LegalRuleML: From Metamodel to Use Cases
(A Tutorial)

Tara Athan[1], Harold Boley[2], Guido Governatori[3],
Monica Palmirani[4], Adrian Paschke[5], and Adam Wyner[6]

[1] Athan Services, USA
taraathan@gmail.com
[2] Faculty of Computer Science
University of New Brunswick, Canada
harold.boley@ruleml.org
[3] NICTA, Australia
guido.governatori@nicta.com.au
[4] CIRSFID
University of Bologna, Italy
monica.palmirani@unibo.it
[5] Corporate Semantic Web
Freie Universitaet Berlin, Germany
paschke@inf.fu-berlin.de
[6] Department of Computing Science
University of Aberdeen, UK
adam@wyner.info

1 Motivation and Background

Several XML-based standards have been proposed for describing rules (RuleML, RIF, SWRL, SBVR, etc.), or specific dialects (RuleML family [1,2]). In 2009, the Legal Knowledge Interchange Format (LKIF [4]) was proposed to extend rule languages to account for the specifics of the legal domain and to manage legal resources. To further develop the representation of the law in XML-based standards, the OASIS LegalRuleML TC held its first technical meeting on 19 January 2012 [9]. The objective of the TC is to extend the RuleML family with features specific to the formalisation of norms, guidelines, policies, and legal reasoning [3].

The work of the LegalRuleML Technical Committee has been focusing on four specific needs:

1. To close the gap between natural language text description and semantic norm modelling, in order to realise an integrated and self-contained representation of legal resources that can be made available on the Web as XML representations [8].
2. To integrate technologies such as NLP and Information Extraction (IE) with Semantic Web technologies such as graph representation and web-based ontologies and rules.
3. To provide an expressive XML standard for modelling normative rules that is able to satisfy the requirements of the legal domain. This enables use of a legal reasoning level on top of the ontological layer in the W3C Semantic Web stack. This approach

seeks also to fill the gap between regulative norms, guidelines, and business rules in order to capture and model the processes in them and to make them usable for the workflow and business layer [6,7].

4. To support the Linked Open Data approach to modelling with respect to the semantics of raw legal data (acts, contracts, court files, judgements, etc.) and also the rules together with their functionality and usage. Without rules, legal concepts constitute just a taxonomy [10].

The LegalRuleML TC work has been addressing these four main goals and has provided means to semantically model norms, guidelines, judgements, and contracts. The outcome is intended to define a standard (expressed with XML Schema Definition Language (XSD) and Relax NG) that is able to represent the peculiarities of the legal normative rules easily and meaningfully.

2 Methodology of the Tutorial

This tutorial presents the principles of the LegalRuleML applied to the legal domain and discuss why, how, and when LegalRuleML is well-suited for modelling norms. To provide a framework of reference, we present a comprehensive list of requirements for devising rule interchange languages that capture the peculiarities of legal rule modelling in support of legal reasoning. The tutorial comprises syntactic, semantic, and pragmatic foundations, a LegalRuleML primer, a comparison with related other approaches, as well as use case examples from the legal domain.

3 LegalRuleML Model

A key tenet of LegalRuleML is that the concepts and features of the language should provide a conceptually faithful representation of legal textual provisions and the norms they encode. To this end the language captures the following functionalities and peculiarities of the legal domain.

– qualification of norms: legal documents can contains different types of norms (constitutive, technical, prescriptive, etc.). Some norms are intended to define the terms used in the document, others to produce normative effects, and others to describe legal procedures.
– defeasibility of rules; norms are often written in a way that they admit exceptions. Defeasiblity allows for a natural representation of exceptions and permits terms to be defined in an open textured fashion.
– deontic operators: the function of prescriptive rules is to describe the normative effects that they produce (e.g., obligations, permissions, prohibitions, . . .), the parties related to them, and the conditions under which such effects are produced.
– temporal management of the rules and temporal expressions within the rules: norms are affected over the time in their validity and efficacy. LegalRuleML is able to define temporal instants and intervals that can be used to build complex legal events and situations (e.g. date of publication, interval of suspension, interval of efficacy

but not applicability). These temporal parameters are called external temporal characteristics of the norm, and thus permit the representation of the temporal information of the rules. The temporal characteristics can be associated with different rules or with a part of the rule.
- jurisdiction of norms: norms emanated from different authorities, different locations, and different times. Relative to such differences, norms can produce different effects. To properly model such contextual dependence, LegalRuleML associates rules with the jurisdictions where the rules hold.
- isomorphism between rules and natural language normative provisions: norms have a lifecyle - they are created, enter into force, can be modified, and can be repealed. Where the language of the provisions changes, so too must the corresponding formal expression in LegalRuleML (isomorphism). LegalRuleML provides for this isomorphism by maintaining a link between the units of natural language textual provisions and the sets of rules.
- authorial tracking of the rules: rules constitute an interpretation of the textual provisions and so of the norms. Accordingly, it is important to trace who is author of the interpretation to establish a level of trust in a ruleset or to identify the context in which a ruleset is suitable to be used.

4 Metamodel of the Rule Properties

The LegalRuleML syntax is modelled using a metamodel founded on RDF triples. This permits us to serialize LegalRuleML XML into RDF assertions for Semantic Web interoperability and Linked Open Data integration. The tutorial presents this aspect of the design. Rules have properties expressed in separate blocks in a generic way. We have provided a mechanism for defining an identifier of type xsd:ID for property values, which is used as the fragment identifier in an IRI that may be efficiently referenced through a relative IRI or CURIE. This ”internal” IRI may act as an alias for exernal IRIs, entities identified in external non-IRI identifier systems, and entities having no external identifiers:

• `<References>` and `<LegalSources>` for referencing the textual provisions that are modelled by the rules:

```
<lrml:LegalSource key="ref9"
    sameAs="http://www.law.cornell.edu/wiki/lexcraft/
    section_identifiers_lii"
/>
```

• `<TimeInstants>` and `<TemporalCharateristics>` for capturing the external temporal dimensions of the rules are represented. For example, here we show the period for entering into force and the period of efficacy:

```
<lrml:TimeInstants>
    <ruleml:Time key="t1">
      <ruleml:Data xsi:type="xs:dateTime">
```

```
        2012-07-21T00:00:00Z
      </ruleml:Data>
    </ruleml:Time>
</lrml:TimeInstants>
<lrml:TemporalCharacteristics key="tblock1">
    <lrml:TemporalCharacteristic key="nev1">
      <lrml:forRuleStatus iri="lrmlv:Efficacious"/>
      <lrml:hasStatusDevelopment iri="lrmlv:Starts"/>
      <lrml:atTimeInstant keyref="#t1"/>
    </lrml:TemporalCharacteristic>
</lrml:TemporalCharacteristics>
```

• <Agent> and <Authority> are two classes for defining the agent and the authority of the rules in order to represent the provenance of the rules:

```
  <lrml:Agents>
    <lrml:Agent key="aut1"
      sameAs="unibo:person.owl#m.palmirani"/>
  </lrml:Agents>
  <lrml:Authorities>
    <lrml:Authority key="congress"
        sameAs="unibo:organization.owl#congress">
      <lrml:type iri="lrmlv:Legislature"/>
    </lrml:Authority>
  </lrml:Authorities>
```

• The <Context> block associates property values to rules (in the example to rule1) and also adds other important metadata such as jurisdiction, role, and strength (defeasible, defeater, strict):

```
<lrml:Context key="ruleInfo1" hasCreationDate="#t8">
    <lrml:appliesTemporalCharacteristics keyref="#tblock1"/>
    <lrml:appliesStrength iri="lrmlv:Defeasible"/>
    <lrml:appliesRole>
        <lrml:Role iri="lrmlv:Author">
          <lrml:filledBy keyref="#aut1"/>
        </lrml:Role>
    </lrml:appliesRole>
    <lrml:appliesAuthority keyref="#congress"/>
    <lrml:appliesJurisdiction keyref="jurisdictions:us"/>
    <lrml:toStatement keyref="#rule1"/>
</lrml:Context>
```

This mechanism is flexible; it permits us to represent relationships with arity higher than two (e.g. multiple authors, multiple textual sources), which guarantees multiple interpretations over time of the same rule without redundancy. However, it is always possible to transform them into an RDF triples serialization.

5 LegalRuleML Skeleton

LegalRuleML may be composed using the following main blocks skeleton:

– declaration of the internal identifiers for property values in the top of the XML.
– association of the property values to the rules using `<lrml:Association>`.
– rules, both constitutive and prescriptive, are modelled in one `<lrml:Statements>` block.
– facts are modelled inside of a second `<lrml:Statements>` block.

```
<lrml:LegalRuleML>
  <lrml:References>
    <lrml:Reference/>
  </lrml:References>

  <lrml:Context key="ruleInfo1-v2">
     <lrml:Association>
       <lrml:appliesSource
             keyref="#sec2.1-list1-itm31-par1-v2"/>
       <lrml:toTarget keyref="#rulebase-v2"/>
       </lrml:Association>
   </lrml:Context>

   <lrml:Statements key="rulebase-v2">
     <lrml:ConstitutiveStatement key="rule1a-v2">
       <ruleml:if> ...</ruleml:if>
       <ruleml:then>... </ruleml:then>
      </lrml:ConstitutiveStatement>
   </lrml:Statements>

   <lrml:Statements key="facts-v1">
     <lrml:FactualStatement key="fact1">
       <ruleml:Atom key=":atom11">
         <ruleml:Rel iri="#rel5"/>
         <ruleml:Ind iri="#JohnDoe"/>
       </ruleml:Atom>
     </lrml:FactualStatement>
   </lrml:Statements>
</lrml:LegalRuleML>
```

6 Tutorial Use Cases

The use of LegalRuleML is illustrated with some concrete application scenarios:

– in the eHealth domain, LegalRuleML can be used to model privacy issues and security policies for managing document access according to the profile and the

authorizations of the operator. By using LegalRuleML, it is possible to filter sensitive data, according to the law/regulation, and to create particular views of the same health record or document based on the role of the querying agent.
- in the open data domain, LegalRuleML could model the creative commons licences of datasets to permit an automatic IPR compatibility check among different datasets, in particular to evaluate if different datasets could be combined for producing a commercial application.
- in patent law, LegalRuleML can model the judgments and the regulations in order to support the industrial decisions.

References

1. 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)
2. 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)
3. 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)
4. Gordon, T.F.: Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF). In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) Computable Models of the Law. LNCS (LNAI), vol. 4884, pp. 162–184. Springer, Heidelberg (2008)
5. Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: OASIS LegalRuleML. In: Verheij, B. (ed.) Proceedings of 14th International Conference on Artificial Intelligence and Law (ICAIL 2013). ACM (2013)
6. Governatori, G., Rotolo, A.: Changing legal systems: Legal abrogations and annulments in defeasible logic. The Logic Journal of IGPL (2010)
7. Grosof, B.: Representing e-commerce rules via situated courteous logic programs in RuleML. Electronic Commerce Research and Applications 3(1), 2–20 (2004)
8. 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)
9. Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley, H., Paschke, A.: LegalRuleML: XML-Based Rules and Norms. In: Olken, F., Palmirani, M., Sottara, D. (eds.) RuleML - America 2011. LNCS, vol. 7018, pp. 298–312. Springer, Heidelberg (2011)
10. 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 (2005)