

Legal Rules, Text and Ontologies Over Time

Monica Palmirani¹, Tommaso Ognibene, Luca Cervone¹

¹ CIRSIFID, University of Bologna.
{monica.palmirani, tommaso.ognibene, luca.cervone}@unibo.it

Abstract. The current paper presents the “Fill the gap” project that aims to design a set of XML standards for modelling legal documents in the Semantic Web over time. The goal of the project is to design an information system using XML standards able to store in an XML-native database legal resources and legal rules in an integrated way for supporting legal knowledge engineers and end-users (e.g., public administrative officers, judges, citizens).

Keywords: Legal Reasoning, Akoma Ntoso, LKIF-core, LegalRuleML.

1. Introduction

“Fill the gap” [28] is a project funded by the CIRSIFID-University of Bologna in 2009 as an extension of the outcomes of the ESTRELLA¹ IST project (IST-2004-027655) with the aim for performing a platform where legal documents are modelled using XML standards and the ontology layer is used as the interconnection technique between the pure text of the document and the embedded legal knowledge, including rules representing the norms expressed by the textual document. The ontology is used for modelling the legal concepts and to represent the properties and the T-Box axioms of the main legal values (e.g., copyright, work, etc.), including geo-spatial (e.g., jurisdiction) and legal temporal dimensions (e.g., enforceability, efficacy, applicability of the norms). The text, annotated in XML using the Akoma Ntoso standard [35] and the metadata, extracted using parsers and NLP techniques [22], are connected manually to the ontology framework [14][10] and finally, the rules, formalized in *defeasible logic*, are connected to the textual provisions and to general and abstract legal concepts modelled in the legal ontology.

In this way several applications are possible: i) to improve information retrieval of legal documents; ii) to facilitate navigation over time when norms and texts change; iii) to foster semantic indexing, classification and query; iv) to enhance the expressivity of formal models of legal reasoning; v) to provide appropriate legal explanations using textual provisions in order to justify the proof of reasoning and vi) finally to annotate legal resources with the legal knowledge output of the proof for permitting open data sharing (e.g., Linked data). The aim of this paper is to present a first demo of this approach applied on a fragment of the US code, Title 17, Section 504 in order to provide a *proof-of-concept* of the applicative architecture.

¹ <http://www.leibnizcenter.org/current-projects/estrella>

2. Relevance of the Problem

The last twenty years have seen a growing interest in the development of XML standards, methodologies and models for the management of legal knowledge [21][33][8]. This interest covers not only “proper” law documents, such as legislation, but it embraces all source documents with a relevant legal content, such as the vast area of internal regulations of public bodies/private companies, codes of conduct, codes of practice, often cited as “soft law”. This is particularly true in some financial/legal domains [6], such as those involving banks and insurance companies and as well as new emerging sectors like patent law , cloud computing and privacy [17], credit card company regulations and telecommunication policies.

The Computer Science and Law community itself [30] dedicated the last two decades to modelling legal norms using different logics and formalisms. The methodology used starts from a reinterpretation of the legal source text by a Legal Knowledge Engineer who extracts the norms, applies models and theory using a logic representation and finally represents them with a particular formalism. In the last ten years several Legal XML standards have arisen for describing legal text and rules (RuleML, RIF, SWRL, etc.). In the meantime the Semantic Web, in particular Legal Ontology research, combined with NLP extraction of semantics, has given a great impulse to modelling the legal concepts [23][10][14][31]. Certainly, one of the main challenges is to acquire the ability to capture, with the help of NLP techniques [30][36], all relevant legal knowledge embedded in a legal document and to represent it in an appropriate formal model. This enables the descriptiveness, meaning and semantics of the source document to be retained, and at the same time, the knowledge is machine-readable and computable.

In this scenario there is an urgent need to close the gap between the text description, represented using XML techniques, and the norms formalized with logical rules in order to realize an integrated and self-contained representation. There are four main reasons:

- legal knowledge is currently presented in a disjointed way in the original text that inspired the logical modelling. This disconnection between legal document management and logic representation of the embedded rules strongly affects the real usage of the legal document knowledge in favour for citizens, public administrations and business (e.g., contracts, assurance regulation, bank soft law, etc.);
- management of changes to the legal document over time, especially act, regulations and contracts that by nature are variable and subject to frequent modifications, significantly affecting the coordination between the text and the rules that should be remodelled;
- the legal validity of the text as authentically approved by the empowered bodies (e.g., contractors) should be preserved by any manipulation. On the other hand, it is important to connect legal document resources, which themselves include many legality values (e.g., authenticity, integrity, evidence in trial, written form, etc.), with the multiple interpretations coming from legal knowledge modelling;
- a theory of legal document modelling able to separate clearly the many layers of representation of the resource: content (text), structure of the text, metadata on the

document, ontology on the legal concepts expressed in the text, legal content modelling (regulatory part of the text) are fundamental to preserving over time the digital legal text enriched by many semantic annotations, including logic representation of the rules.

3. Filling the Gap: from Text to Rules

The first distinction that we should provide to understand goals in the legal domain is to distinguish between three conceptual layers:

- **norms** (abstract mandatory commands concerning rights or duties)
- **textual provisions** (sequences of texts) and
- **rules** (elaboration of the text in logical rules).

The **norm**, following Kelsen's definition [19], is an abstract mandatory command concerning rights or duties. The norm usually is expressed in written using legal text or in an oral way (e.g., social norm, oral contract) or in other representations (e.g., symbolic road signs).

The **textual provisions** (or simply *provisions*) are the instantiation of the general norms in one possible textual representation (sentence, article, paragraph).

The **legal rules** are interpretation of the provision(s) formalized using logical rules in the form of antecedent and consequent. Sometime several provisions determine a single rule or a single provision includes multiple rules.

Usually in the state of the art AI&Law scholars focused their attention only on the rules modelling and on the foundational logical theory, and apart the isomorphism principles [4] the connection with the text over time and the ontology aspects has been neglected. There is a theoretical and important debate in the AI&Law community on the interpretation of the legal textual provisions expressed in natural language and on canonization of the rules using logical formalisms [5]. The prevalent theory is now oriented towards *hybrid interpretation* [32] (rather than pure *textualism*, or pure *interpretation*). We want to make visible in the text a "scintilla of evidence" that there is a minimal but reasonable interconnection, following the legal theory of interpretation, with a logical rule in a formal representation. This exercise sometimes forced the legal knowledge expert to split the original provision in two or more rules, or to duplicate the rules or to compact several sentences in one unique rule. In this scenario we have to manage an N:M relationship between norms, textual provisions and ontology that we want to capture and to represent maintaining the strong separation between these three levels.

The law changes over time and consequently change the rules and the ontological classes (e.g., the definition of EU citizenship changed in 2004 with the annexation of 10 new member states in the European Community). It is also fundamental to assign dates to the ontology and to the rules, based on an analytical approach, to the text, and analyze the relationships among sets of dates. The semantic web cake recommends that content, metadata should be modelled and represented in separate and clean layers. This recommendation is not widely followed from too many XML schemas, including those in the legal domain. The layers of content and rules are often confused to pursue a short annotation syntax, or procedural performance

parameters or simply because a neat analysis of the semantic and abstract components is missing.

Therefore in our vision all legal resources present a complex multilayered information architecture [2] that includes several perspectives of analysis:

- **TEXT.** The perspective of the document officially approved by a legally competent authority. This text is the only legal binding.
- **TEXT'S STRUCTURE.** The perspective of the document that describes the way the text is organized.
- **METADATA.** Any additional information that was not deliberated by the legally competent authority. Metadata can describe the document itself (e.g., by way of keywords), its workflow (e.g., procedural steps in the bill), its lifecycle (the document's history), or its identification (e.g., by way of an URI).
- **ONTOLOGY.** Any information specifying the legal or institutional setting in which the document plays a role—e.g., information identifying the document as a judgment or opinion about the legal system's concepts—or any legal concept which is invoked in the text and which needs modelling (e.g., jurisdiction, enter into force, applicability, etc.).
- **LEGAL RULES.** The legal interpretation and modelling of the text's meaning. The transformation of the norms in logical rules for permitting legal reasoning. Several XML standards are present in the state of the art for managing rules (RIF, RuleML, SWRL), nevertheless RuleML seems to provide a flexible language able to describe different possible theories or logical models (propositional, predicative, argumentative, non-monotonic, deontic, defeasible, etc.) but it is not fitted for the legal domain. For this reason we are starting a new technical group in OASIS called LegalRuleML with the aim to perform a specific RuleML module oriented to the legal peculiarities including defeasibility, deontic, temporal reasoning, qualification of the norms, institutions (e.g., authority, agents, authors), jurisdiction [29].

The aim of this project is to connect all the layers and to manage the temporal dimensions of the provisions (text layer) and rule levels to make them connected accordingly with respective change over time. In other words our aim is to track the modifications of the text especially concerning the temporal dimensions (e.g., application of the norm) and to model the changes in the rules accordingly with the change management of the text. Vice versa it often happens that rules are changed by external events or by the result of legal reasoning, so the validity of the text is affected by these inferred knowledge (e.g., international treaty rules suspended by war).

Secondarily, we would like to build a reasoner that is able to deduct the correct rules to apply depending on the time parameters of the facts. If we have a precedent in criminal law of the High Court in 1994, a new law in 2010 and while the facts of the case occurred in 2005, but the judge must decide the case-law in 2012, the reasoner has to take in consideration and compare two scenarios: the rules coming before the 2005 (date of the facts) and the rules coming after the 2005 till 2012 for permitting to the judge to evaluate the best opportunities for the criminal (principle of *favor rei*). The reasoner should be able to manage temporal reasoning on the rules at a meta-level in order to pre-filter the pertinent rules according to the temporal parameters and to build dynamically new rules for qualifying the rules (e.g., defeasible and defeater).

We use the Akoma Ntoso XML standard [35] for implementing the first three levels of the Semantic Web cake (mark-up the text, the structure and the legal metadata) and to provide hooks and mechanisms for referring to external ontologies and to legal knowledge modelling using URI and idRef to the proper nodes. LKIF-Core [10] provides general mechanisms for coping with the ontology level and it is able to manage the events, the roles, the authors, and other more fine grained legal knowledge models. Finally LegalRuleML (we used a draft preliminary version not official approved by the OASIS TC²) is able to model normative rules [29]. The fundamental part of this multilayer architecture is the URI reference based naming convention [16][3] that functions as the interface between levels. A specific resolver is able to point out the different sources and rules, over time, in correct way. This mechanism is included in the *RuleViewer* module presented in section 7.

4. Applicative Scenario

The applicative scenario that this project aims to manage is showed in the following picture (see Fig.1) and organized in the following steps:

- i. legal text is marked-up in XML legal standard, in our case in Akoma Ntoso;
- ii. legal concepts derived by legal text are modelled in OWL and updated on the base of the changes over time;
- iii. rules interpreted by legal text, and integrated by the legal ontologies, are modelled in LegalRuleML;
- iv. a native XML database stores all the files and it is able to manage Xpath and Xquery in easy way on all the files XML, RDF, OWL;
- v. Drools engine provide the level of reasoning;
- vi. general assertions derived by knowledge base reasoning process (e.g., an invalid section) could be exported in RDF in order to enrich the original XML text and in such way to improve the sharable knowledge on the web.

This approach is based on several requirements near to the concept “Fill the Gap”:

- the **granularity** of the legal document marked-up in XML must be related, following the isomorphic principle [4], to the logic rules representation and to the ontological statements (word, paragraphs, etc.). For this purpose we recall the URI references of Akoma Notoso XML nodes in a special metadata block of LegalRuleML called <lrml: sources>, where we specify the provisions connected with the atoms and rules. This **relationship between rules and text** is an **N:M cardinality**, so the current state of LegalRuleML implements a mechanism able to represent the multiple associations;
- the **contextual information** related to the rules has to manage in atomic way in order to favour the correspondence between rule and the textual provision. This ensures that where there is a modification or cessation of a fragment of text, it is possible to manipulate only a node of the LegalRuleML XML tree without affecting the consistency of the other rules. This principle enables rule and its

² We take into consideration the version available at the date of the paper submission: <http://www.oasis-open.org/apps/org/workgroup/legalruleml/download.php/45888/2.1defeasibility.002.002.doc>

metadata to be encapsulated as an atomic *object*. For this reason we proposed a separate and atomic block <lrml:ruleInfo> in LegalRuleML (under discussion in the TC).

- the relationship between the document, rules and the ontology layers needs to be implemented considering the bidirectionality of the information and, among the other issues, the evolution of the legal concept over time. The system is not able to implement this feature in the current version;
- the legal document **changes over time**, therefore in the rule modelling layer, as well as in the legal ontology and in the text layers, it is necessary a mechanism for managing the dynamicity over time and a temporal logic able to manage retroactivity effect of the norms (e.g., annulment, forking of temporal lines, etc.) and applicability of the law (e.g., an Act about the earthquakes in Italy is applicable only to the events of 20 and 29 May 2012). The **legal temporal model** should be able to manage three main legal axes (enforceability, efficacy and applicability, so a rule could be effective but not applicable when concrete conditionals are not satisfied). For this purpose Akoma Ntoso and LegalRuleML include a temporal model event-based;
- the **deontic operators** need to be managed jointly with the temporal parameters for permitting the correct application of the obligations, rights, permissions and violations accordingly with the fact (e.g, crime) to evaluate;
- the non-monotonic dimension of the law, (e.g., the exceptions that are present often in legal documents including contract) would strongly suggests that a **defeasible reasoning** approach should be adopted.

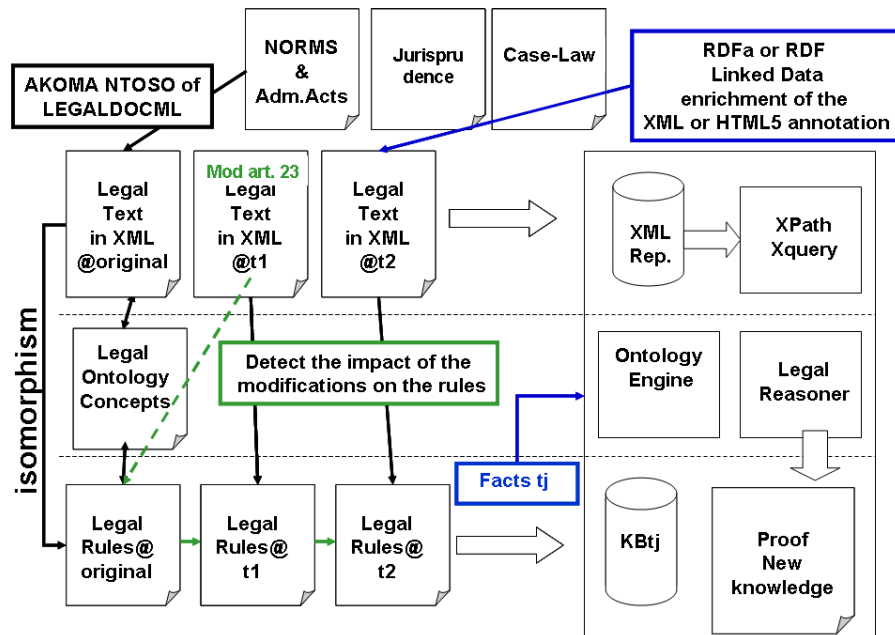


Fig. 1 – Applicative Scenario

5. Architecture Design

For implementing these goals we will adopt the following methodology and apply it to a fragment of the US copyright domain (sec. 504) in order to model, describe and represent the different levels of legal knowledge information:

- **text, structure, metadata:** legal documents (e.g., contracts) will be marked-up in Akoma Ntoso using a new web editor (*Rawe*) based on a previous .specialized editors features (Norma-Editor [24]) that was able to extract structure, references and metadata using parsers [2];
- **legal domain ontology:** an ontology for modelling and defining macro-concepts specific for the legal domain (e.g., IPR issues, jurisdiction, penalty, etc). The ontological classes built manually define the legal theoretical concepts describing their properties, relationships with other contents and their spatial and chronological dimensions. The ontology are stored in the native XML database that is able to manage versioning and evolution over time of OWL2 ontologies [7][1];
- **NLP tools:** the team uses NLP techniques for extracting the legal knowledge embedded in legal texts and to represent them as XML structured elements or as previously defined ontological structures. NLP techniques facilitate the development of parsers able to fill the gap between the text and its semantic level and accelerating the mark up process that usually is a time consuming task [25];
- **rules modelling and reasoning:** the legal document (e.g., law, judgments, contracts, etc.) will be modelled by the legal knowledge engineer in LegalRuleML. Rules using the web editor. The rules represented in LegalRuleML are imported inside of an inference engine properly customized by CIRSFID (based on Drools ver. 5.4) with a specific dialog interface for entering the facts. The rule engine aims to use a hybrid technology [9][34] (see among the others OntoRule project) using semantic rule reasoning taking benefits from the OWL2 ontology;
- **native XML DB repository and rule viewer.** All legal resources (text, metadata, ontologies and rules) are stored and delivered on the Web using a native XML database (based on eXist). In this way all the legal resources will be interconnected and presented for gathering the legal knowledge through an information retrieval engine, a reasoning engine and an application layer [26][27];
- **presentation of rules an text.** Finally the text and rules are presented using a web interface (*RuleViewer*) in order to connect text and rules for the end-user.

Some modules of this architecture (see Annex) are presented in this paper demo.

- A **specialized web editor** for marking up legal text and normative rules (*Rawe*³) in synchronized way, realizing so the isomorphism principle.

³ <http://sinatra.cirsfid.unibo.it/rawe/>

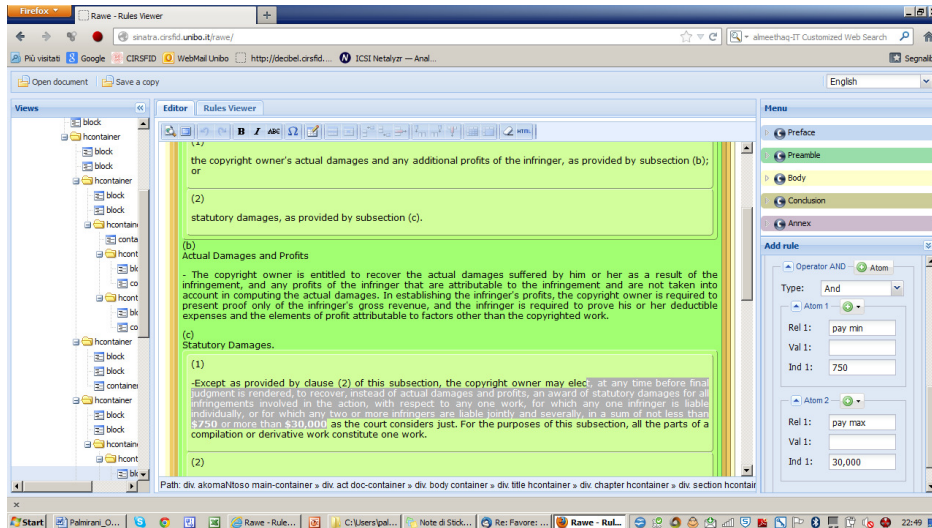


Fig. 2 – Rawe web editor for marking up legal text and normative rules.

The rules are converted to the LegalRuleML emerging standard and the text to Akoma Ntoso.

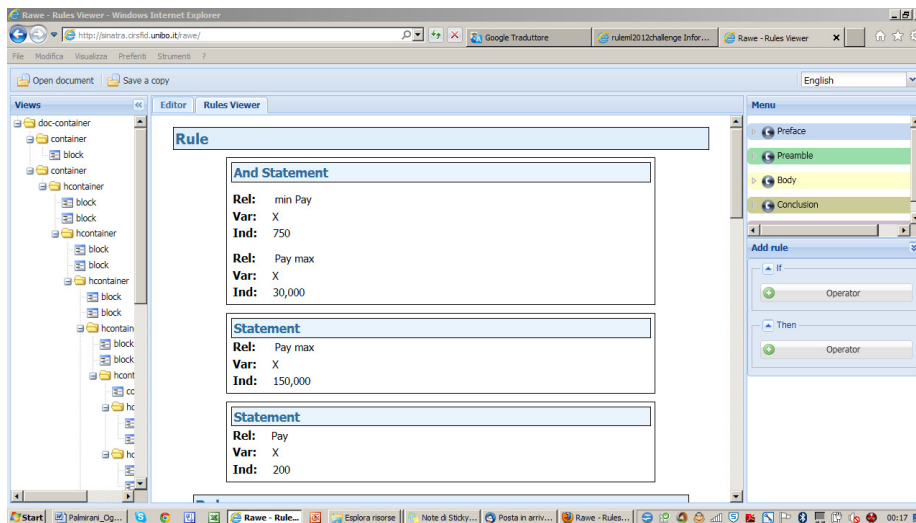


Fig. 3 – Rule Viewer inside of Rawe editor.

- In a second step the editor submits the text and the rules in XML format (respectively in Akoma Ntoso and in LegalRuleML) to the *eXist* native XML database.
- An interface API queries *eXist* and extracts the LegalRuleML files. In particular those files are used for populating the temporal information in a dynamic way stored in the Drools knowledge base.

- The Drools reasoner simulates defeasibility rules and reasoning.
- Finally a rule viewer (*RuleViewer module*⁴) presents the rules connected with the legal text, including version management over time.

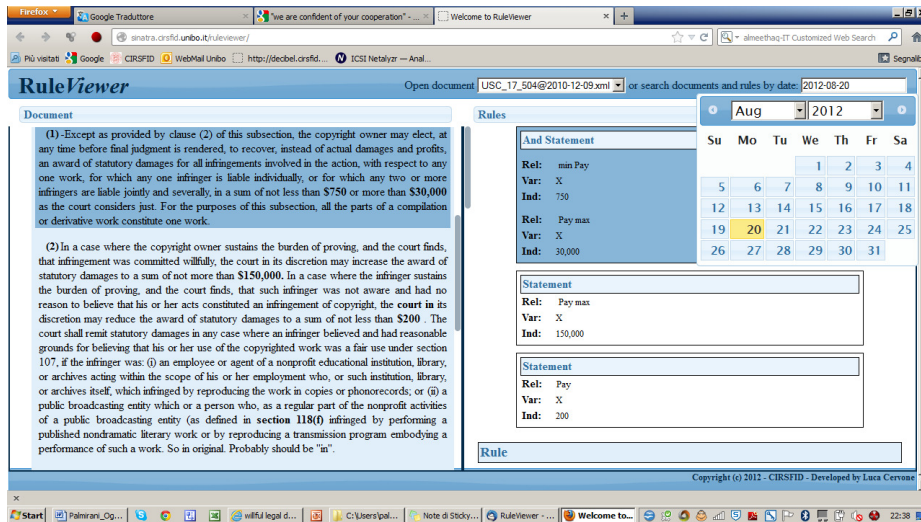


Fig. 4 – Text and RuleViewer.

6. Pilot Case

We definitely aim to provide a demonstration environment for testing the applicative scenario, using a pilot case coming from the US code, Title 17, sec. 504 on the copyright infringement.

17 USC Sec. 504

(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 \$250 or more than \$10,000 as the court considers just. For the purposes of this subsection, all the parts of a compilation or derivative work constitute one work.

(2) In a case where the copyright owner sustains the burden of proving, and the court finds, that infringement was committed willfully, the court in its discretion may increase the award of statutory damages to a sum of not more than \$50,000. In a case where the infringer sustains the burden of proving, and the court finds, that such infringer was not aware and had no reason to believe that his or her acts constituted an infringement of

⁴ <http://sinatra.cirsfid.unibo.it/ruleviewer/>

copyright, the court at its discretion may reduce the award of statutory damages to a sum of not less than \$100.

This provision was modified three times and it is just a chance that the modifications are not retroactive (i.e., in legal domain it is necessary to use a non-monotonic temporal model). The table below lists the modifications with the temporal intervals of efficacy:

Interval name	Interval of efficacy of the norm	Statutory Damages	Willfully	Bona Fides
timeBlock1	[1976-10-19, 1995-03-01[\$250 <= statutoryDamages <= \$10,000	Max \$50,000	Min \$100
timeBlock2	[1995-03-01, 2001-02-01[\$500 <= statutoryDamages <= \$20,000	Max \$100,000	Min \$200
timeBlock3	[2001-02-01, •[\$750 <= statutoryDamages <= \$30,00	Max \$150,000	Min \$200

The goal is to insert in the system a fact in a due date t1 and to check which statutory damages the infringer must pay. We are interested also to show the text of the proper version according to the rules applied and the time.

6.1. Modelling Text and Rules

We have translated the textual provisions (all the three versions) in XML using Akoma Ntoso and the corresponding rules in LegalRuleML⁵. The XML files were posted to the system, stored in *eXist* repository and also converted in Drools Rule Language for permitting to manage the rules in the Drools suite. In Drools a rule has the following format:

```
rule
    // attributes
    when
        // conditions
    then
        // actions
end
```

It is not a classical conditional structure IF-THEN, because Drools uses WHEN-THEN model. Drools implements a version of *Rete* algorithm and executes the conclusions whenever the patterns in the conditions are matched by a fact. Drools supports several temporal reasoning constructs (e.g., Allen's time model) as showed in the table below:

⁵ Using the preliminary draft version of the OASIS TC.

	Point-Point	Point-Interval	Interval-Interval
A before B			
A meets B			
A overlaps B			
A finishes B			
A includes B			
A starts B			
A coincides B			

Fig. 5 – Allen’s temporal model managed by Drools.

Anyway, the implementation of the temporal reasoning by Allen’s model in Drools is effective only when dealing with business rules and business events such as commercial markdowns that live in the present and die in the future. On the contrary, when dealing with legal events and rules, there is not only present and future, but a complex mix of past, present and future to deal with (e.g., modifications in the past with retroactive effects). For this reason the temporal attributes (metadata) implemented by Drools to manage events’ and rules’ lifecycles are not enough in our case. This limitation has been partially supplied by adding temporal constraints as patterns to be matched.

```
@ID("statutory_damages_01")
date-effective "31-OCT-1988"           // start efficacy
//date-expires "dd-MMM-dddd"         // limitation of action (?)
```

Fig. 6 – date-effective and date-expires.

The rule attributes “date-effective” and “date-expires” are almost useless when dealing with legal rules (see Fig. 6). Drools is open source and written entirely in Java, so that we implemented a data structured using POJO. We inserted some instances in the reasoner in the following way:

```
infringements = new ArrayList<Infringement>();
infringements.add(new Infringement(1, 1, "1", new DateTime(1989,1,1,0,0,0,0),
    null, false, false, null));
infringements.add(new Infringement(1, 1, "1", new DateTime(1999,1,1,0,0,0,0),
    null, false, false, null));
infringements.add(new Infringement(1, 1, "1", new DateTime(2009,1,1,0,0,0,0),
    null, false, false, null));
```

Fig. 7 – Instances.

Furthermore, in the DRL (Drools Rule Language) file, we added besides the mere instances some semantics of Drools:

```
declare Infringement
  @role( event )
  @timestamp( getTimestamp() )
  @duration( getDuration() )
end
```

Fig. 8 – Data model of the instance query.

In this way, the reasoner considers the instances as events, with a precise *timestamp* and duration. Though the attributes “@timestamp” and “@duration” suffer the same limitations as the rule’s temporal attributes as “@stated” before. Every rule refers to a precise legal provision with precise temporal parameters for its efficacy. We can add these temporal conditions in every rule as in the following listing (see Fig. 9). Or we can consider the temporal parameters as meta rules that control the rule flow. For this we use the Drools & jBPM RuleFlow (see Fig. 10). In the following example we force the rule engine to fire (or rather, to give a chance to fire) only the rules that match with given certain temporal parameters. In this manner we can deal with rule versioning in a fast way (especially in case of modifications of the relative norms).

```
rule "rule of the statutory damages with burden of proving and willful infringement [from 03-01-1995 to 01-01-2001]"
when
  Date( $startEfficacy: time ) from new Date(getDate("3-1-1995")) // 03-01-1995
  Date( $endEfficacy: time ) from new Date(getDate("1-1-2001")) // 01-01-2001

  $infringement : Infringement
  (
    $infringement.getTimestamp() after $startEfficacy, // temporal reasoning
    $infringement.getTimestamp() before $endEfficacy,

    $infringement.getBurdOfProving(), // normative reasoning
    $infringement.getMalicious()
  )
then
  System.out.println("The event happened in " + $infringement.getTimeStart() +
    " therefore I apply the rule in force [from 03-01-1995 to 01-01-2001].");
  System.out.println("Since the burden of proving is " + $infringement.getBurdOfProving()
    + " and the willfulness is " + $infringement.getMalicious() + " then the rule applied is 'burden of proving + willfulness'");
  System.out.println();
end
```

Fig. 9 – Temporal conditions in Drools Rules.

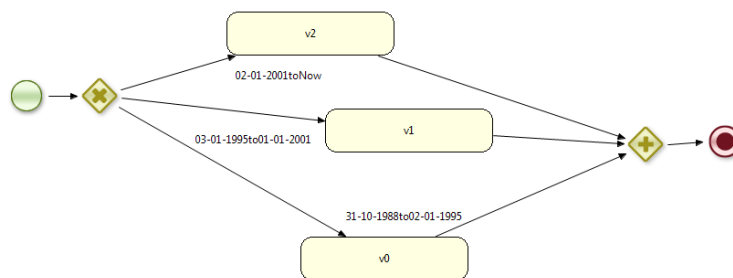


Fig. 10 – Workflow module for modelling the priority of the rules in Drools Rules.

Nevertheless, this kind of rule flow management cannot deal in an effective way with the phenomenon of multiple versions of legal rules applicable at the same time. Or rather, it can, by using an OR type instead of an XOR type, but it needs also a more accurate combination between the two rule set versions.

Drools has various way to deal with complex rule sets. In order to deal with conflicting rules we tried to apply the construct named “salience”. It allows to give to every rule an order of priority. This way the rule engine shall fire firstly the rules with the highest priority. The disadvantages is to define since the beginning the priority of all rules manually.

6.2. Goals

The system manages the following request for each fact on the basis of the below table of variables shown below:

Starting Time of the infringement	Ending Time of the infringement	Status of the willfulness	Status of the proving
T1	T2	Willful Not-willful	Burden of Proving Not burden of proving

Fig. 11 – Data model of the dialogue.

We need also the *status of the work* (public domain or not public domain) that for now it is stored in the local database in static way, but for the future we will insert the appropriate rules coming from the Title 17, US code, and we will deduct in real-time and dynamically the status of the work from the reasoning.

We have entered several cases testing different intervals of time, different status of wilful and proving. In the pilot case this produces the following output:

```
The event happened in 1998-01-01T00:00:00.000+01:00 therefore I must apply the rule in force [from 03-01-1995 to 01-01-2001].
Since the burden of proving is false and the willfulness is false then the rule applied is 'not burden of proving'

The event happened in 1994-01-01T00:00:00.000+01:00 therefore I must apply the rule in force [from 31-10-1988 to 02-01-1995].
Since the burden of proving is false and the willfulness is false then the rule applied is 'not burden of proving'

The event happened in 1989-01-01T00:00:00.000+01:00 therefore I must apply the rule in force [from 31-10-1988 to 02-01-1995].
Since the burden of proving is true and the willfulness is true then the rule applied is 'not burden of proving'
```

Fig. 12 – Output of the test.

For example, the event with *timestamp* “1989-01-01T00:00:00.000+01:00” has the field “burden of proving” set true and the field “willfulness” set true, therefore the legal rule to be applied should be the rule “burden of proving + willfulness”. Nevertheless, as the rule “not burden of proving” has the highest priority and it is a general rule with no constraints, the rule engine fires it and executes the consequences. In a second moment, when the rule engine fires the rules with a lower

priority, it fires our right rule, the rule “burden of proving + willfulness” and this leads to *override* the consequences of the general rule fired before.

Such kind of priority as a conflict resolution strategy between rules may be consequent to the following legal argument: first we apply the general rule, then we apply the more specific rule and override the general rule consequences with the more specific rule consequences.

6.3. Reasoner

Drools rule engine is made for dealing with thousands of facts and thousands of rules fired real time by different entry points. Its aim is to build expert systems able to take into consideration all the events happened during the time and matching the appropriate conclusions by means of rulebases. For our legal domain, the best option is to build a legal expert system very focalized on concrete, business application in order to exploit all the best features of Drools (If the aim is to exploit all the best feature of a Business Rule Management System). For example if we want a system for dealing with US Copyright law, with Drools we can build a rule engine always active, 24/7, that process several precise classes of events (such as possible violations, death of authors, payment of royalties) and compute different kinds of rules (we can have more abstract rules that just represents the logic within the norms, or we can have more concrete rules that realize a business policy conforming to the norms). In technical terms, Drools offers:

- Several clock types, for example, a real-time clock or a pseudo clock. The real-time clock is the default and should be used in normal circumstances. But in the legal domain the pseudo clock is especially useful as we have complete control over time and we can better deal with complex temporal reasoning schemas.
- The declaration and usage of events with both semantics: point-in-time events and interval-based events. This is useful in the legal domain since an infringement or, more generally, a crime, can be committed in a precise single moment, or in an interval of time (e.g., stalking crime that is a set of harassment during an interval of time).

7. Front-End Component: RuleViewer

At the end the system front-end *RuleViewer*⁶ (see Fig. 5) shows the list of the versions of the US Code, Title 17, section 504 over times (seven versions). The end-user can navigate to one version (using a calendar or selecting the list of versions) and to detect immediately, by *mouse-hover* mechanism, the rules involved in a due fragment of text. Vice versa it is possible to navigate the rules in the right window and the end-user can see the correspondent fragment of legal text for integrating the legal interpretation. An important feature of *RuleViewer* is to provide a common environment where the models for representing legal knowledge are close to the

⁶ <http://sinatra.cirsfid.unibo.it/ruleviewer/>

document text, thus bridging the gap between legal reasoning and textual representation. This means that each document (or each document package) can be connected with a set of rules representing the norms found in the legal document. It is for this purpose that we use the LegalRuleML language in our project in a very preliminary version⁷. The relationship between LegalRuleML files and the legal textual Akoma Ntoso documents is thus managed using a many-to-many (N:M) cardinality expressed in the <lrml:source> element that is able to connect, using URI references, rules and original textual sources. This makes it possible to navigate from a document to its corresponding norms and vice-versa, also considering the versioning. This can be done using the latest Web technologies (AJAX) and the mouse-hover technique. But this also means that searching for the relations dependent on a rule can be a quite complex task, typically involving a very large body of XML documents. The best way to query LegalRuleML resources related to a document is to use Xpath or Xquery languages and to foster the < lrml:source> < lrml:ruleInfo> LegalRuleML blocks and the temporal parameters related to the rules. So, if we store the LegalRuleML files in a particular instance of the *eXist* database, we can use the above-mentioned technology to do complex queries in a very customizable way and with good results. This feature of connecting documents with their rules and vice-versa is helpful especially where change management is concerned, for in this way we can navigate a set of documents stilled at a time *t* and extract a subset of rules already coordinated with respect to the timeline. An inference engine can process for purposes of reasoning.

8. Conclusion

We have presented the foundational architecture design of a project called “Fill the Gap” which aims to integrate and to interconnect legal text marked up in Akoma Ntoso with rules marked up with the emerging LegalRuleML and ontology expressed in LKIF-Core. Three modules are presented: i) a specialized web editor for marking up text and rules in XML; ii) an eXist repository with an API for Xpath and Xquery; iii) a prototype in Drools for implementing defeasible reasoning using first priority mechanism and after workflow steps definitions for simulating the hierarchy relationship among the rules (*override* mechanism); iv) a front-end visualization of the legal text and rules in *side-by-side* windows. We have also refined the temporal aspects, but the legal time parameters don't follow the canonical sequence of Allen's events because we need to manage the retroactive effects. Finally we have tested the Drools engine for producing some results. Unfortunately Drools needs new modules for managing defeasibility (not only simulating it) and temporal reasoning closer to the legal domain time parameters imported dynamically by LegalRuleML files. We are satisfied by these discovery because for the future we intend to proceed with the following experiments: 1) to refine the editor especially on the rule modelling side and for including boxes dedicated to metadata; 2) to extend the eXist API versus Drools;

⁷ Documents of the OASIS LegalRuleML TC are available at https://www.oasis-open.org/committees/documents.php?wg_abbrev=legalruleml. The mailing list describing the work in progress can be browsed at <https://lists.oasis-open.org/archives/legalruleml/>.

3) to extend the Drools environment with the following modules: i) to import in a more effective way in Drools the LegalRuleML rules and to export from the editor them in the LegalRuleML syntax (when fixed by the OASIS TC); ii) to manage defeasibility in a more abstract way, not embedded in the Drools code; iii) to implement an abstract legal temporal model of reasoning. The goal is also to understand if we can reach the same performance and expressiveness of SPINdle [20] using some Drools extensions; 4) to refine the front-end interface for the rule viewer (e.g., graphs representation) and to improve the Rest/API from and to Drools.

Acknowledgement. A particular thanks go to Guido Boella, University of Turin, for the precious inputs on the research topics and on the organization of the paper for improving readability. I would like to thank Llio Humphreys for the proof-reading, not limited to the English matter but fruitfully integrated with her competences in the legal informatics and logic.

References

- [1] Ashley K. D.: Ontological requirements for analogical, teleological, and hypothetical legal reasoning. In: ICAIL 2009, pp. 1-10, 2009.
- [2] Barabucci G., Cervone L., Palmirani M., Peroni S., Vitali F.: Multi-layer Markup and Ontological Structures in Akoma Ntoso. In: LNCS 6237/2010, pp. 133-149, Springer, 2010.
- [3] Bekiari C., Doerr M. and Le Boeuf P.: International Working Group on FRBR and CIDOC CRM Harmonization. 2008. FRBR object-oriented definition and mapping to FRBRER (v. 0.9 draft). Accessed 20 August 2009. http://cidoc.ics.forth.gr/docs/frbr_oo/frbr_docs/FRBR_oo_V0.9.pdf.
- [4] Bench-Capon T. and Coenen F.: Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law*, 1(1):65–86, 1992.
- [5] Boella G., Governatori G., Rotolo A., Torre L.V.D.: A Formal Study on Legal Compliance and Interpretation. In: AICOL Workshops(2009), Springer, 162-183, 2011.
- [6] Boella G., Humphreys L., Martin M., Rossi P., and van der Torre L.: Eunomos, a legal document and knowledge management system to build legal services. In Proceedings of AI Approaches to the Complexity of Legal Systems Workshop (AICOL), Berlin, Springer, 2012.
- [7] Boer A., Hoekstra R., de Maat E., Hupkes E., Vitali F., Palmirani M., Rátai B.: CEN Metalex Workshop Agreement (2009-08-28 proposal). <http://www.metalex.eu/WA/proposal>.
- [8] 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*, Springer, Heidelberg (2008), pp. 21-41, 2008..
- [9] Bragaglia S., Chesani F., Ciampolini A., Mello A., Montali M., Sottara D.: An Hybrid Architecture Integrating Forward Rules with Fuzzy Ontological Reasoning. *HAI* (1) 2010: pp. 438-445, 2010.
- [10] 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.
- [11] Brighi R., Lesmo L., Mazzei A., Palmirani M., Radicioni D.: Towards Semantic Interpretation of Legal Modifications through Deep Syntactic Analysis. *JURIX 2008*: 202-206, 2008.
- [12] Gordon T. F., Governatori G., Rotolo A.: Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain. *RuleML 2009*: pp. 282-296, Springer, 2009.
- [13] 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, 2008.
- [14] Hoekstra R., Breuker J., Di Bello M., Boer A.: The LKIF Core Ontology of Basic Legal Concepts. In: Casanovas P., Biasiotti M.A., Francesconi E., Sagri M.T. (eds.), *Proceedings of LOAIT 2007*, 2007.
- [15] <http://www.akomantoso.org/> naming convention of the URI
- [16] <http://www.ifla.org/publications/functional-requirements-for-bibliographic-records>

- [17] Hu Y., Wu W., Cheng D.: Towards law-aware semantic cloud policies with exceptions for data integration and protection. WIMS 2012: 26, 2012.
- [18] Karam N., Paschke A.: Patent Valuation Using Difference in ALEN. Description Logics 2012, 2012.
- [19] Kelsen H.: *Reine Rechtslehre*, 2d. ed., Wien, 1960.
- [20] Lam H., Governatori G.: The Making of SPINdle. RuleML 2009 proceeding, pp. 315-322, 2009.
- [21] 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.
- [22] Mazzei A., Radicioni D., Brighi R.: NLP-based extraction of modificatory provisions semantics. ICAIL 2009: pp. 50-57, ACM, 2009.
- [23] Mommers L.: Ontologies in the Legal Domain. In: Poli R., Seibt J. (eds.), *Theory and Applications of Ontology: Philosophical Perspectives*, Springer 2010, pp. 265-276, 2010.
- [24] Palmirani M., Brighi R.: An XML Editor for Legal Information Management. Proceeding of the DEXA 2003, Workshop on E-Government, Praga, 1-5 September, pp. 421-429. Springer-Verlag Berlin Heidelberg, 2003.
- [25] Palmirani M., Brighi R.: Model Regularity of Legal Language in Active Modifications. AICOL Workshops 2009: pp. 54-73, Springer, 2009.
- [26] Palmirani M., Cervone L., Vitali F.: Legal metadata interchange framework to match CEN metalex. ICAIL 2009, pp. 232-233, 2009.
- [27] Palmirani M., Cervone L.: Legal Change Management with a Native XML Repository. A cura di G. Governatori. *Legal Knowledge and Information Systems. JURIX 2009. The Twenty-Second Annual Conference*. Rotterdam. 16th-18th December 2009, pp. 146-156, Amsterdam: ISO press, 2009.
- [28] Palmirani M., Contissa G., Rubino R: Fill the Gap in the Legal Knowledge Modelling. In *Proceedings of RuleML 2009*, pp. 305-314, Springer, 2009.
- [29] Palmirani M., Governatori G., Rotolo A., Tabet S., Boley H., Paschke A.: *LegalRuleML: XML-Based Rules and Norms*. RuleML America 2011: 298-312, Springer, 2011.
- [30] Proceeding of the 13th International Conference on Artificial Intelligence and Law, Pittsburgh 6-10 June, 2011, ACM, NY, 2011.
- [31] Sartor G.: Legal Concepts as Inferential Nodes and Ontological Categories. In *Artif. Intell. Law* 17(3) 2009, pp. 217-251, 2009.
- [32] Sartor G.: *Legal Reasoning: A Cognitive Approach to the Law*. Vol. 5. *Treatise on Legal Philosophy and General Jurisprudence*. Berlin: Springer, 2005.
- [33] Sartor G.; Palmirani M.; Francesconi E.; Biasiotti M. (eds.): *Legislative XML for the Semantic Web. Principles, Models, Standards for Document Management*, Dordrecht/Heidelberg/London/New York, Springer, 2011, Law, Governance and Technology Series, Vol. 4., 2011.
- [34] Sottara D., Mello P., and Proctor M.: A configurable rete-oo engine for reasoning with different types of imperfect information. *IEEE Trans. Knowl. Data Eng.*, 22(11): pp. 1535-1548, 2010
- [35] Vitali F., Palmirani M.: Akoma Ntoso Release Notes. [<http://www.akomantoso.org>]. Accessed 20 June 2012.
- [36] Wilcock G.: *Introduction to Linguistic Annotation and Text Analytics* Morgan & Claypool Publishers 2009.

Annex

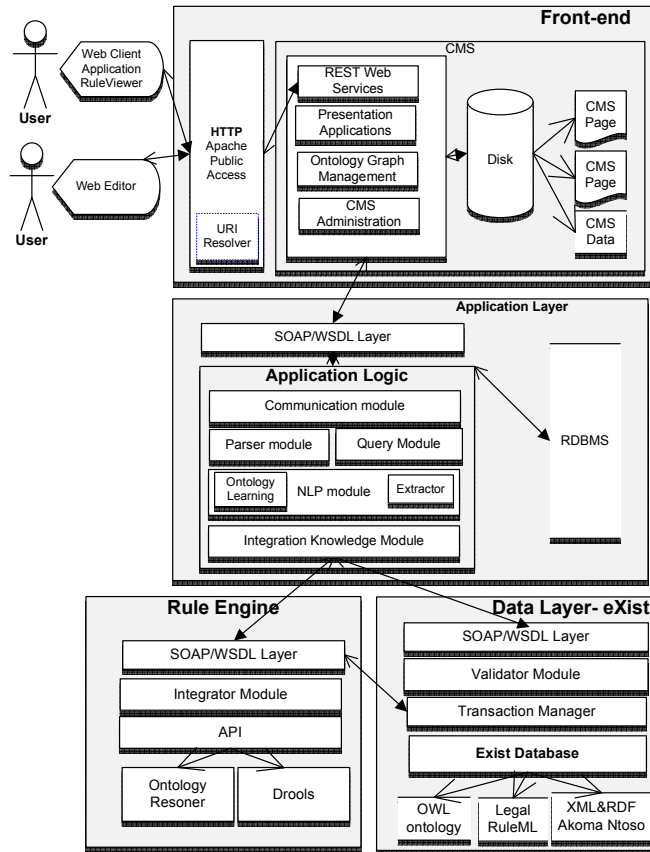


Fig. 13 – System Architecture