interdites et où l'interdiction de lire un fichier est plus restrictive que la permission de le lire. 2b est basé sur les licences CC. 2c est inspiré par le vocabulaire ODRl où les actions peuvent être autorisées, obligées, interdites ou non spécifiées (i.e., undefined). Fig. 2d montre un treillis de restrictivité où une action recommandée ou autorisée est moins restrictive que la même action lors qu'elle est autorisée et recommandée.

5. <https://www.w3.org/TR/odrl-vocab/#term-derive>

6. <https://www.w3.org/TR/odrl-vocab/#term-DerivativeWorks>

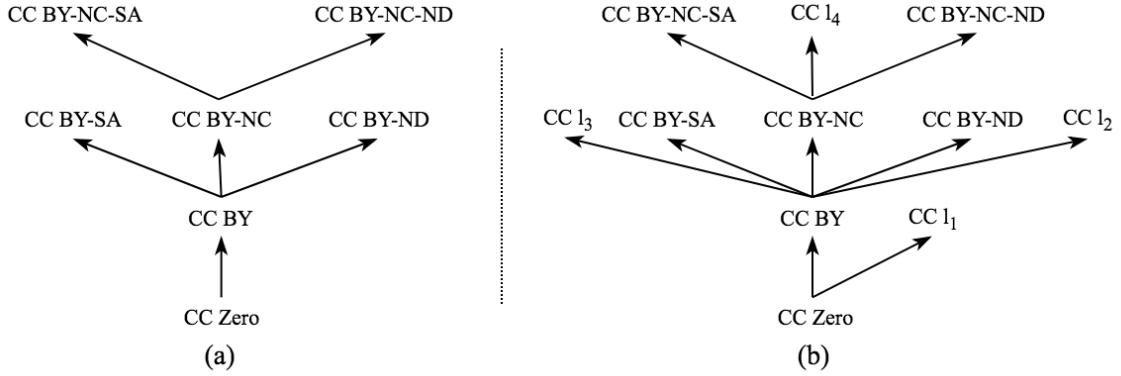


FIGURE 3: Sous-graphes de compatibilité de CC_CaLi : (a) contient les 7 licences officielles CC et (b) contient en plus d'autres licences valides.

2.2 Un ordre CaLi pour licences de Creative Commons

Creative Commons propose 7 licences juridiquement vérifiées, gratuites et faciles à comprendre. Elles sont largement utilisées sur le Web. Ces licences utilisent 7 actions qui peuvent être autorisées, obligées ou interdites (cf. Figure 2b). Dans notre modèle CC_CaLi , nous modélisons un ordre de compatibilité de licences complet pour toutes les licences possibles et valides utilisant les 7 actions de Creative Commons.

Ainsi, CC_CaLi est un modèle CaLi $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ tel que :

- $\mathcal{A}$ est l'ensemble des 7 actions $\{cc : Distribution, cc : Reproduction, cc : DerivativeWorks, cc : CommercialUse, cc : Notice, cc : Attribution, cc : ShareAlike\}$;
- $\mathcal{LS}$ est le treillis de restrictivité des statuts de la Figure 2b ;
- $C_{\mathcal{L}}, C_{\rightarrow}$ sont les ensembles de contraintes issues du vocabulaire CC et réinterprétées sous forme de fonctions.

$C_{\mathcal{L}} = \{\omega_{\mathcal{L}_1}, \omega_{\mathcal{L}_2}\}$ invalide une licence (1) quand $cc : CommercialUse$ est obligée, ou (2) quand $cc : ShareAlike$ est interdit :

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

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

$C_{\rightarrow} = \{\omega_{\rightarrow_1}, \omega_{\rightarrow_2}\}$ identifie les relations de restrictivité qui ne sont pas des relations de compatibilité (1) quand $cc : ShareAlike$ est obligée ou (2) quand $cc : DerivativeWorks$ est interdit. En effet, $cc : ShareAlike$ exige que la ressource redistribuée soit protégée par la même licence. L'interdiction de $cc : DerivativeWorks$, n'autorise pas la distribution d'une ressource ou licence modifiée. Ces deux actions sont donc en contradiction avec la définition de la compatibilité.

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

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

D'autres contraintes peuvent être définies pour être encore plus proche des propriétés du langage CC REL⁷. Mais pour notre exemple, ces contraintes sont suffisantes.

7. <https://creativecommons.org/ns>

Nous représentons les licences sous forme d'une fonction $l : \mathcal{A} \rightarrow S$, où $\mathcal{A}$ est un ensemble d'actions et $\mathcal{LS} = (S, \leq_S)$ est un treillis de restrictivité de statuts. Par exemple, la licence CC BY-NC-SA peut être définie comme :

$$CCBYNC(a) = \begin{cases} \text{Permission} & \text{if } a \in \{cc : \text{Distribution}, cc : \text{Reproduction} \\ & \quad cc : \text{DerivativeWorks}, cc : \text{ShareAlike}\}; \\ \text{Duty} & \text{if } a \in \{cc : \text{ShareAlike}, cc : \text{Notice}, cc : \text{Attribution}\}; \\ \text{Prohibition} & \text{if } a \in \{cc : \text{CommercialUse}\}. \end{cases}$$

La taille du treillis de restrictivité CC_CaLi est de 3^7 licences, mais le nombre de licences valides est de 972. Ceci est dû aux contraintes $C_{\mathcal{L}}$. En effet, il est possible d'avoir 5 actions dans n'importe quel statuts et 2 actions ($cc : \text{CommercialUse}$ et $cc : \text{ShareAlike}$) dans uniquement 2 statuts : $3^5 * 2^2$.

La Figure 3 montre deux sous-graphes de CC_CaLi . 3a montre uniquement les 7 licences officielles de CC tandis que 3b montre en plus d'autres licences valides. Grâce à $\omega_{\rightarrow 1}$, la relation de restrictivité entre CC BY-SA et CC BY-NC-SA n'est pas identifiée comme une relation de compatibilité. Et grâce à $\omega_{\rightarrow 2}$, la relation de restrictivité entre CC BY-ND et CC BY-NC-ND n'est pas identifié comme une relation de compatibilité non plus. A noter qu'une licence interdisant $cc : \text{DerivativeWorks}$ n'est pas compatible avec elle même.

La relation de compatibilité de la Figure 3a est conforme au graphe de compatibilité qu'il est possible de générer à partir de la table de compatibilité des licences Creative Commons ⁸.

3 Implementation et expérimentation

Notons que la taille d'un ordre CaLi est exponentiel $|\mathcal{LS}|^{|\mathcal{A}|}$. Cependant, il n'est pas nécessaire de générer le treillis complet pour l'utiliser. Dans l'article (Moreau *et al.*, 2019b), nous proposons un algorithme de tri pouvant ordonner n'importe quel sous-ensemble de $\mathcal{L}_{\mathcal{A}, \mathcal{LS}}$ en approximativement $n^2/2$ comparaisons, n étant le nombre de licences dans notre sous-ensemble i.e., $O(n^2)$. Cet algorithme est capable d'insérer une licence dans un ensemble de licences partiellement ordonné en temps linéaire $O(n)$.

L'implémentation en Python de notre algorithme et les détails de nos expériences sont disponibles sur GitHub ⁹.

Nous montrons l'utilité de notre modèle avec un moteur de recherche de jeux de données basé sur des licences ¹⁰. Il utilise l'ordre $ODRL_CaLi$ capable d'ordonner par compatibilité des licences décrites à l'aide de l'ontologie ODRL.

$ODRL_CaLi$, est un modèle CaLi $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ tel que :

- $\mathcal{A}$ est l'ensemble des 72 actions proposées dans ODRL ¹¹ ;
- $\mathcal{LS}$ est le treillis de restrictivité des statuts où (i) les statuts possibles sont les trois règles ODRL : Permission, Duty (i.e. obligation), Prohibition (i.e. interdiction) ¹² ou Undefined (pour les actions n'apparaissant pas dans la licence), et où (ii) les relations de restrictivité entre les statuts es celui de la Figure 2c ;
- Enfin, $C_{\mathcal{L}}, C_{\rightarrow}$ sont les ensembles de contraintes issues du vocabulaire ODRL.

$C_{\mathcal{L}} = \{\omega_{\mathcal{L}_1}, \omega_{\mathcal{L}_2}, \omega_{\mathcal{L}_3}\}$ invalide une licence de la même manière que dans l'ordre CC_CaLi avec en plus (3) quand la sémantique d'une action autorisée ou obligée est incluse (*odrl : includedIn*) dans une action interdite (e.g., si *CommercialUse* est permis alors, l'action *use* ne

8. https://wiki.creativecommons.org/wiki/Wiki/cc_license_compatibility

9. <https://github.com/benjmor/CaLi-Search-Engine>

10. <http://cali.priloo.univ-nantes.fr/ld/>

11. <https://www.w3.org/TR/odrl-vocab/#actionConcepts>

12. <https://www.w3.org/TR/odrl-model/#rule>

doit pas être interdite, car l'usage commerciale d'une ressource implique son utilisation) :

$$\omega_{\mathcal{L}_3}(l_i) = \begin{cases} \textit{Faux} & \text{si } a_i \text{ odrl :includedIn } a_j \\ & \text{et } (l_i(a_i) = \textit{Permission} \text{ ou } l_i(a_i) = \textit{Duty}) \\ & \text{et } l_i(a_j) = \textit{Prohibition}; \\ \textit{Vrai} & \text{sinon.} \end{cases}$$

Un exemple de sous-ensemble de licences partiellement ordonné selon l'ordre *ODRL_CaLi*¹³ alimente notre moteur de recherche de jeux de données. Il contient un ordre partiel avec les licences les plus utilisées sur les plateformes DataHub¹⁴ et Opendatasoft Data Network¹⁵. Le même moteur de recherche est mis à disposition pour de dépôts GitHub et contient les licences les plus utilisées sur cette plateforme¹⁶.

4 Conclusion

Nous proposons un modèle basé sur un treillis pour définir la relation de compatibilité entre licences. Notre approche se base sur la relation de restrictivité affinée avec des contraintes pour prendre en compte la sémantique des actions. Nous avons montré la faisabilité de notre approche à travers deux ordres CaLi, le premier utilisant le vocabulaire Creative Commons et le second utilisant le vocabulaire ODRL. Nous avons expérimenté la production d'ordres CaLi avec l'implémentation d'un algorithme de tri par insertion dont la complexité est de $n^2/2$ comparaisons. Nous avons implémenté le prototype d'un moteur de recherche de jeux de données basé sur les licences montrant l'utilité de notre contribution. Notre modèle de compatibilité n'a pas l'intention de fournir un avis juridique mais il permet d'exclure les licences qui violerait une licence particulière.

Remerciements

Les auteurs remercient Margo Bernelin et Sonia Desmoulin-Canselier (laboratoire de Droit et Changement Social - UMR CNRS 6297) pour nos discussions sur ce travail.

Références

- DAVEY B. A. & PRIESTLEY H. A. (2002). *Introduction to Lattices and Order*. Cambridge university press.
- MOREAU B., SERRANO-ALVARADO P., PERRIN M. & DESMONTILS E. (2019a). A License-Based Search Engine. In *Extended Semantic Web Conference (ESWC), Poster&Demo*.
- MOREAU B., SERRANO-ALVARADO P., PERRIN M. & DESMONTILS E. (2019b). Modelling the Compatibility of Licenses. In *Extended Semantic Web Conference (ESWC)*.
- SENEVIRATNE O., KAGAL L. & BERNERS-LEE T. (2009). Policy-Aware Content Reuse on the Web. In *International Semantic Web Conference (ISWC)*.

13. <http://cali.priloo.univ-nantes.fr/ld/graph>

14. <https://old.datahub.io/>

15. <https://data.opendatasoft.com/pages/home/>

16. <http://cali.priloo.univ-nantes.fr/rep/graph>