Legal Knowledge and Information Systems
M. Palmirani (Ed.)

Modelling Legal Knowledge for GDPR Compliance Checking

Monica PALMIRANI, Guido GOVERNATORI

[1] *CIRSFID, University of Bologna.*
monica.palmirani@unibo.it
[2] *Data61, CSIRO, Australia.*
guido.governatori@data61.csiro.au

Abstract. In the last fifteen years, Semantic Web technologies have been successfully applied to the legal domain. By composing all those techniques and theoretical methods, we propose an integrated framework for modelling legal documents and legal knowledge to support legal reasoning, in particular checking compliance. This paper presents a proof-of-concept applied to the GDPR domain, with the aim to detect infringements of privacy compulsory norms or to prevent possible violations using BPMN and Regorous engine.

Keywords. LegalRuleML, Akoma Ntoso, Legal Ontology, Legal Compliance Checking, GDPR Rule Modelling

1. Introduction

Over the last fifteen years, Semantic Web technologies have been successfully applied to the legal domain by defining unique identifier naming conventions for legal resources (e.g., ELI, ECLI, URN:LEX) [1], legal document vocabularies for the representation of sources (e.g., Metalex/CEN, Akoma Ntoso[2][11][15]), legal ontologies for modelling legal concepts (e.g., LKIF ontology,[3] PrOnto for GDPR[4][14]), and legal rule-based languages for modelling norms (e.g., LegalRuleML[5][1][2]). However, such components, and the related research communities, are not integrated enough to produce a robust and scientific framework that can be usable in real applications and that takes the needs of end users into account. By composing all these techniques and theoretical methods, we propose an integrated framework for modelling legal documents and legal knowledge to support legal reasoning, and in particular to check for compliance. This paper presents a proof-of-concept of this framework carried out in the Cloud for Europe (C4E) European project, where these techniques (Akoma Ntoso, PrOnto, LegalRuleML) have been applied to the GDPR domain with the aim of

[1] ELI—European Legislation Identifier https://eur-lex.europa.eu/eli-register/about.html; ECLI—European Case Law Identifier https://e-justice.europa.eu/content_european_case_law_identifier_ecli-175-en.do; URN:LEX—https://datatracker.ietf.org/doc/draft-spinosa-urn-lex/
[2] Metalex/CEN—http://www.metalex.eu/; Akoma Ntoso—Architecture for Knowledge-Oriented Management of Any Normative Texts using Open Standards and Ontologies https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=legaldocml
[3] LKIF—Legal Knowledge Interchange Format, https://github.com/RinkeHoekstra/lkif-core
[4] GDPR–General Data Protection Regulation (EU) 2016/679.
[5] LegalRuleML—https://www.oasis-open.org/committees/legalruleml/

detecting infringements of privacy rules (*ex-post analysis*) or of preventing possible violations (*ex-ante analysis*) using BPMN[6] [7] and the Regorous engine [9].

2. The Cloud for Europe Project

The Cloud for Europe project (C4E)[7] aims to design a cloud computing platform for eGovernment sevices compliant with GDPR rules. Cloud computing gives rise to several legal issues related to data protection rules: jurisdiction defines which legal system is applicable (e.g., in Germany, Section 130 of the Criminal Code bans Nazi symbolism, but no such ban exists in the USA); the geographic location of servers defines special rules (e.g., Cross-Border Data Transfer, Arts. 44–50 GDPR); security attacks in a cloud computing environment could cause multiple data breaches affecting different servers and consequently different data processors.

The solution we propose is an innovative architecture for managing legal compliance checking for public-sector cloud-computing network services. The GDPR includes several provisions that have a significant impact on this domain:

i) It defines constraints that must be included *by design* as part of information-system specifications and implementation (e.g., obligations, rights, permissions, prohibitions, penalties, remedies).

ii) It shapes policies (e.g., security, privacy) and business processes (e.g., the nodes of the brokers admitted for the transmission of data out of EU borders) that affect processing workflow.

iii) It changes over time, and this produces dynamic situations where a prompt reaction is fundamental (e.g., data breach, notification). A classic static rule engine is not enough to achieve a feasible legal-compliance-checking environment: it is necessary to include a defeasible, deontic, and temporal logic model connected with the original legal texts if (evidence based) reports are to be produced that can justify the outcomes of compliance-checking activities.

Our integrated framework prototypes (see Fig. 1) are capable of managing four main crucial functional requirements in legal compliance checking:

1. **Managing changes** made to the legal document over time, especially as this applies to acts, regulations, and contracts, which by nature are variable and subject to frequent change, significantly affecting coordination between the text and the rules that need to be remodeled. Our framework uses two different specialized Web editors integrated in a dashboard with a harmonized design interface: one for marking up the legal text in Akoma Ntoso, the other for modelling legal norms in LegalRuleML. The legal rules, the texts, and the legal ontologies are connected with one another other via FRBR (Functional Requirements for Bibliographic Records). Native NoSQL and XML databases store the legal sources marked up in Akoma Ntoso and LegalRuleML. The application level (server and client side) is able to maintain the legislative repositories updated (*point-in-time*) and to discover new pertinent law available in the Web. A legal temporal model is implemented in an application and data model based on three main parameters: a norm's time of entry into force, its time of efficacy, and its time of application to a specific case. This temporal model is extensively used in a coordinated manner in Akoma Ntoso and in LegalRuleML XML standards. It ensures that all legal

[6] BPMN— Business Process Model and Notation, http://www.bpmn.org/

[7] https://www.fokus.fraunhofer.de/en/dps/projects/cloudforeurope

rules affected by a legislative change are detected. In particular, if a legislative text changes, we can also detect the business processes that have been affected or are missing, so as to make it possible to promptly update the entire system.

2. **Modelling Legal concepts** using the PrOnto ontology for the data protection legal domain, and in particular for the GDPR.

3. The **legal reasoning** component uses the legal sources previously marked up using the Web editors. The legal reasoning engine is based on defeasible and temporal logic specific to the pertinent legal domain, and it is also scalable and computable with the relevant volume of rules [9]. It ensures legal compliance checking by means of a specific algorithm making it possible to answer queries submitted by cloud service providers or by the national service brokerage infrastructure. When a fact or a service is required, the cloud computing infrastructure asks the legal reasoning engine to verify the legality of the operation using, among the other resources, the contract's general provisions, the relevant case law, and soft-law policies. The result of legal reasoning is a report detailing violation [3], remedies, and possible alternative solutions that need to be interpreted by decision-makers (cloud actors).

4. **Business process** integration with legal reasoning is necessary in order to guarantee the correct application of technical operations, events, and processes connected with cloud computing services. To that end we have a special editor for modelling business processes using BPMN 2.0. This module is invoked in each legal action to determine whether the legal rules are also consistent with the real applicative scenario. Legal reasoning is also invoked when a law changes and the need arises to check whether business processes are still compliant with the new modified legislative scenario. If not, an alert is sent to the business process designer to update the workflow component that works with the cloud computing service platform and with the content management system.

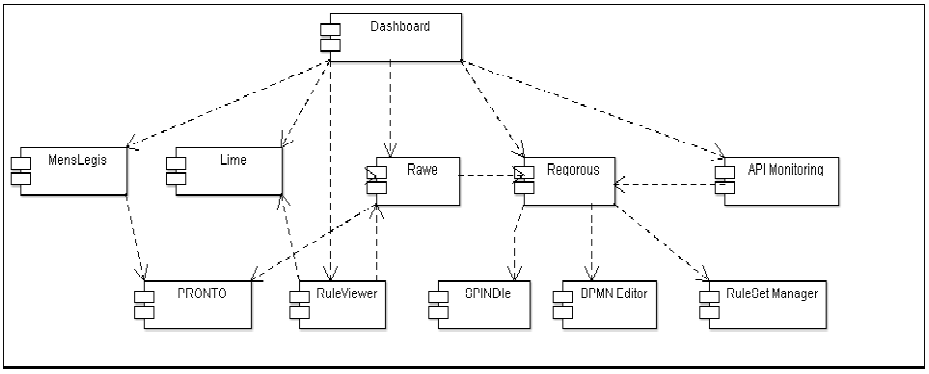

Figure 1 - Legal Tools Architecture

The architecture (Fig. 1)[8] was implemented and a prototype was tested under the Cloud 4 Europe project. Three legal experts manually modelled the rules and checked the BPMN. In our framework, presented below, we find LIME and RAWE [12], which are two web editors (JavaScript) capable of semi-automatically marking up the text in Akoma Ntoso and the manually formalized norms in LegalRuleML. PrOnto is a legal ontology for modelling GDPR concepts and axioms. It feeds concepts and predicates to the legal rule-modelling layer in order to make the formalization consistent and

[8] http://sinatra.cirsfid.unibo.it/c4eu-dashboard/

harmonized. Regorous [9] is a tool (written in Java) that makes it possible to design BPMN 2.0 and to connect each step of the process with the legal rules. Regorous provides an API to SPINdle [10], a defeasible legal reasoning engine. Regorous presents at the end the results of compliance checking in a user interface for the end user.

3. Use-Case Scenario

A student wants to access an online service provided by a public school platform in cloud computing. The platform provides an online environment where student and parents can access grades, information, administrative communications, and courseware. Students can also upload their material connected with training activities so as to share it with other students and teachers. Additionally, the platform includes a chatline with specialized school staff (e.g., psychology counselling, health service). Art. 8 GDPR reads as follows:

"Article 8 - Conditions applicable to child's consent in relation to information society services

1. Where point (a) of Article 6(1) applies, in relation to the offer of information society services directly to a child, the processing of the personal data of a child shall be lawful where the child is at least 16 years old. Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental responsibility over the child. Member States may provide by law for a lower age for those purposes provided that such lower age is not below 13 years."

In order to access to the service/platform the student must (i) agree to the general service conditions for authentication and (ii) provide the consent for the controller's processing of personal data (Art. 4 GDPR), including sensitive data (Art. 6 GDPR).
The BPMN modelling of the process above is illustrated in Fig. 2 below.

4. Legal Knowledge Modelling Framework

The lifecycle of legal-knowledge management starts by modelling Art. 8 GDPR in Akoma Ntoso so as to describe the structure of the provisions, the normative references, and the legal concepts using the PrOnto ontology and also the temporal parameters (e.g., entry into force). After this step the rules are modelled and connected with the BPMN. Finally, the Regorous engine provides the result of compliance checking.

4.1. LegalRuleML Metamodel Extension

The current LegaRuleML metamodel is very elementary and intended to design LegalRuleML constructs. However, it is a good starting point for developing extensions suitable for other goals, like compliance checking. Table 1 shows an extension of the LegalRueML metamodel included in the PrOnto ontology. It includes relationships between deontic operators, disjoint-class axioms, and better modelling of remedies, violations, and penalties.

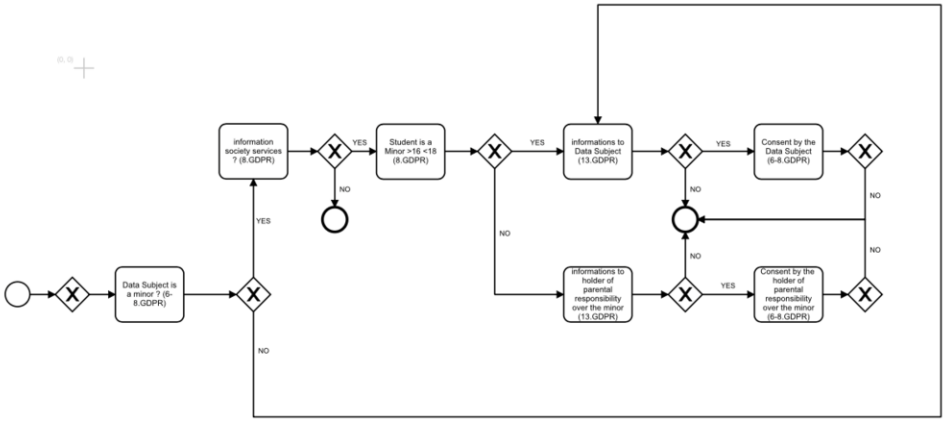

Figure 2 – BPMN modelling of an eGovernment service

In particular, some predicates and axioms are included in PrOnto for modelling deontic relationship like `repairs`, which connects a PenaltyStatement with a LogicalFormulaStatement. Another example is the restriction applied to the `generates` property in order to model the Obligation generated by Right for the AuxiliaryParty. In LegalRuleML we have Reparation and PenaltyStatement, where Remedy links a PenaltyStatement with a PrescriptiveStatement, and the PenaltyStatement is a Deontic Specification.

Table 1. Some axioms in the extension of the LegalRuleML ontology

Reparation <owl:ObjectProperty rdf:about="https://w3id.org/ontology/pronto#repairs"> <rdfs:domain rdf:resource="http://docs.oasis-open.org/legalruleml/ns/v1.0/metamodel#PenaltyStatement"/> <rdfs:range rdf:resource="http://docs.oasis-open.org/legalruleml/ns/v1.0/metamodel#PrescriptiveStatement"/> </owl:ObjectProperty>
Restriction: Obligation *hasHeld* CounterParty generated by a Right <owl:ObjectProperty rdf:about="https://w3id.org/ontology/pronto#generates"> <rdfs:domain rdf:resource="http://docs.oasis-open.org/legalruleml/ns/v1.0/metamodel#Right"/> <rdfs:range> <owl:intersectionOf rdf:parseType="Collection"> <owl:Class rdf:about="http://docs.oasis-open.org/legalruleml/ns/v1.0/metamodel#Obligation"/> <owl:Restriction> <owl:onProperty rdf:resource="https://w3id.org/ontology/pronto#hasHeld"/> <owl:someValuesFrom rdf:resource="http://docs.oasis-open.org/legalruleml/ns/v1.0/metamodel #AuxiliaryParty"/> </owl:Restriction> </owl:intersectionOf> </rdfs:range> </owl:ObjectProperty>

In the scenario of Art. 8 GDPR, we have the following OWL-DL:

Table 2. Legal Axioms

Legal concepts OWL-DL Axioms	Individual(PrOnto:child type(PrOnto:person)) Individual(PrOnto:child type(PrOnto:person)) SubClassOf(PrOnto:data_controller PrOnto:data) SubClassOf(PrOnto: personal_data_processing PrOnto: process) Individual(PrOnto: information_society_service PrOnto:process) Individual(PrOnto: obligation_to_obtain_consent PrOnto:obligation) ObjectProperty(PrOnto:has_at_least16years domain(PrOnto:child) range(PRONTO:status))

4.2. Legal Rule Modelling: LegalRuleML Formalization

Art. 8 is modelled using the RAWE graphic tool implemented using Scratch diagrams[9] (see Fig. 3) to help legal experts approach logic formalization. The idea is that the following logic rule is directly modelled using visual diagrams even in order to properly connect the PrOnto legal ontology:

Table 3. Logic rule modelling

Logic rule modelling	IF *personal_data_processing*(d,x) ∧ *child*(x) ∧ *at_least16years*(x) ∧ *information_society_service*(s,d) ∧ *data_controller*(y,s) THEN *obligation_to_obtain_consent*(y,x,s)

Thanks to the official legal text in a single window, we can also model the legal rules in LegalRuleML connected to the ontology terms previously marked up in the text (see Fig. 4). Secondly, the XML id attribute connects the rules with the original legal official text. This helps to detect the rules that need to be updated when a legal text changes. LegalRuleML manages temporal defeasible logic to detect the correct set of rules point-in-time. The Art. 8 GDPR admits being trumped by domestic regulation.[10] At present in Europe different age limitations are in place (e.g, age 13 in Spain; 14 in Italy;[11] 15 in France).[12] LegalRuleML makes it possible to use defeasible operators (e.g., <lrml:appliesStrength iri="lrmlv:Defeasible"/>), implementing hierarchies between rules, jointly with metadata that tracks jurisdictions [2] (e.g., <lrml:appliesJurisdiction keyref="jurisdictions:it"/>).

4.3 Checking for Compliance

The LegalRuleML representation of norms is imported in Regorous via an API. Thus, the RAWE LegalRuleML is also imported in the Regorous editor with the corresponding BPMN previously designed by the person in charge of the eGov service. A legal expert annotates the BPMN tasks with the terms present in the LegalRuleML

[9] https://scratch.mit.edu/

[10] "Member States may provide by law for a lower age for those purposes provided that such lower age is not below 13 years."

[11] Legislative Decree n. 101 of 10 August 2018.
http://www.gazzettaufficiale.it/atto/serie_generale/caricaDettaglioAtto/originario?atto.dataPubblicazioneGazzetta=2018-09-04&atto.codiceRedazionale=18G00129&elenco30giorni=false

[12] https://www.betterinternetforkids.eu/web/portal/practice/awareness/detail?articleId=3017751

rules. The ex-ante checking is guaranteed on the basis of the BPMN. The live monitoring is guaranteed using the flow of data coming from the eGov service. Using these log files as facts, we can check the operations' compliance with the rules in Art. 8 GDPR.

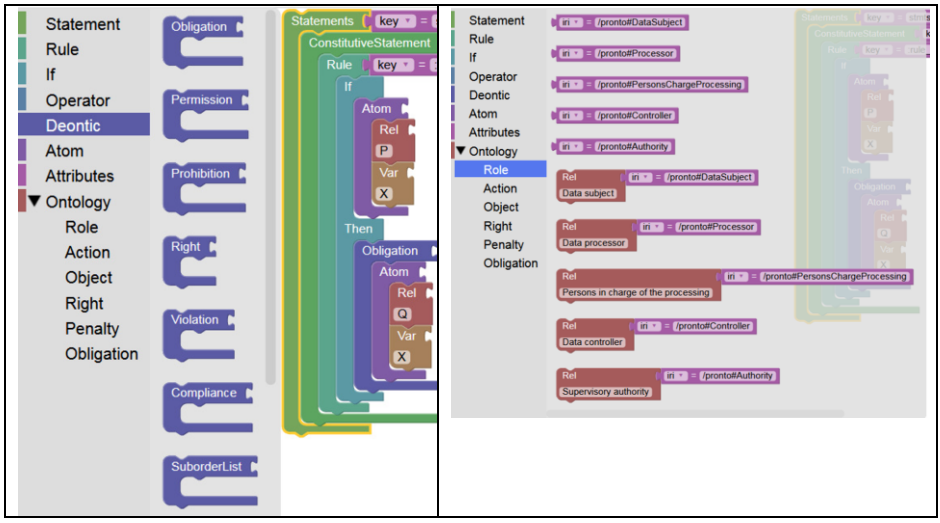

Figure 3 – RAWE web editor for modelling legal rules using graphic diagrams

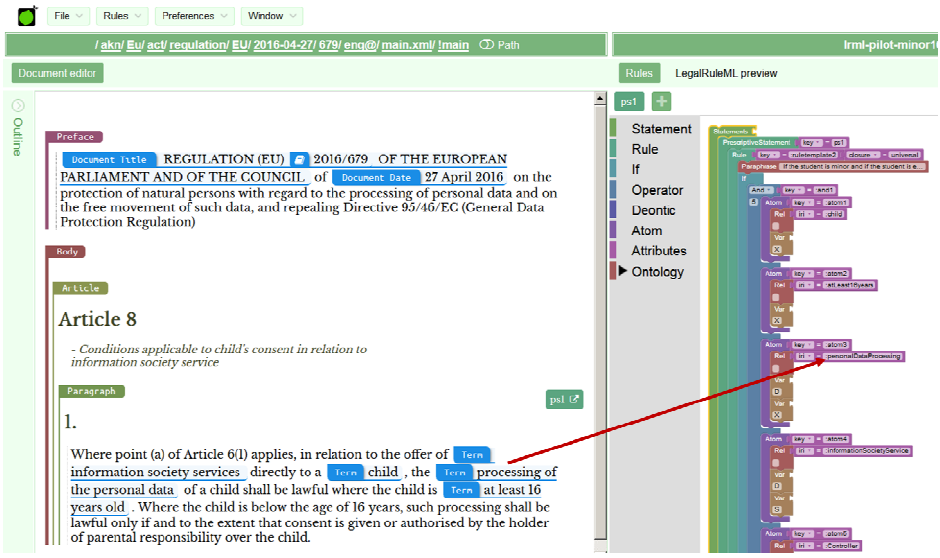

Figure 4 – RAWE web editor for modelling the connection with text, ontology concepts, and rules

The Regorous [9] mechanism for managing compliance checking is as follows: for ex-ante (design time) compliance checking, Regorous dynamically generates the traces in the process (corresponding to the possible ways in which the process can be executed), and for every step it computes the state of the process after executing the corresponding task, so as to then make a call to SPINdle [10] to determine the obligations,

prohibitions, and permissions in force at that particular step. It then compares the state with the legal requirements in force to determine whether they have been fulfilled or violated, or whether the status is still pending. It repeats the procedure for all tasks in the trace, and then for the full set of traces. At the end it reports the compliance status according to the following conditions:

- A trace is compliant if no task in it results in a violation.
- A trace is weakly compliant if every violation is compensated for.
- A process is (weakly) compliant if, and only if, all its execution traces are (at least weakly) compliant.
- A process is partly compliant if, and only if, there is at least one compliant trace.

The live monitor (run-time compliance checking) uses the log files generated by the underlying process-execution workflow, and it first extracts the information from it and passes them as facts for each executed task. At this stage, we can use the same mechanism as the design-time procedure, noting that the log for an instance corresponds to a single trace in the process.

5. Related Work

Over the last decade, the problem of modelling legal knowledge has been addressed by different scholars [15], but unfortunately in a patchy manner, without any integrated vision that combines different technologies (e.g., Semantic Web, AI and Law, NLP) into a single usable framework. Several different standards for modelling text and rules arise (RuleML, SWRL, RIF, LKIF, ODRL, etc.), but they are not specific to the task of modelling the complexity of different legal contexts. Unlike any of these projects, our approach (i) connects text with rules for tracking changes over time, (ii) uses a legal reasoning level on top of the ontological layer of the Semantic Web stack, (iii) permits multiple alternative interpretations, and (iv) connects BPMN with the legal reasoning layer. None of Semantic Web previously mentioned languages and technologies are compliant with the guidelines established in [5] [13] for representing legal knowledge and legal reasoning. In addition, these approaches have severe limitations when it comes to modelling legal reasoning, since they do not provide a conceptually sound model of legal reasoning [6]. A good project is [4], but it is not totally integrated with LegalRuleML.

6. Conclusions

We have presented an integrated framework for checking compliance with legal rules, focusing in particular on the GDPR. We have used Akoma Ntoso to model the text, PrOnto to model legal concepts, LegalRuleML to model norms, and Regorous to combine BPMN and facts with the rules expressed in LegalRuleML and to provide the final report. We have provided an integrated user interface and dashboard based on diagrams to help legal engineers align with the legal text as it evolves over time. This framework makes it possible to track the changes a legal text goes through over time, and hence detect the legal rules that need to be updated. From the text it imports constraints on the metadata level: these constraints include temporal information and

jurisdiction. Finally, the legal ontology connected with the text is imported into the legal rules, without redundancy or errors, so as to maintain a coherent taxonomy of predicates in the rule base.

Acknowledgement

This work was supported by Cloud for Europe project no. 610650 FP7-ICT, and was also partly supported by the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreement no. 690974, "MIREL: MIning and REasoning with Legal Texts."

Annex A

X is a "child," D is "personal data, S is the "information society service" and Y is the "controller."

```
<lrml:LegalRuleML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://docs.oasis-open.org/legalruleml/examples/compactified/ex9-alternatives-compact#"
xmlns:lrml="http://docs.oasis-open.org/legalruleml/ns/v1.0/" xmlns:ruleml="http://ruleml.org/spec"
xmlns:rulemlmm="http://ruleml.org/1.0/metamodel#" xml:base="http://docs.oasis-
open.org/legalruleml/examples/compactified/ex9-alternatives-compact"
xsi:schemaLocation="http://docs.oasis-open.org/legalruleml/ns/v1.0/ ./xsd-schema/compact/lrml-
compact.xsd">
omissis
    <lrml:Statements>
        <lrml:PrescriptiveStatement key="ps1">
            <ruleml:Rule key=":ruletemplate2" closure="universal">
                <lrml:Paraphrase> If the student is minor and if the student is emancipated, in any case,
he/she can provide autonomous consent, if it is considered an action of ordinary
administration</lrml:Paraphrase>
                <ruleml:if>
                    <ruleml:And key=":and1">
                        <ruleml:Atom key=":atom1">
                            <ruleml:Rel iri=":child"/>
                            <ruleml:Var>X</ruleml:Var>
                        </ruleml:Atom>
                        <ruleml:Atom key=":atom2">
                            <ruleml:Rel iri=":atLeast16years"/>
                            <ruleml:Var>X</ruleml:Var>
                        </ruleml:Atom>
                        <ruleml:Atom key=":atom3">
                            <ruleml:Rel iri=":personalDataProcessing"/>
                            <ruleml:Var>D</ruleml:Var>
                            <ruleml:Var>X</ruleml:Var>
                        </ruleml:Atom>
                        <ruleml:Atom key=":atom4">
                            <ruleml:Rel iri=":informationSocietyService"/>
                            <ruleml:Var>D</ruleml:Var>
                            <ruleml:Var>S</ruleml:Var>
                        </ruleml:Atom>
                        <ruleml:Atom key=":atom5">
                            <ruleml:Rel iri=":Controller"/>
                            <ruleml:Var>Y</ruleml:Var>
                            <ruleml:Var>S</ruleml:Var>
                        </ruleml:Atom>
                    </ruleml:And>
                </ruleml:if>
                <ruleml:then>
                    <lrml:Obligation iri=":obligation">
                        <ruleml:Atom key=":atom6">
                            <ruleml:Rel iri=":ObtainConsent"/>
```

```
                                    <ruleml:Var>X</ruleml:Var>
                                    <ruleml:Var>Y</ruleml:Var>
                                    <ruleml:Var>S</ruleml:Var>
                                </ruleml:Atom>
                            </lrml:Obligation>
                        </ruleml:then>
                    </ruleml:Rule>
                </lrml:PrescriptiveStatement>
            </lrml:Statements>
        </lrml:LegalRuleML>
```

```
<lrml:hasQualification>
    <lrml:Overrides over="#ps2" under="#ps1"/>
</lrml:hasQualification>
```

References

[1] Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: OASIS LegalRuleML. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law, pp. 3–12, New York (2013)

[2] Athan, T.; Governatori, G.; Palmirani, M.; Paschke, A.; Wyner, A., LegalRuleML: Design Principles and Foundations, in: Reasoning Web. Web Logic Rules 2015, LNCS, vol. 9203, pp. 151–188, Springer, Heidleberg (2015)

[3] Elgammal, A., Türetken, O., van den Heuvel, W.J.: Using patterns for the analysis and resolution of compliance violations. Int. J. Coop. Inf. Syst. 21(1), 31–54 (2012)

[4] Gandon, T.F., Governatori G., Villata S., Normative Requirements as Linked Data, JURIX2017, Amsterdam, IOS, (2017)

[5] Gordon, T.F., Governatori, G., Rotolo, A.: Rules and norms: Requirements for rule interchange languages in the legal domain. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 282–296. Springer, Heidelberg (2009)

[6] Governatori, G., Hashmi, M.: No time for compliance. In: Hallé, S., Mayer, W. (eds.) EDOC 2015, pp. 9–18. IEEE Press (2015)

[7] Governatori, G., Mustafa M., Lam H.-P., Villata S. and Palmirani, M.: Semantic Business Process Regulatory Compliance Checking using LegalRuleML, In Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2016), November, 2016, Bologna, Italy (2016)

[8] Governatori, G.: On the relationship between Carneades and defeasible logic. In: van Engers, T. (ed.) Proceedings of the 13th International Conference on Artificial Intelligence and Law (ICAIL 2011), pp. 31–40. ACM Press (2011)

[9] Governatori, G.: The Regorous approach to process compliance. In 2015 IEEE 19th International Enterprise Distributed Object Computing Workshop, pages 33–40. IEEE Press (2015)

[10] Lam, H.-P., Governatori, G.: The making of SPINdle. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009. LNCS, vol. 5858, pp. 315–322. Springer, Heidelberg (2009)

[11] Palmirani M, Vitali F, Akoma-Ntoso for Legal Documents, in: Legislative XML for the Semantic Web. Principles, Models, Standards for Document Management, BERLIN, Springer Verlag, 2011, pp. 75 – 100.

[12] Palmirani, M., Cervone, L., Bujor, O., Chiappetta, M.: RAWE: an editor for rule markup of legal texts. In: Fodor, P., Roman, D., Anicic, D., Wyner, A., Palmirani, M., Sottara, D., Lévy, F. (eds.) CEUR Workshop Proceedings, Seattle, USA, 11–13 July 2013, vol. 1004, CEUR-WS.org (2013)

[13] Palmirani, M., Governatori, G., Contissa, G.: Temporal dimensions in rules modelling. In: Winkels, R. (ed.) JURIX. Frontiers in Artificial Intelligence and Applications, vol. 223, pp. 159–162. IOS Press, Amsterdam (2010)

[14] Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: PrOnto: Privacy Ontology for Legal Reasoning. EGOVIS2018, 7th International Conference, EGOVIS 2018, Regensburg, Germany, September 3-5, 2018, Proceedings. LNCS 11032, Springer, pp. 139-152 (2018)

[15] Sartor, G., Palmirani, M., Francesconi, E., Biasiotti, M.: Legislative XML on Semantic web. Springer, Heidelberg (2011)