Legal Knowledge and Information Systems
M. Araszkiewicz and V. Rodríguez-Doncel (Eds.)

doi:10.3233/FAIA190321

Legal Compliance in a Linked Open Data Framework

Enrico FRANCESCONI [a,1], Guido GOVERNATORI [b]
[a] *Publications Office of the European Union (Luxembourg); Istituto di Informatica Giuridica e Sistemi Giudiziari – IGSG-CNR (Italy)*
[b] *Data61, CSIRO, Australia*

Abstract An approach for legal compliance representation and checking within a Linked Open Data framework is presented. It is based on modeling deontic norms in terms of ontology and ontology property restrictions. It is also shown how the approach can handle norm defeasibility. Such methodology is implemented by decidable fragments of OWL 2, while legal reasoning is implemented by available decidable reasoners.

Keywords. Legal reasoning, Norm compliance, Semantic Web, OWL 2

1. Introduction

Machine readable rules represent a precondition for developing legal information systems with automatic reasoning facilities. Approaches were proposed to formalize reasoning on deontic notions [1], norm compliance [2] or legal argumentation [3]. In the Semantic Web, languages as OWL/RDF(S) for modeling real world scenarios, and mainly SWRL or RIF for legal rules are typically used. Recently LegalRuleML for legal rules modeling and defeasible reasoning has been proposed [5]. The Linked Open Data (LOD) approach to the Semantic Web is producing a growing amount of RDF triples for concepts, rules and facts. LOD principles recommend OWL/RDF(S), while implementing OWL 2 decidable profiles[2] allows to use available reasoners [6]. In this paper we discuss a legal reasoning framework [7] based on the distinction between the concepts of *Provision* and *Norm*. In particular, an approach for norm compliance in the LOD framework, based on decidable OWL 2 profiles, is here presented and tested. In Section 2 the distinction between *Provision* and *Norm* is discussed [8]; in Sections 3 and 4 norms modeling by ontologies able to implement defeasible norm compliance reasoning within a decidable framework is described and tested; in Section 5 some conclusions are reported.

2. Provisions and Norms

The legal order can be seen as a legal discourse composed by linguistic entities or *speech acts* [9] with descriptive or prescriptive functions. Every linguistic entity can be seen in

[1]Corresponding Author: E. Francesconi, e-mail: enrico.francesconi@igsg.cnr.it
[2]like OWL 2 DL, OWL 2 EL, OWL 2 QL and OWL 2 RL

a twofold perspective: as a set of signs, organized in words and sentences representing a normative statement, typically called *Provision* [10] [11], as well as the meaning for application of such normative statement, typically called *Norm* [8] [13]. Provisions and related norms have different roles and properties pertaining to the different domain they operate at. A provision represents the building block of the legal order. A norm can either modify the text of other provisions (as for the amendments) or can introduce restrictions on the real world (in case of obligations, for example).

Let's consider two examples of rules:

R1: *The supplier shall communicate to the consumer all the contractual terms and conditions*

R2: *According to a [country] law one cannot drive over 90 km/h*

Both rules are speech acts, namely *Provisions* in specific regulations. The *Provision Model* [11] [12] describes provisions in terms of *types* (as *Definition*, *Duty/Obligation*, *Right/Permission*) and *attributes* (as the *Bearer* or the *Counterpart* of a *Right/Permission*). According to the Provision Model, R1 can be classified as an *Obligation* of a *Supplier* towards *Consumer*, while R2 as an *Obligation* for any *Driver* in the related specific country. A Provision Model annotation can support advanced legal information retrieval (see [12]). When we consider the application of R1 and R2 on specific facts, we actually talk about *Norms*. Real world scenarios and facts can be effectively represented by ontologies and related individuals, respectively. Norms, providing constraints on the reality, can be modeled as restrictions on ontology properties. Legal compliance checking is a process aiming to verify if a fact, occurring in the real world, complies with a legal norm (namely the related restrictions).

Hereinafter we illustrate our approach for modeling norms with the aim of implementing legal compliance checking in a decidable framework.

3. Modeling norms for legal compliance checking

The scenario of R1 can be modeled by an ontology including a class Supplier, having a boolean property hasCommunicatedConditions (see Fig. 1, myo: is a namespace for a fictitious ontology "MyOntology"). Norm R1, expressing an obligation, states that suppliers must communicate purchasing conditions to the consumers. In our approach norm R1 is represented as a restriction on the property hasCommunicatedConditions able to identify the class SupplierR1CompliantIndividuals of individuals for which the value of the property under consideration is "true" (Fig. 1). The individuals of the class Supplier complying with this norm are all and only those belonging to the subclass SupplierR1Compliant. Such a representation results in the OWL 2 DL profile, allowing us to use an OWL 2 DL decidable reasoner, as for example Pellet[3]. The inferred model produced by Pellet establishes the rdfs:subClassOf relation between SupplierR1Compliant and Supplier. Therefore, compliance checking according to R1 is a problem of checking if an individual of type Supplier belongs also to the class SupplierR1Compliant. As a concrete example let's consider the two individuals myo:s1 and myo:s2 (Fig. 1) of Supplier. myo:s1 is an individual not compliant with R1, while myo:s2 is complaint with R1. The following SPARQL query:

[3]https://github.com/stardog-union/pellet

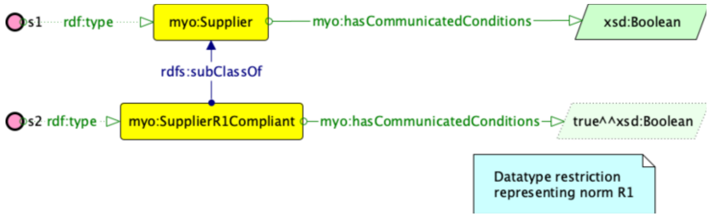
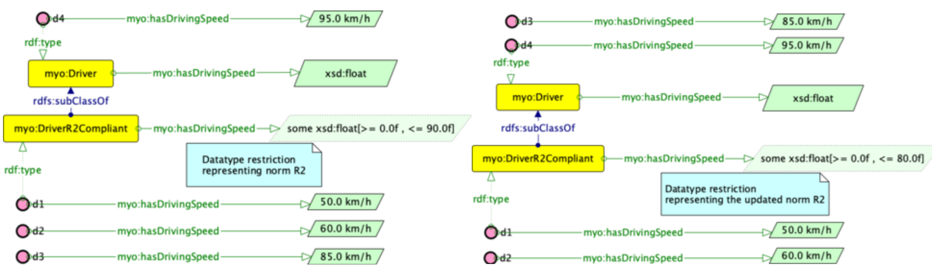


Figure 1. Norm R1 as restriction on the property hasCommunicatedConditions and examples of compliant and non-compliant individuals (the subclass relation between SupplierR1Compliant and Supplier is inferred).

```
SELECT ?x WHERE { ?x rdf:type myo:SupplierR1Compliant }
```

is able to select the individuals which are complaint with R1 (in our case s2).

In case of R2, the vehicles circulation scenario can be modeled in terms of an ontology including a class Driver, having a datatype property hasDrivingSpeed with range in the xsd:float datatype (Fig. 2a). Norm R2, expressing an obligation, states that, ac-



(a) Original norm R2 (b) New version of norm R2

Figure 2. Norm R2 represented as restriction on property hasDrivingSpeed and examples of compliant and non-compliant individuals.

cording to the related country law, one cannot drive over 90 km/h. In our approach norm R2 is represented as a restriction on the property myo:hasDrivingSpeed able to identify the class DriverR2CompliantIndividuals of individuals for which the values of the property under consideration are in the range [0.0, 90.0] km/h (Fig. 2a). In OWL 2 this can be expressed by the xsd:minInclusive and xsd:maxInclusive datatype bound properties. The individuals of the class Driver complying with this norm are all and only those belonging to the subclass DriverR2Compliant. Also such R2 modeling results in the OWL 2 DL decidable profile. The inferred model establishes a rdfs:subClassOf relation between DriverR2Compliant and Driver. Therefore, compliance checking for norm R2 is a problem of checking if an individual of type Driver belongs also to the class DriverR2Compliant. As a concrete example, let's consider the four individuals of the class Driver shown in Fig. 2a. The individual myo:d4 is not compliant with R2 having speed 95.0 Km/h $\geq$ 90.0 Km/h (Fig. 2a). The following query:

```
SELECT ?x WHERE { ?x rdf:type myo:DriverR2Compliant }
```

is able to select the individuals complaint with R2 (here myo:d1, myo:d2 and myo:d3).

In both the previous examples norm compliance checking is performed in a LOD framework within a decidable profile.

4. Modeling norms defeasibility for legal compliance checking

Defeasibility is a broad concept in the legal domain including "contrary to evidence" reasoning in argumentation systems [4], as well as reasoning with norm conflicts or norm exceptions [14] in normative systems. Two examples, one dealing with norm conflict and one with norm exception, are here modeled within a description logic framework able to provide support for defeasible reasoning, for example in norm compliance checking.

As first example let's consider rule R2, modeled in Section 3, and the following new version of rule R2, introducing a more strict driving speed limit at 80.0 Km/h:

R2 : *According to a [country] law one cannot drive over 80 km/h*

The *new version of R2* can defeat the previous compliance conclusions, in the sense that individuals, compliant with the *previous version of R2*, might not be compliant with it anymore. To cope with this norm change, the same model can be updated (without any change in class or property names) by changing the original datatype property restriction on myo:hasDrivingSpeed with a new one expressed by the *new version of R2*, as shown in Fig. 2b. Without changing anything on the individuals, their membership to the class myo:DriverR2Compliant changes so that, for example, the individual myo:d3, compliant with the previous version of R2 (Fig. 2a), is no more compliant with the new version of it (Fig. 2b). Therefore, the query able to select compliant individuals remains the same:

```
SELECT ?x WHERE { ?x rdf:type myo:DriverR2Compliant }
```

which retrieves the only now compliant individuals d1 and d2.

As second example, let's consider the following rule R3[4] which establishes the limits for engaging credit activities in Australia, composed by the following 3 statements ($R3 = R3a \cup R3b \cup R3c$):

R3a) *It is forbidden to engage in a credit activity without a credit license.*

R3b) *It is permitted to engage in a credit activity if acting on behalf of a principal and the principal holds a credit activity provided the principal has not been elected to the parliament.*

R3c) *It is permitted to engage in a credit activity if acting on behalf of a body corporate and the person has been appointed as representative of the body corporate.*

The defeasibility of norm R3 consists in an exception (R3a) which can defeat the previous compliance conclusions about the engagement of an agent in a credit activity, and in the exceptions of exception to it (R3b and R3c) which can defeat the conclusions about the prohibition established by R3a. The whole scenario addressed by norm R3 can be modeled through an ontology (Fig. 3) describing a class Agent and a specific subclass AgentEngagingCreditActivity of those agents who engage in a credit activity. Also in this case the deontic concepts Prohibition and Permission, expressed in R3a, R3b and R3c, are represented as restrictions on the datatype properties having domain AgentEngagingCreditActivity and expressing the conditions which the norm operates on. The individuals of the class Agent can engage a credit activity, thus belonging to the subclass AgentEngagingCreditActivity. According to the constraints expressed in R3, individuals can:

p1. have a credit license (hasCreditLicence)
p2. act on behalf of a principal (isActingOnBehalfOfPrincipal)
p3. have principal holding a credit activity (isPrincipleHoldingCreditActivity)

[4] section 29 of the Australian Consumer Credit Protection Act

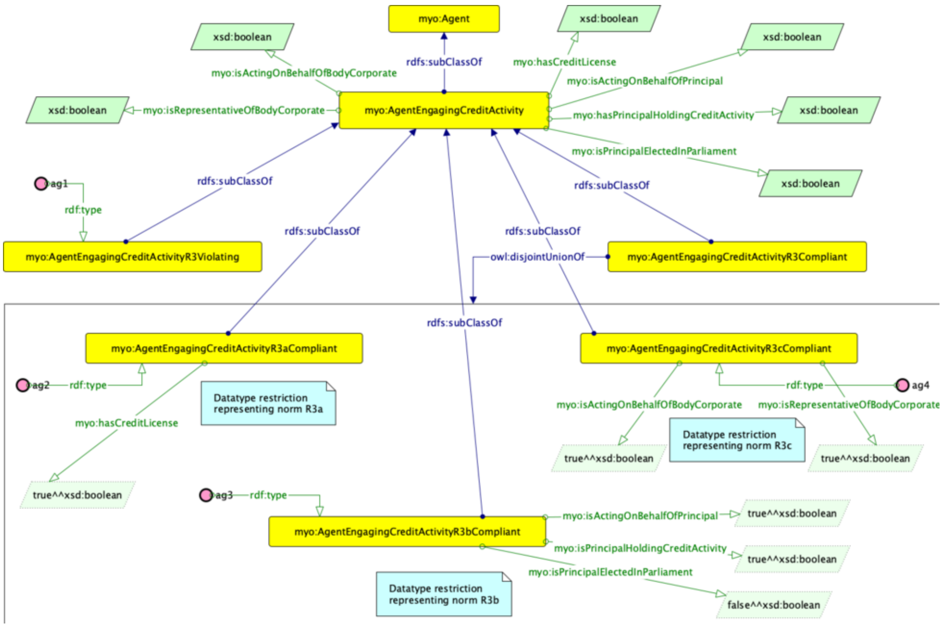


Figure 3. Norm R3 as restriction on AgentEngagingCreditActivity's properties (subclass relations between classes of compliant or violating individuals and AgentEngagingCreditActivity are inferred)

p4. have principal elected in Parliament (isPrincipleElectedInParliament)
p5. act on behalf of a body corporate (isActingOnBehalfOfBodyCorporate)
p6. act as representative of a body corporate (isRepresentativeOfBodyCorporate)

Norm R3a states that engaging in a credit activity "is forbidden without a credit license". Therefore, if an individual of the class AgentEngagingCreditActivity has a credit license (hasCreditLicence = "true") the activity is permitted. This is modeled as restriction on the property hasCreditLicence so to create a subclass AgentEngagingCreditActivityR3aCompliant of individuals having "true" as value of the property hasCreditLicence. In Fig. 3 the individual ag2 is compliant to the norm R3a.

Norm R3b states that the activity is permitted also when the individual "is acting on behalf of a principle" and "the principle holds a credit activity" and "the principle is not elected in Parliament". This is modeled through a multiple restriction on the properties isActingOnBehalfOfPrincipal = true, isPrincipleHoldingCreditActivity = true and isPrincipleElectedInParliament = false, to create a subclass of individuals for which the previous three restrictions contemporarily hold. In Fig.3, ag3 is compliant with norm R3b. Very similar considerations can be made for R3c modeling, concerning restrictions on the properties expressed by the conditions for individuals compliant with R3c.

The individuals compliant with the whole R3, composed by R3a, R3b and R3c, are therefore those belonging to the class AgentEngagingCreditActivityR3Compliant, obtained as disjoint union of the classes AgentEngagingCreditActivityR3aCompliant, AgentEngagingCreditActivityR3bCompliant, AgentEngagingCreditActivityR3cCompliant.

In all the other cases, engaging in a credit activity is forbidden. Therefore, the individuals which do not respect a combination of restrictions on properties of compliant individuals, violate norm R3, namely they belong to the class AgentEngagingCreditActivityR3Violating. In Fig. 3, ag1 violates norm R3. The combination of the property restrictions p1, ..., p6 able to identify individuals violating norm R3 can be obtained by the negation of the combination of properties of compliant individuals. In the case of R3, and applying the De Morgan laws, we obtain:

$$\neg[p1 \vee (p2 \wedge p3 \wedge \neg p4) \vee (p5 \wedge p6)] = \neg p1 \wedge (\neg p2 \vee \neg p3 \vee p4) \wedge (\neg p5 \vee \neg p6)$$

In order to verify which individuals are compliant or are violating R3, the following queries on the inferred model are respectively sufficient:

```
SELECT ?x WHERE {?x rdf:type myo:AgentEngagingCreditActivityR3Compliant}
SELECT ?x WHERE {?x rdf:type myo:AgentEngagingCreditActivityR3Violating}
```

5. Conclusions and future developments

In this paper we have presented an approach for legal compliance checking within a LOD framework. It is based on the representation of deontic norms in terms of domain ontology and ontology properties restrictions. The approach is implemented by decidable fragments of OWL 2, able to guarantee computational tractability and the possibility of using available reasoners. We have also shown how this approach can handle norm defeasibility. A development of this work will be the identification of specific knowledge modeling patterns able to represent defeasible deontic norms for legal compliance.

References

[1] J. M. Broersen, C. Condoravdi, N. Shyam, and G. Pigozzi, eds., *Deontic Logic and Normative Systems - 14th Int. Conference, DEON 2018*, (Utrecht, The Netherlands), College Publications, July 3-6 2018.

[2] R. Muthuri, G. Boella, J. Hulstijn, S. Capecchi, and L. Humphreys, "Compliance patterns: harnessing value modeling and legal interpretation to manage regulatory conversations," in *Proc. of ICAIL 2017*, (London, United Kingdom,), pp. 139–148, ACM, June 12-16 2017.

[3] H. Prakken and G. Sartor, "Law and logic: A review from an argumentation perspective," *Artificial Intelligence*, no. 227, pp. 214–245, 2015.

[4] T. Athan, G. Governatori, M. Palmirani, A. Paschke, and A. Wyner, "LegalRuleML: Design principles and foundations," in *The 11th Reasoning Web Summer School*, 2015.

[5] G. Governatori, M. Hashmi, H. Lam, S. Villata, and M. Palmirani, "Semantic business process regulatory compliance checking using LegalRuleML," in *Knowledge Engineering and Knowledge Management*, no. 10024 in LNAI, pp. 746–761, Springer International, 2016.

[6] F. Gandon, G. Governatori, and S. Villata, "Normative requirements as linked data," in *Proceeding of the JURIX Conference* (A. Wyner and G. Casini, eds.), vol. 302, pp. 1–10, IOS Press, 2017.

[7] E. Francesconi, "Reasoning with deontic notions in a decidable framework," in *Knowledge of the Law in the Big Data Age* (G. Peruginelli and S. Faro, eds.), vol. 317 of *Frontiers in Artificial Intelligence and Applications*, pp. 63–77, IOS Press, 2019.

[8] A. Marmor, *The Language of Law*. No. 978-0-19-871453-8, Oxford University Press, 2014.

[9] J. Searle, *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[10] J. Raz, *The Concept of a Legal System*. Oxford University Press, 1980.

[11] C. Biagioli, *Modelli Funzionali delle Leggi. Verso testi legislativi autoesplicativi*, vol. 6 of *Legal Information and Communications Technologies Series*. European Press Academic Publishing, 2009.

[12] E. Francesconi, "A description logic framework for advanced accessing and reasoning over normative provisions," *Int. Journal on Artificial Intelligence and Law*, vol. 22, no. 3, pp. 291–311, 2014.

[13] H. Kelsen, *General Theory of Norms*. Clarendon Press, Oxford, 1991.

[14] G. Casini, T. A. Meyer, K. Moodley, and U. S. I. Varzinczak, "Introducing defeasibility into owl ontologies," in *Proceedings of the International Semantic Web Conference*, pp. 409–426, 2015.