Semantic Model for Legal Resources: Annotation and Reasoning over Normative Provisions

Enrico Francesconi

ITTIG-CNR, via de’ Barucci 20, Florence (Italy)

Abstract. A Semantic Web approach in the legal domain is presented in terms of a model of normative provisions and related axioms. In particular, relation between provisions are identified and modelled by introducing design patterns able to describe Hohfeldian legal fundamental relations and by a query approach able to deal with relations between provisions instances. Examples of semantic annotation of legal textual resources using RDF/OWL standards, as well as advanced access and reasoning facilities over provisions using SPARQL, are shown. The main benefit of the approach is represented by the ability to keep the complexity of the problem within a description logic computational tractability.

Keywords: Legal Semantic Web, Normative Provisions, Provision relation, Hohfeldian reasoning, Description Logic

1. Introduction

The legal domain is one of the most challenging area for developing applications based on the Semantic Web principles, because of the complex nature of legal information and document workflow, as well as the peculiarities of legal users’ information needs, requiring advanced information retrieval and reasoning services.

Such development can be conceived according to two converging perspectives: top-down and bottom-up.

The top-down perspective aims at defining semantic models for the Law to be used for the mark-up of legal resources and for reasoning over it. In literature several models (classification) of legal rules have been proposed, from the traditional Hohfeldian theory of legal concepts until more recent legal philosophy theories\(^1\), while computational models of them have been implemented [11].

The bottom-up perspective, on the other hand, aims to enrich legal resources with semantics at different levels of abstraction. This perspective tends to adhere to the Open Linked Data principles, which actually promotes a bottom-up approach to the Semantic Web, aimed to publish data, using URI naming conventions and RDF format to describe the available semantics, exploiting XML granular mark-up if available. The aim is to promote data exposition in a shared format at an available level of interoperability, leaving the effort of semantic enrichment and interconnection to a virtuous trend stimulated by the interest to data consuming. The bottom-up approach can be implemented manually, in a growing analytic effort of legal resources semantic mark-up, or automatically/semi-automatically, by using tools to extract knowledge from such resources, primarily textual, in order to detect and formalize rules [9,8] expressed in natural language, by using machine learning and natural language processing techniques [10].

In this paper a top-down/bottom-up combined approach is presented. It is based on a top-down definition of a semantic model for legislation in terms of normative provisions, presented in Sections 2, which can be used within a bottom-up, Linked Open Data framework, to provide a level of semantic refinement to legislative resources at an available minimum level

---

\(^1\) due to Rawls, Hart, Ross, Bentham, Kelsen, among the others.
of structural mark-up. In this context, in Section 3 possible kinds of relations between provisions are introduced. In Sections 4, 5 and 6 an extension of the normative provisions model through a description logic pattern able to deal with relations between provisions as the Hohfeldian fundamental relations [13] are presented. In Section 7 an example of how this approach can support Hohfeldian inferences for improving provisions accessibility within a description logic framework is presented with respect to a European directive case-study. In Section 8 specific relations which can be identified between provision instances are discussed, while in Section 9 an implementation of reasoning facilities over relations between provision instances with respect to the same European directive case-study is shown. Finally in Section 10 some conclusions are reported.

2. A model of normative provisions

In this section a brief recall of the model of normative provisions presented firstly in [3] and [4] is given. According to [3] [4] laws and regulations may be seen as a set of provisions, carried by speech acts [16] [14]. Following this perspective, fragments of a legislative text are, at the same time, sentences, paragraphs, or provisions, according to whether they are seen from a formal or semantic viewpoint. Provisions can be described in terms of provision types (as Definition, Procedure, Duty, Right, Power, as well as more technical ones as Insertion, Abrogation, Substitution, etc.) and related attributes (for example the Bearer of a Right), reflecting the lawmaker directions. Provision types and attributes can be considered as a sort of metadata model able to analytically describe fragments of legislative texts, hence the name of Provision Model [3].

The details of the Provision Model is widely described in [3] and [4]; in this paragraph the semantic organization of the model is briefly recalled. In the Provision Model, provision types are organised into two main groups: Rules (introducing entities or expressing deontic concepts) and Rules on Rules (different kinds of amendments). Adopting a typical law theory distinction, well expressed by Rawls, Rules consist in:

- Constitutive rules, which introduce or assign a juridical profiles to entities of a regulated reality;
- Regulative rules, which discipline actions or the substantial and procedural defaults (remedies).

On the other hand, Rules on Rules can be distinguished in:

- Content amendments, which modify literally the content of a norm, or their meaning without literal changes;
- Temporal amendments, which modify the times of a norm (come-into-force and efficacy time);
- Extension amendments, which extend or reduce the cases on which the norm operates.

The values of provision attributes can be expressed by lexical units, or by concepts derived from thesauri/ontologies, able to provide additional information on the entities of the regulated domain [2] [12]. An example of an ontology dealing with a domain regulated by national and EU legislations, as the consumer protection one, has been developed within the DALOS project [1].

For example, the following fragment (article 5, paragraph 1) of the European Directive 2002/65/EC, concerning the distance marketing of consumer financial services:

The supplier shall communicate to the consumer all the contractual terms and conditions and the information referred to in Article 3(1) and Article 4 on paper or on another durable medium available and accessible to the consumer in good time before the consumer is bound by any distance contract or offer.

besides being considered as a formal partition (a paragraph) of the related directive, can also be viewed as a semantic component (a provision) and qualified as a Duty, whose attributes are:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasBearer</td>
<td>“Supplier”</td>
</tr>
<tr>
<td>hasObject</td>
<td>“Contractual terms and conditions…”</td>
</tr>
<tr>
<td>hasAction</td>
<td>“Communication”</td>
</tr>
<tr>
<td>hasCounterpart</td>
<td>“Consumer”</td>
</tr>
</tbody>
</table>

where attributes values can be literals or concepts in a ontology, like the DALOS one, for example.

3. Relations between provisions

Relations between provisions can be identified concerning either the general characteristics of the norms or their applications to a specific domain to be regu-
lated. In the Provision Model they can be respectively distinguished in functional relations, involving provision types and attributes, as the Hoefeldian relations between duties and rights, typically described in the theory of norms, and functional-thematic relations involving provision types and values of corresponding provision attributes, as the relations between the duty of a specific bearer to accomplish a specific action towards a given counterpart as well as the procedure to fulfill it.

While the first kind of relations can be described at the level of Provision Model (see Sections 4-7) and are inherited by the related instances, the second kind can be identified and described at the level of provision instances only (see Sections 8-9).

As reported in [4] relations between provisions can be established directly by the legislator through references, internal or external of the including act, or can be deduced by reasoning over provisions. The relations established by the legislator can be easily detected and assume specific roles especially in handling amendments, in the automatisms that can be conceived to produce consolidated versions of the texts, as well as a guide in the consultation of legislative texts. The relations that can be deduced by reasoning are obviously more difficult to identify. This difficulty represents one of the main sources of ambiguities in the application of the law, therefore they will be hereinafter discussed.

4. Functional relations between provisions

Functional relations between provisions can be described as axioms on provision types and attributes, as the Hoefeldian fundamental relations regarding Right/Duty, Liberty/No-right, Power/Liability, Immunity/Disability, as well as the relation between the Duty of a subject (duty Bearer) towards a Counterpart, which can be viewed as an implicit Right of the duty Counterpart towards the duty Bearer.

A description of legislative texts in terms of provisions allows advanced access services on legislation, able to implement reasoning facilities based on the theory of norms. A typical example can be a service able to implement the previously mentioned Hohfeldian reasoning by accessing the rights of a subject, either explicitly expressed or inferred. An implementation of such retrieval and reasoning scheme is here presented within a Description Logic framework.

Let’s consider the Hoefeldian relation between Duty and Right to show our approach. For example article 5 paragraph 1 of the European Directive 2002/65/EC reported in Section 2, can be considered a provision of type Duty involving ‘Supplier’ and ‘Consumer’. In terms of Provision Model, such duty of the ‘Supplier’ towards the ‘Consumer’ can be expressed as follows: Duty(hasBearer='Supplier', hasCounterpart='Consumer') which corresponds to

Right(hasBearer='Consumer', hasCounterpart='Supplier').

This Hoefeldian relation underlines an equivalence between Duty and Right, as long as the values of the duty Bearer and Counterpart are swapped, assuming symmetric roles in the Right provision, therefore involving equivalence relations between provision types and attributes.

However, describing these relations in the Provision Model by establishing the equivalence relations Duty ≡ Right[4] [5] and hasBearer ≡ hasCounterpart would imply equivalence relations between any duties and rights, irrespective to the attribute types and values, as well as between all the provision types sharing equivalence relations between such attributes, which might produce inconsistent results in a provisions retrieval system.

For example a query aiming to retrieve provisions having Right(hasBearer = 'Supplier'), would also give back Duty provisions having Duty(hasBearer = 'Supplier') because they satisfy the axiom Duty ≡ Right. Similarly, the previously mentioned query would retrieve back Right provisions having Right(hasCounterpart = 'Supplier'), since they satisfy the axiom hasBearer ≡ hasCounterpart.

5. Extension of the Provision Model

To avoid these problems, while relying on Description Logic expressivity as implemented in OWL-DL, an extension of the Provision Model is proposed.

Firstly provision attributes are specified according to the related provision types, for example hasBearer and hasCounterpart attributes are distinguished in terms of hasDutyBearer and hasDutyCounterpart as properties of Duty, and hasRightBearer and hasRightCounterpart as properties of Right.

[4] Hereinafter, provision types as OWL classes (starting with capital letters) and provision attributes as OWL properties (starting with lowercase letters) are written in serif font. The namespace is omitted for simplicity.
A model extension can also be provided by observing that a Right, in Hohfeldian 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 attributes are swapped. Therefore the Provision Model can be extended in terms of Duty and Right 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).

Attributes 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 (see Fig. 1 for the extension of the Right provision type and attributes; a similar extension can be figured out for Duty and other provisions).

6. Hohfeldian relations in the Provision Model

To describe our approach for implementing Hohfeldian relations between provisions, we consider:

- as examples of correlative deontic concepts the couple Duty/Right;
- as examples of correlative potestative concepts\(^5\) [15] (expressing necessity conditions) the couple Power/Liability.

Similar considerations can be given for the deontic couple Liberty/No-right and the correlative potestative one Disability/Immunity because they can be respectively derived as negation of the opposite deontic and potestative couples, respectively.

Fig. 2. Sub-class and asserted equivalence relations between deontic correlated provisions.

To represent the Hohfeldian fundamental relations between Duty and Right, firstly an equivalence relation between their explicit and implicit views is established: ImplicitRight \(\equiv\) ExplicitDuty and ImplicitDuty \(\equiv\) ExplicitRight. In Fig. 2 the established sub-class (Section 5) and equivalence relations between Duty and Right in their explicit and implicit views are summed up.

As for the couple Duty/Right, an equivalence between Power and Liability represents a Hohfeldian relation as long as the values of the power Bearer and Counterpart are swapped, assuming symmetric roles in the Liability provision, therefore involving, in the Provision Model, equivalence relations between provision types and attributes. Such group of relations is isomorphic to the one between Duty and Right therefore the same pattern can be applied to represent Power/Liability relation, as summed up in Fig. 3.

Moreover, equivalence relations between implicit/explicit Duty and Right attributes, as well as between implicit/explicit Power and Liability attributes, can be established. In Fig. 4 the asserted sub-property and equivalence relations between hasDutyBearer and hasRightCounterpart in their explicit and implicit views are summed up. The same holds for the asserted sub-property and equivalence relations between hasPower-

\(^5\)also called ‘anankastic’ in [7]
Fig. 3. Sub-class and asserted equivalence relations between anackastic correlated provisions.

Bearer and hasLiabilityCounterpart in their explicit and implicit views.

The reader can imagine a symmetric view for the relations between a right bearer and a duty counterpart, as well as between a liability bearer and a power counterpart in their explicit and implicit views.

Note that the proposed pattern does not interfere with the equivalence relation between Right and Duty, as well as Power and Liability, which still hold. In fact, for the couple Right/Duty as example (but similar consideration can be given for Power/Liability), 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, given that the union of the disjoint explicit and implicit subclasses covers completely the related superclass (see Section 5).

Therefore reasoning properties are preserved, but the expressivity of the model is improved to provide enhanced retrieval services. The proposed pattern in fact aims to introduce:

1. Properties equivalence, allowing direct swapping on attributes contents for addressing provision relations, without the need of using conditional statements (ex: if (hasDutyCounterpart == 'Consumer'))

2. Abstract classes (namely classes not used for mark-up, in our case “implicit” classes) so to provide different views (implicit and explicit views) on the same provision, as well as retrieval services able to access implicit provisions only (ex: provision instances where ImplicitRightBearer == 'Consumer');

Moreover, by providing equivalence relations between symmetric implicit and explicit classes and attributes, the proposed pattern is able to avoid inconsistent deductions (as for example that bearers and counterparts freely mix in the same provision). In fact, equivalence axioms applied on provision implicit and explicit classes and related properties produce inferential deductions (for example attribute mixing) which keep semantic consistency. For example, given the following description of an explicit right:

a) ExplicitRight(hasExplicitRightBearer = 'Consumer')

which is given that:

ExplicitRight ≡ ImplicitDuty

hasExplicitRightBearer ≡ hasImplicitDutyCounterpart

the following consistent deductions, describing the same provision instance, can be obtained:

b) ExplicitRight(hasImplicitDutyCounterpart = 'Consumer')

c) ImplicitDuty(hasExplicitRightBearer = 'Consumer')

d) ImplicitDuty(hasImplicitDutyCounterpart = 'Consumer')

which are semantically consistent.

For example being “Consumer” an ExplicitBearer of an ExplicitRight, is also to be consider an Implicit-Counterpart of the same provision, viewed as an ImplicitDuty.

Finally, it is worth to stress that the introduced axioms are not dealing with relations between different provision instances expressed in a legislative text (which could be better described in terms of existential restrictions, as for example: ‘for every explicit duty there is an implicit right where bearer and counterpart are swapped’ or in terms of relations between attribute values (Section 8)), but they deal with different views (explicit and implicit views) of the same provision instance. In this perspective all the deductions derived from the established equivalence relations between classes, as well as the deductions derived from mixing provision qualified properties, are valid, as previously discussed.

7. Provision functional relations case-study

In this section an example of how this approach can be used for a provision retrieval system able to
deal with functional relations is shown. In particular a case study of Hohfeldian reasoning over provisions is shown.

7.1. Semantic annotation


According to the Provision Model and a domain ontology like DALOS, the semantic of such document fragments, identified by the including document URI and specific IDs, can be summed up as in Tab. 1 (this semantic description is limited to the attributes useful to demonstrate the approach).

<table>
<thead>
<tr>
<th>Partition ID</th>
<th>Provision Type</th>
<th>Provision Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>art2.par1.suc</td>
<td>Definition</td>
<td>hasDefinitionDefiniendum='Supplier'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasDefinitionDefiniendum=’Consumer’</td>
</tr>
<tr>
<td>art2.par1.subb</td>
<td>Definition</td>
<td>hasExplicitDutyBearer=’Supplier’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasExplicitDutyAction=’Communication’</td>
</tr>
<tr>
<td>art2.par1;par1</td>
<td>ExplicitDuty</td>
<td>hasExplicitDutyObject=’ContractualTerms’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasExplicitDutyCounterpart=’Consumer’</td>
</tr>
<tr>
<td>art5.par1</td>
<td>Procedure</td>
<td>hasProcedureBearer=’Supplier’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasProcedureAction=’Communication’</td>
</tr>
<tr>
<td>art5.par2</td>
<td>Equipment</td>
<td>hasEquipmentObject=’ContractualTerms’</td>
</tr>
<tr>
<td>art5.par3</td>
<td>ExplicitRight</td>
<td>hasExplicitRightBearer=’Consumer’</td>
</tr>
<tr>
<td>art6.par1</td>
<td>ExplicitDuty</td>
<td>hasExplicitDutyBearer=’MemberStates’</td>
</tr>
<tr>
<td>art7.par1;spa1</td>
<td>ExplicitPower</td>
<td>hasExplicitPowerBearer=’Supplier’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasExplicitPowerCounterpart=’Consumer’</td>
</tr>
<tr>
<td>art7.par1;spa2</td>
<td>ExplicitPower</td>
<td>hasExplicitPowerBearer=’Supplier’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasExplicitPowerCounterpart=’Consumer’</td>
</tr>
<tr>
<td>art7.par2</td>
<td>ExplicitPower</td>
<td>hasExplicitPowerBearer=’MemberStates’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasExplicitPowerCounterpart=’Supplier’</td>
</tr>
<tr>
<td>art7.par3</td>
<td>ExplicitLiability</td>
<td>hasExplicitLiabilityBearer=’Supplier’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hasExplicitLiabilityCounterpart=’Consumer’</td>
</tr>
</tbody>
</table>

Having defined the prv and cl namespaces for Provision Model and DALOS consumer law domain ontology, respectively, an RDF/OWL semantic annotation excerpt of such fragments (art. 5 paragraph 1 is reported as example) can be the following:

Table 1
Semantics of Directive 2002/65/EC excerpt
7.2. Querying the system

Having an OWL-DL description of the Provision Model and provision instances, a provisions management system can be given inference facilities through an OWL-DL reasoner able to derive an inferred model. In this case-study the Pellet\(^6\) Java based OWL-DL reasoner is used. The result is a Provision Model where inferences are calculated from the associated axioms. At this stage an RDF triple store of provisions can be queried using SPARQL. Let’s assume to query the Directive excerpt in Section 7.1 in order to demonstrate the approach and, as first example, a query able to retrieve consumer’s rights:

\[
\text{SELECT } ?\text{par} \text{ WHERE } \{ ?\text{par} \text{ prv:hasExplicitRightBearer cl:Consumer} \}
\]

where prv: and cl: are the namespaces of the Provision Model and DALOS consumer law ontology, respectively, and ?\text{par} is the variable which will contain the identifier of the retrieved provision instances (paragraphs).

In case the non-inferred model is queried, no provisions are retrieved since only ExplicitRight and related explicit attributes are used for provision annotation. To obtain the rights explicitly expressed, the query has to be specified asking for provisions whose hasExplicitRightBearer value is cl:Consumer. In this case, paragraph with id="art5;par3" is correctly retrieved.

In case the inferred model is queried, all the inferred provisions are retrieved, either annotated as ExplicitRight of Consumer or implicitly deduced by provision relations. Since Hohfeldian relations have been implemented in the Provision Model, the result will be a Hohfeldian reasoning over provisions. By exploiting the established rdfs:subClass and owl:equivalentClass relations between provisions type and attributes, the system will act as virtually expanding the query and obtaining the results as shown in Tab. 2.

<table>
<thead>
<tr>
<th>Virtual query expansion</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>?\text{par} prv:hasExplicitRightBearer cl:Consumer</td>
<td>art5;par3</td>
</tr>
<tr>
<td>?\text{par} prv:hasImplicitRightBearer cl:Consumer</td>
<td>art5;par1</td>
</tr>
<tr>
<td>[≡ prv:hasExplicitDutyCounterpart] art6;par1</td>
<td>art6;par1</td>
</tr>
</tbody>
</table>

Table 2

Virtual query expansion corresponding to inferences on provision axioms.

Moreover, the distinction between implicit/explicit provisions and attributes allows to select, for example, among all the Rights of a Bearer, only those which are not explicitly expressed in the text. The related query able will be

\[
\text{SELECT } ?\text{par WHERE } \{ ?\text{par prv:hasImplicitRightBearer cl:Consumer} \}
\]

which will retrieve the ExplicitDuty where hasExplicitDutyCounterpart is Consumer (being hasImplicitRightBearer ≡ hasExplicitDutyCounterpart); in the case-study of Section 7.1 the following paragraphs are retrieved: id="art5;par1", id="art6;par1".

Similar considerations can be made about querying potestative provisions, like retrieving consumer’s powers.

8. Functional-thematic relations between provisions

As previously introduced, another kind of relations between provisions can be identified: we call them functional-thematic relations because they deal with functional characteristics of the provisions and thematic aspects of the domain they apply to, therefore

\(^6\)http://clarkparsia.com
such relations can be described at the level of provision instances only.

An example of them can be the relation existing between a Duty of a Bearer to accomplish a specific Action towards a Counterpart, the Procedure describing how to fulfill this obligation, the Exceptions to it, as well as the Sanction such Bearer may face if he does not fulfill his obligation. In the excerpt of Directive 2002/65/EC in Section 7.1, Art. 5 paragraph 1 and 2 represent a Duty and the correlated Procedure (see also Tab. 1).

Another example of relations between provision instances can be the one existing between a Definition introducing a specific entity through its attribute Definiendum and other provisions involving such entity. In the excerpt of Directive 2002/65/EC in Section 7.1, Art. 2 paragraph 1 letter c) and Art. 5 paragraph 1 and 2, as well as other provisions involving the ‘Supplier’, represent correlated provision instances.

In terms of the Provision Model such relations between provision instances hold only if there is identity between a number of values in corresponding attributes in different correlated provisions. In this respect [4] distinguishes between strong and weak relations between provision instances according to whether there is identity between all the values in corresponding attributes of provision instances or between only some of them. The number of attributes and values in identity relation gives the degree of strength/weakness of the relation [4].

Relations between provision instances are particularly interesting for providing users with advanced retrieval services. For example, while querying for his duties, a ‘Supplier’ might be also informed about the procedures to fulfill such a duty and possible sanctions in case of non-compliance.

9. Provision thematic relations case-study

In this section a possible implementation of the reasoning facilities dealing with thematic relations between provision instances is shown. On the basis of the semantic annotation proposed in Section 7.1, an example of SPARQL queries able to deal with thematic relations between provision instances of the excerpt of Directive 2002/65/EC is given. For example a SPARQL query able to retrieve the supplier’s duties is:

```
SELECT ?par WHERE { ?par prv:hasDutyBearer cl:Supplier }
```

Firstly, in case the inferred model is queried, the paragraphs with id="art5-par1" and id="art5-par3" are retrieved, thus obtaining a Hoefeldian reasoning, retrieving both explicit and implicit duties of the ‘Supplier’. At this stage the system can analyse the thematic aspects of the retrieved provisions, by collecting provisions attribute values and construct a query able to check whether correlated provision instances of type Procedure regarding the ExplicitDuty at id="art5-par1" are available (the same can be done for the implicit duty at id="art5-par3"). Such query can be as follows:

```
SELECT ?par WHERE { ?par prv:hasDutyBearer cl:Supplier.
  ?par prv:hasDutyAction cl:Communication.
  ?par prv:hasDutyObject cl:ContractualTerms.
  ?par prv:hasDutyCounterpart cl:Consumer
}

UNION

{?par prv:hasProcedureBearer cl:Supplier.
  ?par prv:hasProcedureAction cl:Communication.
  ?par prv:hasProcedureObject cl:ContractualTerms.
  ?par prv:hasProcedureCounterpart cl:Consumer
}
```

This query will retrieve provision instances of type Duty with id="art5-par1" and the correlated Procedure at id="art5-par2". Such query can be also extended to search for correlated sanctions (Redress).

10. Conclusions

The combination of Provision Model and domain ontologies can be an effective approach to the Semantic Web for the legal domain. Relations between provisions can be also exploited to provide advanced retrieval services and reasoning facilities over norms. Functional relations between provisions, as the Hoefeldian fundamental relations, have been described by extending the Provision Model to represent either implicit or explicit provision types and attributes. Similarly an approach dealing with both functional and thematic relations between provisions has been presented. The main benefit of this approach is represented by the ability to keep the complexity of the problem within a description logic computational tractability.


