Licenses Compatibility and Composition in the Web of Data

Serena Villata⋆ and Fabien Gandon

INRIA Sophia Antipolis, France
{firstname.lastname}@inria.fr

Abstract. The absence of clarity concerning the licensing terms does not encourage the reuse of the data by the consumers, thus preventing further publication and consumption of datasets, at the expense of the growth of the Web of Data itself. In this paper, we propose a general framework to attach the licensing terms to the data queried on the Web of Data. In particular, our framework addresses the following issues: (i) the various license schemas are collected and aligned taking as *reference* the Creative Commons schema, (ii) the compatibility of the licensing terms concerning the data affected by the query is verified, and (iii) if compatible, the licenses are combined into a composite license. The framework returns the composite license as licensing term about the data resulting from the query.

1 Introduction

Heath and Bizer [5] underline that "the absence of clarity for data consumers about the terms under which they can reuse a particular dataset, and the absence of common guidelines for data licensing, are likely to hinder use and reuse of data". Therefore, all Linked Data on the Web should include explicit licensing terms, or waiver statements [8]. Data consumers need to know the licenses on the data to avoid misusing and even illegal reuse of the data. Following these highlights, several machine-readable licensing terms have been proposed, leading to a huge amount of unrelated but comparable ways of expressing the licenses. The scenario we consider in this paper is visualized in Figure 1. When consumers query the Web of Data, results from different datasets, and thus released under different licensing terms, are provided. Our objective is to build a composite license which combines the elements from the distinct licenses associated to the selected data. The research questions we answer in this paper are: (1) How to verify the compatibility among the licensing terms associated to a query result?, and (2) How to compose, if compatible, the distinct licensing terms for creating a composite license?

We answer the research questions by adopting Semantic Web languages only, and reusing the Creative Commons (CC) licenses schema [1]. In particular, we choose the CC vocabulary to define the anatomy of our licenses. Licenses

⋆ The author acknowledges the support of the DataLift Project ANR-10-CORD-09.

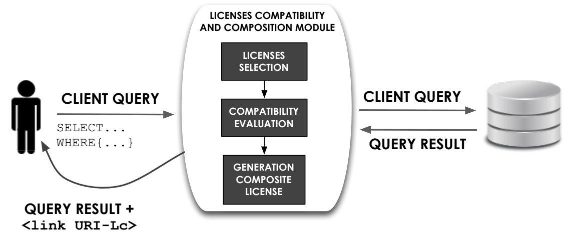


Fig. 1. The architecture of the license compatibility and composition framework.

are composed by *models* (`cc:Permission`, `cc:Requirement`, `cc:Prohibition`) which specify the permissions allowed by the license, the required behaviors and the prohibited use and reuse of the data. The models are composed by *elements* like `ShareAlike`, `Attribution`, and many others, which specify the precise permissions, requirements, and prohibitions associated to the license. We choose this vocabulary because it provides a general schema for licenses specification. We are aware that there are works which cannot be considered as creative works [8, 5]. For addressing this issue and covering a wider range of machine-readable license specifications, we align CC with the other schemas representing licenses.

The set of licenses is evaluated as compatible if a number of compatibility rules are satisfied. These rules specify the subsumption relations among the licenses' elements and evaluate the compatibility among Permissions, Requirements and Prohibitions elements. If the pre-requisite of licenses compatibility is satisfied, the licenses are composed. We propose rules to deal with the composition of the licenses whose elements subsume other elements, and of the constraints expressed by the licenses. We define some basic heuristics to address the composition. Using these heuristics, the Composite License Generation algorithm returns the composite license. This license will be the only license attached to the data returned by the query. The composite license is retuned together with the query result using the standard SPARQL query results XML format by means of the `<link>`[1] element.

In this paper, we do not introduce yet another licensing schema. We consider RDF machine-readable license specifications only, and we do not consider MPEG-21[2]. Moreover, we do not provide verification compliance about the reuse of the data by the consumer. We extend and adapt existing proposals for licenses compatibility and composition in the area of service license analysis [4] to the Web of Data scenario.

The reminder of the paper is as follows: Section 2 compares the existing literature with the proposed approach. Section 3 formalizes the licenses structure and presents the alignment with the other vocabularies. Section 4 defines the compatibility rules and the evaluation algorithm, and Section 5 shows how the licenses are composed.

[1] `http://www.w3.org/TR/rdf-sparql-XMLres/#head`
[2] `http://mpeg.chiariglione.org/standards/mpeg-21/mpeg-21.htm`

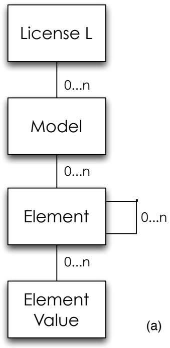
2 Related work

The Creative Commons Rights Expression Language (ccREL) [1] is the standard recommended by Creative Commons for machine-readable expression of copyright licensing terms and related information. Miller et al. [8] propose the Open Data Commons waivers and licenses[3] that try to eliminate or fully license any rights that cover databases and data. The Waiver vocabulary[4] defines properties to use when describing waivers of rights over data and content where a waiver is the voluntary relinquishment or surrender of some known right or privilege. Iannella [6] presents the Open Digital Rights Language (ODRL), a language for the Digital Rights Management community for expressing rights information over content. The ODRL-S language [5] extends ODRL by implementing the clauses of service licensing. Gangadharan et al. [4, 3] address the issue of service license composition and compatibility analysis. They specify a matchmaking algorithm which verifies whether two service licenses are compatible. In case of a positive answer, the services are composed and the framework determines the license of the composite service. Even if the Compatibility Evaluation algorithm we propose follows the basic steps of [4, 3], there are several differences between the two approaches. The application scenarios are different (service composition vs Web of Data) and this opens different problems. In particular, the subsumption and compatibility rules we define are different, and this affects also the algorithm definition. The composite license mirrors the same differences. Truong et al. [9] address the issue of analyzing data contracts which leads to the definition of a contract composition where first the comparable contractual terms from the different data contracts are retrieved, and second an evaluation of the new contractual terms for the data mashup is addressed. This work concentrates on data contracts and not on data licenses. However, there are also some common points like the use of RDF for the representation of data licenses/contracts. Krotzsch and Speiser [7] present a semantic framework for evaluating ShareAlike recursive statements, developing a general policy modelling language for supporting self-referential policies as expressed by CC. We address another kind of problem that is the composition of the licenses constraining the data into a unique composite license.

3 Creative Commons licensing language

The Creative Commons licenses are a family of license models for publishing resources on the Web, enabling the reuse of the data instead of the forbidden *by default* approach of traditional copyright specifications. The basic anatomy of such licenses[6] is composed by a *Work* which is a potentially copyrightable work, a *License* that is a set of permissions/requirements/prohibitions to the users of a *Work*, and a *Jurisdiction* that is the legal jurisdiction of a license.

[3] `http://opendatacommons.org/licenses/`
[4] `http://vocab.org/waiver/terms/.html`
[5] `http://odrl.net/Profiles/Services/WD-20080402.html`
[6] `http://creativecommons.org/ns`



Permissions	Requirements	Prohibitions
Reproduction	Notice	Commercial Use
Distribution	Attribution	High Income Nation Use
Derivative Works	Share Alike	
Sharing	Source Code	
	Copyleft	

(b)

Fig. 2. The basic structure of our licenses (a), and the CC models and elements (b).

Figure 2.a shows the structure of a license in our framework. This structure is as desired general enough to include possible extensions to the CC vocabulary. Each license $\mathcal{L}$ is composed by a set of models. In particular, the models are `cc:Permission`, `cc:Requirement`, and `cc:Prohibition`, and express the kind of actions respectively permitted, required and prohibited on the data.

Figure 2.b shows the set of elements associated to each model in CC. Our general model of licenses takes into account for generality purposes also the possible values assigned to the elements.

Definition 1. *A license $\mathcal{L}(R)$ for a named graph R is a finite set of models m, each of which further consists of a set of elements el.*

A huge amount of vocabularies includes properties and classes about the licensing terms for data reuse. In order to achieve interoperability, we align the CC vocabulary with these vocabularies, in some cases very heterogeneous, including licensing terms. A summarization of the alignment about licensing terms[7] is visualized in Figure 3. Note that in the CC vocabulary it holds that `cc:license subPropertyOf dcterms:license`, and `cc:License subClassOf dcterms:LicenseDocument`. The alignment considers the following vocabularies: the Description of a Project vocabulary (`doap`)[8], the Ontology Metadata vocabulary (`omv`)[9], the Data Dictionary for Preservation Metadata (`premis`)[10], the Vocabulary Of Attribution and Governance (`voag`)[11], the NEPOMUK Informa-

[7] The research of the vocabularies including licensing terms has been addressed using the LOV dataset (available at `http://labs.mondeca.com/dataset/lov/`).
[8] `http://usefulinc.com/ns/doap`
[9] `http://omv2.sourceforge.net/index.html`
[10] `http://bit.ly/MVTZkr`
[11] `http://voag.linkedmodel.org/schema/voag`

tion Element Ontology (`nie`)[12], the Music Ontology (`mo`)[13], the Good Relations vocabulary (`gr`)[14], and the myExperiment Base Ontology (`meb`)[15].

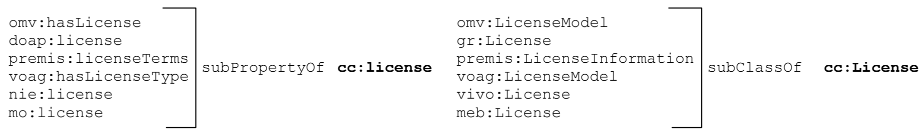


Fig. 3. Alignment between the CC vocabulary and other vocabularies.

4 Licenses compatibility analysis

For providing a definition of the Compatibility Evaluation algorithm, we need to define a set of compatibility rules assessing the possible compatibility among the elements composing the licenses, following [4] for service licenses. First, there are certain elements which are broader in scope of permission than other elements. Consider the following example where two named graphs[16] R_1 and R_2 respectively are associated to different licenses where the elements are `cc:Reproduction` and `cc:Sharing`. The consumer's query returns triples from both R_1 and R_2, thus the compatibility between the two licenses has to be verified. *Reproduction* permits making multiple copies, and *Sharing* permits commercial derivatives [1]. In our model these two licenses are evaluated as compatible because `cc:Sharing` subsumes `cc:Reproduction`. In this context, subsumption means that there is a compatibility if a certain element el_i is more permissive, i.e., it accepts more, than the other element el_j. For instance, we have that `cc:Sharing` is more permissive than `cc:Reproduction` where only multiple copies are allowed but no commercial derivatives. The way the two licenses then are combined into the composite license $\mathcal{L}_c$ depends on the redefinition rule applied by the data provider (Section 5). The subsumption rules for the `cc:Permission` elements are shown in Table 1.

Table 2.a shows whether an element el_1 is compatible with another element el_2, i.e., $el_1 \bigcirc el_2$, for `cc:Permission` elements. Rules are expressed under the form of truth tables where elements are evaluated as compatible T, or incompatible F. The rationale behind Table 2.a is that the elements are compatible if there is a subsumption relation between them. Thus, the only case in which the compatibility does not hold is when the elements are `cc:Sharing` and

[12] `http://www.semanticdesktop.org/ontologies/2007/01/19/nie/`
[13] `http://purl.org/ontology/mo/`
[14] `http://purl.org/goodrelations/v1`
[15] `http://rdf.myexperiment.org/ontologies/base/`
[16] The discussion about the use of named graphs in RDF 1.1 can be found at `http://www.w3.org/TR/rdf11-concepts`

Subsumption	More permissive	Less permissive
$DerivativeWorks \succ Sharing$	$DerivativeWorks$	$Sharing$
$Distribution \succ Reproduction$	$Distribution$	$Reproduction$
$DerivativeWorks \succ Reproduction$	$DerivativeWorks$	$Reproduction$
$Sharing \succ Reproduction$	$Sharing$	$Reproduction$
$DerivativeWorks \succ Distribution$	$DerivativeWorks$	$Distribution$

Table 1. Compatibility rules for subsumption relation for `cc:Permission` elements.

`cc:Distribution` where no subsumption relation is present because `cc:Sharing` allows for non-commercial distribution only, while `cc:Distribution` allows commercial distribution as well. Note that compatibility in our rules is bi-directional, where (el_1 compatible with el_2) is equivalent to (el_2 compatible with el_1)[17].

As in [4], a possible situation which may arise is that one license specifies clauses which are not specified by the other license, e.g., one license specifies `cc:Prohibition` and the other license does not specify this clause. Table 2.b shows the compatibility rules for `cc:Requirement`, `cc:Prohibition` and `cc:Permission` elements against *Unspecified* elements[18]. Concerning `cc:Requirement`, the requirement for specification of *Attribution* does not affect the compatibility with *Unspecified* (the same holds for *Notice*, *SourceCode*, and *CopyLeft*), and *ShareAlike* affects the composite license $\mathcal{L}_c$ by requiring that $\mathcal{L}_c$ is similar to the original license. Concerning `cc:Prohibition`, these elements are incompatible with *Unspecified*. For instance, commercial use is the default setting of the licenses, thus we cannot assume compatibility if commercial use is denied by NonCommercial[19]. The same holds for the `cc:HighIncomeNationUse` element. Concerning `cc:Permission`, we follow the "conservative" approach in handling *Unspecified* elements where unspecification means a denial of compatibility [3].

Finally, the Compatibility Evaluation algorithm has to verify whether and how `cc:Requirement` elements and `cc:Prohibition` elements are compatible due to the constraints they specify. Assume two licenses $\mathcal{L}_1$ and $\mathcal{L}_2$ where $\mathcal{L}_1$ contains a clause specifying the requirement for `cc:Attribution` and $\mathcal{L}_2$ contains a clause specifying the prohibition `cc:CommercialUse`. Are these clauses compatible? The answer is that, despite Table 2.b, they are compatible as far as the composition of the two licenses includes both the elements, i.e., it satisfies both the `cc:CommercialUse` clause and the `cc:Attribution` clause. This means that these license sets are non-disjoint. Table 3 shows the compatibility rules applied to the requirements `cc:Attribution` and `cc:ShareAlike` and the

[17] Differently from software compatibility, it is not a matter of having the same element changed in different versions, but to have different elements to be compared, e.g., Sharing and Reproduction.

[18] We interpret unspecified elements as "do not care", as in [4].

[19] Note that in this section we refer to the element `cc:CommercialUse` as *NonCommercial* for keeping clear that it is a prohibition.

el_1	el_2	$el_1 \bigcirc el_2$
$Sharing$	$DerivativeWorks$	T
$Reproduction$	$Distribution$	T
$Reproduction$	$DerivativeWorks$	T
$Reproduction$	$Sharing$	T
$Distribution$	$DerivativeWorks$	T
$Sharing$	$Distribution$	F

el_1	el_2	$el_1 \bigcirc el_2$
$Notice$	$Unspecified$	T
$Attribution$	$Unspecified$	T
$ShareAlike$	$Unspecified$	T
$SourceCode$	$Unspecified$	T
$CopyLeft$	$Unspecified$	T
$NonCommercial$	$Unspecified$	F
$HighIncomeNationUse$	$Unspecified$	F
$Reproduction$	$Unspecified$	F
$Distribution$	$Unspecified$	F
$DerivativeWorks$	$Unspecified$	F
$Sharing$	$Unspecified$	F

Table 2. (a) Compatibility rules among `cc:Permission` elements, (b) Compatibility rules among `cc:Requirement`, `cc:Prohibition` and `cc:Permission` elements against *Unspecified*.

`cc:CommercialUse` prohibition where, with a slight abuse of notation, we write $\mathcal{L}_1 \odot \mathcal{L}_2 = el_1 \wedge el_2$ to indicate that the composition $\odot$ of the licenses will include both the elements.

el_1	el_2	$el_1 \bigcirc el_2$	$\mathcal{L}_1 \odot \mathcal{L}_2$
$Attribution$	$ShareAlike$	T	$el_1 \wedge el_2$
$Attribution$	$NonCommercial$	T	$el_1 \wedge el_2$
$ShareAlike$	$NonCommercial$	T	$el_1 \wedge el_2$

Table 3. Compatibility rules among `cc:Requirement` and `cc:Prohibition` elements.

The rules detailed in Tables 1-3 provide the basic components of our Compatibility Evaluation algorithm. Let $\mathcal{L}(C) = \{\mathcal{L}_1(R_1), \ldots, \mathcal{L}_n(R_n)\}$ be the set of licenses associated to the n named graphs affected by the consumer's query where $\mathcal{L}_i(R_i)$ is the license associated with named graph R_i, then our objective is to compose these licenses into the composite license $\mathcal{L}_c$. Definition 2 defines that two licenses are compatible if for all their clauses l_i, these clauses are compatible.

Definition 2. *A license $\mathcal{L}_i$ is compatible with another license $\mathcal{L}_j$, $\mathcal{L}_i \bigcirc \mathcal{L}_j$, if $\forall l_{i=1,\ldots,n} \in \mathcal{L}_i, \forall l_{j=1,\ldots,m} \in \mathcal{L}_j$ it holds that $l_i \bigcirc l_j$.*

The Compatibility Evaluation algorithm (Figure 4) compares the clauses of the licenses to be composed. Two licenses are compatible if the models in both the licenses are compatible[20]. The models are compatible if (i) the models are the same ($m_i \equiv m_j$), (ii) the models are composed by elements which satisfy

[20] In this paper, we do not consider the presence of sub-elements and of numerical values assigned to the elements, e.g., financial use, however the algorithm can easily be extended for treating these cases as well.

Algorithm 1: LicensesCompatibility

Input: Licenses $\mathcal{L}_i$, $\mathcal{L}_j$
Output: `true` if $\mathcal{L}_i \bigcirc \mathcal{L}_j$, `false` otherwise
if $\forall m_i : m_i \in \mathcal{L}_i, \exists m_j : m_j \in \mathcal{L}_j \leftarrow$ `ModelsCompatibility`$(m_i, m_j) = true \wedge$
$\forall m_j : m_j \in \mathcal{L}_j, \exists m_i : m_i \in \mathcal{L}_i \leftarrow$ `ModelsCompatibility`$(m_i, m_j) = true$ **then**
  $compatible = true$;
else
  $compatible = false$;
return $compatible$;

Algorithm 2: ModelsCompatibility

Input: Models m_i, m_j
Output: `true` if $m_i \bigcirc m_j$, `false` otherwise
if $((m_i \equiv m_j) \vee$ `IntersectionRules`$(m_i, m_j))\wedge$
$\forall el_i : el_i \in m_i, \exists el_j : el_j \in m_j \leftarrow$ `ElementsCompatibility`$(el_i, el_j) = true \wedge$
$\forall el_j : el_j \in m_j, \exists el_i : el_i \in m_i \leftarrow$ `ElementsCompatibility`$(el_i, el_j) = true$ **then**
  $compatible = true$;
else
  $compatible = false$;
return $compatible$;

Algorithm 3: ElementsCompatibility

Input: Elements el_i, el_j
Output: `true` if $el_i \bigcirc el_j$, `false` otherwise
if $((el_i \equiv el_j)$
$\vee$`SubsumptionRules`$(el_i, el_j) \vee$ `UnspecifiedRules`$(el_i, el_j) \vee$ `IntersectionRules`$(el_i, el_j))\wedge$
$\forall el_i : el_i \in m_i, \exists el_j : el_j \in m_j \leftarrow$ `ElementsCompatibility`$(el_i, el_j) = true \wedge$
$\forall el_j : el_j \in m_j, \exists el_i : el_i \in m_i \leftarrow$ `ElementsCompatibility`$(el_i, el_j) = true$ **then**
  $compatible = true$;
else
  $compatible = false$;
return $compatible$;

Fig. 4. The compatibility evaluation procedure.

the compatibility rules (Table 3), and (iii) their elements are compatible. The elements compatibility is checked with respect to the following features: (i) the elements are the same ($el_i \equiv el_j$), (ii) the elements satisfy the subsumption rules (Table 1) (`SubsumptionRules`(el_i, el_j)), (iii) the elements satisfy the compatibility rules against *Unspecified* (Table 2.b) (`UnspecifiedRules`(el_i, el_j)), (iv) the elements satisfy the compatibility rules (Table 3) (`IntersectionRules`(el_i, el_j)), and (v) all the nested elements (if any) are compatible.

Example 1. Assume we want to analyze two licenses $\mathcal{L}_1$ and $\mathcal{L}_2$, visualized in Figure 5. Following our algorithm, we first compare the two licenses at the model level. Both licenses contain the model `cc:Permission`, and both of them permit `cc:Reproduction`. As the models are of the same type and the elements are of the same type as well, then the compatibility is verified. License $\mathcal{L}_1$ contains the model `cc:Requirement` with the element `cc:Notice`. This model is not specified by license $\mathcal{L}_2$, but following the compatibility rules of Table 2.b, the algorithm evaluates that the element `cc:Notice` is compatible with an *Unspecified* element. Compatibility is maintained. Finally, the algorithm evaluates the model `cc:Prohibition` of $\mathcal{L}_2$, looking for the same model and the same

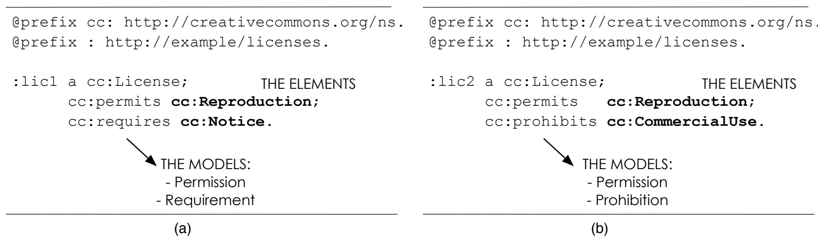


Fig. 5. An example of compatibility evaluation which leads to a `false` result.

element (`cc:CommercialUse`) in license $\mathcal{L}_1$. The model `cc:Prohibition` is not specified in $\mathcal{L}_1$. From Table 2.b, the algorithm evaluates `cc:CommercialUse` and *Unspecified* incompatible. The compatibility evaluation of the licenses leads to a `false` answer.

The following properties hold for the license compatibility relation we define.

Property 1 (Symmetry). Let $\mathcal{L}_i, \mathcal{L}_j$ be two compatible licenses, then $\forall l_i \in \mathcal{L}_i, \forall l_j \in \mathcal{L}_j$ it holds that if $l_i \bigcirc l_j$ then $l_j \bigcirc l_i$.

The proof follows from the definition of the Compatibility Evaluation algorithm and the composition rules. The symmetry property ensures that $\mathcal{L}_i \bigcirc \mathcal{L}_j \equiv \mathcal{L}_j \bigcirc \mathcal{L}_i$.

Property 2 (Associativity). Let $\mathcal{L}_i, \mathcal{L}_j, \mathcal{L}_k$ be three licenses then it holds that $(\mathcal{L}_i \bigcirc \mathcal{L}_j) \bigcirc \mathcal{L}_k = \mathcal{L}_i \bigcirc (\mathcal{L}_j \bigcirc \mathcal{L}_k)$.

Proof. (Associativity $\Rightarrow$) Let the clause $l \in (\mathcal{L}_i \bigcirc \mathcal{L}_j) \bigcirc \mathcal{L}_k$, then either $l \in (\mathcal{L}_i \bigcirc \mathcal{L}_j)$ or $l \in \mathcal{L}_k$ (or both). This means that either $l \in \mathcal{L}_i$, or $l \in \mathcal{L}_j$, or $l \in \mathcal{L}_k$. Thus $l \in \mathcal{L}_i$ or $l \in (\mathcal{L}_j \bigcirc \mathcal{L}_k)$. This means that $l \in \mathcal{L}_i \bigcirc (\mathcal{L}_j \bigcirc \mathcal{L}_k)$ where $(\mathcal{L}_i \bigcirc \mathcal{L}_j) \bigcirc \mathcal{L}_k \subset \mathcal{L}_i \bigcirc (\mathcal{L}_j \bigcirc \mathcal{L}_k)$.
(Associativity $\Leftarrow$) Let the clause $l \in \mathcal{L}_i \bigcirc (\mathcal{L}_j \bigcirc \mathcal{L}_k)$, then either $l \in \mathcal{L}_i$ or $l \in (\mathcal{L}_j) \bigcirc \mathcal{L}_k)$ (or both). This means that either $l \in \mathcal{L}_i$, or $l \in \mathcal{L}_j$, or $l \in \mathcal{L}_k$. Thus $l \in (\mathcal{L}_i \bigcirc \mathcal{L}_j)$ or $l \in \mathcal{L}_k$. This means that $l \in (\mathcal{L}_i \bigcirc \mathcal{L}_j) \bigcirc \mathcal{L}_k$ where $\mathcal{L}_i \bigcirc (\mathcal{L}_j \bigcirc \mathcal{L}_k) \subset (\mathcal{L}_i \bigcirc \mathcal{L}_j) \bigcirc \mathcal{L}_k$.

5 Licenses Composition

The result of the Compatibility Evaluation algorithm is an answer of the kind `true/false`. If the answer is `false`, then the licenses considered for composition are not compatible. In this case, we leave to the data provider to decide the strategy to deal with this situation: (i) the data queried by the consumer is returned without attaching any license, (ii) the data is returned together with the more constraining license among $\mathcal{L}(C)$, and (iii) the data is returned together with the less constraining license among $\mathcal{L}(C)$.

If the licenses, instead, are compatible then we compose the licenses in $\mathcal{L}(C)$ such that the resulting composite license $\mathcal{L}_c$ satisfies the licensing clauses specified by the licenses in $\mathcal{L}(C)$. In particular, the definition of the composite license satisfies the following properties, where $\odot$ is the composition relation. First, the composite license $\mathcal{L}_c$ can be generated only if all the licenses composing it are compatible (Section 4).

Property 3. Let $\mathcal{L}_1, \ldots, \mathcal{L}_n$ be n licenses to be composed in a composite license $\mathcal{L}_c$. If it holds that $\mathcal{L}_1 \bigcirc \mathcal{L}_2 \bigcirc \ldots \bigcirc \mathcal{L}_n$, then $\mathcal{L}_c = \mathcal{L}_1 \odot \ldots \odot \mathcal{L}_n$.

Second, the composite license, if allowed, is consistent with the set of licenses used to compose it, due to the Compatibility Evaluation module.

Property 4. Let $\mathcal{L}_1(R_1), \ldots, \mathcal{L}_n(R_n)$ be the licenses associated to named graphs $R_1, \ldots, R_n$. The composite license $\mathcal{L}_c$ for $R_1, \ldots, R_n$ is consistent with $\mathcal{L}_1(R_1), \ldots, \mathcal{L}_n(R_n)$.

We now define the algorithm which generates $\mathcal{L}_c$ from the set of *compatible* licenses. The first step consists in providing the redefinition rules the algorithm applies in case a subsumption relation holds among these license elements. Table 1 shows the redefinition rules given the subsumption relation among the elements. We identify two kinds of redefinition: a redefinition where the more permissive element is included in $\mathcal{L}_c$, and another redefinition which follows the converse. The second step consists in detailing the composition which results from the rules expressed in Table 3. The composition of these license elements is achieved by assigning *both* these elements to the composite license $\mathcal{L}_c$. These rules are necessary to keep the consistency of $\mathcal{L}_c$ w.r.t. the licenses composing it. These rules overcome the compatibility rules against *Unspecified*, and the heuristics which guide the composition. This leads to the generation of some popular licensing terms: Attribution-ShareAlike, Attribution-NonCommercial, and ShareAlike-NonCommercial, namely. This is a minimum set of *composed* licenses. However, other composition rules can be defined, depending on the requirements of the data provider.

The algorithm for Composite License Generation is listed in Figure 6. The pre-requisite for license composability is that the licenses in $\mathcal{L}(C)$ are compatible with each others. If so, $\mathcal{L}(C)$ are composed as follows: the `ElementsSelectionHeuristic` procedure adopts the heuristic h to combine the elements of each license together in a single license $\mathcal{L}_c$.

Algorithm 4: Composite License Generation Algorithm

Input: Compatible Licenses $\mathcal{L}_1, \ldots, \mathcal{L}_n$
Output: Composite license $\mathcal{L}_c$
$\mathcal{L}_c \leftarrow$ `ElementsSelectionHeuristic`$_h(\mathcal{L}(C))$;
return $\mathcal{L}_c$;

Fig. 6. The procedure for generating the composite license.

The three basic heuristics we consider in our framework are the following [2]:

OR-composition : If there is at least one of the licenses involved in the composition that owns a clause then also the composite license owns it;

AND-composition : if all the licenses involved in the composition own a clause then also the composite license owns it;

Constraining-value : the most constraining clause among the ones offered by the licenses is included in the composite license.

In this paper, we do not argue in favor or against these three possibilities, and a qualitative evaluation of the *best* composition heuristics is left as future work. We leave to the data provider the choice of her best strategy for composing the licenses, e.g., AND-composition typically leads to a shorter and simpler license, while OR-decomposition leads to a more complex license where all the clauses present in $\mathcal{L}(C)$ are included.

```
@prefix cc: http://creativecommons.org/ns.
@prefix : http://example/licenses.

:lic1 a cc:License;
      cc:permits cc:Reproduction;
      cc:requires cc:ShareAlike.
```

(a)

```
@prefix cc: http://creativecommons.org/ns.
@prefix : http://example/licenses.

:lic2 a cc:License;
      cc:permits cc:DerivativeWorks;
      cc:prohibits cc:CommercialUse.
```

(b)

```
@prefix cc: http://creativecommons.org/ns.
@prefix : http://example/licenses.

:licComposite a cc:License;        THE ELEMENTS
              cc:permits cc:DerivativeWorks;
              cc:requires cc:ShareAlike;
              cc:prohibits cc:CommercialUse.

                 THE MODELS:
                 - Permission
                 - Requirement
                 - Prohibition
```

(c)

Fig. 7. Compatibility evaluation and composition which leads to a `true` result.

Example 2. Assume we want to analyze two licenses $\mathcal{L}_1$ and $\mathcal{L}_2$, visualized in Figure 7.a-b. Following our Compatibility Evaluation algorithm, we first compare the two licenses at the model level. Both licenses contain the model `cc:Permission`, the first element is `cc:Reproduction`, and the second element is `cc:DerivativeWorks`. Even if the two elements are not the same, we know from Table 1 that there is a subsumption relation between the element `cc:Reproduction` and `cc:DerivativeWorks`. This means that we can redefine one element as the other element when we build the composite license. The first license does not contain the model `cc:Prohibition` and the second license does not contain the model `cc:Requirement`. However, the two respective elements `cc:ShareAlike` and `cc:CommercialUse` are compatible due to Table 3, and then the two licenses are compatible. Thus composition is allowed, following Table 3 and the composition rules defined above. $\mathcal{L}_c$ is obtained using the more permissive permission (Table 1), and the OR-composition heuristic (Figure 7.c). $\mathcal{L}_c$ permits `cc:DerivativeWorks` (from the redefinition rules of Table 1), requires `cc:ShareAlike`, and prohibits `cc:CommercialUse`.

6 Conclusions

In this paper, we present a framework to associate licensing terms to the data on the Web of Data. The result of the consumer's query is not only the requested data but also its licensing terms. We adopt the CC licenses vocabulary because it provides a general schema for data licensing, and we align it with the other vocabularies including license specifications. First, we verify the compatibility among the licenses associated to the various named graphs affected by the consumer's query. Second, given the compatibility among the composed licenses, we construct $\mathcal{L}_c$ which aggregates the clauses of the distinct licenses following the heuristics we propose. The composite license is finally returned to the consumer together with the requested data.

The first point to be addressed is a legal and social validation of the proposed framework, answering research questions like "Can data be licensed at all depending on the jurisdiction?" and "Can non CC licenses be related to CC licenses in a legally correct way to extend the framework to non CC licenses and waivers?". This means to evaluate, not only quantitatively the performance of the algorithms to retrieve remote licensing statements referenced in the data, but also the legal value of the composite license. We will address an evaluation of the disparate named graphs that might be used in a "typical" query, and their relative proportions that have compatible or incompatible licenses. Moreover, we plan to integrate our framework with the W3C provenance working group's activities[21]. Finally, we are defining more complex heuristics like the one looking for the minimal composite license.

References

1. H. Abelson, B. Adida, M. Linksvayer, and N. Yergler. *ccREL: The Creative Commons Rights Expression Language*, 2008.
2. M. Comerio. *Web Service Contracts: Specification, Selection and Composition.* PhD thesis, University of Milano-Bicocca, 2009.
3. G. R. Gangadharan, H. L. Truong, M. Treiber, V. D'Andrea, S. Dustdar, R. Iannella, and M. Weiss. Consumer-specified service license selection and composition. In *ICCBSS, IEEE*, pages 194–203, 2008.
4. G. R. Gangadharan, M. Weiss, V. D'Andrea, and R. Iannella. Service license composition and compatibility analysis. In *ICSOC, LNCS 4749*, pages 257–269, 2007.
5. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space.* Morgan & Claypool, 2011.
6. R. Iannella. *Open Digital Rights Language (ODRL)*, 2002.
7. M. Krötzsch and S. Speiser. Sharealike your data: Self-referential usage policies for the semantic web. In *ISWC, LNCS 7031*, pages 354–369, 2011.
8. P. Miller, R. Styles, and T. Heath. Open data commons, a license for open data. In *LDOW*, 2008.
9. H. L. Truong, G. R. Gangadharan, M. Comerio, S. Dustdar, and F. De Paoli. On analyzing and developing data contracts in cloud-based data marketplaces. In *APSCC, IEEE*, pages 174–181, 2011.

[21] `http://www.w3.org/2011/prov/`