Combining Ontologies and Rules to Model Judicial Interpretation

Marcello Ceci[1]
[1] CIRSFID, University of Bologna.
m.ceci@unibo.it

Abstract. This research aims to define an integrated methodology for modelling judgments, starting from legal texts and capturing not only structural parts but also arguments used by judges to reach conclusions. The goal is to build a complete ontology framework capable of detecting and modelling knowledge from the judgement's text. The formalized judgements provide the necessary metadata for the rule layer to enable argumentation towards the acceptance or rejection of a given interpretation. For pursuing this goal it is important to integrate the legal ontology construction with a rule formalization following legal reasoning-oriented theory and defeasible logics, just like the Carneades application (presented here) follows Walton's argumentation theory. The XML interchange uses OWL, RuleML, and the emerging LegalRuleML.

Keywords: OWL, Case-law, Legal Reasoning, LKIF-Rules, LegalRuleML

1 Introduction

Precedent is a main element of legal knowledge worldwide: by settling conflicts and sanctioning illegal behaviours, judicial activity enforces law provisions within the national borders, therefore supporting the validity of laws as well as the sovereignty of the government that issued them. Moreover, precedents (or case-law) are a fundamental source for law interpretation and it paradoxically happens that the exercise of jurisdiction can influence the scope of the same norms it has to apply, both in common law and civil law legal systems – even if to different extents.

The goal of the present research is to define a framework for case-law semantics, exploiting Semantic Web technologies to achieve isomorphism between the text fragment (the only binding legal expression) and the legal rule, thus "filling the gap" between document representation and rules modelling [15]. The formalization of the general structure of case-law, the metadata connected with the judicial legal concepts and the ontology set[1] constitute the basis for a semantic tool enriching the XML mark-up of precedents and supporting legal reasoning [17]. We believe that the features of OWL2 could greatly improve legal concepts modelling and reasoning [4], once properly combined with rule modelling. Our aim is hence to formalize the legal concepts and the argumentation patterns contained in the judgment in order to check,

[1] An ontology is a shared vocabulary, a taxonomy and axioms representing a domain of knowledge by defining objects and concepts with their properties, relations and semantics.

validate and reuse the elements of judgement as expressed by the text and the argumentation contained in it. To achieve this, four models are necessary:

- a *document metadata structure*, capturing the main parts of the judgment to create a bridge between text and semantic annotation of legal concepts;
- a *legal core ontology*, describing the legal domain's main elements in terms of general concepts through an LKIF-Core extension;
- a *legal domain ontology*, modelling the legal concepts of a specific legal domain concerned by the case-law, including a set of sample precedents;
- *argumentation modelling and reasoning*, representing the structure and dynamics of argumentation.

2 Research Methodology

This research is based on a middle-out methodology: top-down for modelling the core ontology, bottom-up for modelling the domain ontology and the argumentation rules. The research starts from the analysis of a sample set of 27 decisions of different grade Italian case-law (tribunal, court of appeal, Cassation Court) concerning consumer law[2].

While DL is very powerful and semantically rich (making OWL an acknowledged standard for the modelling of document metadata), monotonic logics are not sufficient for modeling legal rules in a verifiable way, while at the same time maintaining the structure of the legislation and regulation they are ought to represent: legislation is typically organized as general rules subject to exceptions, and arguments made by applying legal rules are often defeasible. Moreover, the application of laws depends on time, and various legal rules may conflict with each other: these conflicts are resolved using legal principles about priority relationships between rules. Metarules such as these, however, need defeasible logics in order to be properly managed.

The research relies on the previous efforts of the community in the field of legal knowledge representation [2] and rule interchange for applications in the legal domain [9]. The issue of implementing logics to represent judicial interpretation has already been faced [1], albeit only for the purposes of an sample case. The aim of the present research is to apply these theories to a set of real legal documents, stressing the OWL axioms definitions as much as possible in order to enable them to provide a semantically powerful representation of the legal document and a solid ground for an argumentation system using a defeasible subset of predicate logics.

3 The Ontology Set

The Core Ontology (an extension of the LKIF-Core Ontology) introduces a model of the legal domain whose main elements are: legal rules, legal concepts, material circumstances, judicial claims, judicial interpretations, adjudications[3].

---

[2] The matter is specifically disciplined in Italy through the "Codice del Consumo" (Consumer Law) and articles 1341-1342 of the Civil Code. This discipline is also present in all foreign legal systems, which will allow an extension of the research to foreign decisions and laws.

[3] Even though the core ontology should be domain-generic and not modeled upon a specific legal subject, the sample model was conceived only to successfully represent the interaction in the civil law subject, when contracts, laws and judicial decisions come into play.

Following this structure, the metadata taken from judicial documents are represented in the Domain Ontology. The modeling was carried out manually by an expert in the legal subject, which actually represents the only viable choice in the legal domain: automatic information retrieval and machine learning techniques, in fact, do not yet ensure a sufficient level of accuracy, even if some progress in the field has been made, for example in applying NLP techniques to recognize law modifications [14]. Building a legal domain ontology is similar to writing a piece of legal doctrine, thus it should be manually achieved in such a way as to maintain a reference to the author of the model, following an open approach.

Such a layered ontology creates an environment where the knowledge extracted from the decision's text can be processed and managed, in such a way as to enable a deeper reasoning on the interpretation instances grounding the decision itself. Example of this deeper reasoning include: finding relevant precedents which were not explicitly cited in the decision; validating the adjudications of the judge on the claims brought forward by the parties during the trial on the basis of applicable rules, accepted evidence and interpretation; suggesting legal rules/precedents /circumstances that could bring to a different adjudication of the claim. The layered structure of the ontology set also allows an efficacious scaling from legal concepts to factors, up to dimensions and legal principles: all these concepts can be represented in the domain ontology, and the hook of the core concepts to the LKIF taxonomy should allow semantic alignment between domain ontologies when different authors are concerned (yet, the ontology set alignment has not been tested by the present research).

4 Legal Argumentation Modeling

The last part of the research relies on argumentation modeling to perform legal reasoning on the knowledge contained in the ontologies, in order to evaluate their capabilities. As a first test of the ontology set, the research developed a pilot case using the Carneades Argumentation System[4] [5] in order to verify the correctness of the OWL representation of precedents and to test how far defeasible rules can simulate judicial reasoning. Carneades implements Walton's argumentation schemes [10] to reconstruct and evaluate past arguments in natural language texts, but also as templates for manual generation of arguments graphs representing ongoing dialogues. It can therefore be used for studying argumentation from a computational perspective, but also to develop tools supporting practical argumentation processes. It is capable of importing knowledge from the ontology set [8] and of applying rules on them [7]. The new version of Carneades (2.x, under development) uses the Clojure language, while the latest complete version (1.0.2) relies on the LKIF-Rule language [2].

4.1. The implementation

The present research developed applications [3] for Carneades in both LKIF-Rules and Clojure languages. Implementing Carneades involved the following interventions:

- enriching the semantic content of the ontology set by representing finer-grained knowledge contained in the decision's text, in an environment which does not overload the OWL reasoners, compromising computability;

[4] Carneades is a set of open source software tools for mapping and evaluating arguments, under development since 2006.

- modeling a rule system representing the dynamic relationships created by judicial interpretation and law application;
- importing knowledge from the ontology set in such a way as to allow successful interaction with the rule set and the Carneades model.

4.2. Results and Issues

The so-built system is capable of creating argumentation graphs pro or con a given legal statement: for example, it can state whether a contract clause can be judged as inefficacious, under which norms, and following which judicial precedents and judicial interpretations. These arguments are brought forward by the system not only when all the premises for the argument are accepted in the knowledge base: Carneades is in fact capable of "suggesting" incomplete arguments [3], thus highlighting critical aspects of the case which have not been taken into consideration by the judge (in the precedent case) or by the user (in the query).

As a result of their application to the ontology set, the LKIF-Rule and Clojure rule languages showed to be unsatisfactory in:

- identifying a border between semantic representation and syntactic modeling;
- importing ontologies: in the Carneades application no distinction is made between stated and inferred knowledge, and no feedback of new knowledge into the ontology reasoner is possible;
- providing basic legal deontics operators to represent obligation, violation, reparation and penalty;
- providing defeasibility logic operators defining the hierarchy between rules;
- modeling temporal dimensions to represent the three axes of enforceability, efficacy, applicability;
- assigning IDs for single parts of the rules, necessary to reach full isomorphism between the rule and the source legal document(s);
- qualifying rules with metadata such as author, data of creation of the rule, jurisdiction of the rule, etc.

4.3. The Boundary Between Ontology and Rules.

The critical point in importing ontologies, adding factors and writing rules was the design and management of information between the ontologies and the rules: some of the axioms already modeled in the ontologies, in fact, could better meet their potentialities if modeled as an LKIF or Clojure rules instead. The issue, anyway, should be solved with general criteria, since the two systems use different logics (description logic for OWL vs. first-order predicate logic for LKIF and Clojure). This suggests the distinction between static information (thesauri, taxonomies, administrative and procedural data) to be included in the ontology, and legal concepts (legal statuses, subsumptions, inclusion of a material circumstance into the scope of a norm) to be modeled as rules for argument evaluation.

To manage defeasibility of arguments (and thus rules) Carneades includes proof standards [6], which can in a way be interpreted as a kind of priority relation in defeasible logics [11]. The new Carneades 2.x also includes, in its Clojure-based rule system, the metadata block <strict> which allows to specify (through the values true/false) if a <scheme> is either strict of defeasible. However, this block does not ensure full expressivity of the defeasible logics constituents, since it does not represent neither defeaters nor metarules.

5 A New Approach: LegalRuleML

The situation presented above is likely to present itself over and over again as long as modeling decisions (i.e. the introduction of elements of defeasible logics) are taken just for the purposes of a single application. In order to fully exploit its potentialities, AI&Law systems (such as legal argumentation systems) should instead rely on open and shared standards. The research community joined the efforts towards the definition of a standard for the syntax of legal rule extending LKIF-Rules with a modeling of temporal parameters, giving birth to the LKIF++ language [16]. Soon realizing that a standard in syntactic representation of norms would require a shared rule language to be built from scratch, the OASIS consortium started the development of a brand new syntax for legal rules, explicitly relying on the acquired standards in the underlying layers of the Semantic Web cake.

5.1. RuleML and LegalRuleML

LegalRuleML [18] is an extension of RuleML, an XML based language for the representation of legal rules using formal semantics [12]. LegalRuleML introduces features which are fundamental for modeling legal rules: isomorphism, defeasible logics, jurisdiction and authority, legal temporal parameters, legal deontics operators, qualifications, semantic of negation, behaviors. The language also allows to include elements and statements compliant with external ontologies. The syntax and structure of this language are a work in progress: therefore all tags, elements, attributes recalled here follow a syntax proposal presented mostly by Palmirani[5] in the TC, and some of them may not correspond to the definitive syntax of LegalRuleML[6]. The requirements expressed are nevertheless important for tackling some of the problems encountered using LKIF-Rules and Carneades.

5.2. Achieving isomorphism

Modelling the legal rules in the LegalRuleML language highlighted the potentialities of this tool in achieving isomorphism and representing defeasibility and temporal parameters. In the present proposal, the <ruleInfo> section introduces detailed information on the context of the rule. This rule-centric metadata approach favours the isomorphism with the legal text during the change management and the encapsulation of all information related to the rule in a unique XML node:

```
<lrml:ruleInfo id="ruleInfo2" appliesTo="#rule2">
   <lrml:sources id="sourceBlock2">
      <lrml:source element="#atom1" idRef="#art1341-com2"/>
      <lrml:source element="#atom2" idRef="#art1341-com1"/>
      <lrml:source element="#atom3" idRef="#art1341-com1"/>
```

[5] General contents of the proposal can be found at https://www.oasis-open.org/apps/org/workgroup/legalruleml/download.php/46379/1.2PRINCIPLES.004.doc.

[6] 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/.

```
      <lrml:source element="#atom4" idRef="#art1341com2"/>
    </lrml:sources>
    <lrml:strength iri="&dfsont;defeater"/>
    <lrml:jurisdiction iri="&jurisdictions;italy"/>
    <lrml:author idRef="#aut1"/>
    <lrml:times idRef="#t1"/>
    <lrml:creationDateTime idRef="#e1"/>
  </lrml:ruleInfo>
```
[7]

This section represents a context data container, since it introduces metadata which can be used in multiple circumstances to classify the rules. The bond between the text fragment, the legal rule and the author of the model is ensured by a list of `<sources>` for every element which constitutes the rule and by the element `<author/>`, linking the rule and its parts to specific individuals (a fragment of text and a person respectively), identified through an IRI. In this way it is possible to explicitly refer to the source documents of each part of a rule, also when a rule takes origin from multiple documental sources, allowing a clear distinction of which part of the rule comes from which document. At the same time, different rule authors are allowed to model the same text fragment in different ways, being always clear which author modelled which rules on a certain legal document.

5.3. Introducing Defeasible Logics and Temporal Parameters to the Rule

Inside the `<ruleInfo>` section, the element `<strength/>` defines the role of each rule in the defeasible logics dynamics (strict/defeasible/defeater). Priority relations are built through the `<Overrides>` element[8], in the following form:

```
<Overrides id="ovr1">
   <Rule keyref="#rule_3"/>
   <Rule keyref="#rule_2"/>
</Overrides>
```
[9]

The `<ruleInfo>` section contains also a `<times>` element, not indeed a normal attribute but rather a section introducing a whole different layer: it contains information on the time periods of the legal rule's coming into force, efficacy, application. The representation of temporal dimension of legal rules using three axes is a crucial addition towards the automatic management of legal rules in connection

[7] The proposal can be found at http://www.oasis-open.org/apps/org/workgroup/legalruleml/download.php/45887/2.8isomorphism.001.doc.

[8] This tag is developed jointly with RuleML (http://ruleml.org/) and in particular with the Defeasibility RuleML TG (http://ruleml.org/1.0/defeasible.html).

[9] This implementation of defeasibility should allow a better management of exceptions than in LKIF-Rule, where exceptions had to be made explicit in the rule syntax. If an exception presents itself in the form of two legal fragments not explicitly referring to each other but rather disposing opposing legal consequences (i.e. efficacy vs. inefficacy) it is possible to model these rules independently, and then create a priority relation reflecting the actual hierarchy between the two norms. Moreover, this solution allows a relative management of hierarchy, without the need to assign arbitrary "weight" values to each rule. The proposal can be found at https://www.oasis-open.org/committees/download.php/46454/2.1.1defeasibility.006.doc. Meaningful comments by TC member Tara Athan can be found at http://www.oasis-open.org/apps/org/workgroup/legalruleml/download.php/45888/2.1defeasibility.002.002.doc.

with their legally binding documents [16], and LegalRuleML allows to specify these time coordinates for each rule, starting from the identification of the relevant points in time through a list of `<events>` such as:

```
<lrml:event id="e2" value="1942-04-21T01:01:00.0Z"/>.
```

Events' IDS are recalled by the `<timeBlock>` element, which adds information on the event which occurs (`start`, `end`) and on the axis which is affected:

```
<lrml:timeBlock id="t1">
    <lrml:time start="#e2" refType="&lkif;#efficacy"/>
</lrml:timeBlock>
```
[10]

6 Conclusions

The project presented here represents an effort towards the acquisition of an acknowledged standard for the rule layer of the semantic web layer cake, while at the same time trying to improve the state-of-the-art of legal knowledge representation by facing its main issues: the gap between document representation and rule modeling, and the need for a shared standard in the logic layer to represent legal reasoning. The Carneades Argumentation System has been used as a test field for the ontology set and the rules design, highlighting potentialities and issues of the approach.

Under these points of view, the approach proposed by the research represents a step towards the filling of that gap, relying on existing standards to achieve the isomorphism between legal document and rules; at the same time, the research defines the requirements for a reasoning engine to be capable of semantically managing knowledge coming from the ontology and applying legal rules to it. This engine will probably not be a closed one, embedded in some argumentation tool (as in Carneades): It should rather consist of a set of libraries to be implemented into existing engines in order to introduce a complete management of defeasibility and a standard language for interaction between these rules and OWL-encoded knowledge. The intention, in the upcoming research on this behalf, is to rely on a Drools[11] application under construction by CIRSFID and on NICTA's SPINdle[12] [13].

Acknowledgements

A particular acknowledgement is due to Tom Gordon, provider of meaningful research inputs during my stage in Berlin, FOKUS, in the last autumn. Thanks also to Adam Wyner who gave me important feedbacks for improving the readability of the paper and also, as a legal domain expert, fruitful comments on the ontology modeling choices. Thanks go also to Harold Boley and Tara Athan of the RuleML Initiative for his helpful hints. Last but not least, I am grateful to my tutor and mentor Monica Palmirani, for her guidance in this challenging research.

---

[10] The proposal can be found at http://markmail.org/message/oy34tkzzr3r2ldhz?q=temporal+list:org%2Eoasis-open+list:org%2Exml+list:org%2Eebxml.

[11] Drools (www.drools.org) is a production rule system based on the Rete algorithm.

[12] SPINdle is an open source, Java-based defeasible logic reasoner which conducts efficient and scalable reasoning on defeasible logic theories.

References

1. Boella G., Governatori G., Rotolo A., van der Torre L., A Logical Understanding of Legal Interpretation. In: KR 2010.
2. 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.
3. Ceci, M., Gordon, T.: Browsing Case-Law: An Application of the Carneades Argumentation System, RuleML Challenge 2012 (under submission).
4. Gangemi, A.: Design Patterns for Legal Ontology Construction. In: Noriega, P., Bourcier, D., Galindo, F. (eds.), Trends in Legal Knowledge. The Semantic Web and the Regulation of Electronic Social Systems, pp. 171-191. European Press Academic Publishing (2007)
5. Gordon, T., Walton, D.: The Carneades Argumentation Framework: using presumptions and exceptions to model critical questions. In: Dunne, P.E.: Computational Models of Argument. Proceedings of COMMA 2006: 1st International Conference on Computational Models of Argument, The University of Liverpool, UK, 11th - 12th September 2006. IOS Press, Amsterdam (2006)
6. Gordon, T., Prakken, H., Walton, D.: The Carneades model of argument and burden of proof. In: Artificial Intelligence, Vol.171 (2007), No.10-15, pp.875-896.
7. Gordon, T.: Constructing Legal Arguments with Rules in the Legal Knowledge Interchange Format (LKIF). In: Casanovas, P.: Computable models of the law: Languages, dialogues, games, ontologies (Lecture Notes in Artificial Intelligence 4884), pp. 162-184. Springer, Heidelberg (2008)
8. Gordon, T.: Combining Rules and Ontologies with Carneades. In: Proceedings of the 5th International RuleML2011@BRF Challenge, CEUR Workshop Proceedings (2011), 103-110.
9. Gordon, T., Governatori, G., Rotolo, A., Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain. In: Rule Interchange and Applications, International Symposium, RuleML 2009, BERLIN, Springer pp. 282 - 296 (2009).
10. Gordon, T., Walton, D., Legal reasoning with argumentation schemes. Ind: Proceedings of the Twelfth International Conference on Artificial Intelligence and Law, ACM Press, New York, 2009, pp. 137-146.
11. Governatori, G.: On the relationship between Carneades and Defeasible logic. In: ICAIL '11 Proceedings of the 13th International Conference on Artificial Intelligence and Law, ACM New York (2011)
12. Lee, J.K., Sohn, M.M.: The eXtensible Rule Markup Language, Communications of the ACM, Volume 46, Issue 5, pp. 59-64 (2003)
13. Lam, H.P., Governatori, G.: The making of SPINdle. In: Governatori, G., Hall, J., Paschke, A. (eds.): RuleML 2009. LNCS 5858, 315–322.Springer, Berlin (2009)
14. Palmirani, M., Brighi, R., Model Regularity of Legal Language in Active Modifications. In: LNCS 6237/2010, pp. 54-73.
15. Palmirani, M., Contissa, G., Rubino, R.: Fill the Gap in the Legal Knowledge Modelling. In: Proceedings of RuleML 2009, pp. 305-314.
16. Palmirani, M., Governatori, G., Contissa, G.: Temporal Dimensions in Rules Modelling. In: JURIX 2010, pp.159-162.
17. Palmirani, M., Ceci, M.: Ontology framework for judgement modelling. In: AICOL 2011 Proceedings. Springer, 2012 (under publication).
18. Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley, H., Paschke, A.: LegalRuleML: XML-Based Rules and Norms. In: RuleML 2011, pp. 298-312.