Legal Knowledge and Information Systems
A. Wyner and G. Casini (Eds.)

doi:10.3233/978-1-61499-838-9-1

Normative Requirements as Linked Data

Fabien GANDON [a,1,2], Guido GOVERNATORI [b] and Serena VILLATA [a]
[a] *Université Côte d'Azur, Inria, CNRS, I3S, France*
[b] *Data61, CSIRO, Australia*

Abstract. In this paper, we propose a proof of concept for the ontological representation of normative requirements as Linked Data on the Web. Starting from the LegalRuleML ontology, we present an extension of this ontology to model normative requirements and rules. Furthermore, we define an operational formalization of the deontic reasoning over these concepts on top of the Semantic Web languages.

Keywords. Linked data, Semantic Web, Deontic rules, Ontology

1. Introduction

The Linked Data principles [3] provide a standard approach to weave a Web of data, linking datasets across the world and virtually in any domain. The semantic Web frameworks additionally provide standard means to publish data (RDF [4]), ontological knowledge (RDFS [5] and OWL [6] schemata), and to query and reason on them (SPARQL [7]). Despite existing approaches to model legal ontological knowledge [9,1,2], little work has been devoted towards the definition of an end-to-end framework to represent, publish and query ontological knowledge from the legal domain using such standards. In this paper, we study how Semantic Web frameworks could apply to the formalization, publication and processing of legal knowledge, and in particular, *normative requirements* and *rules*.

A linked data based deontic representation and reasoning allows us to (a) rely on Web standard to represent, exchange and foster interoperability between deontic rule bases and reasoning systems, (b) rely on existing standards (e.g. SPARQL) and infrastructures (e.g. triple stores) to implement deontic systems, and (c) combine linked data and semantic Web reasoning and formalisms (e.g., OWL) with deontic reasoning to support more inferences.

Our research question is: *Can we represent and reason on the deontic aspects of normative rules with standard Semantic Web languages?* We focus here on two sub-questions: *For which aspects schema-based reasoning (RDFS, OWL) is relevant?* and *Can we operationally formalize other deontic reasoning rules with RDF and SPARQL?*

We first survey the related work to show that current legal vocabularies on the Semantic Web do not provide the expressiveness we need (Section 2). Then we specify and formalize of the ontology we require (Section 3). We describe how normative requirements

[1] Corresponding Author: Fabien Gandon, Inria, Wimmics, 2004 rt des Lucioles, 06902 France; E-mail: fabien.gandon@inria.fr, http://fabien.info.

[2] The authors have received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 690974 for the project "MIREL: MIning and REasoning with Legal texts".

can be represented as Linked Data (Section 4), and why the states of affairs should be represented as RDF 1.1 named graphs (Section 5). Relying on this modeling, we show that some aspects of deontic reasoning cannot be covered by the OWL formalization whilst they can be captured with SPARQL rules (Section 6). We experiment this approach with a proof of concept (Section 7) before concluding.

2. Related Work

We performed a search[3] on LOV [8], a directory of Semantic Web vocabularies and schemata, to see how legal concepts are covered in published ontologies. Among the retrieved vocabularies, we identified that:

- the General Ontology for Linguistic Description (GOLD) includes a "Deontic Modality" concept [4] but it is essentially defined from a linguistic point of view with the goal to perform natural language analysis.
- the Public Procurement Ontology (PPROC) has the notion of "Contract additional obligations" which is a class limited to describing the additional obligations a contract requires[5].
- the Open Standards for Linking Governments Ontology (OSLO) includes an upper class "permission", but attached to the role of an individual in a society[6].
- the notions of rights, permissions and licenses are mentioned in schemata such as Dublin Core[7], Creative Commons[8] or ODRL[9] but to describe the possible uses of a digital resource and they remain at a descriptive non-formalized level.

Current ontologies are often limited to a specific domain of application and have very shallow coverage of deontic concepts. They are not designed with the goal to support deontic reasoning above Semantic Web frameworks. Their primitives are designed to annotate resources with the goal of documenting or supporting some degree of interoperability, but they are not intended to support Semantic Web based reasoning and processing of the normative requirements and rules. Closer to our goal is the LegalRuleML Meta Model [9] providing primitives for deontic rule and normative requirement representation (Permission, Obligation, Prohibition). We started from this model and extended it with a new ontology focusing on the deontic aspects, integrating notions from an existing abstract formal framework for normative requirements of regulatory compliance [10], and previous on modal defeasible reasoning for deontic logic on the Semantic Web [11]

3. Ontological extension of the LegalRuleML Meta Model

In this section, we first describe the competency questions that motivate our extension of the LegalRuleML ontology, and then we detail the core concepts of our new legal ontology as well as their formalization in OWL.

[3] Keywords include: obligation, prohibition, permission, rights and licences.
[4] http://purl.org/linguistics/gold/DeonticModality
[5] http://linguistics-ontology.org/gold/DeonticModality
[6] http://purl.org/oslo/ns/localgov#Permission
[7] http://dublincore.org/
[8] https://creativecommons.org/ns
[9] http://w3c.github.io/poe/vocab/

3.1. Motivating scenarios and competency questions

Among the many approaches to design an ontology [12], the writing of motivating scenarios is a very usual initial step of specifications to capture problems that are not adequately addressed by existing ontologies [13]. The motivating scenario for us here is to support the annotation, detection and retrieval of normative requirements and rules. We want to support users in information retrieval with the ability to identify and reason on the different types of normative requirements and their statuses. This would be possible through ontology population approaches, but the lack of an existing ontology covering these aspects slows this process, as well as the further development of more advanced applications in legal computer science.

In a second step of ontology specification, a standard way to determine the scope of the ontology is to extract from the scenarios the questions and answers it should be able to support if it becomes part of knowledge-based system. These so-called *competency questions* [13] place demands on the targeted ontology, and they provide expressiveness requirements. The competency questions we target for this ontology are:

- What are the instances of a given requirement and its sub-types, e.g. obligation?
- Is a requirement violated by one or more states of affairs, and if so, which ones?
- Is a given description of rules and states of affairs coherent?
- Which rules, documents and states of affairs are linked to a requirement and how?

3.2. Core primitives

To support the competency questions and relying on definitions from LegalRuleML [9] and deontic reasoning [10,11], we identified a set of core primitives for an ontology capturing the different aspects of normative requirements, and supporting the identification and classification tasks. We called that ontology Normative Requirement Vocabulary (NRV), and made it available and dereferenceable following the Linked Data principles. The namespace is `http://ns.inria.fr/nrv#` with the preferred prefix `nrv` respectively submitted both to LOV [8] and to `http://prefix.cc`.

The top class of the ontology is the `Normative Requirement` which is defined as the set of the requirements implying, creating, or prescribing a norm. Then we have a number of upper classes to capture different features of the requirements:

- `Compensable Requirement`, `Non Compensable Requirement`, `Compensated Requirement` are classes of requirements with different compensation statuses.
- the classes `Violable requirement`, `Non Violable Requirement`, `Violated Requirement` and `Compliant Requirement` characterize the requirements with respect to their relation to a `Compliance` or a `Violation`.
- the other classes follow the same logic, and they distinguish requirements with respect to their perdurance, persistence, co-occurance and preemptiveness.

Using these upper classes, we positioned and extended three primitives from the LegalRuleML Meta Model (i.e., `Prohibition`, `Permission`, `Obligation`), each one inheriting from the appropriate super classes we introduced. For instance, `Permission` inherits from `Non Violable Requirement` and `Non Compensable Requirement`, while `Obligation` inherits from `Violable Requirement` and `Compensable Requirement`. Specializations of these classes are then used to introduce the notions of `Achievement`, `Maintenance` and

`Punctual`. For the complete list of classes and their definitions, we refer the reader to the online documentation available at the namespace URL. These primitives and definitions provide the taxonomic skeleton of our NRV ontology.

3.3. Formalization

In this section, we provide some formalization details (ontological commitment) and their translation into OWL (computational commitment). We will use the TriG syntax [14] for RDF, and the prefixes we use in the rest of this article are:

```
lrmlmm: http://docs.oasis-open.org/legalruleml/ns/v1.0/metamodel#
owl: http://www.w3.org/2002/07/owl#
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs: http://www.w3.org/2000/01/rdf-schema#
rulemm: http://docs.oasis-open.org/legalruleml/ns/v1.0/rule-metamodel#
xml: http://www.w3.org/XML/1998/namespace
xsd: http://www.w3.org/2001/XMLSchema#
nrv: http://ns.inria.fr/nrv#
nru: http://ns.inria.fr/nrv-inst#
```

We captured the disjointedness expressed in the upper classes representing exclusive characteristics of normative requirements (compensable / non-compensable, violable / non-violable, persistent / non persistent):

```
:NormativeRequirement a rdfs:Class ;
  owl:disjointUnionOf ( :CompensableRequirement :NonCompensableRequirement ) ;
  owl:disjointUnionOf ( :ViolableRequirement :NonViolableRequirement ) ;
  owl:disjointUnionOf ( :PersistentRequirement :NonPersistentRequirement ) .
```

We initially considered the disjointedness of a compliant requirement and a violated requirement, however this disjointedness is not global but local to a state of affairs and therefore it does not translate to a general disjointedness of classes, i.e., a requirement may be violated by a state of affairs but compliant with an other one at the same time. However, this led us to capture this issue as a property disjointedness, since a requirement cannot be violated and be compliant with the same state of affairs at the same time:

```
:hasCompliance a owl:ObjectProperty ; rdfs:label "has for compliance"@en ;
 rdfs:domain :ViolableRequirement ; rdfs:range lrmlmm:Compliance ;
 owl:propertyDisjointWith :hasViolation .
```

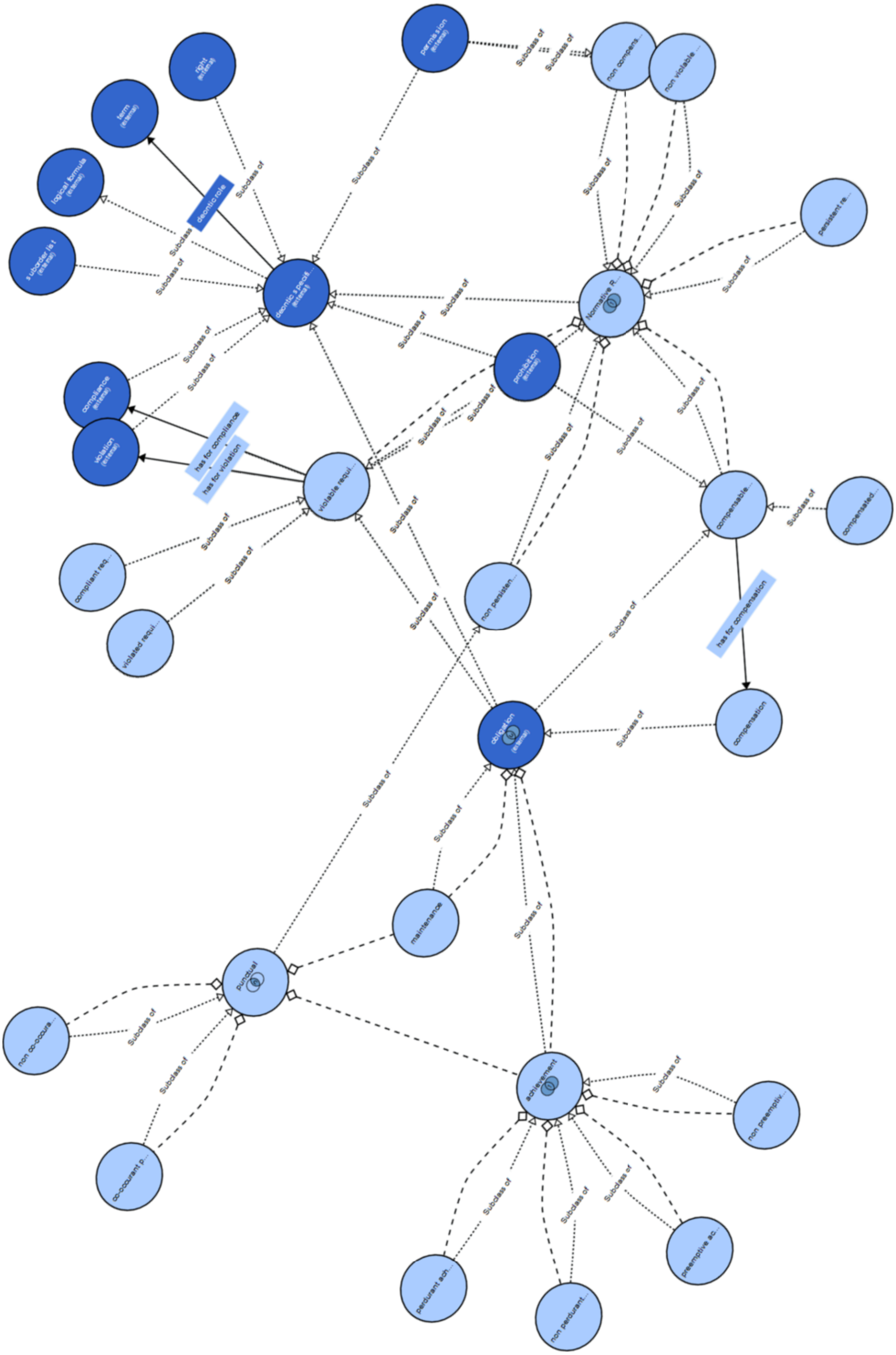
Obligations are an example of non disjoint union between achievements and maintenances, since a punctual requirement is both an achievement and a maintenance:

```
lrmlmm:Obligation a rdfs:Class ;
  rdfs:subClassOf :ViolableRequirement ;
  rdfs:subClassOf :CompensableRequirement ;
  owl:unionOf ( :Achievement :Maintenance )  .

:Achievement a rdfs:Class ; rdfs:label "achievement"@en ;
  owl:disjointUnionOf ( :PreemptiveAchievement :NonPreemptiveAchievement ) ;
  owl:disjointUnionOf ( :PerdurantAchievement :NonPerdurantAchievement ) ;
  rdfs:subClassOf lrmlmm:Obligation .

:Maintenance a rdfs:Class ; rdfs:label "maintenance"@en ;
  rdfs:subClassOf lrmlmm:Obligation .
```

Figure 1. Overview of the NRV ontology and its core primitives, in particular `Prohibition`, `Permission`, `Obligation` and a number of upper classes to capture different features of a `Normative Requirement`.



Violated and compensated requirements could be defined with restrictions on the properties `hasViolation` and `hasCompensation`:

```
:ViolatedRequirement a rdfs:Class ;
    rdfs:subClassOf :ViolableRequirement ;
    owl:equivalentClass  [ a owl:Restriction ;
       owl:onProperty :hasViolation ;
       owl:minCardinality 1 ] .

:CompensatedRequirement a rdfs:Class ;
    rdfs:subClassOf :CompensableRequirement ;
    owl:equivalentClass  [ a owl:Restriction ;
       owl:onProperty :hasCompensation ;
       owl:minCardinality 1 ] .
```

We could now be tempted to define a compliant requirement with the following restrictions:

```
1 :CompliantRequirement a rdfs:Class ; rdfs:label "compliant requirement"@en ;
2   rdfs:subClassOf :ViolableRequirement ;
3   owl:equivalentClass  [ a owl:Restriction ;
4     owl:onProperty :hasCompliance ;
5     owl:minCardinality 1 ] .
6   owl:equivalentClass  [ a owl:Restriction ;
7     owl:onProperty :hasViolation ;
8     owl:maxCardinality 0 ] .
```

However we removed the second part (lines 6-8) of the restriction since it re-introduces a disjunction between the compliant and violated requirement classes. The notions of compliance and violation are not generally disjoint but only disjoint locally to a state of affair, i.e., a normative requirement can be violated and compliant at the same time but with respect to different states of affairs. However, OWL definitions cannot rely on RDF 1.1 named graphs, which we will use for representing states of affairs. Therefore we will need another mechanism to capture this kind of constraints.

Because we used disjoint unions, the ontology is in OWL DL, i.e., $\mathcal{SHOIN}^{(\mathcal{D})}$, more precisely in the $\mathcal{AL}(\mathcal{U})\mathcal{C}(\mathcal{H})\mathcal{RN}$ family, i.e., $\mathcal{AL}$ attributive language, ($\mathcal{U}$ concept union), $\mathcal{C}$ complex concept negation, ($\mathcal{H}$ role hierarchy), $\mathcal{R}$ limited complex role inclusion axioms, reflexivity, irreflexivity, role disjointedness, and $\mathcal{N}$ cardinality restrictions.

We decided to declare the signature of properties (e.g., `hasViolation`, `hasCompensation`) at the ability level (e.g., violable requirement, compensable requirement), and not at the effective status level (e.g., violated requirement, compensated requirement) because each status will be local to a state of affairs. Therefore, in the end, we avoided too strong restrictions and signatures. If we remove cardinality restrictions, unions and disjointedness, the ontology becomes compatible with OWL EL and OWL RL which could be interesting for implementations relying on rule-based systems, especially when we consider the extensions proposed in the following sections.

4. Requirements as Linked Data

Using the LegalRuleML Meta Model and the NRV ontology we can now start to represent normative requirements as Linked Data. Let us introduce two examples. The first one is a rule stating that according to Australian law one cannot drive over 90km/h:

```
<http://gov.au/driving-rule> a lrmlmm:Source ;
   rdfs:label "driving rules in Australia"@en .
nru:LSS1 a lrmlmm:Sources ;
   lrmlmm:hasLegalSource <http://gov.au/driving-rule> .
nru:LRD1 a lrmlmm:LegalRuleMLDocument ;
   lrmlmm:hasLegalSources nru:LSS1 ;
   lrmlmm:hasAlternatives [ lrmlmm:fromLegalSources nru:LSS1 ;
                            lrmlmm:hasAlternative nru:PS1 ] ;
   lrmlmm:hasStatements nru:SS1 .
nru:SS1 a lrmlmm:Statements ;
  lrmlmm:hasStatement nru:PS1 .
nru:PS1 a lrmlmm:PrescriptiveStatement, lrmlmm:Prohibition ;
  rdfs:label "can’t drive over 90km"@en .
```

The second example is a rule stating that employees of CSIRO must wear their badges:

```
<http://csiro.au/security-rule> a lrmlmm:Source ;
  rdfs:label "security rules in CSIRO"@en .
nru:LSS2 a lrmlmm:Sources ;
  lrmlmm:hasLegalSource <http://csiro.au/security-rule>  .
nru:LRD2 a lrmlmm:LegalRuleMLDocument ;
  lrmlmm:hasLegalSources nru:LSS2 ;
  lrmlmm:hasAlternatives [ lrmlmm:fromLegalSources nru:LSS2 ;
                           lrmlmm:hasAlternative nru:PS2 ] ;
  lrmlmm:hasStatements nru:SS2 .
nru:SS2 a lrmlmm:Statements ;
  lrmlmm:hasStatement nru:PS2 .
nru:PS2 a lrmlmm:PrescriptiveStatement, lrmlmm:Obligation ;
  rdfs:label "you must wear your badge inside CSIRO facilities"@en .
```

5. State of affairs as named graphs.

The ability to define contexts and group assertions was one of the main motivations for having named graphs in RDF 1.1 [15]. The notion of state of affairs at the core of deontic reasoning is naturally captured by named graphs where all the statements of each state of affairs are encapsulated as RDF triples in a named graph, identifying that precise state of affairs. We provide here four examples of states of affairs respecting (2 and 3) or breaking (1 and 4) the rules of the normative statements described above. The core idea is to represent each state of affairs as a named graph typed as a factual statement of LegalRuleML.

```
:StateOfAffairs1 a lrmlmm:FactualStatement .
GRAPH :StateOfAffairs1 { rdfs:label "Tom" ;
  :Tom :activity [ a :Driving ;
                  :speed "100"^^xsd:integer ;
                  rdfs:label "driving at 100km/h"@en ] . }
:StateOfAffairs2 a lrmlmm:FactualStatement .
GRAPH :StateOfAffairs2 {
  :Jim :activity [ a :Driving ; rdfs:label "Jim" ;
                   :speed "90"^^xsd:integer ;
                   rdfs:label "driving at 90km/h"@en ] . }
:StateOfAffairs3 a lrmlmm:FactualStatement .
GRAPH :StateOfAffairs3 { rdfs:label "Jane" ;
 :Jane :location [ rdf:value :CSIRO ;
                   :start "2017-07-18T09:30:10+09:00"^^xsd:date ;
```

```
                    :end   "2017-07-18T17:00:10+09:00"^^xsd:date ] ;
        :badge   [ rdf:value :CSIRO ;
                    :start "2017-07-18T09:30:10+09:00"^^xsd:date ;
                    :end   "2017-07-18T17:00:10+09:00"^^xsd:date ] . }
:StateOfAffairs4 a lrmlmm:FactualStatement .
GRAPH :StateOfAffairs4 { rdfs:label "Steve" ;
 :Steve :location [ rdf:value :CSIRO ;
                     :start "2017-07-18T09:30:10+09:00"^^xsd:date ;
                     :end   "2017-07-18T17:00:10+09:00"^^xsd:date ] ;
        :badge    [ rdf:value :CSIRO ;
                     :start "2017-07-18T10:30:10+09:00"^^xsd:date ;
                     :end   "2017-07-18T17:00:10+09:00"^^xsd:date ] . }
```

6. Deontic reasoning as SPARQL rules

Since the notion of named graph that appeared with RDF 1.1 (2014, [4]) is absent from OWL 2 (2012, [6]) and its constructors, we need to implement the reasoning on states of affairs by other means. The SPARQL language is both a standard and a language able to manipulate named graphs so we propose to use SPARQL rules. In this section, we explore the coupling of OWL reasoning with SPARQL rules to formalize and implement some deontic reasoning. Description Logics (DL) support reasoning on the description of concepts and properties of a domain (terminological knowledge or T-Box) and of their instances (assertional knowledge or A-box). They are the basis of the Web Ontology Language (OWL). The classical inferences supported by DL are instance checking, relation checking, subsumption checking, and consistency checking [16]. While these inferences are useful to reason about deontic knowledge (e.g., a compensable requirement must also be a violable requirement), they do not cover all the inferences we want to support here in particular deontic rules (e.g., a requirement is violated by a state of affairs if, during a specific period of time, a given constraint does not hold). These rules rely on complex pattern matching including, for instance, temporal interval comparison that go beyond OWL expressiveness. As a proof of concept, the following rules check the violation or compliance of the statements made by the previous states of affairs. The core idea is to add to each named graph of each state of affairs the deontic conclusions of the legal rules relevant to it. By relevant we mean here that the state of affairs describes a situation that falls under the application conditions of that legal rule. The following rules update compliance and violation for the driving speed requirement:

```
DELETE {  graph ?g { nru:PS1 nrv:hasCompliance ?g } }
INSERT {  graph ?g { nru:PS1 a nrv:ViolatedRequirement ;
                     nrv:hasViolation ?g } }
WHERE  {  graph ?g {  ?a a :Driving ; :speed ?s .  }
          FILTER (?s>90) }  ;
DELETE { graph ?g { nru:PS1 a nrv:ViolatedRequirement ;
                    nrv:hasViolation ?g } }
INSERT { graph ?g { nru:PS1 nrv:hasCompliance ?g } }
WHERE  { graph ?g {  ?a a :Driving ; :speed ?s . }
         FILTER (?s<=90) }
```

The following rules update compliance and violation for the CSIRO badge requirement:

```
INSERT { graph ?g { nru:PS2 a nrv:ViolatedRequirement ; nrv:hasViolation ?g }}
WHERE  { graph ?g { ?x :location [ rdf:value ?o ; :start ?ls ; :end ?le ]
```

```
         optional { ?x :badge    [ rdf:value ?o ; :start ?bs ; :end ?be ] .
         FILTER (?bs<=?ls && ?be>=?le) } }
        FILTER (  ( ! bound (?bs))   ) }     ;
INSERT { graph ?g { nru:PS2 nrv:hasCompliance ?g } }
WHERE  { graph ?g { ?x :location [ rdf:value ?o ; :start ?ls ; :end ?le ]
                    ?x :badge    [ rdf:value ?o ; :start ?bs ; :end ?be ] .  }
         FILTER (?bs<=?ls && ?be>=?le) }
```

The following rules update compliance for the state of affairs after violations were checked:

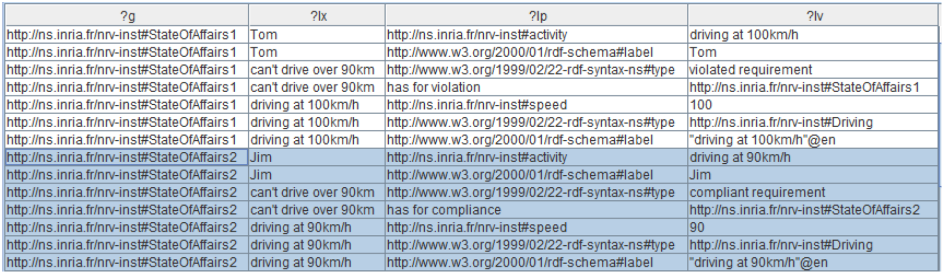
```
INSERT {  graph ?g {?n a nrv:CompliantRequirement }  }
WHERE  {  ?g  a lrmlmm:FactualStatement .
          ?n a nrv:ViolableRequirement .
          graph ?g {  ?n  nrv:hasCompliance ?g }
          minus { graph ?g {  ?n  nrv:hasViolation ?g } } }   ;
DELETE {  graph ?g {?n a nrv:CompliantRequirement }  }
WHERE  {  ?g  a lrmlmm:FactualStatement .
          ?n a nrv:ViolableRequirement .
          graph ?g {  ?n  nrv:hasViolation ?g } }
```

7. Proof of concept and experimentation

To validate and experiment with the ontology, the Linked Data and the rules, we used two established tools:

- the latest version of the Protégé platform [17] and the reasoners it includes were used to check the NRV OWL ontology which was found coherent and consistent.
- the latest version of CORESE [18] was used to load the LegalRuleML and NRV ontologies, the Linked Data about the rules and the states of affairs, and the SPARQL rules to draw the conclusions as shown in Figure 2 for the two first states of affairs concerning speed limitation.

Figure 2. Extract of the quadruples (N-Quads) produced by CORESE after all the reasoning on the two first states of affairs concerning speed limitation showing one violated state (white background) and one compliant one (blue background). The columns indicate the named graph of the state of affairs (`?g`), the subjects (`?lx`), the predicates (`?lp`), and the objects (`?lv`) of the triples in this named graph.



?g	?lx	?lp	?lv
http://ns.inria.fr/nrv-inst#StateOfAffairs1	Tom	http://ns.inria.fr/nrv-inst#activity	driving at 100km/h
http://ns.inria.fr/nrv-inst#StateOfAffairs1	Tom	http://www.w3.org/2000/01/rdf-schema#label	Tom
http://ns.inria.fr/nrv-inst#StateOfAffairs1	can't drive over 90km	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	violated requirement
http://ns.inria.fr/nrv-inst#StateOfAffairs1	can't drive over 90km	has for violation	http://ns.inria.fr/nrv-inst#StateOfAffairs1
http://ns.inria.fr/nrv-inst#StateOfAffairs1	driving at 100km/h	http://ns.inria.fr/nrv-inst#speed	100
http://ns.inria.fr/nrv-inst#StateOfAffairs1	driving at 100km/h	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://ns.inria.fr/nrv-inst#Driving
http://ns.inria.fr/nrv-inst#StateOfAffairs1	driving at 100km/h	http://www.w3.org/2000/01/rdf-schema#label	"driving at 100km/h"@en
http://ns.inria.fr/nrv-inst#StateOfAffairs2	Jim	http://ns.inria.fr/nrv-inst#activity	driving at 90km/h
http://ns.inria.fr/nrv-inst#StateOfAffairs2	Jim	http://www.w3.org/2000/01/rdf-schema#label	Jim
http://ns.inria.fr/nrv-inst#StateOfAffairs2	can't drive over 90km	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	compliant requirement
http://ns.inria.fr/nrv-inst#StateOfAffairs2	can't drive over 90km	has for compliance	http://ns.inria.fr/nrv-inst#StateOfAffairs2
http://ns.inria.fr/nrv-inst#StateOfAffairs2	driving at 90km/h	http://ns.inria.fr/nrv-inst#speed	90
http://ns.inria.fr/nrv-inst#StateOfAffairs2	driving at 90km/h	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://ns.inria.fr/nrv-inst#Driving
http://ns.inria.fr/nrv-inst#StateOfAffairs2	driving at 90km/h	http://www.w3.org/2000/01/rdf-schema#label	"driving at 90km/h"@en

8. Conclusions

In this paper, we addressed the issue that current vocabularies on the Semantic Web do not provide the expressiveness required to support deontic reasoning on normative requirements and rules. As a contribution, we specified and formalized an ontology extending LegalRuleML, and we showed how it can be used to represent normative requirements as Linked Data with states of affairs represented as RDF 1.1 named graphs. Relying on this modeling, we proposed an approach based on SPARQL rules to cover some of the deontic aspects outside the expressiveness of OWL 2, and we experiment this approach with a proof of concept based on two established tools of the Semantic Web community. Future work includes extensive population and testing of the ontology on larger datasets and cases. In particular, we intend to go beyond the proof of concept by evaluating this end-to-end approach based on the Semantic Web languages on a business process compliance checking scenario [10]. Extensions of this work also include the possibility to represent differentiated classes of validity that would correspond to the actual structure of our legal system and non-binary modes that would be fit to process proportionality of legal principles. The introduction of a complete rule-based system is part of our future directions as well.

References

[1] Hoekstra, R.; Breuker, J.; Bello, M. D.; and Boer, A. 2007. The LKIF core ontology of basic legal concepts. In *Proc. of the Workshop on Legal Ontologies and AI Techniques*.

[2] Sartor, G.; Casanovas, P.; Biasiotti, M.; and Fernndez-Barrera, M. 2013. *Approaches to Legal Ontologies: Theories, Domains, Methodologies*. Springer.

[3] T. Heath, C. Bizer, *Linked data: Evolving the web into a global data space*, Synthesis lectures on the semantic web: theory and technology 1.1 (2011): 1-136.

[4] World Wide Web Consortium, *RDF 1.1 concepts and abstract syntax*, W3C Recommendation, 25 February 2014

[5] World Wide Web Consortium, *RDF Schema 1.1*, W3C Recommendation, 25 February 2014

[6] World Wide Web Consortium, *OWL 2 Web Ontology Language*, Document Overview (2nd Edition), W3C Recommendation 11 December 2012

[7] World Wide Web Consortium, *SPARQL Query Language for RDF*, W3C Recommendation 15 January 2008

[8] Vandenbussche, Pierre-Yves, et al. "Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web." Semantic Web 8.3 (2017): 437-452.

[9] Athan, Tara, et al., OASIS LegalRuleML, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, ACM, 2013

[10] M. Hashmi, G. Governatori, M.T. Wynn, Normative requirements for regulatory compliance: An abstract formal framework. *Omnes Information Systems Frontiers* 18(3), pp. 429-455, 2016

[11] K. Efstratios, N. Bassiliades, G. Governatori, G. Antoniou, A Modal Defeasible Reasoner of Deontic Logic for the Semantic Web, *International Journal on Semantic Web and Information Systems*, (IJSWIS) 7 (2011): 1, doi:10.4018/jswis.2011010102

[12] F. Gandon, *Ontology Engineering: a Survey and a Return on Experience*, RR-4396, INRIA. 2002.

[13] M. Uschold, M. Gruninger, Ontologies: Principles, methods and applications, *The knowledge engineering review*, 11(2), 93–136, 1996, Cambridge University Press.

[14] World Wide Web Consortium, *RDF 1.1 TriG*, W3C Recommendation, 25 February 2014

[15] F. Gandon, O. Corby, Name That Graph or the need to provide a model and syntax extension to specify the provenance of RDF graphs., *W3C Workshop — RDF Next Steps*, Jun 2010, Palo Alto, United States.

[16] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, Applications*. CUP, 2003.

[17] N.F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen, Creating semantic web contents with protege-2000, *IEEE Intelligent Systems*, 16.2 (2001): 60-71.

[18] O. Corby, R. Dieng-Kuntz and C. Faron-Zucker, Querying the semantic web with corese search engine, *In Proc. of the 16th European Conf. on Artificial Intelligence*, pp. 705-709 (2004), IOS Press.