D2: A LegalRuleML specialisation for reporting obligations, as well as the legal rationale that motivates the design choices

Monica Palmirani, University of Bologna, ALMA-AI
May 2024

1. Introduction

The goal of task D2 is to define LegalRuleML annotation for the reporting obligation called also reporting requests.

LegalRuleML is a markup language XML (OASIS standard[1]) for modelling legal rules, especially using defeasible logic. The defeasible logic is used for managing non-monotonic logic where exceptions could affect the premises of a set of rules. Like this example:

http://data.europa.eu/eli/reg/2022/869/oj

In this case there is a basic rule R1 in Art. 3

R1 "1. Regional groups (Groups) shall be established in accordance with the process set out in Section 1 of Annex III."

The R1 is derogated with this paragraph 1 of the Art. 24 (R2):
R2 "1. In the case of Cyprus and Malta, which are not interconnected to the trans-European gas network, a derogation from Article 3, [omissis]" shall apply, without prejudice to Article 32(2)."

However in paragraph 5 of Art. 24 we have a derogation of the R2.
R3 " 5. The derogation set out in paragraph 1 shall apply until Cyprus or Malta, respectively, is directly interconnected to the trans-European gas network or until 31 December 2029, whichever is the earlier.
To cope with these objectives, we have analysed the database KOEL and the typologies of conditions."
A hierarchy of rules is defined.

```
R3>R2>R1

R3 overrides R2, R2 overrides R1
```

In the reporting request (obligations is reserved as nomenclature to the deontic operator) we find some conditions that it is better to model using LegalRuleML because we could have derogations over time.
The following example is a conditioned reporting request that was modified in 2008-03-21.

/akn/eu/act/directive/2006-05-17/06-43/eng@2008-03-21/!main~art_48__para_3
http://data.europa.eu/eli/dir/2006/43/2023-01-05

The original paragraph 3 of art. 48 was

> *"3. The Committee shall adopt its Rules of Procedure."*

In the 2008-03-21 the paragraph 3 is modified as following:

> *"3. By 31 December 2010 and, thereafter, at least every three years, the Commission shall review the provisions concerning its implementing powers and present a report to the European Parliament and to the Council on the functioning of those powers. The report shall examine, in particular, the need for the Commission to propose amendments to this Directive in order to ensure the appropriate scope of the implementing powers conferred on the Commission. The conclusion as to whether or*

[1] http://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/legalruleml-core-spec-v1.0.html

not an amendment is necessary shall be accompanied by a detailed statement of reasons. If necessary, the report shall be accompanied by a legislative proposal to amend the provisions conferring implementing powers on the Commission."

The new paragraph includes two conditions:

i) to justify in case no amendments are necessary;
ii) to accompany the report with a legislative proposal if it is necessary.

These two patterns are quite regular.

If no amendments are necessary, THEN obligation to present a justification.
If necessary amendment THEN obligation to present a legislative proposal.

We propose two patterns in LegalRuleML for favouring the automatic detection and serialization. Before to start with the serialization of these two patterns we want to introduce the pillars of LegalRuleML.

2. LegalRuleML Principles and Standard

The LegalRuleML is an XML standard defined inside of OASIS international standardization body (www.oasis-open.org) that aims to produce a rule interchange language for modelling legal norms. It is an extension of a particular module of RuleML families, Consumer RuleML (http://wiki.ruleml.org/index.php/Specification_of_Consumer_RuleML_1.02). Consumer RuleML was designed to uniform Deliberation RuleML and Reaction RuleML families and so to provide a consistent language good for the Semantic Web applications. LegalRuleML reuses some structures of Deliberation RuleML (e.g., Atom, Rule) and others from Reaction RuleML (e.g., Data, Interval). The figure below show the relationship of LegalRuleML with the RuleML families.

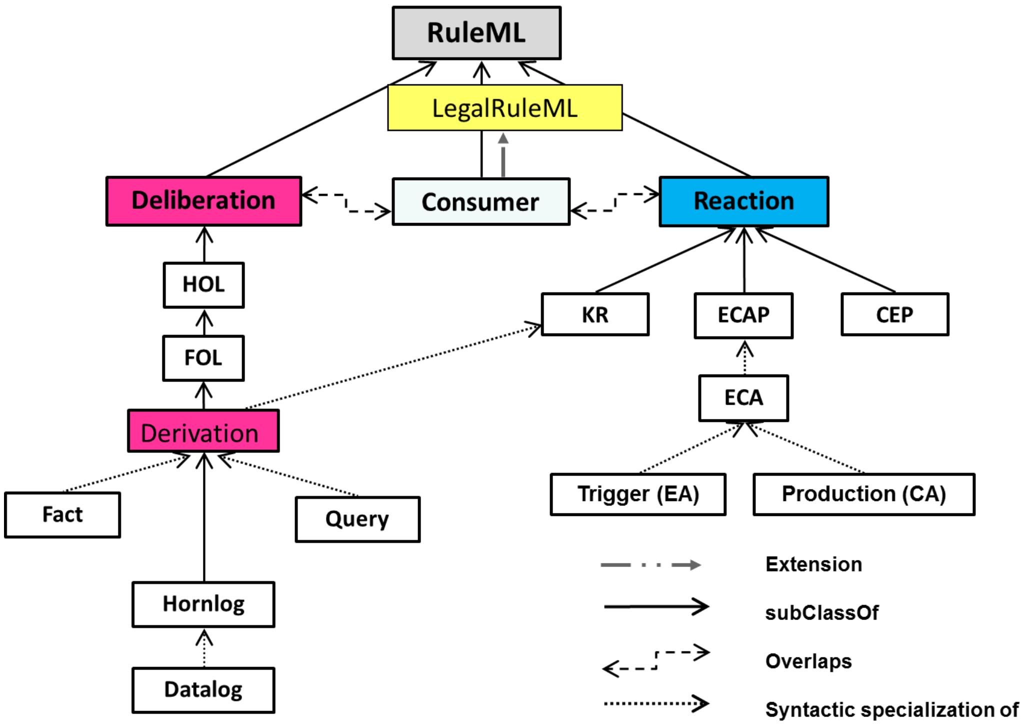


Figure 1 – LegalRuleML relationship in the RuleML families.

LegalRuleML intends to provide XML structures to model in formal manner the legal norms. It is designed for Semantic Web applications (e.g., legal ontology of relationship between norms, for permitting SPARQL queries) or for logic reasoning (e.g., compliance checking) according with the legal logic theories described in the D1.1 of WP1 and in the Chapters above in this D1.2, including deontic, defeasibility, temporal reasoning. In the last decade, several Legal XML standards have been proposed to represent legal texts [30] with XML-based rules (RuleML, SWRL, RIF, LKIF, etc.) [16, 18]. At the same time, the Semantic Web, in particular Legal Ontology research combined with semantic norm extraction based on Natural Language Processing (NLP)

[15], has given a strong impetus to the modeling of legal concepts [8, 10, 11]. However in the state of the art does not exist a uniform and consistent XML rule language for modelling legal norms. Based on this, the work of the LegalRuleML focuses the design on three specific needs:

1. To close the gap between legal texts, which are expressed in natural language, and semantic norm modeling. The authentic legal documents (e.g., official gazette) are available more and more in digital format on the Web and the logic representation of the norms has to be connected with the fragment of texts that generate the modelization. The tractability between text and legal rules is fundamental for preserving the authenticity, the integrity, the legal validity of the logic assertions, and so to create trust and confidence in the legal operators (e.g., judges) about the legal reasoning process.
2. To provide an expressive XML standard for modeling normative rules that satisfies legal domain requirements. The requirements include deontic operators, defeasible structure, temporal parameters, metadata mechanisms.
3. To apply the Linked Open Data [9] approach to model raw data in the law (acts, contracts, court files, judgments, etc.) and to extend it to legal concepts and rules along with their functionality and usage. Legal concepts need to be integrated with legal rules and LegalRuleML permits this connection without conflict [36] with legal ontologies.

LegalRuleML principles

LegalRuleML is designed following some main principles.

Multiple Semantic Annotations: A legal rule may have multiple semantic annotations, where these annotations represent different legal interpretations. Each such annotation appears in a separate annotation collection using metadata that provide the interpretation with respect to provenance (agent and role), applicable jurisdiction, logical interpretation of the rule, additional sources that support this interpretation.

Tracking the LegalRuleML Creators: As part of the provenance information, a LegalRuleML document or any of its fragments can be associated with its creators. This is important to establish the authority and trust of the knowledge base and annotations.

Linking Rules and Provisions: LegalRuleML includes a mechanism, based on IRI, that allows many to many (N:M) relationships among the rules and the textual provisions. Multiple rules are embedded in the same provision, several provisions contribute to define the same rule. This mechanism may be managed in the metadata collections avoiding redundancy in the IRI definition.

Temporal Management: Textual provisions and legal rules change in time according to at least three temporal legal axis: time of enter into force of the textual provisions, time of commencement of operational of the norms, the time of application of the norms to the facts. Legal RuleML represents these temporal issues using special metadata. In particular, the rules can be associated to temporal parameters which can vary over time. In this way we can produce all the elements necessary for implementing the temporal reasoning theory described in the D1.1.

Formal Ontology Reference: LegalRuleML is independent from any legal ontology and logic framework. However it includes a mechanism, based on IRIs, for pointing to reusable classes of a specified external ontology. In this manner any <Rel> element is associated to a predicate defined in the legal domain ontology. Also the different type of deontic operators (e.g., [OM]: maintenance, [OP]: puntuale, [OAPP]: achievement preemptive persistent, [OAPNP]: achievement preemptive non-persistent, [OANPP]: achievement nonpreemptive, persistent, [OANPNP]: achievement, non preemptive, non-persistent see [40][42]). LegalRuleML legal knowledge is also mappable to RDF triples for Linked Data reuse using a RDFS metamodel.

LegalRuleML functionalities

The features that LegalRuleML is capable to manage are the following:
– defeasibility of rules and defeasible logic [13,31,34];
– deontic operators (e.g., obligations, permissions, prohibitions, rights);
– semantic management of negation;
– temporal management of rules and temporality in rules [21,22,29];

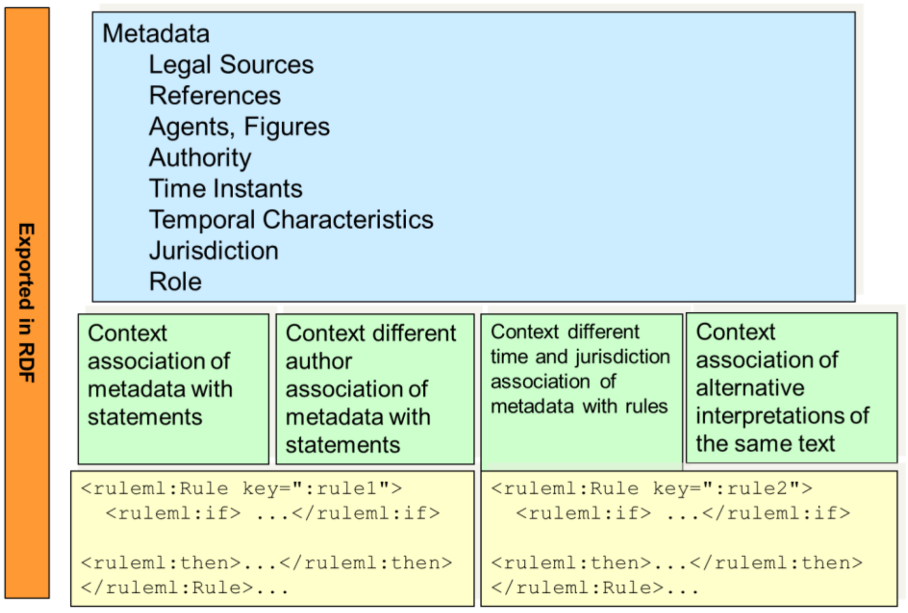
– classification of norms (i.e., constitutive, prescriptive);
– jurisdiction of norms;
– isomorphism between rules and natural language normative provisions [7];
– identification of parts of the norms (e.g., bearer, conditions);
– authorial tracking of rules.

LegalRuleML vocabulary

LegalRuleML vocabulary is designed for modelling three main blocks:

- Rules: where the legal norms are modelled in logic formalism;
- Metadata: where the fundamental data concerning the norms are stored. In this block we find IRI of the authentic legal documents, temporal parameters, jurisdiction declaration, information about agents, locations, or roles played by the agents;
- Contex: that is a bridge between rules and metadata in order to minimize the redundancy and for permitting n-ary relations between rules and metadata.

The following figure shows the organization of a LegalRuleML XML document following the abovementioned blocks.


Exported in RDF
Metadata
Legal Sources
References
Agents, Figures
Authority
Time Instants
Temporal Characteristics
Jurisdiction
Role
Context association of metadata with statements
Context different author association of metadata with statements
Context different time and jurisdiction association of metadata with rules
Context association of alternative interpretations of the same text

```
<ruleml:Rule key=":rule1">
  <ruleml:if> ...</ruleml:if>

<ruleml:then>...</ruleml:then>
</ruleml:Rule>...
```

```
<ruleml:Rule key=":rule2">
  <ruleml:if> ...</ruleml:if>

<ruleml:then>...</ruleml:then>
</ruleml:Rule>...
```

Figure 2 – LegalRuleML main blocks.

Constitutive and Prescriptive Norms

In LegalRuleML, legal rules are captured by the broader class of Statement and the hasTemplate property links a prescriptive or constitutive statement to its template, a fragment of RuleML syntax with root ruleml:Rule that denotes a class of rules. In Legal Theory norms are classified mostly in two main categories: constitutive norms and prescriptive norms. Constitutive norms usually provide definition of the terms and legal concepts used in a given jurisdiction. The function of constitutive norms is to define and create so called institutional facts [36], where an institutional fact is how a particular concept is understood in a specific institution.

The conclusion (head) of a constitutive rule cannot be a deontic formula, nor can it be a compound formula that contains a deontic formula.

```
<lrml:ConstitutiveStatement key="cs1">
        <ruleml:Rule key=":key1">
                <lrml:hasStrength>
strength of the rule
</lrml:hasStrength>
                <ruleml:if>
formula, including deontic formula
</ruleml:if>
```

```
                        <ruleml:then>
non-deontic formula
</ruleml:then>
            </ruleml:Rule>
</lrml:ConstitutiveStatement>
```

Prescriptive norms command obligations, prohibitions, permissions, etc. of a legal system, along with the conditions under which the obligations, prohibitions, permissions, etc. hold.

LegalRuleML uses deontic operators to capture such notions. Deontic operators are meant to qualify formulas. The head of a prescriptive rule is a list of deontic formulas which is called a suborder list and represented in LegalRuleML by the `<lrml:Suborder>` element.

```
<lrml:PrescriptiveStatement key="ps1">
            <ruleml:Rule key=":key1">
                        <lrml:hasStrength>
strength of the rule
</lrml:hasStrength>
                        <ruleml:if>
formula, including deontic formula
</ruleml:if>
                        <ruleml:then>
                                    <lrml:SuborderList>
list of deontic formulas
</lrml:SuborderList>
                        </ruleml:then>
            </ruleml:Rule>
</lrml:PrescriptiveStatement>
```

Deontic Structures

Standard deontic logic assumes the following relationships between the operators:

$$[\mathrm{OBL}]p \equiv \neg[\mathrm{PERM}]\neg p$$

If p is obligatory, then its opposite, $\neg p$, is not permitted.

$$[\mathrm{PRO}]p \equiv [\mathrm{OBL}]\neg p$$

If p is prohibited then its opposite is Obligatory. Alternatively, a Prohibition (PRO) of p can be understood as Obligation (OBL) of the negation of p.

The operators of Obligation, Prohibition and Permission are typically considered the basic ones, but further refinements are possible, for example, two types of permissions have been discussed in the literature on deontic logic: *weak permission* (or *negative permission*) and *strong permission* (or *positive permission*).
Weak permission corresponds to the idea that some *A* is permitted if ¬*A* is not provable as mandatory. In other words, something is allowed by a legal norm only when it is not prohibited by that legal norm [38]. The concept of strong permission is more complicated, as it amounts to the idea that some *A* is permitted by a legal norm if and only if such a legal norm explicitly states that *A* is permitted, typically as an exception to the prohibition of *A* or the obligation of its contrary, i.e., ¬*A*. It follows that a strong permission is not derived from the absence of a prohibition, but is explicitly formulated in a permissive (prescriptive) norm [2]. An example of an explicit positive permission is manifested by a “U-turn permitted” sign exposed at a traffic light, which derogates the (general) prohibition to U-turn at traffic lights.

Refinements of the concept of obligation have been proposed as well. For example it is possible to distinguish between *achievement* and *maintenance* obligations, where an *achievement* obligation is an obligation that is fulfilled if what the obligation prescribes holds at least once in the period when the obligation holds, while a *maintenance* obligation must be obeyed for all the instants when it holds (see [18] for a classification of obligations).

LegalRuleML is neutral about the different subclasses of the deontic operators and it permits to declare an IRI to a proper ontology.

```
<lrml:Obligation key="oblig1" iri="http://example.org/deontic/vocab#achievementobligation">
...
```

```
</lrml:Obligation>
```

The second method is to use an Association to link a Deontic Specification to its meaning using the applyModality element, namely:

```
<lrml:Association>
        <lrml:appliesModality iri="http://example.org/deontic/vocab#maintenanaceobligation"/>
        <lrml:toTarget keyref="#oblig101"/>
</lrml:Association>
```

Furthermore, Obligations, Prohibitions and Permissions in LegalRuleML are directed operators [24], thus they have parties (e.g. Bearer), specifying, for example, who is the subject of an Obligation or who is the beneficiary of a Permission.

```
<lrml:Obligation iri="http://example.org/deontic/vocab#obl1">
        <ruleml:slot>
                <lrml:Bearer iri="http://example.org/deontic/vocab#oblBearer"/>
                <ruleml:Ind>Y</ruleml:Ind>
        </ruleml:slot>
        <ruleml:Atom key=":atom2">
                <ruleml:Rel iri="#rel2"/>
                <ruleml:Ind>X</ruleml:Ind>
        </ruleml:Atom>
</lrml:Obligation>
```

Violation, Suborder, Penalty and Reparation.

Obligations can be violated. A violation is, basically, a situation where we have ([OBL]p) and $\neg p$

One of the characteristics of norms is that having violated them, a penalty can be introduced to compensate for the violation, where a penalty is typically a Deontic Specification. To model this feature of norms and legal reasoning [20] introduced what is called here a suborder list, and [16] showed how to combine them with defeasible reasoning for the modelling of (business) contracts. As we have mentioned above, a suborder list is a list of deontic formulas. Syntactically, a suborder list of one element can be rendered in LegalRuleML as just the element.

Obligations and Prohibitions should not be preceded by Permissions and Rights in a suborder list, for the semantics of suborder lists is such that an element holds in the list only if all the elements that precede it in the list have been violated. It is not possible to have a Violation of a Permission, so it cannot serve a purpose in the suborder list.

For example, given the rules:

$$\mathit{body} \Rightarrow [\mathrm{OBL}]A$$
$$\neg A \Rightarrow [\mathrm{OBL}]B$$

Here the body of the second rule is the negation of the content of the obligation in the head of the first rule. It is possible to merge the two rules above in the following rule:

$$\mathit{body} \Rightarrow [\mathrm{OBL}]A,\ [\mathrm{OBL}]B$$

obligation of B. This suggests that suborder lists provide a simple and convenient mechanism to model penalties. It is not uncommon for a legal text (e.g., a contract) to include sections about penalties, where one penalty is provided as compensation for many norms. To model this and to maintain the isomorphism between a source and its formalisation, LegalRuleML includes a <PenaltyStatement> element, the scope of which is to represent a statement of a penalty as a suborder list (including the trivial non-empty list of a single element).

```
<lrml:PenaltyStatement key="pen1">
      <lrml:SuborderList>
list of deontic formulas
      </lrml:SuborderList>
</lrml:PenaltyStatement>
```

LegalRuleML not only models penalties, but aims to connect the penalty statement with the corresponding Reparation element:

```
<lrml:Reparation key="rep1">
        <lrml:appliesPenalty keyref="#pen1"/>
        <lrml:toPrescriptiveStatement keyref="#ps1"/>
</lrml:Reparation>
```

Alternatives

In the legal interpretation theory [37] norms are interpreted by the judges in order to apply them to the concrete cases. Sometime the legal interpretation theories conflict and diverge from each other [11,23,33]. Linguistic elements are added to this also for different reasons such as jurisdiction (e.g., national and regional level) or for competences (e.g., civil or criminal court). LegalRuleML endeavours not to account for how different interpretations arise, but to provide a mechanism to record and represent them. We have four different templates: i) same textual provision associated to different legal rules; ii) partial textual provision and other different textual fragments associated to different legal rules:, iii) one provision associated to set of legal rules; iv) set of provisions associated to set of legal rules.

The element <lrml:Alternatives> permits to express all these interpretation templates. The following LegalRuleML fragments illustrate how to represent the four cases above:

Case 1

```
<lrml:Alternatives key="alt1">
        <lrml:fromLegalSources>
                <lrml:LegalSources>
                        <lrml:hasLegalSource keyref="#ref1"/>
                </lrml:LegalSources>
        </lrml:fromLegalSources>
        <lrml:hasAlternative keyref="#ps1"/>
        <lrml:hasAlternative keyref="#ps2"/>
</lrml:Alternatives>
```

Case 2

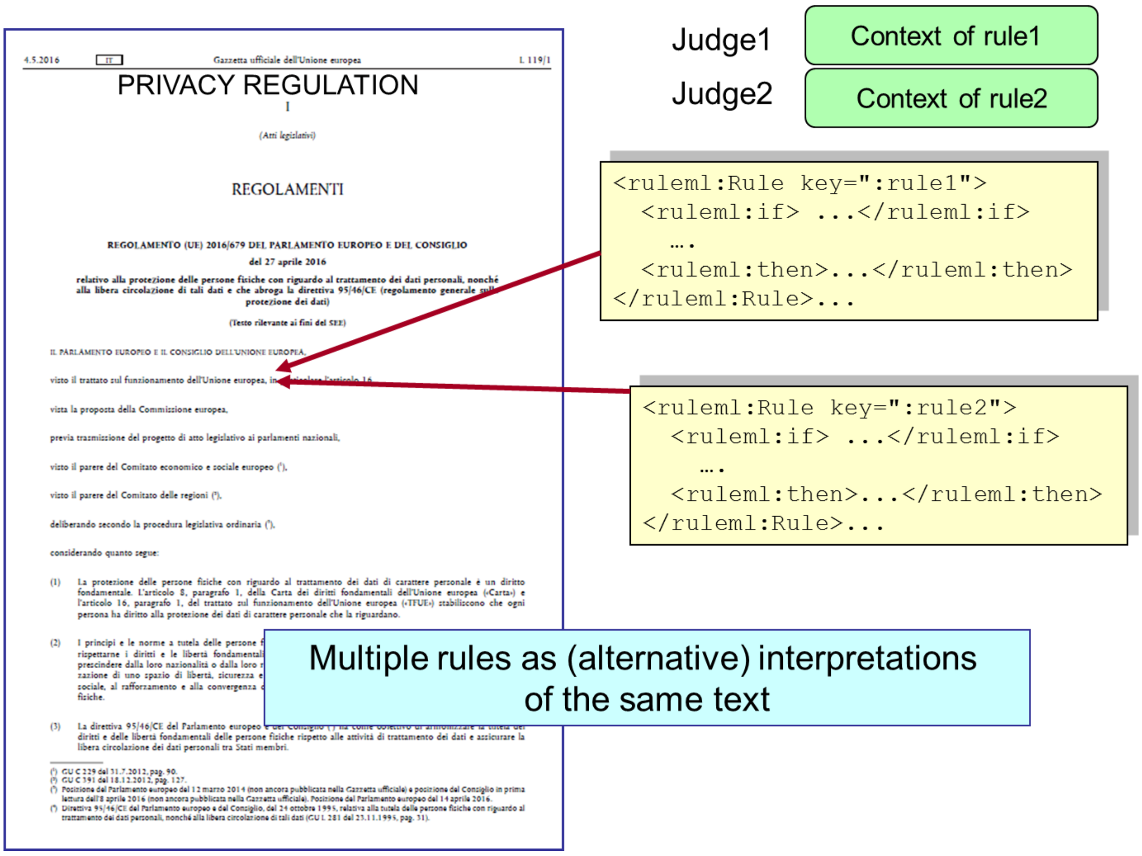
```
<lrml:Alternatives key="alt2">
        <lrml:LegalSources>
                <lrml:hasLegalSource keyref="#ref1"/>
                <lrml:hasLegalSource keyref="#ref2"/>
        </lrml:LegalSources>
        <lrml:hasAlternative keyref="#ps1"/>
        <lrml:hasAlternative keyref="#ps2"/>
</lrml:Alternatives>
```

Case 3

```
<lrml:Alternatives key="alt3">
        <lrml:LegalSources>
                <lrml:hasLegalSource keyref="#ref1"/>
        </lrml:LegalSources>
        <lrml:hasAlternative keyref="#ss1"/>
        <lrml:hasAlternative keyref="#ss2"/>
</lrml:Alternatives>
<lrml:Statements key="ss1">
        <lrml:ConstitutiveStatement keyref="#ps1"/>
        <lrml:ConstitutiveStatement keyref="#ps2"/>
</lrml:Statements>
<lrml:Statements key="ss2">
        <lrml:ConstitutiveStatement keyref="#ps3"/>
</lrml:Statements>
```

Case 4

```
<lrml:Alternatives key="alt3">
        <lrml:LegalSources>
                <lrml:hasLegalSource keyref="#ref1"/>
                <lrml:hasLegalSource keyref="#ref2"/>
        </lrml:LegalSources>
        <lrml:hasAlternative keyref="#ss1"/>
        <lrml:hasAlternative keyref="#ss2"/>
</lrml:Alternatives>
<lrml:Statements key="ss1">
        <lrml:ConstitutiveStatement keyref="#ps1"/>
        <lrml:ConstitutiveStatement keyref="#ps2"/>
</lrml:Statements>
<lrml:Statements key="ss2">
        <lrml:ConstitutiveStatement keyref="#ps1"/>
        <lrml:ConstitutiveStatement keyref="#ps3"/>
        </lrml:Statements>
```


PRIVACY REGULATION
Judge1
Context of rule1
Judge2
Context of rule2
Multiple rules as (alternative) interpretations of the same text

```
<ruleml:Rule key=":rule1">
  <ruleml:if> ...</ruleml:if>
   ….
  <ruleml:then>...</ruleml:then>
</ruleml:Rule>...
```

```
<ruleml:Rule key=":rule2">
  <ruleml:if> ...</ruleml:if>
   ….
  <ruleml:then>...</ruleml:then>
</ruleml:Rule>...
```

Figure 3 – Alternative interpretation of the same fragment of legal text.

4. LegalRuleML metadata

LegalRuleML includes a block for modelling metadata in order to qualify the legal rules, like the legal sources, jurisdiction, temporal parameters. Those metadata can be imported to other XML standards like Akoma Ntoso or Metalex/CEN.

Sources and Isomorphism

<lrml:References> is the collection dedicated to record non-IRI based identifier sources, and the attribute refIDSystemName is able to annotate the naming convention used. In the following example we refer to the Akoma Ntoso relative IRI of the section 8 of the GDPR, following the naming convention of the XML-schema in Akoma Ntoso:

```
<lrml:References refType="http://example.legalruleml.org/lrml#LegalSource">
        <lrml:Reference refersTo="ref1" refID="/akn/eu/act/regulation/eu/2016-04-27/2016-679/!main#art_8"
refIDSystemName="AkomaNtoso3.0-2017-06-06"/>
</lrml:References>
```

<lrml:LegalSource> is the construct dedicated to record the IRI based identifier sources. The following example define the source of the IRI of the European Regulation on Privacy, GDPR, using ELI naming convention:

```
<lrml:LegalSources>
        <lrml:LegalSource key="ref2" sameAs="http://eur-lex.europa.eu/eli/reg/2016/679/oj"/>
</lrml:LegalSources>
```

The list of the resources connected with the legal rules that are modelled in a LegalRuleML document are defined once usint <lrml:References> and <lrml:LegalSource>, this minimizes redundant definitions of the resources and avoids errors. The <lrml:Association> construct links LegalSources and References with rules (or fragment of rule), thus implementing the N:M relationship.

Jurisdiction and Authority

The *jurisdiction* is a legal concept that represents the geographic area (e.g., international, supra-national, national, regional, etc.) or the subject-matter (e.g., criminal law, public law, etc.) over which an authority applies its legal power. In LegalRuleML we have Jurisdiction element that annotates this concept.

.

```
<lrml:Jurisdictions>
        <lrml:Jurisdiction key="eu" sameAs="http://example.org/jurisdiction#europeanUnion
</lrml:Jurisdictions>
```

We can use Jurisdiction also to specify a limited subject-matter, for instance, legal rules which are applicable only to the European Patent Office

```
<lrml:Jurisdictions>
        <lrml:Jurisdiction key="epo" sameAs="http://example.org/jurisdiction#europeanPatentOffice"/>
</lrml:Jurisdictions>
```

Similarly, authority qualifies the rules with respect to the authenticity of the provenance of the formal model. Authority is a person or organization with the power to create, endorse, or enforce Legal Norms.

```
<lrml:Authorities>
        <lrml:Authority key="euParliament" sameAs="eu:organization.owl#europeanParliament">
                <lrml:hasType iri="lrmlv:Legislature"/>
        </lrml:Authority>
</lrm:Authorities>
```

Agent, Figure, Role

An Agent is an entity that acts or has the capability to act. An Agent could be a physical person, a database, or a bot; for this reason we have the sub-element <lrml:hasType> that expresses the category of agent.

```
<lrml:Agents>
        <lrml:Agent key="at" sameAs="http:example.org/agents#AntonioTajani">
                <lrml:hasType iri="http://example.org/types#Person"/>
        </lrml:Agent>
</lrml:Agents>
```

The Agent usually is the author of the legal rules model and he/she/it can act in a particular function (e.g., as European Parliament President). A Figure in LegalRuleML is an instantiation of a function.

```
<lrml:Figures>
        <lrml:hasMemberType iri="http://example.org/figure-types#LegislativeFigure"/>
        <lrml:Figure key="eupp">
                <lrml:hasFunction iri="http://example.org/functions#europenParliamentPresident"/>
                <lrml:hasActor keyref="#at"/>
        </lrml:Figure>
</lrml:Figures>
```

In the end we associate the Actor that fills a Role (using <lrml:filledBy>) for a particular rule.

```
<lrml:Roles>
        <lrml:Role key="role1" iri="http://example.org/roles#author">
                <lrml:filledBy keyref="#at"/>
                <lrml:forExpression keyref="#rule1_a"/>
        </lrml:Role>
</lrml:Roles>
```

Using this mechanism we can filter all the rules modelled by a particular Actor when he/she/it acts as a particular figure.

Time and Events

Legal texts are often amended as a society or judicial system evolves. Norms and rules are valid in a particular interval of time and with respect to three main legal axes: when they come into force (entry or enforceability), when they effect the intended or desired result (efficacy or commencement), and when they apply (applicability).
The temporal dimensions of the complex events that are the content of the textual provision (e.g., when a person is to present a tax application) are modelled using the normal rules, but for the external temporal parameters that governs the rule itself, LegalRuleML provides a special mechanism for representing temporal situations. We model the intervals and temporal parameters that define the period of enter into force or efficacy or applicability of the rules. Secondly it is fundamental to link link the temporal parameters to any part of a rule (e.g., Atom, Rel, Ind, if, then, etc.) with a very fine granularity. The following fragment shows the definition of the instant time using the <ruleml:Time> element wrapped by the <lrml:Times> collection elements. The first time is the enter into force of the original version of the Italian Privacy Law (196/2003 Act); the second date is the date of efficacy of the second amended and updated version of the same Act:

```
<lrml:Times>
        <ruleml:Time key="t1">
                <ruleml:Data xsi:type="xs:dateTime">2004-01-01T01:01:00.0Z </ruleml:Data>
        </ruleml:Time>
        <ruleml:Time key="t2">
                <ruleml:Data xsi:type="xs:dateTime">2011-05-13T01:01:00.0Z</ruleml:Data>
        </ruleml:Time>
</lrml:Times>
```

The time instants are combined in intervals according with the legal TemporalCharacteristics, e.g., enter into force, efficacy, applicability. In the following case the tblock1 defines the interval [t_1, t_2] enter into force of the version 1.

```
<lrml:TemporalCharacteristics key="tblock1">
        <lrml:TemporalCharacteristic key="e1-b">
                <lrml:forStatus iri="lrmlv:enterIntoForce"/>
                <lrml:hasStatusDevelopment iri="lrmlv:Starts"/>
                <lrml:atTime keyref="#t1"/>
        </lrml:TemporalCharacteristic>
        <lrml:TemporalCharacteristic key="e1-e">
                <lrml:forStatus iri="lrmlv:enterIntoForce"/>
                <lrml:hasStatusDevelopment iri="lrmlv:Ends"/>
                <lrml:atTime keyref="#t2"/>
        </lrml:TemporalCharacteristic>
</lrml:TemporalCharacteristics>
```

After this definition of the time interval or instant, it is possible to associate them to the legal rules using the <lrml:Association> element or the <lrml:Context> element for associating the temporal parameters with any part of the rule formalization.

Association and Context

To avoid redundancy, we have the element <Association> which can be used to group meta information referring to several rules or portions of them. In the following example we have two associations inside of the collection element <Associations>. The first <Association> applies the temporal parameters of tblock1 to the prescriptive statements 1 and 2. In the second one authority and jurisdiction properties are applied to prescriptive statements 1 and 2:

```
<lrml:Associations key="sourceBlock1">
        <lrml:Association>
                <lrml:appliesTemporalCharacteristics keyref="#tblock1"/>
                <lrml:toTarget keyref="#ps1"/>
                <lrml:toTarget keyref="#ps2"/>
        </lrml:Association>
        <lrml:Association>
                <lrml:appliesAuthority keyref="ex:#euParliamentPresident"/>
                <lrml:appliesJurisdiction keyref="ex:#eu"/>
                <lrml:toTarget keyref="#ps1"/>
                <lrml:toTarget keyref="#ps2"/>
        </lrml:Association>
</lrml:Associations>
```

The same it is possible with all the metadata of LegalRuleML: Authority, LegalSource, Jurisdiction, Strenght, TemporalCharacterisctis, etc.
To represent such parameters to the rules, we use the <lrml:Context> element, which permits the description of all the characteristics that are linked to a particular rule (e.g., rule1).
The mechanism combines the relationships and the target rules, and it acts as a bridge between metadata and rules or fragments of them.

```
<lrml:Context key="ruleInfo4" hasCreationDate="#t1">
        <lrml:appliesSource keyref="#art_8"/>
        <lrml:appliesTemporalCharacteristics keyref="#tblock1"/>
        <lrml:appliesStrength iri="lrmlv:Defeater"/>
        <lrml:appliesAuthority keyref="authorities:euParliamentPresident "/>
        <lrml:appliesJurisdiction keyref="jurisdictions:eu"/>
        <lrml:appliesAssociations keyref="#assoc1"/>
        <lrml:appliesAlternatives keyref="#alt2"/>
        <lrml:inScope keyref="#rule1"/>
</lrml:Context>
```

5. LegalRuleML metamodel

LegalRuleML provides also a metadata model for transforming the information stored inside of the XML file in RDF triples. This permits to reuse the legal knowledge included in the legal rules description for different purposes: i) improve information retrieval of the legal documents; ii) filter the legal rules respect some parameters (e.g., jurisdiction, author, time); iii) combine legal knowledge with Linked Open Data Cloud[2] information. RDFS [8] is used to define the LegalRuleML metamodel, and graphs of the RDFS schemas.

LegalRuleML RDFS

LegalRuleML has an RDFS model for transforming the metadata in RDF triples. The most important parts are related to the qualification of the legal statement (ConstitutiveStatement and PrescriptiveStatement) and to the deontic operators (Obligation, Right, Permission, Prohibition, Compliance, etc.). The PrescriptiveStatement is disjoined with ConstitutiveStatement, ReparationStatement is disjoined with PenaltyStatement and FactualStatement and in general LogicalFormulaStatement is disjoined with RulesStatement.

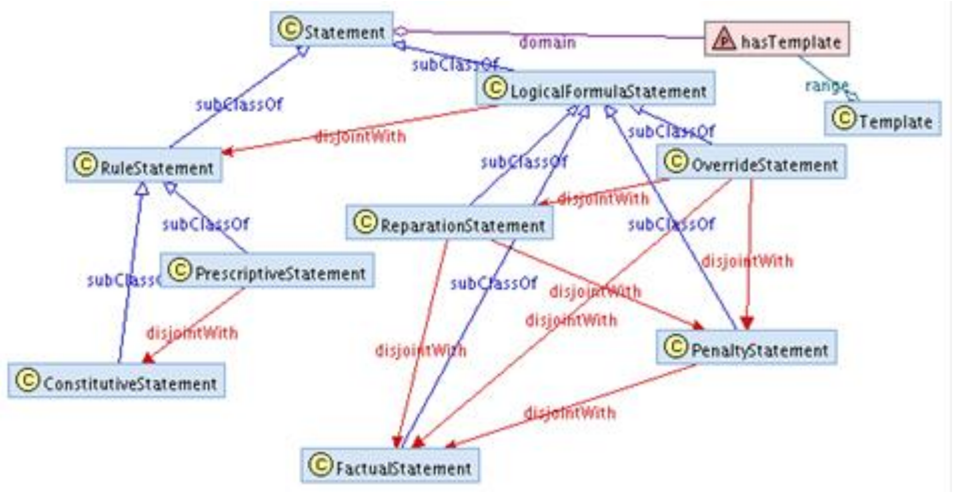


Figure 4 – Metamodel of the Statement qualifications.

[2] http://lod-cloud.net/

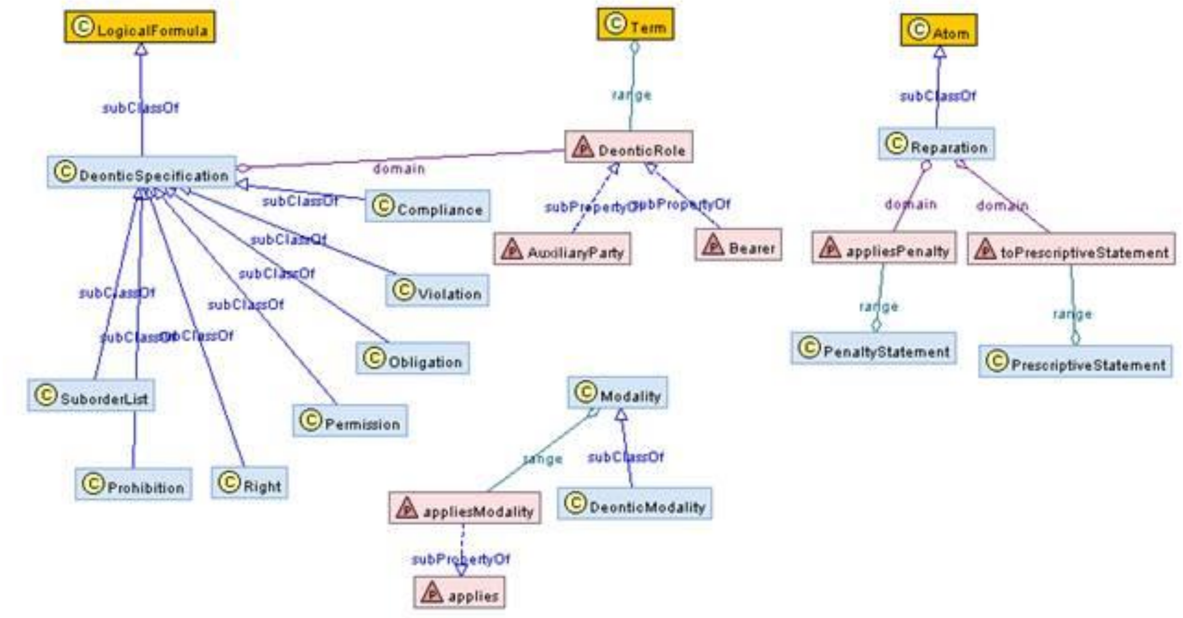


Figure 5 – Metamodel of the deontic operators.

One limitation of this metamodel is the plain relationships representation between deontic operators. There is lack of axioms on relationships, no disjoined axiom for the is-a relationship, no existential quantifier restrictions.

6. Pattern of conditions in the request

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="triplifyMerger-ids.xsl"?>
<!DOCTYPE RDF [
  <!ENTITY base "http://docs.oasis-open.org/legalruleml/examples/compactified/ex8-defeasible-compact">
  <!ENTITY lrml "http://docs.oasis-open.org/legalruleml/ns/v1.0">
  <!ENTITY lrmlv "http://docs.oasis-open.org/legalruleml/ns/v1.0/vocab">
  <!ENTITY rulemlmm "http://ruleml.org/1.0/metamodel">
  <!ENTITY ruleml "http://ruleml.org/spec">
  <!ENTITY xs "http://www.w3.org/2001/XMLSchema">
  <!ENTITY defeasible-ontology "http://example.org/defeasible/vocab">
  <!ENTITY deontic-ontology "http://example.org/deontic/vocab">
]>
	Definition of the namespaces

<lrml:LegalRuleML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.oasis-open.org/legalruleml/examples/compactified/ex9-alternatives-compact#"
  xmlns:lrml="http://docs.oasis-open.org/legalruleml/ns/v1.0/"
  xmlns:ruleml="http://ruleml.org/spec" xmlns:rulemlmm="http://ruleml.org/1.0/metamodel#"
  xml:base="http://docs.oasis-open.org/legalruleml/examples/compactified/ex9-alternatives-compact"
  xsi:schemaLocation="http://docs.oasis-open.org/legalruleml/ns/v1.0/ ./xsd-schema/compact/lrml-compact.xsd">
  <lrml:Comment> Request4 </lrml:Comment>
  <lrml:LegalReferences>
	Metadata block

```
refType="http://example.org/lrml#LegalSource">
                <lrml:LegalReference refersTo="ref1"
                    refID="/akn/eu/act/directive/2011-06-06/2011-
61/eng@2021-08-02/!main~art_69a__para_1"
                    refIDSystemName="AkomaNtoso3.0"/>
            </lrml:LegalReferences>
            <lrml:LegalSources>
                <lrml:LegalSource key="ls1"
sameAs="http://data.europa.eu/eli/dir/2011/61/2021-08-02"
                    type="eli"/>
            </lrml:LegalSources>

            <lrml:Agents>
                <lrml:Agent key="euCommission"
sameAs="http:example.org/agents#EUCommission">
                    <lrml:hasType
iri="http://example.org/types#organization"/>
                </lrml:Agent>
            </lrml:Agents>

            <lrml:Roles>
                <lrml:Role key="role1"
iri="http://example.org/roles#author">
                    <lrml:filledBy keyref="#euCommision"/>
                    <lrml:forExpression keyref="#atom1"/>
                    <lrml:forExpression keyref="#atom2"/>
                    <lrml:forExpression keyref="#atom3"/>
                    <lrml:forExpression keyref="#atom4"/>
                    <lrml:forExpression keyref="#atom5"/>
                    <lrml:forExpression keyref="#atom6"/>
                    <lrml:forExpression keyref="#atom7"/>
                </lrml:Role>
            </lrml:Roles>

            <lrml:Associations>
                <lrml:Association key="ascs0">
                    <lrml:appliesSource keyref="#ls1"/>
                    <lrml:appliesSource keyref="#ref1"/>
                    <lrml:toTarget keyref="#atom1"/>
                    <lrml:toTarget keyref="#atom2"/>
                    <lrml:toTarget keyref="#atom3"/>
                    <lrml:toTarget keyref="#atom4"/>
                    <lrml:toTarget keyref="#atom5"/>
                    <lrml:toTarget keyref="#atom6"/>
                    <lrml:toTarget keyref="#atom7"/>
                </lrml:Association>
            </lrml:Associations>

            <lrml:Statements>
                <lrml:PrescriptiveStatement key="ps1">
                    <ruleml:Rule key=":ruletemplate1"
closure="universal">
                        <lrml:Paraphrase> Before the entry into force of
the delegated acts referred to in Article 67(6) pursuant to which the rules
set out in Article 35 and Articles 37 to 41 become applicable, the
Commission shall submit a report to the European Parliament and to the
Council, taking into account the result of an assessment of the passport
regime provided in this Directive including the extension of that regime to
non-EU AIFMs.</lrml:Paraphrase>
                        <ruleml:if>
                            <ruleml:And key=":and1">
                                <ruleml:Atom key=":atom1">
                                    <ruleml:Rel
iri=":EUcommission"/>
                                    <ruleml:Var>X</ruleml:Var>
                                </ruleml:Atom>
                                <ruleml:Atom key=":atom2">
                                    <ruleml:Rel
```

LegalReference is used for connecting the URI of AKN fragment involved in the condition.

LegalSource is used for connecting the URI of ELI HTTP URL.

Agent defines the euComission as an agents of ELI.

Role is declaring that the euCommission is the author of the rules.

The Role block of metadata connect the author with the agent, and the destination of the authorship (atoms of the rules).

It is possible to have multiple authors of different atoms (e.g., Eu Parliament).

Association is a block that permits N:M relationships between metadata and the atoms.
An atom is the basic unit of a rule.

Prescriptive rule includes obligation, permission, prohibition operators.

Here the core part of the rule.
If the atoms are true then the obligation is activated.

The AND operator concatenates different atoms.

The atom is composed by
Rel – relationship/predicate

```
iri=":entryIntoForceDelegatedAct"/>
                                                <ruleml:Var>H</ruleml:Var>
                                        </ruleml:Atom>
                                  </ruleml:And>
                              </ruleml:if>
                              <ruleml:then><lrml:Obligation iri=":obligation">
                                  <ruleml:Atom key=":atom3">
                                        <ruleml:Rel iri=":report"/>
                                        <ruleml:Var>X</ruleml:Var>
                                        <ruleml:Var>J</ruleml:Var>
                                  </ruleml:Atom></lrml:Obligation>
                              </ruleml:then>
                          </ruleml:Rule>
                      </lrml:PrescriptiveStatement>
                      <lrml:PrescriptiveStatement key="ps2">
                          <ruleml:Rule key=":ruletemplate2"
closure="universal">
                              <lrml:Paraphrase> That report shall be
accompanied, where appropriate, by a legislative
proposal.</lrml:Paraphrase>
                              <ruleml:if>
                                  <ruleml:And key=":and2">
                                        <ruleml:Atom key=":atom4">
                                                <ruleml:Rel
iri=":EUcommission"/>
                                                <ruleml:Var>X</ruleml:Var>
                                        </ruleml:Atom>
                                        <ruleml:Atom key=":atom5">
                                                <ruleml:Rel iri=":report"/>
                                                <ruleml:Var>J</ruleml:Var>
                                        </ruleml:Atom>
                                        <ruleml:Atom key=":atom6">
                                                <ruleml:Rel
iri=":amendmentsRequired"/>
                                                <ruleml:Var>Y</ruleml:Var>
                                        </ruleml:Atom>
                                  </ruleml:And>
                              </ruleml:if>
                              <ruleml:then>
                                  <lrml:Obligation iri=":obligation">
                                        <ruleml:Atom key=":atom7">
                                                <ruleml:Rel
iri=":produceProposalToAmendments"/>
                                                <ruleml:Var>X</ruleml:Var>
                                                <ruleml:Var>A</ruleml:Var>
                                        </ruleml:Atom>
                                  </lrml:Obligation>
                              </ruleml:then>
                          </ruleml:Rule>
                      </lrml:PrescriptiveStatement>
                    </lrml:Statements>
</lrml:LegalRuleML>
```

Var – variable
Ind – constant
In our case we have only variables.

The second part of the rule defines the obligation

This rule presents the second condition to have the obligation to present a legislative proposal.

```
<!--?xml version="1.0" encoding="UTF-8"?-->
<rdf:rdf xmlns:lrmlmm="http://docs.oasis-
open.org/legalruleml/ns/v1.0/metamodel#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rulemm="http://docs.oasis-open.org/legalruleml/ns/v1.0/rule-
metamodel#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://docs.oasis-
open.org/legalruleml/examples/compactified/ex9-alternatives-compact#"
  xmlns:lrml="http://docs.oasis-open.org/legalruleml/ns/v1.0/"
  xmlns:ruleml="http://ruleml.org/spec"
xmlns:rulemlmm="http://ruleml.org/1.0/metamodel#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

Metadata language in RDF – conversion in RDF of the LegalRuleML XML representation.

```
  xml:base="http://docs.oasis-
open.org/legalruleml/examples/compactified/ex9-alternatives-compact">
  <lrmlmm:legalruleml>
    <lrmlmm:comment>
      <lrmlmm:symbol rdf:parsetype="Literal"> Request4
</lrmlmm:symbol>
    </lrmlmm:comment>
    <lrmlmm:legalreferences>
      <lrmlmm:memberreferencetype
rdf:resource="http://example.org/lrml#LegalSource">
        <lrmlmm:legalreference>
          <lrmlmm:refersto>
            <rdf:description rdf:id="ref1">
              <lrmlmm:refid>/akn/eu/act/directive/2011-06-06/2011-
61/eng@2021-08-02/!main~art_69a__para_1</lrmlmm:refid>

<lrmlmm:refidsystemname>AkomaNtoso3.0</lrmlmm:refidsystemname>
            </rdf:description>
          </lrmlmm:refersto>
        </lrmlmm:legalreference>
        <lrmlmm:hasmembers
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"
        > </lrmlmm:hasmembers>
      </lrmlmm:memberreferencetype>
    </lrmlmm:legalreferences>
    <lrmlmm:legalsources>
      <lrmlmm:legalsource rdf:id="ls1">
        <owl:sameas
rdf:resource="http://data.europa.eu/eli/dir/2011/61/2021-08-02">
          <rdf:type rdf:resource="eli"> </rdf:type>
        </owl:sameas>
      </lrmlmm:legalsource>
      <lrmlmm:hasmembers rdf:resource="http://www.w3.org/1999/02/22-
rdf-syntax-ns#nil"
      > </lrmlmm:hasmembers>
    </lrmlmm:legalsources>

    <lrmlmm:agents>
      <lrmlmm:agent rdf:id="euCommission">
        <owl:sameas
rdf:resource="http:example.org/agents#EUCommission">
          <lrmlmm:hastype
rdf:resource="http://example.org/types#organization"
          > </lrmlmm:hastype>
        </owl:sameas>
      </lrmlmm:agent>
      <lrmlmm:hasmembers rdf:resource="http://www.w3.org/1999/02/22-
rdf-syntax-ns#nil"
      > </lrmlmm:hasmembers>
    </lrmlmm:agents>

    <lrmlmm:roles>
      <lrmlmm:role rdf:id="role1">
        <owl:sameas rdf:resource="http://example.org/roles#author">
          <lrmlmm:filledby rdf:resource="#euCommision">
            <lrmlmm:forexpression rdf:resource="#atom1">
              <lrmlmm:forexpression rdf:resource="#atom2">
                <lrmlmm:forexpression rdf:resource="#atom3">
                  <lrmlmm:forexpression rdf:resource="#atom4">
                    <lrmlmm:forexpression rdf:resource="#atom5">
                      <lrmlmm:forexpression rdf:resource="#atom6">
                        <lrmlmm:forexpression rdf:resource="#atom7"
                        > </lrmlmm:forexpression>
                      </lrmlmm:forexpression>
                    </lrmlmm:forexpression>
                  </lrmlmm:forexpression>
                </lrmlmm:forexpression>
              </lrmlmm:forexpression>
            </lrmlmm:forexpression>
          </lrmlmm:filledby>
        </owl:sameas>
      </lrmlmm:role>
      <lrmlmm:hasmembers rdf:resource="http://www.w3.org/1999/02/22-
rdf-syntax-ns#nil"
      > </lrmlmm:hasmembers>
    </lrmlmm:roles>

    <lrmlmm:associations>
      <lrmlmm:association rdf:id="ascs0">
        <lrmlmm:appliessource rdf:resource="#ls1">
          <lrmlmm:appliessource rdf:resource="#ref1">
            <lrmlmm:totarget rdf:resource="#atom1">
              <lrmlmm:totarget rdf:resource="#atom2">
                <lrmlmm:totarget rdf:resource="#atom3">
                  <lrmlmm:totarget rdf:resource="#atom4">
                    <lrmlmm:totarget rdf:resource="#atom5">
                      <lrmlmm:totarget rdf:resource="#atom6">
                        <lrmlmm:totarget rdf:resource="#atom7">
</lrmlmm:totarget>
                      </lrmlmm:totarget>
                    </lrmlmm:totarget>
                  </lrmlmm:totarget>
                </lrmlmm:totarget>
              </lrmlmm:totarget>
            </lrmlmm:totarget>
          </lrmlmm:appliessource>
        </lrmlmm:appliessource>
      </lrmlmm:association>
      <lrmlmm:hasmembers rdf:resource="http://www.w3.org/1999/02/22-
rdf-syntax-ns#nil"
      > </lrmlmm:hasmembers>
    </lrmlmm:associations>

    <lrmlmm:statements>
      <lrmlmm:prescriptivestatement rdf:id="ps1">
        <rulemm:rule rdf:about="#ruletemplate1">
          <rulemm:closure rdf:resource="universal">
            <lrmlmm:paraphrase> Before the entry into force of the
delegated acts referred to
              in Article 67(6) pursuant to which the rules set out in Article
35 and Articles
              37 to 41 become applicable, the Commission shall submit a
report to the
              European Parliament and to the Council, taking into account
the result of an
              assessment of the passport regime provided in this Directive
including the
              extension of that regime to non-EU
AIFMs.</lrmlmm:paraphrase>
            <rulemm:if>
              <rulemm:and rdf:about="#and1">
                <rulemm:atom rdf:about="#atom1">
                  <rulemm:rel>
                    <owl:sameas rdf:resource="#EUcommission">
                      <rulemm:symbol> </rulemm:symbol>
                    </owl:sameas>
                  </rulemm:rel>
                  <rulemm:var>
                    <rulemm:symbol>X</rulemm:symbol>
                  </rulemm:var>
                  <rulemm:args
                    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"
                  > </rulemm:args>
                </rulemm:atom>
                <rulemm:atom rdf:about="#atom2">
```

```
                  <rulemm:rel>
                    <owl:sameas
rdf:resource="#entryIntoForceDelegatedAct">
                      <rulemm:symbol> </rulemm:symbol>
                    </owl:sameas>
                  </rulemm:rel>
                  <rulemm:var>
                    <rulemm:symbol>H</rulemm:symbol>
                  </rulemm:var>
                  <rulemm:args
                    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"
                  > </rulemm:args>
                </rulemm:atom>
                <rulemm:formulas
                  rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"
                > </rulemm:formulas>
              </rulemm:and>
            </rulemm:if>
            <rulemm:then>
              <rulemm:atom rdf:about="#atom3">
                <rulemm:rel>
                  <owl:sameas rdf:resource="#report">
                    <rulemm:symbol> </rulemm:symbol>
                  </owl:sameas>
                </rulemm:rel>
                <rulemm:var>
                  <rulemm:symbol>X</rulemm:symbol>
                </rulemm:var>
                <rulemm:var>
                  <rulemm:symbol>J</rulemm:symbol>
                </rulemm:var>
                <rulemm:args rdf:resource="http://www.w3.org/1999/02/22-
rdf-syntax-ns#nil"
                > </rulemm:args>
              </rulemm:atom>
            </rulemm:then>
          </rulemm:closure>
        </rulemm:rule>
      </lrmlmm:prescriptivestatement>
      <lrmlmm:prescriptivestatement rdf:id="ps2">
        <rulemm:rule rdf:about="#ruletemplate2">
          <rulemm:closure rdf:resource="universal">
            <lrmlmm:paraphrase> That report shall be accompanied,
where appropriate, by a
              legislative proposal.</lrmlmm:paraphrase>
            <rulemm:if>
              <rulemm:and rdf:about="#and2">
                <rulemm:atom rdf:about="#atom4">
                  <rulemm:rel>
                    <owl:sameas rdf:resource="#EUcommission">
                      <rulemm:symbol> </rulemm:symbol>
                    </owl:sameas>
                  </rulemm:rel>
                  <rulemm:var>
                    <rulemm:symbol>X</rulemm:symbol>
                  </rulemm:var>
                  <rulemm:args
                    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"
                  > </rulemm:args>
                </rulemm:atom>
                <rulemm:atom rdf:about="#atom5">
                  <rulemm:rel>
                    <owl:sameas rdf:resource="#report">
                      <rulemm:symbol> </rulemm:symbol>
                    </owl:sameas>
                  </rulemm:rel>
                  <rulemm:var>
                    <rulemm:symbol>J</rulemm:symbol>
                  </rulemm:var>
                  <rulemm:args
                    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"
                  > </rulemm:args>
                </rulemm:atom>
                <rulemm:atom rdf:about="#atom6">
                  <rulemm:rel>
                    <owl:sameas rdf:resource="#amendmentsRequired">
                      <rulemm:symbol> </rulemm:symbol>
                    </owl:sameas>
                  </rulemm:rel>
                  <rulemm:var>
                    <rulemm:symbol>Y</rulemm:symbol>
                  </rulemm:var>
                  <rulemm:args
                    rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"
                  > </rulemm:args>
                </rulemm:atom>
                <rulemm:formulas
                  rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
ns#nil"
                > </rulemm:formulas>
              </rulemm:and>
            </rulemm:if>
            <rulemm:then>
              <lrmlmm:obligation>
                <owl:sameas rdf:resource="#obligation">
                  <rulemm:atom rdf:about="#atom7">
                    <rulemm:rel>
                      <owl:sameas
rdf:resource="#produceProposalToAmendments">
                        <rulemm:symbol> </rulemm:symbol>
                      </owl:sameas>
                    </rulemm:rel>
                    <rulemm:var>
                      <rulemm:symbol>X</rulemm:symbol>
                    </rulemm:var>
                    <rulemm:var>
                      <rulemm:symbol>A</rulemm:symbol>
                    </rulemm:var>
                    <rulemm:args
                      rdf:resource="http://www.w3.org/1999/02/22-rdf-
syntax-ns#nil"
                    > </rulemm:args>
                  </rulemm:atom>
                </owl:sameas>
              </lrmlmm:obligation>
            </rulemm:then>
          </rulemm:closure>
        </rulemm:rule>
      </lrmlmm:prescriptivestatement>
      <lrmlmm:hasmembers rdf:resource="http://www.w3.org/1999/02/22-
rdf-syntax-ns#nil"
      > </lrmlmm:hasmembers>
    </lrmlmm:statements>
  </lrmlmm:legalruleml>
</rdf:rdf>
```