Profiles of Legal Knowledge Representation and Reasoning in the Semantic Web: an opportunity for AI in the Public Administration

Enrico Francesconi[1,*], Ginevra Peruginelli[2]

[1]Institute of Legal Informatics and Judicial Studies (IGSG-CNR), via de' Barucci 20, Florence, 50127, Italy
[2]Institute of Legal Informatics and Judicial Studies (IGSG-CNR), via de' Barucci 20, Florence, 50127, Italy

Abstract
In this paper an approach for legal knowledge representation and reasoning within a Semantic Web framework is presented. It is based on the distinction between provisions and norms and it is able to provide reasoning facilities for advanced legal information retrieval (like implementing Hohfeldian reasoning) and legal compliance checking for deontic notions. It is also shown how this approach can handle norm defeasibility. Such methodology is implemented by decidable fragments of OWL2[1], while legal reasoning is implemented by available decidable reasoners.

Keywords
Legal Reasoning, Semantic Web, Legal Information Retrieval, Norm Compliance

1. Introduction

Artificial Intelligence (AI) has a very significant impact not only on human beings' lives, but most importantly on our political, legal and economic institutions. In politics, AI can support evidence-based rational decision-making, as well as citizen engagement in policy choices and facilitate political communication and opinion aggregation.

In the legal field, a number of models, standards and applications are being developed to analyze and classify documents, apply complex regulations, suggest or predict the outcome of legal cases, detect or anticipate wrongful conduct, evaluate evidence, analyze sets of legal cases and social data to detect trends and anticipate changes. In this context, AI methods are inspired by, and combine with, the tools and methods of legal theory. The adoption of AI in the legal and socio-political sphere, therefore, contributes to support the development of effective and innovative context-sensitive solutions, thus contributing to democracy and the rule of law.

The semantic web represents one of the main infrastructures for AI, as it provides languages for knowledge representation and reasoning, as well as smart data for implementing intelligent systems.

RDF[1] is the language of the Semantic Web, able to describe a scenario of interest by triples composed by master data (including entities like abstract concepts or real world objects), metadata (namely properties of such entities) and their values (reference data). Such RDF triples are able to provide a semantic description of a specific domain: for example, in the legal one, they can describe facts and legal rules.

In the next future the ability of an information system to process Linked Open Data (LOD) and to show reasoning capabilities will be essential for developing automatic legal assistants, endowed with AI capabilities. In the legal context the availability of machine readable, actionable rules represents therefore a precondition for implementing systems with automatic reasoning facilities for advanced information services. In this contribution we present an approach for legal knowledge representation and reasoning within a Semantic Web framework. It is based on the distinction between provisions and norms and it is able to provide reasoning facilities for advanced legal information retrieval (like implementing Hohfeldian reasoning) and legal compliance checking for deontic notions. It is also shown how this approach can handle norm defeasibility. Such methodology is implemented by decidable fragments of OWL2[2], while legal reasoning is implemented by available decidable reasoners.

2. Provisions and Norms

According to the legal theory point of view, the legal order can be seen as a legal discourse composed by linguistic entities or *speech acts* [1] with descriptive or prescriptive functions. Every linguistic entity can be seen in a twofold perspective: as a set of signs organized in

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
*Enrico Francesconi
enrico.francesconi@igsg.cnr.it (E. Francesconi); ginevra.peruginelli@igsg.cnr.it (G. Peruginelli)
0000-0001-8397-5820 (E. Francesconi); 0000-0002-9331-4476 (G. Peruginelli)

[1]Resource Description Framework

[2]Ontology Web Language

words and sentences, as well as the meaning of such signs. Following the same twofold view for the legal domain, we can distinguish two levels of interpretation of a linguistic entity expressing a legal rule: in terms of a set of signs organized in words and sentences for creating a normative statement, typically called *Provision* [2] [3], as well as in terms of the meaning for application of such normative statement, typically called *Norm* [4], [5].

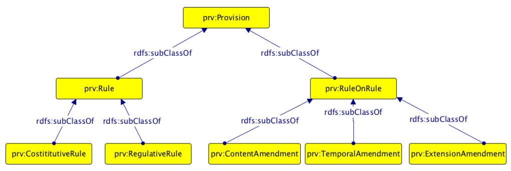

Figure 1: The *Provision Model* top classes ("prv:" is the namespace for provisions)

Provisions have been classified in [3] in terms of provision types, organised into two main groups (Fig. 1): *Rules* and *Rules on Rules*. *Rules* can be *Constitutive Rules* as *Definition* introducing entities, or *Regulative Rules* as the deontic concepts *Duty* and *Right* (or in a more deontic oriented terms *Obligation* and *Permission*), as well as *Power* etc., regulating subject roles and activities. *Rules on Rules* are different kinds of amendments: *Temporal*, *Extension* or *Content amendments*. Each provision type is characterized by specific *properties* (for example the *Bearer* or the *Counterpart* of a *Right*), reflecting the lawmaker directions. Provision types and properties can be considered as a sort of metadata model able to analytically describe fragments of legislative texts, hence the name of *Provision Model* [3].

In this vision, norms represent the way how provisions are applied; as such they represent the product of an interpretative process [6]. Provisions and related norms have, therefore, different roles and properties pertaining to different abstraction levels. Moreover, there may be not a bijective relationship between them: a norm can be expressed by different provisions, as well as it can be valid the opposite, namely one provision can include more norms [7]. They have also different relationships with time. Provisions, as pure textual objects, are the product of lawmaking (legal drafting activity and promulgation) and are characterized by the *in-force* date, namely the starting date of their existence in the legal order. On the other hand, norms are the meaning of provisions, namely their applicative interpretation; as such they are characterized by the *efficacy* date, namely the starting date in which a norm can be concretely applied. For example a taxation rule, in-force at time t_1 (time when the related provision enters the legal order), can express the application of a specific tax starting from time t_2 ($> t_1$). In this case t_2 is the *efficacy* date of the norm. Therefore (see Tab. 1), while it is obvious that we can have cases of provisions in-force and related norms effective, as well as provisions not in-force and related norms not effective, we can also have provisions in-force and related norms not effective, as well as the symmetric case, provisions not in-force and related norms effective (this last one is usually referred as *retro-activity* (efficacy in the past) or *ultra-activity* (efficacy in the future) of a norm.

Provision (in-force)	Norm (effective)
YES	YES
YES	NO
NO	YES[3]
NO	NO

Table 1
Relationship with time of *Provision* and *Norm*

Having different nature, such concepts operate in different domains.

A provision, as pure textual object, represents the building block of the legal order (new provisions can enter or leave the legal order itself). On the other hand, a norm can either modify the text of other provisions (in case of different type of amendments) or can introduce restrictions on the real world (in case of obligations, for example).

Advanced legal information retrieval, able to implement reasoning on deontic notions, is a type of reasoning managing textual information, thus pertaining to *provisions*. A typical example is a system able to implement Hohfeldian reasoning, in which a user submits a query to a legal document collection in order to find the rights of a bearer A towards a counterpart B: following an Hohfeldian reasoning the system should be able to retrieve also the provisions expressed as duties of the bearer B towards the counterpart A, because such duty can also be seen as A's right. An OWL 2 DL approach using the Provision Model for this type of reasoning is illustrated in [8] [9].

On the other hand, legal compliance checking is a process aiming to verify if a fact, occurring in the real world, complies with existing norms. Real world scenarios and facts can be effectively represented in terms of ontologies and related individuals, respectively. Norms, which facts have to be compliant with, provide constraints on the reality, therefore they can be modeled as restrictions on ontology properties. Such modeling can be used for legal compliance checking. Hereinafter we illustrate an OWL 2 DL approach for modeling norms and how such modeling can be used for the aim of legal compliance checking.

[3]retro-activity / ultra-activity

3. Modeling provisions for advanced legal information retrieval

In [8] and [9] it is shown how Hohfeldian relations on deontic and potestative notions can be managed within a description logic computational framework. We recall here the main aspects of the approach to show how *Provisions* can be used to implement an advanced legal provisions retrieval system, endowed with legal reasoning facilities, using a decidable fragment of OWL 2 (in particular OWL 2 DL), therefore exploiting existing decidable reasoners.

In this recall, we show the approach for deontic notions and their relations, sketched in the schema of Fig. 2.[4]

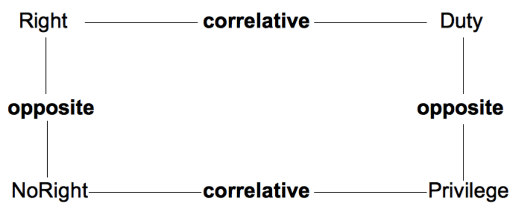

Figure 2: Hohfeldian relations on deontic concepts.

In order to implement an advanced legal provisions retrieval system, it is necessary to describe the relations between provisions at the level of the Provision Model. For example the Hohfeldian relation between *Duty* and *Right* can be effectively represented by observing that a *Right*, in correlative correspondence with a *Duty*, is actually not explicitly expressed in the text, but represents an implicit provision, basically a different view of the *Duty* itself, where the values of the related bearer and counterpart properties are swapped. Therefore, the Provision Model can be extended in terms of Duty and Right[5] implicit and explicit disjoint subclasses, able to represent a complete covering of the related superclass (ex: ExplicitRight and ImplicitRight disjoint subclasses represent a complete covering of the Right superclass).

Properties can also be specified as regards both implicit and explicit provisions, so that hasImplicitDutyBearer and hasExplicitDutyBearer are sub-properties of hasDutyBearer, as well as hasImplicitRightBearer and hasExplicitRightBearer are sub-properties of hasRightBearer.

To represent the hohfeldian fundamental relations between Duty and Right, firstly an equivalence relation between their explicit and implicit views is established: ImplicitRight ≡ ExplicitDuty and ImplicitDuty ≡ ExplicitRight. In Fig. 3 the established sub-class and equivalence relations between Duty and Right in their explicit and implicit views are summed up.

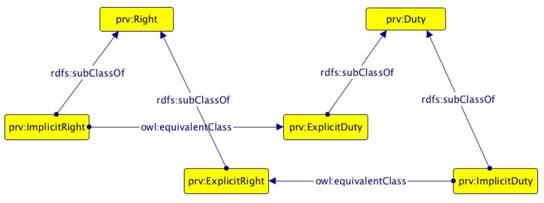

Figure 3: Sub-class and asserted equivalence relations between Duty/Right deontic correlative provisions.

Moreover, equivalence relations between implicit/explicit Duty and Right properties can be established. In Fig. 4 the asserted properties of ExplicitDuty and ImplicitRight and their mutual equivalence relations are shown (hasImplicitRightBearer ≡ hasExplicitDutyCounterpart and hasImplicitRightCounterpart ≡ hasExplicitDutyBearer).

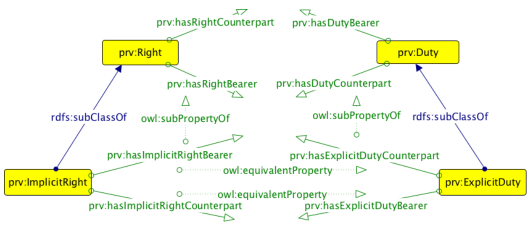

Figure 4: Asserted properties of ExplicitDuty and ImplicitRight and their mutual equivalence relations.

The same holds for the asserted properties of ImplicitDuty and ExplicitRight and their mutual equivalence relations (hasImplicitDutyBearer ≡ hasExplicitRightCounterpart and hasImplicitDutyCounterpart ≡ hasExplicitRightBearer) (Fig. 5) .

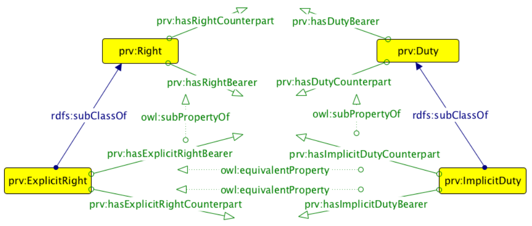

Figure 5: Asserted properties of ImplicitDuty and ExplicitRight and their mutual equivalence relations.

[4]more details on this modeling approach and its application to potestative notions (*Power, Liability, Disability Immunity*), can be found in [8] and [9]

[5]where “prv:”, namespace for provisions, is hereinafter implied

Note that the proposed patterns do not interfere with the relations between Right and Duty, which still hold. In fact, for the couple Right/Duty, an individual of ExplicitDuty is also an individual of Duty, given the axiom rdfs:subClassOf(ExplicitDuty, Duty). Moreover the axiom owl:equivalentClass(ImplicitRight, ExplicitDuty) tells us that such individual is also an ImplicitRight, which is also a Right, given the axiom rdfs:subClassOf(ImplicitRight, Right). Since this is done symmetrically for explicit and implicit duties and rights, we can deduce that Right is equivalent to Duty, namely is another reading of the Duty itself, given that the union of the disjoint explicit and implicit subclasses covers completely the related superclass.

3.1. Example of provision representation and reasoning

In order to show the ability of the Provision Model approach to provide advanced legal information retrieval facilities, based on provisions and related hohfeldian relations, the following example of a legal rule R1 can be used:

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

In terms of the Provision Model, this rule can be seen as a provision of type Duty, which can be represented as ExplicitDuty(Supplier, Consumer), where the arguments of the ExplicitDuty are the explicit bearer (Supplier) and related explicit counterpart (Consumer), respectively. Given the following introduced hohfeldian relations:

$$\begin{aligned}&\text{ImplicitRight} \equiv \text{ExplicitDuty}\\&\text{ImplicitRightCounterpart} \equiv \text{ExplicitDutyBearer}\\&\text{ImplicitRightBearer} \equiv \text{ExplicitDutyCounterpart}\end{aligned}$$

the provision at R1 can also be seen as ImplicitRight(Consumer, Supplier), including related implicit right bearer (Consumer) and implicit right counterpart (Supplier). Therefore, the provision R1 can be retrieved asking for either the duty of the supplier or the right of the consumer.

4. Modeling norms for legal compliance checking

As discussed in Section 2, norms can be viewed as the application of legal provisions, providing constraints on a real world scenario to be regulated. In the Semantic Web a real world scenario is usually represented by a domain ontology. In this context a norm, providing constraints to such scenario, can be modeled in terms of constraints on the domain ontology: for example, in case of obligations (like a duty), as ontology property restrictions.

4.1. Examples of norms representation and compliance checking

Let's consider as example the rule R1. The related scenario can be modeled in terms of an ontology including a class Supplier, having a boolean property hasCommunicatedConditions. Norm R1, expressing a duty for the suppliers states that suppliers must communicate contractual terms and conditions to the consumers: the individuals of the class Supplier complying with this norm are all those ones belonging to the subclass SupplierR1Compliant identified by a restriction on the boolean property hasCommunicatedConditions to have value "true" (see Fig. 6, where myo: is a fictitious namespace representing MyOntology).

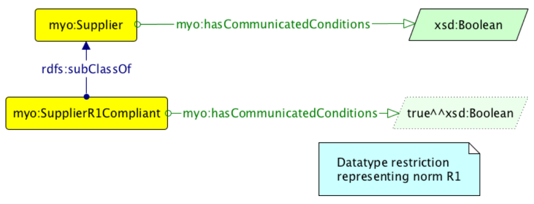
myo:Supplier
myo:hasCommunicatedConditions
xsd:Boolean
rdfs:subClassOf
myo:SupplierR1Compliant
myo:hasCommunicatedConditions
true^^xsd:Boolean
Datatype restriction representing norm R1

Figure 6: Norm R1 represented as restriction on the Supplier's property hasCommunicatedConditions (note that the subclass relation between SupplierR1Compliant and Supplier is inferred).

Such a representation for the real world scenario and related norm expressed by R1 results in the OWL 2 DL, decidable profiles. This allows us to use a OWL 2 DL decidable reasoner in order to implement reasoning facilities, preparing the ground for compliance checking with respect to R1. The inferred model establishes a rdfs:subClassOf relationship between SupplierR1Compliant and Supplier (as shown in Fig. 6), where SupplierR1Compliant is the class of all the individuals of type Supplier having "true" as value of the property hasCommunicatedConditions. Therefore, compliance checking according to the norm R1 is a problem of checking if an individual of type Supplier belongs to the class SupplierR1Compliant.

As an example let's consider the following two individuals myo:s1 and myo:s2 of the class Supplier: myo:s1 is an individual not compliant with R1, while myo:s2 is complaint with R1. The following SPARQL query

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

is able to select the individuals which are complaint with R1 (in our case s2). Legal reasoning in terms of norm compliance checking is therefore performed within a decidable computational complexity profile.

4.2. Norm compliance and defeasibility

In this section we show how the presented approach for norm representation and compliance checking can handle norm defeasibility. Let's consider the following legal rule R2:

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

In the 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.

Norm R2, expressing an obligation on the vehicles circulation scenario, states that, according to the related country law, one cannot drive over 90 km/h: the individuals of the class Driver complying with this norm are those ones belonging to the subclass DriverR2Compliant having value $\in$ [0.0, 90.0] Km/h on the datatype property hasDrivingSpeed (Fig. 7).

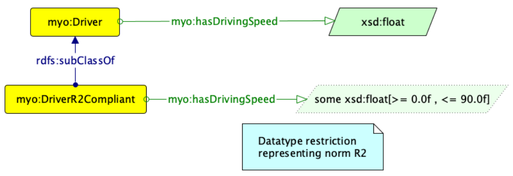

Figure 7: Norm R2 represented as restriction on the Driver's property hasDrivingSpeed (note that the subclass relation between DriverR2Compliant and Driver is inferred).

In other terms the norm R2 is represented as restriction on the property hasDrivingSpeed able to identify the class DriverR2Compliant which is equivalent to the class of the individuals for which the values of the property under consideration are in the range [0.0, 90.0] km/h. In order to represent such constraints the following restriction on the datatype property myo:hasDrivingSpeed to values (inclusively) between 0.0 and 90.0 can be expressed by the xsd:minInclusive and xsd:maxInclusive datatype bound properties. Such a representation results in the OWL 2 DL decidable profile.

As in the previous example, the inferred model establishes a rdfs:subClassOf relationship between DriverR2Compliant and Driver (as shown in Fig. 7), where DriverR2Compliant is the class of all the individuals of type Driver having values of the property hasDrivingSpeed in the interval [0.0, 90.0] km/h. Therefore, compliance checking according to the norm R2 is a problem of checking if an individual of type Driver belongs to the class DriverR2Compliant.

As a concrete example, let's consider four individuals myo:d1... myo:d4 of the class Driver, as represented in Fig. 8.

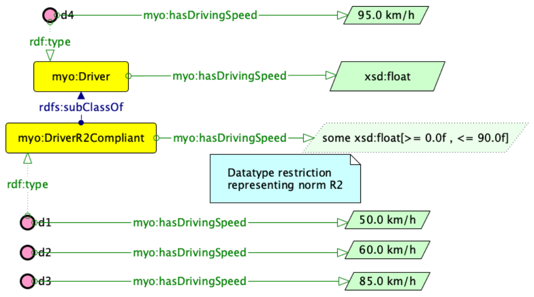

Figure 8: Four individuals of the class Driver: myo:d1, myo:d2, myo:d3 are compliant with R2, myo:d4 is not compliant with R2

In this list of individuals, the individual myo:d4 is not compliant with R2 (having speed 95.0 Km/h $\geq$ 90.0 Km/h). The following query:

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

is able to select the individuals which are complaint with R2 (in our case myo:v1, myo:v2, myo:v3).

Using the same example, we can now show how this compliance checking modeling approach can cope with norm defeasibility. Defeasibility is the property of an argumentation system for which a conclusion is open to revision in case evidence to the contrary is provided [10]. This particularly holds in legal reasoning which is a typical case of *non-monotonic* reasoning, where norm conflicts or norm exceptions might breach a previous conclusion.

Let's consider rule R2, as previously modeled, and the following new version of rule R2, introducing a more strict driving speed limit at 80 Km/h:

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

The *new version of R2* (Fig. 9) can defeat the previous compliance conclusions, in the sense that individuals, which were compliant with the *old version of R2* (Fig. 8), might not be compliant with it anymore (this is the case in the example in Fig. 9 of the individual d3). In order to cope with this change, the same model can be updated (without changing anything on the names of the classes) just by changing the original restriction on the datatype property hasDrivingSpeed with a new one expressed by the *new version of R2*, as shown in Fig. 9. Without changing anything on the individuals, their membership to the class DriverR2Compliant changes accordingly so that, for example, the individual d3, compliant with the *old version of R2* (Fig. 8), is no more compliant with the *new version of R2* (Fig. 9). Therefore, the query able to select compliant individuals remains the same:

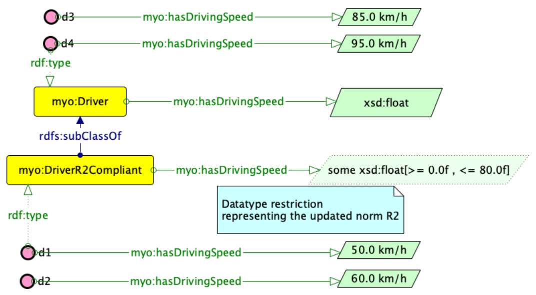

Figure 9: New version of norm R2 represented as restriction on the Driver's property hasDrivingSpeed and examples of compliant and non-compliant individuals.

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

which is able to retrieve the only individuals d1 and d2, compliant with the *new version of R2.*

5. Conclusions and future developments

In this paper we have presented a legal reasoning approach based on the distinction between the concepts of provisions and norms, able to deal with different types of legal reasoning, in particular advanced legal information retrieval, as well as norms compliance checking. The method is based on the use of decidable fragments of OWL 2, able to guarantee the computational tractability of the approach. This represents an essential property of a legal reasoning system in the Semantic Web, characterized by a huge amount of Linked Open Data in the form of triples.

Nowadays, within the public administration, AI is no longer just theory, but it's becoming an increasingly important option. Moreover, the Public Administration (PA) may play a central role in AI systems development, because they produce a large amount of public data, and legal data in particular, whose accessibility and reuse can be improved by applying semantic web technologies, as shown in this paper. This aspect is considered so strategic for business and public administration that each European country has developed its own national strategy. Applications for public administrations are aimed to create data infrastructures able to exploit the potential of big data that the PA generates, to simplify and personalize the offer of public services and to innovate administrations.

References

[1] J. Searle, Speech Acts: An Essay in the Philosophy of Language, ISBN 978-0521096263, Cambridge University Press, 1969.

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

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

[4] R. Guastini, Le Fonti del Diritto. Fondamenti teorici., Giuffrè, Milano, 2010.

[5] A. Marmor, The Language of Law, 978-0-19-871453-8, Oxford University Press, 2014.

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

[7] G. Pino, Teoria analitica del diritto, 9788846744517, ETS, 2016, pp. 144–183.

[8] E. Francesconi, Semantic model for legal resources: Annotation and reasoning over normative provisions, Semantic Web journal: Special Issue on Semantic Web for the legal domain 7 (2016) 255–265.

[9] E. Francesconi, A description logic framework for advanced accessing and reasoning over normative provisions, International Journal on Artificial Intelligence and Law 22 (2014) 291–311.

[10] T. Athan, G. Governatori, M. Palmirani, A. Paschke, A. Wyner, LegalRuleML: Design principles and foundations, in: The 11th Reasoning Web Summer School, DOI: 10.1007/978-3-319-21768-0_6, 2015.