LegalRuleML: XML-Based Rules and Norms

Monica Palmirani[1], Guido Governatori[2], Antonino Rotolo[1], Said Tabet[3],
Harold Boley[4], and Adrian Paschke[5]

[1] CIRSFID, University of Bologna, Italy
{monica.palmirani,antonino.rotolo}@unibo.it
[2] NICTA, Brisbane, Australia
guido.governatori@nicta.com.au
[3] EMC Corporation, Office of the CTO, Hopkinton, MA, USA
said.tabet@emc.com
[4] Institute for Information Technology, National Research Council Canada
Fredericton, NB, Canada
harold.boley@nrc.gc.ca
[5] Computer Science Department, Freie Universitaet Berlin, Germany
paschke@inf.fu-berlin.de

Abstract. Legal texts are the foundational resource where to discover rules and norms that feed into different concrete (often XML-based) Web applications. Legislative documents provide general norms and specific procedural rules for eGovernment and eCommerce environments, while contracts specify the conditions of services and business rules (e.g. service level agreements for cloud computing), and judgments provide information about the legal argumentation and interpretation of norms to concrete case-law. Such legal knowledge is an important source that should be detected, properly modeled and expressively represented in order to capture all the domain particularities. This paper provides an extension of RuleML called LegalRuleML for fostering the characteristics of legal knowledge and to permit its full usage in legal reasoning and in the business rule domain. LegalRuleML encourages the effective exchange and sharing of such semantic information between legal documents, business rules, and software applications.

Keywords: Legal Rules, Legal XML Standards, Semantic Web, Legal Reasoning, LegalRuleML.

1 Rationale

The AI & Law community dedicated a good part of the last twenty years to model legal norms using different logics and formalisms [30]. The methodology used starts with a re-interpretation of a legal text by a Legal Knowledge Engineer who extracts the norms, applies models and a theory within a logical framework, and finally represents the norms using a particular formalism. In the last decade, several Legal XML standards were proposed to describe and represent legal texts [23; 35; 4] with XML based rules (RuleML, SWRL, RIF, LKIF, etc.) [12; 7]. In the meantime, the Semantic Web, in particular Legal Ontology research combined with semantic norm

F. Olken et al. (Eds.): RuleML 2011 - America, LNCS 7018, pp. 298–312, 2011.

extraction based on Natural Language Processing (NLP) [10; 8; 25; 24], gave a great impulse to the modeling of legal concepts [7]. In this paramount scenario, there is urgent need to find a robust and expressive XML annotation, compliant with the Semantic Web technologies, able to meet all the unique particular aspects rising from the legal domain and in the same time close the gap between legal text descriptions, using XML techniques, and norms modeling, in order to realize an integrated and self-contained representation of legal resources available on the Web [26; 22]. This integration is fundamental for fostering Semantic Web advantages applied to legal norms like: NLP, IR, graph representation, Web ontologies and rules, etc.

The second requirement is to capture the processes description embedded into the norms for extracting the business rules and for passing them to other important applications like workflow or business rule engines. There is a gap currently between the norms modeling and business rules, even if the latter are strongly influenced by the former. This knowledge is an important input for several applications in Cloud computing, eGovernment, eCommerce [20; 14; 31], eHealth, etc. Of particular importance in such scenario, the requirement for compliance checking [32; 16; 18; 19] theory and applications.

The third aspect is to permit an agile annotation of all the instruments necessary to capture the legal norms [11] that usually the normal rule XML standard doesn't include. Our goal is to have an expressive XML standard for modeling normative rules that will satisfy the legal domain requirements. This will enable a legal reasoning level on top of the ontological layer, following the Tim Berners-Lee semantic web stack[1]. Finally, particular attention is paid to the Linked Open Data [3] approach to modeling, regarding not only the semantics of raw data (act, contracts, court files, judgments, etc.), but also of rules in conjunction with their functionality and usage. Without rules or axioms, legal concepts represent nothing more than a taxonomy [35].

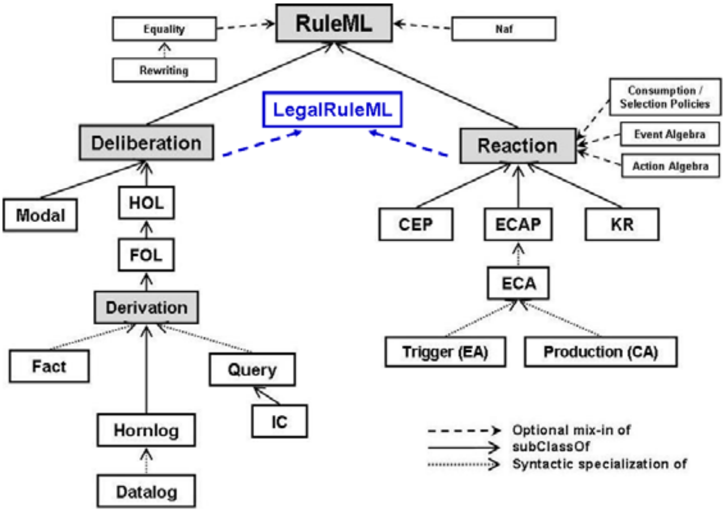

Fig. 1. LegalRuleML position in the current RuleML architecture (adapted from [5])

[1] http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#%2824%29

In this scenario we have extended RuleML [5; 6; 38; 34] to include, in orthogonal way (see Fig. 1), a new dialect capturing all those requirements, not fully incorporated in the original version of RuleML. We call this new dialect LegalRuleML. It is positioned between the Deliberation rules and the Reaction Rules facilitating the modeling of either norms or business rules. This approach provides support for the implementation of reasoning engines combining both norms and business rules.

2 Characteristics of Legal Norms

RuleML provides a good framework to start working towards the above-mentioned goals. We define the main characteristics needed for modeling norms and the essential features needed to apply legal reasoning in effective and computable way. We can divide the characteristics in three main groups: semantic features, logic features, and legal process features.

2.1 Semantic Features

ISOMORPHISM [1] To ease validation and maintenance, there should be a one-to-one correspondence between the rules in the formal model and the units of natural language text which express the rules in the original legal sources, such as sections of legislation. This entails, for example, that a general rule and separately stated exceptions, in different sections of a statute, should not be converged into a single rule in the formal model.

REIFICATION [13] Rules are objects with properties, such as:

- **Jurisdiction**. The limits within which the rule is authoritative and its effects are binding (of particular importance are spatial and geographical references to model jurisdiction).
- **Authority** [29] Who produced the rule, a feature which indicates the ranking status of the rule within the sources of law (whether the rule is a constitutional provision, a statute, is part of a contract clause or is the ruling of a precedent, and so on).
- **Temporal properties** [28, 15, 27] Rules usually are qualified by temporal properties, such as: the time when the norm is enforced and/or has been enacted; the time when the norm can produce legal effects; the time when the normative effects hold.

RULE SEMANTICS. Any language for modeling legal rules should be based on precise and rigorous semantics, which allow for the correct computation of legal effects that should follow from a set of legal rules.

NORMATIVE EFFECTS. There are many normative effects that follow from applying rules, such as obligations, permissions, prohibitions and also more articulated effects such as those introduced, e.g., by Hohfeld (see [36]). Below is a rather comprehensive list of normative effects [33]:

- **Evaluative** indicates that something is good or bad, is a value to be optimized or an evil to be minimized. For example, "Human dignity is valuable", "Participation ought to be promoted";
- **Qualificatory** describes a legal quality to a person or an object. For example, "Joe is a citizen";
- **Definitional** specifies the meaning of a term. For example, "Tolling agreement means any agreement to put a specified amount of raw material per period through a particular processing facility";
- **Deontic** typically imposes the obligation or confers the permission to do a certain action. For example, "x has the obligation to do A";
- **Potestative** assigns powers. For example, "A worker has the power to terminate his work contract";
- **Evidentiary** establishes the conclusion to be drawn from certain evidence. For example, "It is presumed that dismissal was discriminatory";
- **Existential** indicates the beginning or the termination of the existence of a legal entity. For example, "The company ceases to exist";
- **Norm-concerning effects** states modifications of norms such as abrogation, repeal, substitution, and so on.

VALUES [2]. Usually, some values are promoted by legal rules. The modeling of rules sometimes needs to support the representation of values and value preferences, which can also play the role of meta-criteria for solving rule conflicts (given two conflicting rules r1 and r2, value v1, promoted by r1, is preferred to value v2, promoted by r2, and so r1 overrides r2).

2.2 Logic Features

DEFEASIBILITY [13; 29; 37]. When the antecedent of a rule is satisfied by the facts of a case (or via other rules), the conclusion of the rule presumably holds, but is not necessarily true. The defeasibility of legal rules breaks down into the following issues:

- **Conflicts** [29]. Rules can conflict, namely, they may lead to incompatible legal effects. Conceptually, conflicts can be of different types whether two conflicting rules: i) are such that one is an exception of the other (i.e., one is more specific than the other); ii) have a different ranking status; iii) have been enacted at different times.
- **Exclusionary rules** [13; 29; 37]. Some rules provide one way to explicitly undercut other rules, namely, to make them inapplicable.

CONTRAPOSITION [29]. Rules do not counterpose. If the conclusion of a rule is not true, the rule does not sanction any inferences about the truth of its premises.

CONTRIBUTORY REASONS OR FACTORS [37]. It is not always possible to formulate precise rules, even defeasible ones, for aggregating the factors relevant for resolving a legal issue. For example: "The educational value of a work needs to be taken into consideration when evaluating whether the work is covered by the copyright doctrine of fair use."

Rule validity [17]. Rules can be invalid or become invalid. Deleting invalid rules is not an option when it is necessary to reason retroactively with rules, which were valid at various times over a course of events. For instance: The annulment of a norm is usually seen as a kind of repeal, which invalidates the norm and removes it from the legal system as if it had never been enacted. The effect of an annulment applies *ex tunc*: annulled norms are prevented from producing any legal effects, also for past events. An abrogation on the other hand operates *ex nunc*: The rule continues to apply for events that occurred before the rule was abrogated.

2.3 Legal Process Features

Legal procedures. Rules not only regulate the procedures resolving legal conflicts (see above), but also are used for arguing or reasoning about whether or not some action or state complies with other, substantive rules. In particular, rules are required for procedures which regulate methods detecting violations of the law; determine the normative effects triggered by norm violations, such as reparative obligations, namely, which are meant to repair or compensate violations. Note that these constructions can give rise to very complex rule dependencies, because the violation of a single rule can activate other (reparative) rules, which in turn, in case of their violation, refer to other rules, and so forth.

Persistence of normative effects [15]. Some normative effects persist over time unless some other and subsequent event terminates them. For example: "If one causes damage, one has to provide compensation." Other effects hold on the condition and only while the antecedent conditions of the rules hold. For example: "If one is in a public office, one is forbidden to smoke".

An interesting question is whether rule interchange languages for the legal domain should be expressive enough to fully model all the features listed above, or whether some of these requirements can be met at the reasoning level, at the level responsible for structuring, evaluating and comparing legal arguments constructed from rules and other sources.

3 LegalRuleML

To extend RuleML into LegalRuleML, we have defined two more XML-schemas: LegalMeta.xsd module and Legal_operators.xsd module (see Fig.2). Legal_meta.xsd is devoted to model all the legal metadata concerning the legal rules. Legal_operators.xsd defines the legal operators: deontic operators and behaviours. It is also necessary to have a module to connect derivation rules with reaction rules, in order to foster the potentiality of the reaction rules. This paper is a preliminary proposal for testing the rational presented in the § 1 and 2, so in the future we intend to modularize better the schemas in order to improve scalability and maintenance over the time. This proposal aims to open a debate, not to fix a solution, and make possible the mark-up of some pilot cases in order to evaluate the correctness of the solution in the RuleML community.

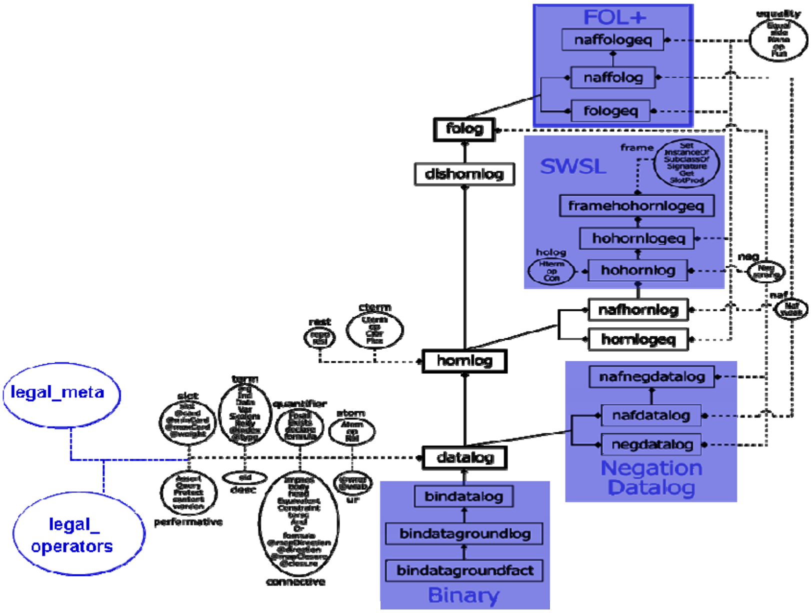

Fig. 2. Legal_metadata module included in the `datalog` component (adapted from [34])

3.1 Legal Meta Data

The root tag of Legal_metadata.xsd is `metaInfo` that includes the following optional metadata:

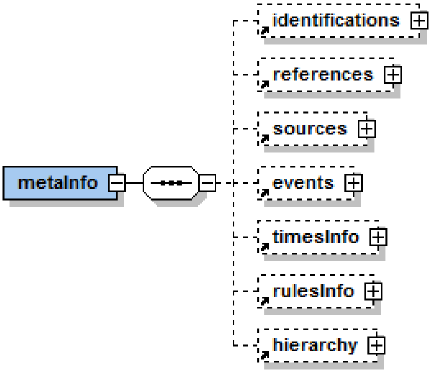

Fig. 3. MetaInfo module organization

- `identification` block provides information on the authors of the rules;
- `references` block provides identification of the textual fragments involved in the rules modeled;

- `sources` block models the connections with the textual fragments and the rules;
- `events` block provides the definition of any temporal event;
- `timesInfo` block adds semantic information to the events;
- `rulesInfo` block models the meta information concerning the rules;
- `hierarchy` block defines the ranging of the rules in the defeasibility logic.

Identification of the Annotators. This part of metadata is modeled to allow a multiple annotation of the rules coming from different authors. In the legal domain it is common to find different interpretations of the norms and equally legitimate under the legal point of view. So the identification of the authors permits to define a trust policy on the base of the context or of the authoritativeness of the annotator. If a constitutional judge annotates a rule, the trustiness is higher rather than an interpretation of a Ph.D. student of a law school. On the other side sometime the rules could be slightly different from the context and the inference engine could take in consideration a particular set of rules on the base of the role expressed by the author (e.g. regional vs. state interpretation). For this reason, we have the attribute `as` for identifying the role of the author in the annotation of the rule.

```
<identifications>
   <identification id="aut1"
uri="http://www.cirsfid.unibo.it/monica.palmirani.owl"
as="author"/>
   <identification id="aut2"
uri="http://www.nicta.com.au/guido.governatori.owl"
as="editor"/>
</identifications>
```

In this section two authors (`aut1`, `aut2`) are defined and connected with their ontology and role (`aut1` is the author of the rule, `aut2` is the editor of the rule). The same could be applied to the authorities, institutions, legal entities, juridical persons.

Sources and References for Isomorphism. The `references` and `sources` blocks are strictly connected together and they provide a solution to the isomorphism requirement. The `references` block defines the entire textual fragment involved in the rules modeling, and the `sources` block connects rules with the appropriate references. Because we have some time N:M relationship with text and rules, this mechanism permits the redundancy of the text resource URI and in the meantime connects one rule to multiple part of the text or vice versa multiple rules to the same fragment of text.

```
<references>
   <reference id="customerContract" uri="http://text1#art1"/>
   <reference id="customerContract2" uri="http://text1#art2"/>
</references>

<sources>
   <source element="#rule1" refersTo="#customerContract"/>
   <source element="#rule1" refersTo="#customerContract2"/>
   <source element="#rule2" refersTo="#customerContract2"/>
</sources>
```

This is particularly true in case of penalty-reparation rule. Usually the definition of the penalty is expressed in one clause and the conditions of reparation in another clause, but together they determine the body and the header of a unique normative rule. In the following fragment we have two citations (clauses 8 and 5) that constitute the body of the rule and the header is in the clause 10.

> *Clause 10, point 1, letter c)*
>
> *If Google does not meet the Google Apps SLA (clause 8), and if Customer meets its obligations under this Google Apps SLA (clause 5), Customer will be eligible to receive the Service Credits of X days.*

This rule is modeled in such way:

```
<references>
   <reference id="GoogleSLA" uri="http://text1#clouse8"/>
   <reference id="GoogleSLA" uri="http://text1#clouse5"/>
   <reference id="GoogleSLA" uri="http://text1#clouse10"/>
</references>

<sources>
   <source element="# rule1_body " refersTo="#GoogleSLA8"/>
   <source element="#rule1_body" refersTo="#GoogleSLA5"/>
   <source element="#rule1_header" refersTo="#GoogleSLA10"/>
</sources>
```

Events and Temporal Parameters. The `events` block detects the events related to a set of norms, in neutral way, without any semantic interpretation. The `timesInfo` block assigns the legal semantic to each group of events. In this way we could connect each atom, body, header, rule with the appropriate `timesInfo` block without any redundancy of data, preserving a compact annotation and high expressiveness. Next, we use attributes not elements in order to avoid both temporal predicates and arbitrary nomenclature to the functions.

Consider now the following clause coming from a SLA contract:

> *A customer is "Premium" if their spending has been min 5000 dollars in the previous year*

We have at least four temporal events in this provision: a) the time when the text creates rights, duties and obligations (e.g. time of *enter into force*, after the signature of the contract); b) the time when the provision is *effective* (e.g. the time when the service starts, 1 Jan 2012); c) the time when the provision is *applicable* (e.g. after at least one year from the efficacy time); d) the temporal conditions included in the provision that is a dynamic dimension (e.g. "previous year"). A new question arise concerning the continuity of the temporal condition: i) the customer have to spend at least one order min 5000 dollars (only one event is sufficient); ii) the customer could aggregate several spending for min 5000 dollars (set of events create the condition); iii) the customer have to maintain their orders min 500 dollars (continuity of condition). For this reason we have introduced an attribute (`perdurant`) with several parameters: and (true for all t_i of an interval), or (true if at least one t_i satisfies the condition), xor (true if only one t_i satisfies the condition) , agg (true if the aggregation of a set of t_i satisfies the condition).

We can model those events as follow:

```
<events>
   <event id="e1" value="2011-08-25T01:01:00.0Z"/>
   <event id="e2" value="2012-01-25T01:01:00.0Z"/>
   <event id="e3" value="2013-01-25T01:01:00.0Z"/>
</events>

<timesInfo>
   <times id="t1">
    <time start="#e1" timeType="efficacy"/>
    <time start="#e1" timeType="inforce"/>
    <time start="#e3" timeType="application"/>
   </times>
   <times id="t2">
    <time duration="-P01Y" timeType="internal" timeType
="application" perdurant="agg"/>
   </times>
</timesInfo>
```

Note the time of application "previous year" is modeled as internal event of the norm and represented using the negative period of duration (-P1Y, following the standard syntax of xsd).

The mechanism presented for modeling the temporal parameters connects times to norms and rules and it fosters effective legal reasoning algorithm about facts occurred in the past, or that happen in the future, with uncertain events and with complex conditionals.

Hierarchy and Type of Norms. The non-monotonic legal reasoning needs to manage the hierarchy of rules [see § 2]. The hierarchy block defines the superiority relationship between two rules: it is a binary operator that creates a meta-rule among existing rules.

Because the superiority relationship depends to some conditions we have several attributes that anchor the association to specific parameters: author and time. It is so possible to have the same rule with different superiority relationship, made in a different time, by a different author.

```
<hierarchy>
   <range id="rng1" function="superior" from="#rule1"
to="#rule2" timesBlock="#t1" author="#aut2"/>
</hierarchy>
```

Semantic Qualification of Rules. In the `rulesInfo` block, we define some properties of the rule like the `ruleType` (e.g. defeasible, defeater, strict, metaRule), the `author` and qualification using the attribute `refersTo`. Fostering the `referesTo` attributes we could connect any external legal concept defined with a given ontology.

```
<rulesInfo>
   <ruleInfo source="#rule1" ruleType="defeasible" refersTo
="/ontology/usaJurisdiction.owl" author="#aut2"/>
```

```
   <ruleInfo source="#rule1" ruleType="strict" refers-
To="/ontology/definition.owl" author="#aut2"/>
</rulesInfo>
```

Let us come back to our example:

> *A customer is "Premium" if their spending has been min 5000 dollars in the previous year.*

The above is modeled as follow in enriched way, ready for legal reasoning base don defeasible logic.

```
<Assert mapClosure="universal">
   <Implies timesBlock="#t2" ruleType="defeasible" id="rule1">
     <then timesBlock="#t1">
       <Atom id="atm1">
         <Rel>premium</Rel>
         <Var>customer</Var>
       </Atom>
     </then>
     <if timesBlock="#t1">
        <Atom id="atm2" timesBlock="#t3">
           <Rel>previous year spending</Rel>
           <Var>customer</Var>
           <Var>x</Var>
           <Data>= 5000$ </Data>
        </Atom>
     </if>
   </Implies>
</Assert>
```

3.2 Legal Operators

In the module Legal_operators.xsd we have defined all the operators needed for managing deontic logic and behaviors like violation and reparation.

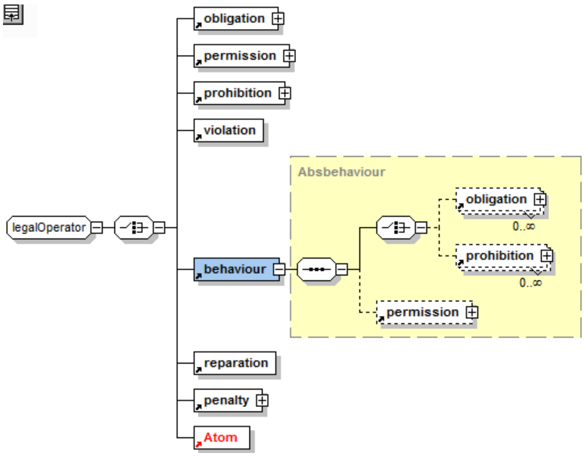

Fig. 4. Legal_operators.xsd elements

Behavior represents a particular sequence of deontic operators that starts with an obligation or a prohibition and ends with a permission.

The `violation` is a unary relationship that refers to the obligation/prohibition subject of the violation. The `reparation` is a unary relationship providing a link to the relevant penalty.

For a better understanding of their usage, we describe an example coming from the US Code related to the infringement of the copyright, Title 18, Chapter 6:

> *§ 602 (b) In a case where the making of the copies or phonorecords would have constituted an infringement of copyright if this title had been applicable, their importation is prohibited.*

To model this example, we first start with the rule 602b where we find in the conclusion a prohibition to import material that infringes the copyright law:

```
<Implies id="rule602b">
   <then>
      <prohibition>
         <Atom id="rule602b-prh1-atm1">
            <Rel>importation is prohibited</Rel>
            <Var>z</Var>
         </Atom>
      </prohibition>
   </then>
   <if>
      <And>
         <Atom id="rule602-if-atm1">
            <Rel>copies or phonorecords</Rel>
            <Var>z</Var>
         </Atom>
         <Atom id="impl602-1-if-atm2">
            <Rel>without the authority of the owner of copyright
</Rel>
            <Var>x</Var>
         </Atom>
      </And>
   </if>
</Implies>
```

After that, we assume as a fact the penalty statement in case of a copyright infringement following the 504 (c)(1):

> *§ 504. Remedies for infringement: Damages and profits*
> *(c) Statutory Damages.—*
> *(1) Except as provided by clause (2) of this subsection, the copyright owner may elect, at any time before final judgment is rendered, to recover, instead of actual damages and profits, an award of statutory damages for all infringements involved in the action, with respect to any one work, for which any one infringer is liable individually, or for which any two or more infringers are liable jointly and severally, in a sum of not less than $750 or more than $30,000 as the court considers just. For the purposes of this subsection, all the parts of a compilation or derivative work constitute one work.*

```
<Atom id="atm504">
        <penalty id="atm504-pnl1">
           <obligation id="obl2" subject="z" beneficiary="y"
timesBlock="#t2">
              <Atom id="atm504-pnl1-atm1">
                 <Rel>award of statutory damages to</Rel>
                 <Var>z</Var>
                 <Data>min $750 </Data>
                 <Data>max $30,000  </Data>
              </Atom>
           </obligation>
        </penalty>
  </Atom>
```

Finally we define a new rule that connects the `reparation` with the violation of the rule602b, and the reparation with the penalty (see the `penalty="#atm504-pnl1"` attribute). We have reparation only if the subject violated the rule602 and has paid the award of statutory damages to the copyright owner.

```
<Implies id="rule602b-rep">
   <then>
      <reparation id="rule602b-rep1" penalty="#atm504-pnl1"/>
   </then>
   <if>
      <violation source="#rule602b"/>
   </if>
  </Implies>
```

3.3 Semantic Qualification of Negation

One of the main problems in legal reasoning is to qualify the negation. To solve this problem, we have customized the module neg_module.xsd and naf_module.xsd in order to include a link to the semantic meaning. The attribute `refersTo` permits to link the markup to specific concept ontology.

```
  <xs:attributeGroup name="Neg.attlist">
   <xs:attributeGroup ref="refersTo"/>
  </xs:attributeGroup>

  <xs:attributeGroup name="Naf.attlist">
   <xs:attributeGroup ref="refersTo"/>
  </xs:attributeGroup>
```

3.3 Extension of the RuleML Modules

To support the application of that metadata, we have extended several modules, like atom_module.xsd that could host the time parameters and the id attribute:

```
<xs:attributeGroup name="Atom.attlist">
      <xs:attributeGroup ref="closure.attrib"/>
      <xs:attributeGroup ref="timesBlock"/>
      <xs:attributeGroup ref="idReq"/>
</xs:attributeGroup>
```

The `connective_module.xsd` is extended in order to define, apart from the time parameters and the id, the type of rule, following the defeasible classification: strict, defeasible, defeater, metaRule.

```
<xs:attributeGroup name="Implies.attlist">
      <xs:attributeGroup ref="closure.attrib"/>
      <xs:attributeGroup ref="direction.attrib"/>
      <xs:attributeGroup ref="material.attrib"/>
      <xs:attributeGroup ref="timesBlock"/>
      <xs:attributeGroup ref="idReq"/>
      <xs:attributeGroup ref="ruleTypeDef"/>
</xs:attributeGroup>
```

In legal_metadata.xsd we define the list of values:

```
<xs:simpleType name="ruleTypeValue">
      <xs:restriction base="xs:token">
         <xs:enumeration value="strict"/>
         <xs:enumeration value="defeasible"/>
         <xs:enumeration value="defeater"/>
         <xs:enumeration value="metaRule"/>
      </xs:restriction>
  </xs:simpleType>
```

4 Conclusion

This paper presents an extension of RuleML customized to support legal requirements. The goal is to have a clear and expressive XML language integrated as part of the RuleML family capable to support the modeling and the representation of legal norms and rules. In this first step we have implemented the following features:

Table 1. List of LegalRuleML features and the extended modules

Features	**LegalRuleML**	**RuleML extension**
Isomorphism	sources and references	legal_meta.xsd
Jurisdiction	refersTo	legal_meta.xsd
Authority	author	legal_meta.xsd
Temporal parameters	event and timesInfo	legal_meta.xsd connective_module.xsd atom_module.xsd
Qualification/Definitional/ Valuable	refersTo	legal_meta.xsd
Semantic of Negation	refersTo	Neg_module.xsd Naf_module.xsd
Deontic operators	legalOperator	legal_operators.xsd connective_module.xsd atom_module.xsd
Defeasible logic	hierarchy and typeRules	legal_meta.xsd
Behaviors	legalOperator	legal_operators.xsd connective_module.xsd atom_module.xsd

LegalRuleML language aims to interoperate with Reaction RuleML modules. Our next steps include a better modularization of the main features from a syntactical point of view, extend the modularization to all the modules of the Declarative Rules and Reactive Rules, and develop a proof of concept implementing a sample set of acts and contracts. The LegalRuleML Initiative has been working with OASIS, especially the LegalXML Technical Committee, on organizing future efforts.

References

1. Bench-Capon, T., Coenen, F.: Isomorphism and legal knowledge based systems. Artificial Intelligence and Law 1(1), 65–86 (1992)
2. Bench-Capon, T.: The missing link revisited: The role of teleology in representing legal argument. Artificial Intelligence and Law 10(1-3), 79–94 (2002)
3. Berners-Lee, T.: Long Live the Web: A Call for Continued Open Standards and Neutrality. Scientific America (2010)
4. Boer, A., Radboud, W., Vitali, F.: MetaLex XML and the Legal Knowledge Interchange Format. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) Computable Models of the Law. LNCS (LNAI), vol. 4884, pp. 21–41. Springer, Heidelberg (2008)
5. Boley, H., Paschke, A., Shafiq, O.: RuleML 1.0: The Overarching Specification of Web Rules. In: Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.) RuleML 2010. LNCS, vol. 6403, pp. 162–178. Springer, Heidelberg (2010)
6. Boley, H., Tabet, S., Wagner, G.: Design rationale for RuleML: A markup language for Semantic Web rules. In: Cruz, I.F., Decker, S., Euzenat, J., McGuinness, D.L. (eds.) Proc. SWWS 2001, The First Semantic Web Working Symposium, pp. 381–401 (2001)
7. Breuker, J., Boer, A., Hoekstra, R., Van Den Berg, C.: Developing Content for LKIF: Ontologies and Framework for Legal Reasoning. In: Legal Knowledge and Information Systems, JURIX 2006, pp. 41–50. ISO Press, Amsterdam (2006)
8. De Maat, E., Winkels, R.: Automated Classification of Norms, in Sources of Law. In: Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D. (eds.) Semantic Processing of Legal Texts. LNCS, vol. 6036, pp. 170–191. Springer, Heidelberg (2010)
9. Francesconi, E., Passerini, A.: Automatic Classification of Provisions in Legislative Texts. In: ICAIL 2007, vol. 15, pp. 1–17. ACM (2007)
10. Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D.: Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language. Springer, Heidelberg (2010)
11. Gordon, T.F., Governatori, G., Rotolo, A.: Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 282–296. Springer, Heidelberg (2009)
12. Gordon, T.F.: Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF). In: Computable Models of the Law, Languages, Dialogues, Games, Ontologies 2008, pp. 162–184. Springer, Heidelberg (2008)
13. Gordon, T.F.: The Pleadings Game; An Artificial Intelligence Model of Procedural Justice. Springer, New York (1995); Book version of 1993 Ph.D. Thesis; University of Darmstadt (1993)
14. Governatori, G., Pham, D.H.: Dr-contract: An architecture for e-contracts in defeasible logic. International Journal of Business Process Integration and Management 5(4) (2009)
15. Governatori, G., Rotolo, A.: Changing legal systems: Legal abrogations and annulments in defeasible logic. The Logic Journal of IGPL (2010)
16. Governatori, G., Milosevic, Z., Sadiq, S.: Compliance checking between business processes and business contracts. In: Proc. EDOC 2006, pp. 221–232. IEEE (2006)

17. Governatori, G., Rotolo, A., Sartor, G.: Temporalised normative positions in defeasible logic. In: Proc. ICAIL 2005, pp. 25–34. ACM Press (2005)
18. Governatori, G., Rotolo, A.: Norm Compliance in Business Process Modeling. In: Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.) RuleML 2010. LNCS, vol. 6403, pp. 194–209. Springer, Heidelberg (2010)
19. Governatori, G.: Representing business contracts in RuleML. International Journal of Cooperative Information Systems 14(2-3), 181–216 (2005)
20. Grosof, B.: Representing e-commerce rules via situated courteous logic programs in RuleML. Electronic Commerce Research and Applications 3(1), 2–20 (2004)
21. Hage, J.C.: Reasoning with Rules – An Essay on Legal Reasoning and its Underlying Logic. Kluwer Academic Publishers, Dordrecht (1997)
22. Hu, Y.-J., Boley, H.: SemPIF: A Semantic Meta-policy Interchange Format for Multiple Web Policies. In: Web Intelligence 2010, pp. 302–307 (2010)
23. Lupo, C., Vitali, F., Francesconi, E., Palmirani, M., Winkels, R., de Maat, E., Boer, A., Mascellani, P.: General xml format(s) for legal sources - Estrella European Project IST-2004-027655. Deliverable 3.1, Faculty of Law. University of Amsterdam, Amsterdam, The Netherlands (2007)
24. Mazzei, A., Radicioni, D.P., Brighi, R.: NLP-Based Extraction of Modificatory Provisions Semantics. In: ICAIL 2009, pp. 50–57. ACM Press, New York (2009)
25. Palmirani, M., Brighi, R.: Model Regularity of Legal Language in Active Modifications. In: Casanovas, P., Pagallo, U., Sartor, G., Ajani, G. (eds.) AICOL-II/JURIX 2009. LNCS, vol. 6237, pp. 54–73. Springer, Heidelberg (2010)
26. Palmirani, M., Contissa, G., Rubino, R.: Fill the Gap in the Legal Knowledge Modelling. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 305–314. Springer, Heidelberg (2009)
27. Palmirani, M., Governatori, G., Contissa, G.: Modelling temporal legal rules. In: ICAIL 2011 Proceedings. ACM (2011)
28. Palmirani, M.: Time Model in Normative Information Systems. In: Proceedings of the Workshop The Role of Legal Knowledge in eGovernment, ICAIL 2005 (2005)
29. Prakken, H., Sartor, G.: A dialectical model of assessing conflicting argument in legalreasoning. Artificial Intelligence and Law 4(3-4), 331–368 (1996)
30. Proceeding of the 13th International Conference on Artificial Intelligence and Law, Pittsburgh, June 6-10. ACM, NY (2011)
31. Rotolo, A., Sartor, G., Smith, C.: Good faith in contract negotiation and performance. International Journal of Business Process Integration and Management 5(4) (2009)
32. Rotolo, A.: Rule-based agents, compliance, and intention reconsideration in defeasible logic. In: Pasche, A. (ed.) RuleML 2011 - Europe. LNCS, vol. 6826, pp. 67–82. Springer, Heidelberg (2011)
33. Rubino, R., Rotolo, A., Sartor, G.: An OWL Ontology of Fundamental Legal Concepts. In: JURIX 2006, pp. 101–110. IOS (2006)
34. RuleML. The Rule Markup Initiative, http://www.ruleml.org (August 20, 2009)
35. Sartor, G., Palmirani, M., Francesconi, E., Biasiotti, M.: Legislative XML on Semantic web. Springer, Heidelberg (2011)
36. Sartor, G.: Legal concepts as inferential nodes and ontological categories. Artif. Intell. Law 17(3), 217–251 (2009)
37. Sartor, G.: Legal Reasoning: A Cognitive Approach to the Law. In: Pattaro, E., Rottleuthner, H., Shiner, R., Peczenik, A., Sartor, G. (eds.) A Treatise of Legal Philosophy and General Jurisprudence, vol. 5. Springer, Heidelberg (2005)
38. Wagner, G., Antoniou, G., Tabet, S., Boley, H.: The abstract syntax of RuleML – towards a general web rule language framework. In: Proc. Web Intelligence 2004, pp. 628–631. IEEE Computer Society (2004)