LegalRuleML Manifesto

Version 1.8, 22 September 2011
Draft version

1. Proponents

Monica Palmirani, CIRSFID, University of Bologna, Italy
Antonino Rotolo, CIRSFID, University of Bologna, Italy, and RuleML director
Guido Governatori, NICTA, Queensland Laboratory, Australia, and RuleML director
Tom Gordon, Fraunhofer Institute for Open Communications Systems, Germany
Harold Boley, Faculty of Computer Science, University of New Brunswick, Institute for Information Technology, National Research Council, Canada, and RuleML co-chair
Said Tablet, EMC Corporation, USA, and RuleML co-chair
Adrian Paschke, Institute for Computer Science, University of Berlin, Germany, and RuleML director
John Hall, Model Systems, UK, and RuleML director
Nick Bassiliades, Aristotle University of Thessaloniki
Jong Hu, National Chengchi University

2. Introduction

The AI & Law community dedicated a good part of the last twenty years to model legal norms using different logics and formalisms [ICAIL 2009]. 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 legal texts [Lupo et. al. 2007] with XML based rules (RuleML, SWRL, RIF, etc.) [Gordon et. al. 2009]. In the meantime, the Semantic Web, in particular Legal Ontology research combined with semantic norm extraction based on Natural Language Processing (NLP) [Francesconi et. al. 2010], gave a great impulse to the modeling of legal concepts [Boer et. al. 2007]. In this scenario, there are three urgent needs:

1. to close the gap between text description, using XML techniques, and norms modeling, in order to realise an integrated and self-contained representation of legal resources available on the Web [Palmirani 2009]. This integration is fundamental for fostering Semantic Web advantages like: NLP, IR, graph representation, Web ontologies and rules, etc.;

2. to have an expressive XML standard for modeling normative rules, able to 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]. This approach seeks also to fill the gap between regulative norms and business rules[2] in order to capture and model the processes embedded in those norms and make them usable for the workflow and business layer [Governatori 2010];
3. finally, particular attention is paid to the Linked Open Data [Berners-Lee 2010] 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 constitute simply a taxonomy [Sartor 2009].

3. LegalRuleML Objectives

3.1 First Goal: Respond to the Need

The aim is to define a general legal document architecture able to describe all of the six layers in Fig. 1 with a uniform standard syntax for integrating, in the clearest conceivable way, all the different layers reducing as much as possible the room for confusion or ambiguity:

- **text**: part of the document officially approved by the authority with legal power;
- **structure:** of the text: part of the document that constitutes the organization of the text;
- **metadata**: any information that was not included by the authority in the deliberative act;
- **ontology**: any conceptualization of the reality in which the document has a role (e.g., for a judgement, the juridical system concepts) or any concept called from the text that needs modeling;
- **legal knowledge representation**: the part of the interpretation and modeling of the meaning of the text under a legal perspective. Several XML languages have been proposed as standards for managing rules (LKIF, RuleML, SWRL, RIF, etc.), with RuleML being a flexible state-of-the-art language able to describe different possible theories or logic models (propositional, predicative, argumentative, non-monotonic, deontic, defeasible, etc.) well-fitted for the legal domain.
- **business and process rule modeling**: An additional layer supported by the interpretation of legal knowledge and providing business and process rule modeling required for policy creation and management activities as a reaction to regulatory interpretation.

3.2 Second Goal: Legal Rule Modeling

In general, many of the drawbacks affecting several existing languages are perhaps due to the fact that there has not yet been an overall and systematic effort to establish a general list of requirements for rule interchange languages in the legal domain or because there is not yet a corresponding widely shared agreement amongst the practitioners working in this field.

The law constitutes a complex area, which can be analyzed into different branches according to the authority that produces legal norms and according to the circumstances and procedures under which norms are created. But, independently of these aspects, it is possible to identify some general features that norms should enjoy.

First of all, it is widely acknowledged in legal theory and AI & Law that norms have basically a conditional structure [Kelsen 1991, Sartor 2005] in the form of:

$$\text{if } A_1 \ldots\ldots A_n \text{ then } B$$

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

where $A_1... A_n$ are the conditions of the norm and B denotes the legal effect which must be pursued when those conditions are true.

This very general view highlights an immediate link between the concepts of norm and rule. However, there are many types of normative rules. For example, von Wright [VonWright:1963a] classified norms into the following main types (among others):

DETERMINATIVE RULES, which define concepts or constitute activities that cannot exist without such rules. These rules in the literature are also called constitutive rules.

TECHNICAL RULES, which state that something has to be done in order for something else to be attained.

PRESCRIPTION RULES, which regulate actions by making them obligatory, permitted, or prohibited. These norms, to be complete, should indicate:

- who (the norm-subjects)
- does what (the action-theme)
- in what circumstances (the condition of application)
- and the nature of their management (the mode).

Many of these aspects have been acknowledged in the field of artificial intelligence and law, where there is now much agreement about the structure and properties of rules [Gordon:1995, Prakken:1996, Hage:1997, Verheij:1996, Sartor:2005]. Important requirements for legal rule languages from the field of AI & Law include the following, divided in three main categories.

SEMANTIC MODELING

ISOMORPHISM [Bench-Capon-Coenen:1992] 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 [Gordon:1995] 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 [Prakken:1996] 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 [Governatori-Rotolo:2010, Palmirani:2009, Palmirani:2010] Rules usually are qualified by temporal properties, such as: the time when the norm is in force 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 correctly computing the 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 [Sartor:2005b]). Below is a rather comprehensive list of normative effects [RubinoEtAl:2006]:

Evaluative, which indicate that something is good or bad, is a value to be optimised or an evil to be minimised. For example, "Human dignity is valuable", "Participation ought to be promoted";

Qualificatory, which ascribe a legal quality to a person or an object. For example, "x is a citizen";

Definitional, which specify 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, which, typically, impose the obligation or confer the permission to do a certain action. For example, "x has the obligation to do A";

Potestative, which attribute powers. For example, "A worker has the power to terminate his work contract";

Evidentiary, which establish the conclusion to be drawn from certain evidence. For example, "It is presumed that dismissal was discriminatory";

Existential, which indicate the beginning or the termination of the existence of a legal entity. For example, "The company ceases to exist";

Norm-concerning effects, which state modifications of norms such as abrogation, repeal, substitution, and so on.

VALUES [Bench-Capon:2002] 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.)

LOGIC MODELING

DEFEASIBILITY [Gordon:1995, Prakken:1996, Sartor:2005b] 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 [Prakken:1996] Rules can conflict, namely, they may lead to incompatible legal effects. Conceptually, conflicts can be of different types, according to 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. Accordingly, rule conflicts can be resolved using principles about rule priorities, such as:

- *lex specialis*, which gives priority to the more specific rules (the exceptions);
- *lex superior*, which gives priority to the rule from the higher authority (see 'Authority' above);
- *lex posterior*, which gives priority to the rule enacted later (see 'Temporal parameters' above).

Exclusionary rules [Prakken:1996,Sartor:2005b,Gordon:1995] Some rules provide one way to explicitly undercut other rules, namely, to make them inapplicable.

CONTRAPOSITION [Prakken:1996] 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 [Sartor:2005b] 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 [Governatori-Rotolo:2010] 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 which occurred before the rule was abrogated.

PROCESS MODELING

LEGAL PROCEDURES. Rules not only regulate the procedures for 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 for 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 we can have that 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 [Governatori :2005a] 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.

3.3 Third Goal: Open Data Rule Modeling

With the (Linked) Open Data approach, Tim Berners-Lee defined a new paradigm for sharing and providing raw data without any previous manipulation by the editor or author, in order to stimulate reuse and new applications by private and public sectors (e.g., the Haiti rescue application after the earthquake coming from the geo information provided by an Open Data portal).

The OpenGov initiative lunched by President Obama with a directive in Jan 2009 is based on the assertion to create “an unprecedented level of openness in government” for ensuring “the public trust and establish a system of transparency, public participation, and collaboration. Openness will strengthen our democracy and promote efficiency and effectiveness in government.” This initiative provided a strong momentum to the Open Data strategy and the web site data.gov is an evident success of this opening move. Also the UK government started contextually the same approach with the data.gov.uk web site for going beyond the classic interchange concept and for fully and truly implementing the interoperability and reuse approach. Tim Berners-Lee in the recent presentation on TED2010[3] stressed the importance of the Linked Data approach for stimulating new applications never seen before and overcoming data silos on the web.

As Google's chief economist Hal Varian has said, the scarce factor is no longer the data, which is essentially free and ubiquitous, but now the "scarce factor is the ability to understand that data and extract value from it." Therefore the new challenge is not providing the raw data per se, but also the intensional level description able to provide the semantic level of the data.

[3] http://blog.ted.com/2010/03/08/the_year_open_d/

So, the Open Data initiative comprises more than providing e-services online and openness of information through some portals, or cooperating in a wiki or blog. It means to provide raw public data for permitting new kinds of usage:

1. elaboration by third parties (citizens and companies)
2. foster the tremendous potentials of those data (e.g., publishers can reuse this information for annotating their editorial commercial products)
3. monitoring the administrative expenditure of money;
4. evaluating the outcomes of the public administration;
5. measuring if the public administration is producing benefits for the society
6. creating new market sectors (e.g., analysis and reuse of public administration data).

For achieving these results, we have to ask for technology not only to open the raw data sets but also to provide semantic tables, vocabularies, glossaries and rules for permitting the correct interpretation of the information. Without the intentional level, a table is simply an arrangement of numbers without any meaning, and apart from the author of the table, no one can reuse this resource in a correct way [Genesereth 2010]. For this reason, a new paradigm is necessary, namely to put on top of the raw data the descriptiveness of schemas, a metadata vocabulary, integrity and behavior rule definitions. Such a new layer is necessary to be able to model the relationships between the extensional (value) and the intentional (semantic) level of the data and the description of the behavior (function) expected by the data. Secondly, data provenance is important to describe how the data were collected, elaborated, aggregated for permitting a correct reuse in the long-term.

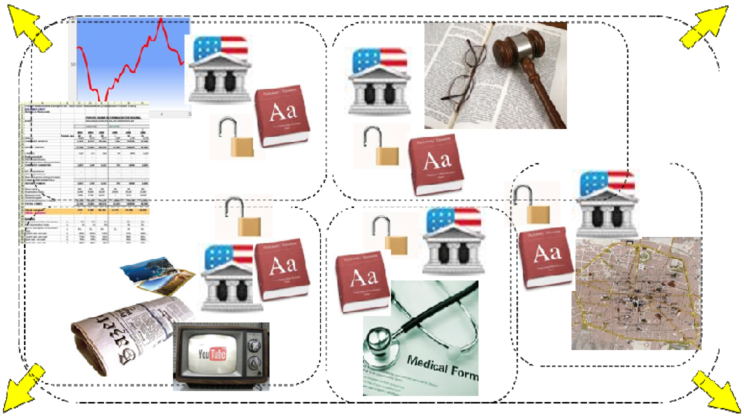

Fig. 1 **Open data scenario: a new open paradigm for sharing raw data and schemas, vocabularies, and rules**

This model means to add, on top of Tim Berners-Lee's semantic web layer, a layer that defines the usage of the information infrastructure as proposed for a pragmatic web [Paschke-Boley:2009] and for a (meta-)policy web [Hu-Boley:2010]. In this way, the knowledge become technology independent and reusable also without the application logic layer that usually provides a specific interpretation and manipulation of the data. Therefore, the modeling of rules (e.g. constitutive and technical rules) that describe how the data are to be combined, is a key point for the future of those data.

In particular, this holds in the legal domain, where the triple "document, author, role" is strictly linked to the validity and authenticity of the legal statements and rules (e.g., an act published by the official gazette in the role of legal publisher makes the law provisions a "Source of Law" and consequently legal binding [Kelsen 1991]).

The open data approach doesn't mean to provide raw data without any supervision or accountability coming from the institutions. The government body should take care of this emerging area not only for increasing the accessibility and reusability of the public information (dissemination), but for conveying the characteristic role of monitoring and leading of the public data. There is a specific duty in charge of the public administration to aggregate, interpret and provide added value to the data: this layer can be covered by RuleML, expressed in a level separate from, but connected to, the legal document. On the other hand, the institutions should support transparency in order to provide raw data without any filter that could manipulate and alter the information.

Finally, defining a long term vocabulary and glossary (in other words, a standard) is a challenge for the preservation of the data semantics over time. We are facing for the first time a massive digitalization of legal data that in the past was only manageable on paper. This means to pay attention to the long term preservation also of the XML standards definition: vocabulary, glossary, methodology for applying the validation, rules of grammar, methodology concerning how to apply the standard, etc., should be preserved because often the standards are not sufficiently descriptive and prescriptive for defining, in unambiguous and undisputable ways, the correctness and the meaning of the markup (usually the XML schema is not self- explainable). We need a standard descriptive and prescriptive enough to persist for generations. RuleML could help in this goal by providing the necessary description about how the standards have to be used now and in future.

With regard to the above-mentioned considerations, we aim to extend RuleML for supporting the description of the intentional level of the *data silos*. Based on that, we would like to investigate how to produce persistent results (e.g., preservation of digital documents) for guaranteeing the authenticity, legality, and validity of these resources over time, using normative and constitutive rule modeling. In other word, we need to define, with the rules, a framework encompassing the history, nature, and type, as well as the proper behavior of the raw data and their standards.

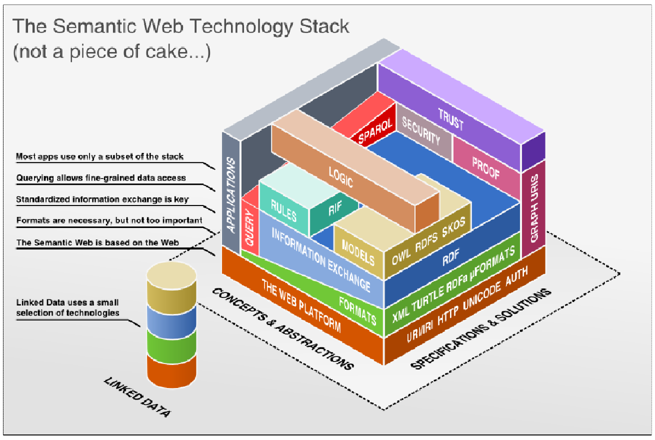
The Semantic Web Technology Stack
(not a piece of cake...)
Most apps use only a subset of the stack
Querying allows fine-grained data access
Standardized information exchange is key
Formats are necessary, but not too important
The Semantic Web is based on the Web
Linked Data uses a small selection of technologies
LINKED DATA
APPLICATIONS
TRUST
SPARQL
SECURITY
PROOF
LOGIC
RULES
RIF
QUERY
GRAPH URIS
INFORMATION EXCHANGE
MODELS
OWL RDFS SKOS
RDF
FORMATS
XML TURTLE RDFa μFORMATS
THE WEB PLATFORM
URI/IRI HTTP UNICODE AUTH
CONCEPTS & ABSTRACTIONS
SPECIFICATIONS & SOLUTIONS

Fig. 2 3D Semantic Web stack

4. LegalRuleML and RuleML

RuleML is an XML based language for the representation of rules. It offers facilities to specify different types of rules from derivation rules to transformation rules to reaction rules. It is capable of specifying queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviors of workflows, services, and agents [Boley-Tabet-Wagner :2001] [Wagner 2004] [Boley-Paschke-Shafiq :2010].

RuleML is intended as the canonical web language for rules, based on XML markup, formal semantics, and efficient implementations. Its purpose is to allow exchange of rules between major commercial and non-commercial rule systems on the Web and various client-server systems located within large corporations. It can facilitate business-to-customer (B2C) and business-to-business (B2B) interactions over the Web, including by using Rule Responder[4] on top of an Enterprise

[4] http://ruleml.org/RuleResponder/

Service Bus. RuleML provides ways of expressing business rules in modular units at various levels of granularity. It allows the deployment, execution, and exchange of rules between different systems and tools. It is expected that RuleML will be the declarative method to describe rules on the Web, in intranets, and other distributed systems.

RuleML has arranged rule types in a hierarchical structure comprising reaction rules (event-condition-action-postcondition rules), transformation rules (functional-equational rules), derivation rules (implicational-inference rules), facts ('premiseless' derivation rules, i.e., derivation rules with empty bodies), queries ('conclusionless' derivation rules, i.e., derivation rules with empty heads) and integrity constraints ('failure-implying rules' for consistency maintenance). The current version of the RuleML hierarchy is shown in Fig. 4.

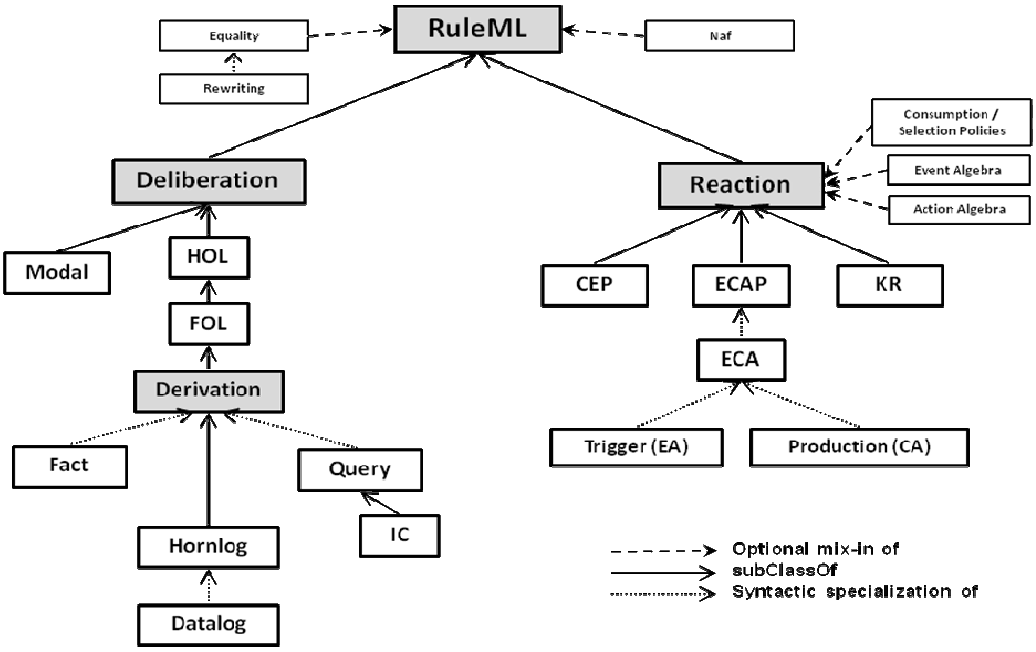

Fig. 4 Family of RuleML languages (cf. [Boley-Paschke-Shafiq:2010])

The way the RuleML/XML specification has achieved flexibility and extensibility is based on the use and composition of XML Schema modules, currently being refined using Relax NG.[5] Each module is meant to implement a particular feature relevant for a specific language or application (e.g., modules for various types of negation, for example, classical negation, and negation as failure). Each module is intended to refer to a semantic interpretation of the feature implemented in the module. While a specific default semantics is always predefined for each RuleML language, the intended semantics of a rulebase can override it by using explicit values for corresponding semantic attributes.

The key strength of RuleML is its extensibility. Thus, despite the current lack of a sublanguage specifically designed for the representation of legal rules, a few authors have proposed a syntactic extension and semantic interpretation for this area, in particular for the representation of (business) contracts [Grosof 2004, Governatori 2005a, Governatori 2009]. The contribution of [Governatori: 2005b] is a proposal of adopting courteous logic programming (a variant of defeasible logic) as execution model for RuleML rulebases, corresponding to the clauses of a contract. Accordingly, Grosof's proposal meets the key requirement of defeasibility for modeling legal rules. Technically, [Grosof 2004] uses derivation rules, but then a courteous logic program, implemented as SweetJess rules, constitutes an executable specification, where the conclusion of a rule can be executed by a

[5] http://wiki.ruleml.org/index.php/Relax_NG

computer program producing effects. Thus the approach bridges the declarative-procedural gap among the various types of rules in the RuleML family.
The limitation of [Grosof 2004] is that it does not consider normative effects (i.e., it is not possible to differentiate between obligations and permissions). This limitation has been addresses by Governatori [Governatori 2005a], where defeasible logic is extended with the standard deontic operators for obligations, permissions, and prohibitions, as well as a new special deontic operator to model violations and penalties for the violations.
Furthermore, [Governatori 2005a] distinguishes between constitutive and prescriptive rules. It provides a RuleML compliant DTD for representing the various deontic elements, and discusses various options for the modeling of such notions in defeasible logic. [Governatori 2009] implements [Governatori 2005a], in a Semantic Web framework with support for RDF databases, to provide an environment to model, monitor and perform business contracts. The modeling approach proposed in [Governatori 2005a] has proven successful for various legal concepts (for example the legal notion of trust [Rotolo 2009]) and it has been extended to cover temporal aspects [Governatori 2005b] and norm dynamics [Governatori 2010], and it has been applied to the study of business process compliance [Governatori 2006].

In this scenario we have extended RuleML to include, in orthogonal way, a new dialect capturing all those requirements, not fully incorporated in the orig-inal 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.

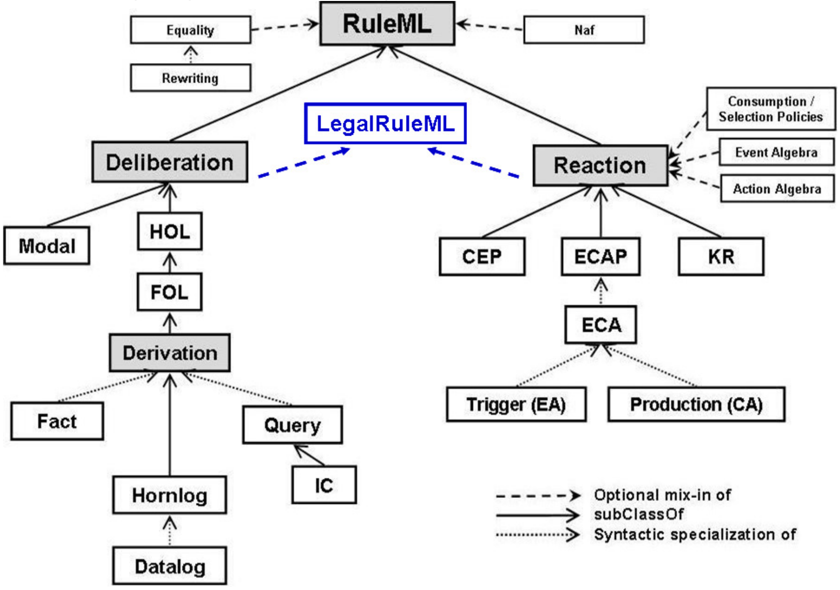

Fig. 5 . LegalRuleML position in the current RuleML architecture (adapted from [Boley-Paschke-Shafiq:2010])

5. Preliminary Proposal: LegalRuleML Modules

To extend LegalRuleML, we have defined two more XML-schemas: `LegalMeta.xsd` module and `Legal_operators.xsd` module. `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.

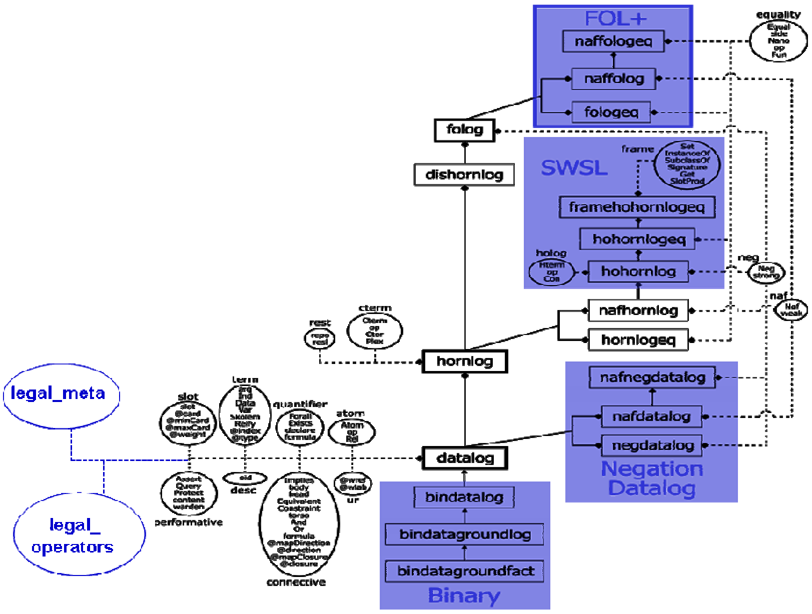

Fig. 1. `legal_metadata` module included in the `datalog` component (adapted from [34])

5.1. Legal Meta Data

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

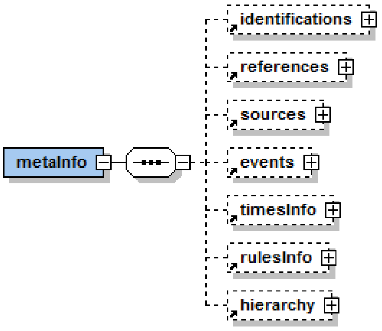

Fig. 2. metaInfo module organization.

— `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"
refersTo="/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>
```

5.2. 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.

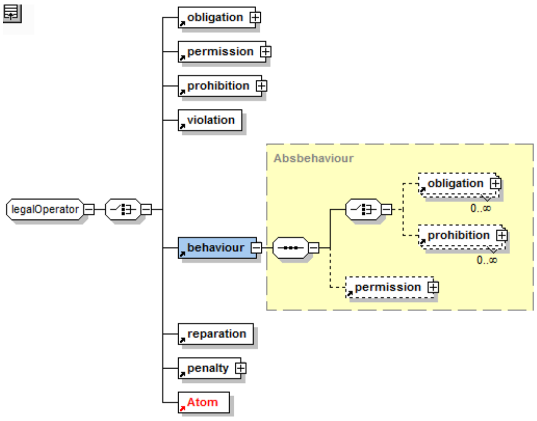

Fig. 3. Legal_operators.xsd elements

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>
```

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>
```

5.3. 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>
```

5.4. *Mapping Functions and Modules*

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.

6. Market target and potential memberships

There is an important market sector that will benefit from LegalRulML approach:

- e-legislative and e-parliament
- e-justice
- e-gov and e-commerce
- banking and insurance domain
- e-health
- Cloud Computing (particularly SLAs, Privacy and Security)

LegalRuleML will have the opportunity to use the unique competencies and skills of several communities focussed on the topic of legal rules:

- RuleML, ICAIL, IAAI. Several annual or biannual conferences could be used for coping with the goal to involve those expert in the standard specifications. In particular the RuleML Symposia and the RuleML Organization (www.ruleml.org) is a strong community.
- ebXML, UBL, tgf TCs may be involved for the join activities and events;
- industries involved in the rule modeling may be involved;
- NIEM community should be involved.

7. LegalRuleML Plan

7.1. LegalRuleML Organization

We intend to build a LegalRuleML/XML module, eventually composed of sub-modules, able to include all the aforementioned requirements and to interchange with the business rule domain.

The plan is to organize the work in four main workpackage:

- LegalRuleML semantic level (e.g. temporal dimension);
- LegalRuleML logic level (defeasibility, deontic, argumentation);
- LegalRuleML integration with business and process rules;
- LegalRuleML open data level.

The semantic and logic levels constitute the core part of the LegalRuleML module. They define the principles of design, the architecture of the module, the main elements for managing patterns, abstract types, group of attributes, general classes, ontology level connection, rule level connection. They have to maintain consistency and clarity in the global design architecture of the module.

The other two areas (integration and open data) can proceed in parallel, strongly coordinated with the core part (semantic and logic levels).

7.2. LegalRuleML Schedule

The schedule is defined as follows, distributed over four years:

First Year	- First draft standard specifications
	- First documentation
	- pilot cases
	- evaluation of the pilot cases
	- refinement of the draft specifications and documentation

Second Year	- publication of the draft standard specification document for the 30-days public review
	- collection of the comments

	- consolidation of the draft version
	- publication of the draft standard specification document for the 30-days public review
	- collection of the comments

Third Year	- OAIS Standard Candidate, Public Review (60-days) and Ballot

Fourth Year (and follow)	- Maintenance of the LegalRuleML - Disseminate and Exploit LegalRuleML - Support applications and tools based on LegalRuleML

FIRST YEAR ROADMAP

WP	Q1	Q2	Q3	Q4
Semantic Level				
	*Definition of the main principles and architecture of the module. *Definition of the basic elements, abstract type, general classes. *Ontology and rules mechanism connection. * Connection with NIEM platform			
		*ID naming convention definition. *URI naming convention definition.		
Logic Level			*Deontic logic elements * Deafisible logic elements	
				* Argumentation
Business rules and processes modeling - Interconnection	* Business rule modeling interconnection definition	* Business rule modeling interconnection requirements	* Business process modeling interconnection specifications	* Business process modeling interconnection specifications

Open Data	*Open data module definition according also with the NIEM platform	*Open data module requirements * Open data module specifications	*Open data module specifications	*Open data module specifications

References

1. Abate F. and Jewell E. J., editors. New Oxford American Dictionary. Oxford UniversityPress, 2001.
2. Bench-Capon T. and Coenen F.: Isomorphism and legal knowledge based systems. Artificial Intelligence and Law, 1(1):65–86, 1992.
3. Bench-Capon T., Coenen F.: Isomorphism and legal knowledge based systems. Artificial Intelligence and Law, 1(1):65–86, 1992.
4. Bench-Capon T.: The missing link revisted: The role of teleology in representing legal argument. Artificial Intelligence and Law, 10(1-3):79–94, September 2002.
5. Benjamins V. R., Casanovas P., Breuker J., and Gangemi A., editors. Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval and Applications. Springer-Verlag, 2005.
6. Berners-Lee T.: Long Live the Web: A Call for Continued Open Standards and Neutrality, Scientific America, 2010.
7. Boer A., Radboud W., Vitali F.: MetaLex XML and the Legal Knowledge Interchange Format, in Computable Models of the Law, Springer, 2008.
8. Boley H., Tabet S., and Wagner G.: Design rationale for RuleML: A markup language for Semantic Web rules. In I. F. Cruz, S. Decker, J. Euzenat, and D. L. McGuinness, editors, Proc. SWWS'01, The first Semantic Web Working Symposium, pages 381–401, 2001.
9. Boley H., Paschke A., Shafiq O.: RuleML 1.0: The Overarching Specification of Web Rules. RuleML 2010: 162-178.
10. Boley H., Paschke A., 2009.
11. 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.
12. Francesconi E., Montemagni S., Peters W., Tiscornia D.: Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language Springer 2010.
13. Genesereth M. R.: Logical Spreadsheets. RuleML 2010.
14. Giovanni S.: Legal concepts as inferential nodes and ontological categories. Artif. Intell. Law 17 (3): pp. 217-251, 2009.
15. Gordon T. F., Guido Governatori, Antonino Rotolo: Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain. RuleML 2009: pp. 282-296, 2009.
16. Gordon T. F.: Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF). Computable Models of the Law, Languages, Dialogues, Games, Ontologies 2008, pp. 162-184, 2008.
17. 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.
18. Governatori G. and Pham D. H.: Dr-contract: An architecture for e-contracts in defeasible logic. International Journal of Business Process Integration and Management, 5(4), 2009.
19. Governatori G. and Rotolo A.: Changing legal systems: Legal abrogations and annulments in defeasible logic. The Logic Journal of IGPL, 2010.
20. Governatori G., Milosevic Z., and Sadiq S.: Compliance checking between business processes and business contracts. In Proc. EDOC 2006, pages 221–232. IEEE, 2006.
21. Governatori G., Rotolo A., and Sartor G.. Temporalised normative positions in defeasible logic. In Proc. ICAIL'05, pages 25–34. ACM Press, 2005.
22. Governatori G.: Representing business contracts in RuleML. International Journal of Cooperative Information Systems, 14(2-3):pp. 181–216, 2005.
23. Grosof B.. Representing e-commerce rules via situated courteous logic programs in RuleML. Electronic Commerce Research and Applications, 3(1):2–20, 2004.
24. Guido G., Rotolo A.: Norm Compliance in Business Process Modeling. RuleML 2010: pp. 194-209, 2010
25. Hage. J. C.: Reasoning with Rules – An Essay on Legal Reasoning and its Underlying Logic. Kluwer Academic Publishers, Dordrecht, 1997.
26. Hu Y.-J., Boley H.: SemPIF: A Semantic Meta-policy Interchange Format for Multiple Web Policies. Web Intelligence 2010: 302-307.
27. http://www.niem.gov/pdf/NIEM_HLVA.pdf
28. IEEE, 2004.
29. Kelsen H.: General theory of norms. Clarendon, Oxford, 1991.
30. Lupo C., Vitali F., Francesconi E., Palmirani M., Winkels R., de Maat E., Boer A., and 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.
31. Palmirani M., Contissa G., Rubino R.: Fill the Gap in the Legal Knowledge Modelling. RuleML 2009: 305-314, 2009.
32. Palmirani M., Governatori G. and Contissa G:. Temporal Dimensions in the Rules: an Evolution of LKIF Rule, Jurix 2010.

33. Prakken H. and Sartor G.: A dialectical model of assessing conflicting argument in legalreasoning. Artificial Intelligence and Law, 4(3-4):331–368, 1996.
34. Proceeding of the 12th International Conference on Artificial Intelligence and Law, Barcelona 8-12 June, 2009, ACM, NY, 243.
35. Rotolo A., Sartor G., and Smith C.: Good faith in contract negotiation and performance. International Journal of Business Process Integration and Management, 5(4), 2009.
36. RuleML. The Rule Markup Initiative. http://www.ruleml.org, 20th August 2009.
37. Sartor G.: Legal reasoning: A cognitive approach to the law. In E. Pattaro, H. Rottleuthner, R. Shiner, A. Peczenik, and G. Sartor, editors, A Treatise of Legal Philosophy and General Jurisprudence, volume 5. Springer, 2005.
38. Sartor G.: Legal Reasoning: A Cognitive Approach to the Law. Vol. 5. Treatise on Legal Philosophy and General Jurisprudence. Berlin: Springer, 2005.
39. Verheij B.: Rules, Reasons, Arguments. Formal Studies of Argumentation and Defeat. Ph.d., Universiteit Maastricht, 1996.
40. Vitali F.: Akoma Ntoso Release Notes. [http://www.akomantoso.org]. Accessed 20 August 2009.
41. Wagner G., Antoniou G., Tabet S., and Boley H.: The abstract syntax of RuleML – towards a general web rule language framework. In Proc. Web Intelligence 2004, pages 628–631.

8. Annexes: Letter of Intents

On behalf of the xxxxxxxx (name of organisation)**,** I herewith confirm our interest to participate in the above metioned LegalRuleML intiative and in particular to open a specific Technical Committee inside of the LegalXML member section.

My institution is active e in the field of the above mentioned RuleML Inc. community since …..
In particular the mission of the xxxxxxxx (name of organisation) is to ……………. (insert the mission of your organisation)

Our role in the initiative will be ……………………… (ex: technical role, mentoring, boarding, to promote the programme, to offer internships, to offer a research supervision to thesis and research works, etc).

Signature
Place and date