

When Data Stands Before the Law: an Experience Report on Representing Financial Rules in SPARQL

Marcello Ceci¹, Nicolas Sannier¹, Sallam Abualhaija¹, Domenico Bianculli¹ and Michael Halling²

¹*SnT – University of Luxembourg, 29 Avenue John F. Kennedy, L-1855 Luxembourg*

²*FDEF – University of Luxembourg, 6 rue Richard Coudenhove-Kalergi L-1359 Luxembourg*

Abstract

Automated compliance checking of data (including financial data) against applicable law is only possible with a formal representation of complex legal rules. The literature in the fields of legal informatics and Requirements Engineering (RE) can count on decades of contributions to the representation of data models, rule languages, and reasoners for legal application. There is however a representational gap between data and legal norms, which prevents a comprehensive approach to legal knowledge representation, resulting in the lack of a standard solution for legal rules representation. This paper reports on our experience regarding the formal representation of complex financial rules, more specifically, the entirety of Article 43 of the Luxembourgish UCITS Law. We used SPARQL to specify the rules to be used for the automatic validation of a real financial dataset. We provide observations regarding (a) the complexity of the resulting SPARQL queries and how SHACL can help address some of this complexity, and (b) the alignment of the rules/queries with the knowledge expressed by the corresponding legal statements. We discuss the implications of these observations and describe the main challenges in achieving a machine-readable representation of legal norms.

Keywords

Automated Compliance Checking, FinTech, Legal Knowledge Representation, Legal Reasoning, SPARQL, SHACL

1. Introduction

Run-time monitoring of investment funds' activities is an important challenge in the financial industry [1]. This is particularly relevant for regulated funds such as UCITS (Undertakings for Collective Investments in Transferable Securities), which are legally required, among other things, to diversify their portfolio composition: they should not, for example, be mainly composed of investments toward the same issuer. Compliance with these requirements can, however, be sensitive to the evolution of markets: a fund operating close to the limits might end up infringing the law not because of the manager's actions, but simply because of market fluctuations affecting the value of the assets in its portfolio. There is therefore a need to monitor the activities of investment funds and the corresponding data as a whole, to ensure compliant operations.

Automated compliance checking shows a prominent need for maintainable models of applicable regulations and advocates for the use of open semantic standards such as RDF and OWL [2]. It also requires close alignment with data management practices [3]. However, there is a certain lack of cohesion between research in Description Logics and practical data modelling [4], which carries over to the field of legal knowledge representation where – we contend – there is a *representational gap* between data and legal norms. As a result of this gap, current commercial applications provide proprietary (ad hoc) solutions assisting asset managers and legal experts in various tasks of the compliance process (e.g., tax and accounting management, portfolio management, risk management, reporting), which are designed as black-box systems that do not rely on open semantics regarding data. Because of this, and because legal experts lack technical expertise¹, these systems often come with consulting services

RuleML+RR'25: Companion Proceedings of the 9th International Joint Conference on Rules and Reasoning, September 22–24, 2025, Istanbul, Turkiye

✉ marcello.ceci@uni.lu (M. Ceci); nicolas.sannier@uni.lu (N. Sannier); sallam.abualhaija@uni.lu (S. Abualhaija); domenico.bianculli@uni.lu (D. Bianculli); michael.halling@uni.lu (M. Halling)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹See the report on Technology Innovation in Legal Services of the UK's Solicitors Regulation Authority: <https://www.sra.org>.

necessary for any customization (e.g., the implementation of additional legal rules or features). The field of automated compliance checking seems thus torn between two needs: the need for legal rigor and the need for viable implementations.

This paper investigates the problem of representing the legal norms expressed by legal statements (i.e., regulations) in a machine-readable format supported by a logical reasoner and aligned with actual data representing the state of affairs. Specifically, the paper reports on our experience of using SPARQL (SPARQL Protocol and RDF Query Language)² for expressing complex norms in the financial sector. We focus on the legal statements contained in Article 43 of the Luxembourgish UCITS Law³, enacting Article 51 of the EU UCITS Directive as it sets asset composition limits for UCITS funds. The paper presents the methodology we followed to define the SPARQL rules, aligned to a conceptual model that we built, as part of previous work, on the basis of a real dataset sourced from a financial information provider [5]. Our experience shows that SPARQL is capable of handling complex financial rules. However, the translation process is far from intuitive, making it challenging for legal experts to understand the rules or its implications. Moreover, the complexity of the resulting rules might lead to unpractical execution times [6].

The paper highlights these challenges, related to the complexity of the representation and the misalignment between the legal statements and the representation. We note that some of them were not highlighted in the literature because they only emerge when dealing with complex legal norms and with application to real data. We show how some of these challenges can be addressed by SHACL (Shapes Constraint Language)⁴. However, not all of them can be solved in the modeling phase (e.g., by splitting the representation of a norm among several rules), as some cases require the input of a legal expert during the execution phase. It seems that compliance checking can only be *semi*-automated, possibly relying on an argumentation framework to support alternative representations of the law.

The rest of the paper is structured as follows: we define the problem of representing financial rules for automated compliance checking in Sect. 2. In Sect. 3 we review the state of the art in the representation of legal knowledge and reasoning. In Sect. 4 we present the methodology that we followed for building the model and the rules, and in Sect. 5 we report on the experience of representing financial rules. In Sect. 6 we reflect on the experience, and provide a research outlook in Sect. 7. Sect. 8 concludes the paper.

2. Preliminaries

In this section, we provide preliminary information regarding the representation of legal knowledge and reasoning.

The peculiarity of law. Legal statements do not behave as normal statements: they do not *describe* an existing objective reality, but rather prescribe a desired intersubjective reality [7]. Any knowledge representation effort aiming to represent the legal norms expressed in these statements for application to data has to take into account the complexities entailed by legal norms, including the regulative [8] and constitutive [9] effects of norms, exceptions [10], regulatory change [11], and legal interpretation [12].

The representational gap. Especially in domains such as finance, the volume and velocity of governing laws and governed financial transactions make run-time compliance checking a challenge. There is a *representational gap* between data (logically organized and deterministic) and legal norms (general and abstract, open-textured [13]), originating in the deductive nature of formal logic, whereas legal reasoning relies on pragmatism, persuasion and critical thinking, involving patterns of beliefs in a process of guesswork and deductive justification [14]. For example, where legal norms would distinguish between “transferable securities” and “money market instruments” at a high level, data would consider several types of bonds, whose qualification to the former or the latter type is often

uk/sra/research-publications/technology-innovation-in-legal-services/

²<https://www.w3.org/TR/sparql11-query/>

³Law of 17 December 2010 relating to undertakings for collective investment, available in English at: https://www.cssf.lu/wp-content/uploads/L_171210_UCI.pdf.

⁴<https://www.w3.org/TR/shacl/>

unclear [5]. The alignment between legal concepts and data thus requires explicit clarifications, as done by financial regulators [15]). As a result, full automation of compliance checking is currently a chimera.

From a research perspective, legal informatics would propose a highly articulated representation of the law in order to ensure its legal soundness, which however results in constructs (e.g., exceptional facts being subcategories of relevant facts in OWL 2 [16]) that are hard (and sometimes downright impossible) to map to actual data. Approaches in software engineering (SE) would not adequately implement legal considerations, with the consequent risk of overlooking aspects such as alternative legal interpretations or inter-relations among norms during the compliance assessment.

In the current state of affairs, creating formal compliance rules remains a task devoted to developers in the lack of a standard approach that would allow legal experts to directly contribute to the process. Large Language Models (LLMs) have the potential to help in that regard, towards the automation of rule creation; however, they currently fall short in creating rules representing the norms expressed by legal statements [17]: so far, successful applications of LLMs in the legal field are rather limited, involving only document generation and analysis, case prediction, and legal research [18].

The need for a model and a rule set. One essential characteristic of automated compliance checking is explainability. This consideration is reinforced for AI-based systems automating legal interpretation and application: since they are categorized as high risk under the AI Act, explainability is critical for their acceptance in the EU [19]. One solution to provide explainability for legal interpretation and application is to use symbolic representations [20] to bridge the formal rules and the understanding of the law by legal experts. Previous contributions to legal knowledge representation and modeling provide the tools necessary for a representation that can fill the aforementioned representational gap [21], expressing the semantics of legal norms in a format that can be mapped to the data to be checked.

This implies the need for a data model and a rule language supported by a reasoner. The *data model* is used to map the legal concepts to a common, coherent representation, and to ensure alignment with the target data [8]. On this regard, automated compliance checking advocates for models as semantic resources to allow reuse and ensure maintainability [2]. The *rules* are needed to validate (and manipulate) data, detecting paths in the data model that allow compliance experts to identify outliers, i.e., breaches [16]. The expressiveness of the rules determines, in turn, the logical complexity to be handled by the *reasoner* in the reasoning layer [10].

3. State of the Art

In this section, we review the state of the art in the representation of data models, rule languages, and reasoners for legal applications, from the perspective of two research communities, namely legal informatics and requirements engineering (RE). RE is a field of SE that investigates regulatory compliance, in particular the specification of legal requirements and rules based on the analysis of legal norms [22].

3.1. Models for Representing Legal Concepts

The field of legal informatics provides approaches for legal compliance checking in the Semantic Web based on modeling deontic norms in terms of ontology classes and ontology property restrictions [16]. Peculiarities of legal knowledge such as norm defeasibility or multiple interpretations are also modeled within the OWL 2 language [16, 23]. However, these models are considered too complex and thus expensive (in terms of construction and execution time) in practical scenarios [4], where the choice falls on databases and SQL-based queries, which only employ first-order logic, considered too limited for legal reasoning [24].

In the field of RE, models such as the taxonomy of semantic metadata of legal provisions [25] or the approach for deriving business processes from the law [22] attempt to tame the complexity of the legal domain with task-automation objectives in mind. However, those models have a high-level, often generic representation of the provisions, and a weak relationship to the concrete state of affairs [26].

3.2. Rule Languages for Representing Norms

Semantic markup languages such as LegalRuleML [27], open formats [16] and representations of the norms complexities [20] have been widely investigated in the field of legal informatics. The level of abstraction at which these rules are formulated, however, does not allow straightforward mapping to actual data for automated compliance checking purposes. Furthermore, not all languages are supported by an automated reasoner, and, for those who are, execution time and scalability are recognized challenges [6].

Complex normative structures have been studied also in the field of RE to support the precise definition of software requirements. While attempts at rule-based formalizations [28] lack consideration for legal peculiarities [29], approaches such as Legal GRL [30] and Nòmos [31] account for alternative representations of legal norms. However, these valiant attempts at bridging the representational gap suffer from two major limitations on either end of the gap: on the legal side, the resulting artifacts (e.g., goal models) are hard to understand for legal experts; on the SE side, since they are meant for compliant design purposes rather than compliance checking, their output requires further processing to be suitable for formal specification, which would further drive the software implementation.

3.3. Legal Reasoning

Regarding the reasoning layer, we note that the simulation of legal reasoning is possible only insofar as it is allowed by the expressiveness of the conceptual models and the rule sets. Many observations regarding the characteristics, challenges and limitations of legal reasoning are thus inherited from the chosen models and rule languages, especially regarding performance and scalability. A recent contribution in the field of legal informatics [10] compares several existing reasoners in terms of their capability to handle exceptions and contrary-to-duty obligations, two important elements of complexity in legal reasoning. However, the rules used to test the reasoners are rather simple, as they do not include complex relations between the involved entities, nor possible impact on other rules. They indeed describe complex information, such as publishing comments on a product evaluation. However, these variables are not automatically detected, as the representation is done independently of the target data.

The most comprehensive solution to representing legal reasoning while accounting for aspects such as alternative or conflicting representations of norms or data consists in the use of argumentation frameworks [32, 33]. However, the execution times are particularly high [6], and no complete approach towards representing legal compliance as an argumentation framework has been attempted yet [34].

3.4. Motivation

As discussed in this section, there is a lack of cohesion among the approaches to legal knowledge representation for software applications:

- the *legal theoretical approach*, investigating interoperable representations of legal concepts and rules, is mostly ignored by the software engineering community due to its detachment from industry concerns or practical data structure [4];
- the *engineering approach*, investigating the automation of steps of the compliance process, is mostly based on ad-hoc models which do not sufficiently account for the complexity of the legal domain; its contributions are thus considered of limited usefulness by the legal community [29]. Here, the field of RE has provided important attempts to bridge the gap, which however have not been considered satisfactory.

As a result, current simulations of legal reasoning are largely incomplete: e.g., they are either not understandable by a legal expert, not mappable to real data, not scalable, not applicable to different contexts, or not maintainable over time. Bridging the gap between raw data and conceptual/rule based representation of a domain of interest is a long-standing problem in contexts like data integration,

data warehousing, data exchange, and ontology-based data management. Typical solutions resort to mappings⁵.

This paper investigates the above-presented representational gap by reporting on an attempt to represent legal norms in a set of rules aligned to a real data set.

4. Methodology

Context. This research is part of a project investigating the automation of run-time monitoring of fund activities [1]. The proposed solution includes the (manual) definition of a conceptual model and the specification of rules to check the compliance of the portfolio composition.

Conceptual model definition. The conceptual model was built, as part of previous work [5], following a methodology [8] where model elements are initially created based on domain concepts taken from the letter of the law, and progressively refined by looking at other sources. Specifically, in this work, we based the refinement on secondary legal sources such as guidelines or Q&As issued by regulatory authorities (e.g., [15]). We also adapted the model to the reality under legal scrutiny, i.e., data regarding portfolio composition of a company. The data came from a proprietary dataset from a major financial information provider, covering periodical (monthly) extracts of fund asset composition over a period of five years. The integration of these extracts into the model went a long way in ensuring that the concepts (and ultimately the rules) were expressed at the right level of granularity.

The process of building the concept model inevitably involved the interpretation of the legal text, e.g., to interpret the concept of “issuer [of a financial instrument]” as an institution, or to correctly represent the relationship between derivatives and their underlying financial instruments (more on legal interpretations in Sect. 6). The model is available in our previous contribution [5] and in the online annex⁶.

Compliance rules specification. We developed the rules as constraints on the model, following a three-step process.

(I) First, we *identified the required actions* by looking at verbs and logical operators within the legal statements.

(II) For each required action, we then *identified the other main rule elements* (namely addressee, pre-condition(s) and constraint(s) – see [11]) and rewrote the legal text in streamlined, logical fashion (e.g., we converted all passive verbs in their active forms) using an *if-then* structure whenever pre-conditions were present.

(III) The last step involved *the actual representation of the rules in SPARQL* and the alignment of the rule variables to the conceptual model. This is also the step where any discrepancies or missing elements in the model/data (non-observable variables) could be detected (see Sect. 6.2).

5. Using SPARQL to Represent Financial Rules

In this section, we report on the application of the process for defining the rules for Art. 43 of the Luxembourgish UCITS law⁷ following the steps introduced in the previous section. The first and second authors, with the validation of the last author, contributed to the first two steps of the translation, while the last step was carried out by the first author only.

Step I. We identified all verbs bound (even implicitly) to deontic modalities (e.g., “may invest”, “may not exceed”) which led us to identify 11 rules, 10 of which (R1–R10) are regulative rules (shown in Table 1 and in their full version in the online annex) and one (R11) is a constitutive rule (a definition). This activity included the integration of regulatory changes and the resolution of cross-references. We note that not all legal statements introduce additional rules: for example, the statement “*The transferable securities and money market instruments referred to in paragraphs 3 and 4 shall not be taken into account*”

⁵See for instance the R2RML recommendation: <https://www.w3.org/TR/r2rml/>.

⁶The annex for this paper is available at: <https://dx.doi.org/10.6084/m9.figshare.29304779>.

⁷English version of the Luxembourgish UCITS Law available at: https://www.cssf.lu/wp-content/uploads/L_171210_UCI.pdf.

for the purpose of applying the limit of 40% referred to in paragraph 2” (Art. 43(5)) only results in an additional (negative) constraint added to the rules expressed in Art. 43(3) and 43(4) (see below for more on inter-relations among norms).

Step II. Successively, we identified the elements of the rules. On this regard, we note that the addressee is always “A UCITS”, the action is always “shall not invest” and the rules do not have pre-conditions (applicability conditions), but rather constraints (compliance conditions, see [11]) ranging over the type of financial instrument targeted, the type of issuer, and the ratio set as upper limit. Table 1, despite not being complete due to space reasons, gives an idea of the investment limits to be represented as constraints. For instance, R1 would state that a “UCITS fund” shall not invest more than “10%” of its “assets” in “transferable securities” (TS) and “money market instruments” (MMI) from a “single issuer” that is not “EU-guaranteed”.

Table 1
Regulative rules expressed in Art. 43

Rule	Financial Instruments	Issued or guaranteed by...	Limit
R1	Transferable Securities (TS) and Money Market Instruments (MMI)	A single issuer not EU-guaranteed	10%
R2	Deposits	A single deposittee	20%
R3	Over-the-counter derivatives (OTC)	A single counterparty credit institution	10%
R4	OTC	A single counterparty credit institution	5%
R5	TS and MMI	All not EU-guaranteed issuers with >5% investment	40%
R6	TS, MMI, Bonds, OTC	A single body	20%
R7	TS and MMI	A single issuer that is EU-guaranteed	35%
R8	Bonds	A single EU credit institution or issuer subject to special supervision	25%
R9	Bonds	Issuers not EU-guar. >5% investment	80%
R10	TS, MMI, Deposits, Bonds, OTC	A single body	35%

Inter-relations among norms. We note that not all constraints of a rule are introduced in the same legal statement. For example, the criterion of R1 on the issuer not being “EU-guaranteed” does not appear in the initial provision of Art. 43(1) but rather as an exception implied by the norm expressed in Article 43(3), which corresponds to R7 in the table. This is due to the fact that the norms expressed by the legal statements often overlap, through exceptions expressed within the same paragraph (e.g., Art. 43(2), “*this limitation does not apply to [...]*”) or through references in other paragraphs or articles (e.g., Art. 43(3), “*the limit laid down in the first sentence of paragraph 1 [...]*”). Figure 1 gives an overview of the inter-relations among the norms expressed in Art. 43. In this work, inter-relations have been integrated into the rules by the authors; see Sect. 6.1 for the possible solutions to represent them.

Step III. Representing the rules of Art. 43 in SPARQL and mapping its variables to the conceptual model was complex, as certain passages in the legal statements require (combinations of) functions which appear non-intuitive. Overall, it took approximately 25 hours to specify the formal representations in SPARQL, with the complex expressions in Table 2 accounting for most of the time. Considering the limited experience of the authors with SPARQL, generative models (in our case, MS Copilot, based on GPT-4) were helpful in refining the rules, even if they were not able to propose fully correct rules.

Example. Due to space limitations, we illustrate our process only using rule R1, while all other rules are available in the online annex. We started our representation of R1 by rewriting a simplified version of the first sentence of Art. 43(1), shown in Figure 2(a). We note that R1 should also include the exceptions coming from Art. 43(3) and (4): for clarity reasons, we did not include them in the figure, but we discuss them below. We then mapped the entities in the rule to the model — see Figure 2(b). Finally, we created the SPARQL query shown in Figure 2(c).

Regarding inter-relations between norms, as mentioned above, R1 has exceptions explicitly introduced

Table 2

Textual Expressions in Art. 43 and their Corresponding Interpretation as Rule Elements

Textual expression	Functional interpretation
(a) “[asset] issued by the same issuer / a counterparty”	Aggregate function [20]
(b) “In other cases [rule]”	Preceding statement’s condition becomes negative condition of rule
(c) “[issuer] in each of which [condition]”	Aggregate function [20], results grouped by issuer
(d) “[reference] does not apply to [condition]”	Condition added to reference
(e) “Where X invests more than [ratio] the total value must be [ratio]”	Aggregate function [20], results not grouped by issuer
(f) “Notwithstanding [reference]”	Rule is cumulative (not alternative) to the referenced one(s)
(g) “The limit may be [ratio][condition]”	Add negation of condition to previous statement, Add rule from condition
(h) “[reference] shall not be taken into account [rule]”	Entities in reference are excluded from rule
(i) “The limits of [reference] shall not be combined”	Compliance to each referenced rule is necessary
(j) “[entity] shall be regarded as [entity]”	New definition (constitutive norm)

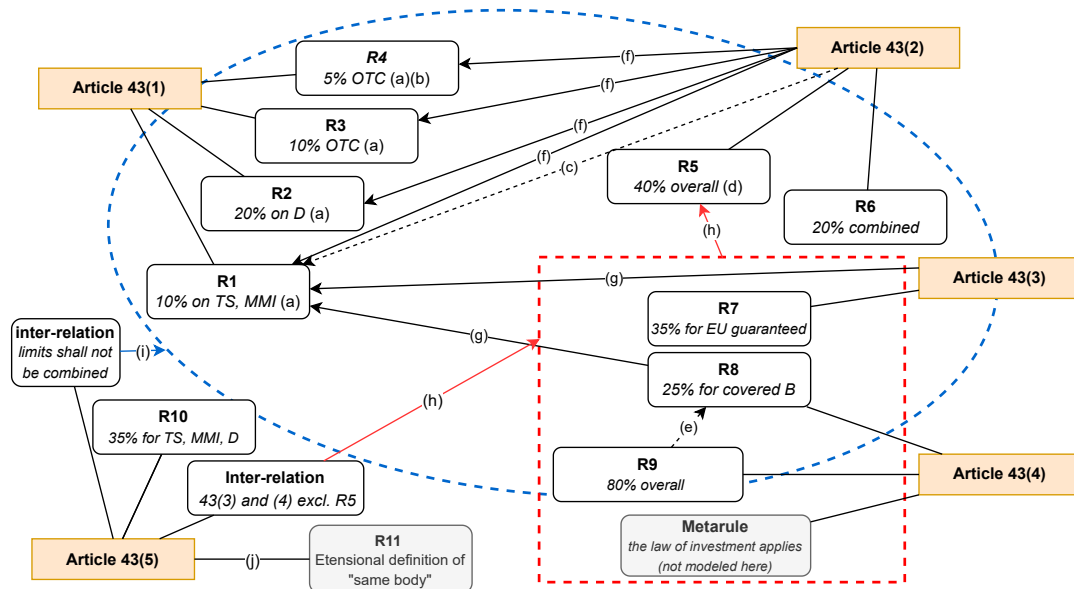


Figure 1: Overview of the rules expressed in Art. 43, with their IDs as introduced in Table 1, and the inter-relations among norms with the letter between parentheses referring to the rows in Table 2. The blue and red shapes indicate the rules affected by the inter-relations expressed in Article 43(5).

in paragraphs (3) and (4) of Art. 43; according to these norms, certain assets (e.g. the transferable securities issued by a EU Member State) are excluded from R1, since a different limit applies to them. As we will further discuss in Sect. 6, one of the possible ways to represent exceptions is to incorporate them in the original rule. The complete (simplified) text of R1 would then be as follows:

A fund shall not invest more than 10% of its assets in ‘transferable securities’ or ‘money market instruments’ issued by the same ‘issuer’ *unless these are guaranteed by a Member State Equivalent Institution, or are covered bonds, or are bonds issued before 8 July 2022 by a credit institution which has its registered office in a Member State and is subject by law to special supervision.*

Similarly, the SPARQL query for R1 would require the addition of the following exclusion filters in

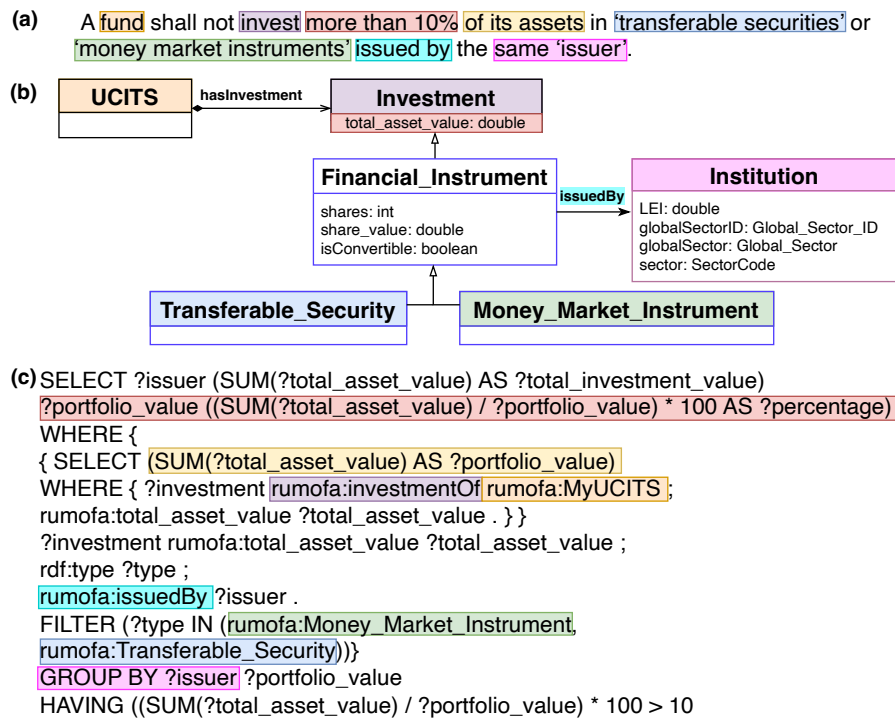


Figure 2: Representation of R1 (not counting exceptions): a. the simplified text of the law; b. the elements of the model involved in the representation; c. the corresponding SPARQL query.

order to implement the exceptions:

```

FILTER NOT EXISTS {?type rdf:type rumofa:CoveredBond}
  FILTER NOT EXISTS {?issuer rumofa:MS_Guaranteed true} UNION {
  ?issuer rdf:type rumofa:MemberStateEquivalent}
  FILTER NOT EXISTS {?investment rumofa:issue_date ?issueDate .
  ?issuer rdf:type rumofa:CreditInstitution ;
  rumofa:prudential_supervision true ;
  rumofa:hasRegisteredOffice ?office .
  ?office rumofa:locatedIn ?country .
  ?country rdf:type rumofa:MemberState .
  FILTER (?issueDate < "2022-07-08"^^xsd:date)}}
  
```

Observations

Following the methodology presented in this paper, the resulting 10 SPARQL queries were validated (using a Virtuoso⁸ endpoint) against sample data that was manually created to cover the various situations expressed in the legal rules. SPARQL proved expressive enough to represent the norms expressed by the legal statements of Art. 43.

We noted how the semantics of certain passages of the legal statements appeared non-intuitive when expressed in a logical format: legal experts notably find it hard to understand the equivalence between the textual expressions and functional representations in Table 2, or between the textual norm of Figure 2(a) and its corresponding query in Figure 2(c). The causes of this would deserve investigation, as they seem to be related to the *representational gap* – see Sect. 2. As a consequence, technical expertise is needed to fully understand and validate the SPARQL query. This technical expertise needs however to be blended with the legal expertise needed to formally represent legal peculiarities such as non-observable variables and alternative legal interpretations of the norm. We discuss these open challenges in the next section.

⁸<https://vos.openlinksw.com/owiki/wiki/VOS/VOSSPARQL>

6. Discussion

In this section, we reflect on the experience of applying the representation process to complex legal rules.

6.1. Logical and Structural Complexity of the Representation

Our experience with Art. 43 shows that formal representations of legal norms are highly articulated, going beyond expressing logical operators or navigating knowledge graphs, as it is the case with the examples shown in the literature. For example, they involve aggregation [20] of the value of investments involving the same issuers (see Table 2). Representing indirect manipulations of the original data is not straightforward, as these operations often conceal legal interpretations (e.g., when classifying investments).

To compute the complexity of a SPARQL query, the only method found in the literature is the one proposed by Buil-Aranda et al. [35]. The method consists in assigning a weight of 1 to each operator and each triple pattern of the query. The resulting complexity calculation is shown in Table 3.

Table 3
Complexity Measures for the SPARQL Queries Representing Art. 43

	R1a ¹	R1b ¹	R2	R3	R4	R5	R6	R7	R8	R9	R10
Complexity	17	35	18	20	23	37	25	28	27	33	12

¹ R1a does not include the exceptions from later statements. R1b includes those exceptions.

Such complexity scores are unseen for SPARQL queries. According to Arias et al. [36]: “Most [SPARQL] queries are simple, i.e., 66.41% of DBPedia queries and 97.25% of SWDF just contain a single triple pattern [...] and the [relationship] chains in 98% of the queries have length one, with the longest path having a length of five”. However, we note that this analysis does not account for nested structures, so a long yet straightforward query with many ANDs would show a complexity similar to a highly nested (and thus less intelligible) query.

In addition to the logical complexity described above, we note the presence of **inter-relations among norms** such as exceptions, introduced in Sect. 5. We have two solutions for handling such inter-relations in our representation:

- The first solution consists of *implementing the relation directly in the affected rule*. We showed an example in Sect. 5, where we added constraints to R1 to exclude the exceptional investments. This is the solution adopted in the literature [10] and is in line with the ontological vision of modalities, where an exceptionally allowed action is represented as a subset of a breach [16]. This solution, however, does not scale well. A rule may repeatedly introduce exceptions such as these into several other rules, and each rule may thus have several of these conditions added, making any human supervision impractical. Also, this part of the rule has different legal sources than the main part of the rule: these additional sources should be added as metadata to the SPARQL query for traceability and change impact management [11].
- An alternative solution is to *allow rules to defeat each other’s consequents* using defeasible logics [37]. This solution implies a hierarchical structuring of the rules to automatically determine the outcome of conflicts, establishing which rules apply and which do not. This alternative has been suggested in the literature [33, 37, 32, 12] and implemented in the SPINdle reasoner [10]. Although the solution is useful for conflicting rules with complementary consequents, it would not be sufficient in our example of Art. 43(3), where the exceptional investments are not only assigned a different threshold, but also excluded from the application of R1.

Further research is required to establish whether these considerations can be generalized beyond exceptions to all inter-relations among norms.

Generally speaking, such a high complexity within and among norms is likely to impact not only the performance of reasoners [6], but also the *usability of the rules for legal experts*. This makes it hard

to establish a collaboration between a legal expert and a developer in writing those rules. While an intermediate representation would help in that regard, we note that the visualization of R1 as a graph is not possible using current standards, as functions such as GROUP BY are not covered and accounted by the literature [38]. The issue of query/data visualization is acknowledged in the legal informatics literature [39] as an important challenge towards the creation of a human-friendly (ideally, visual) intermediate language that can be aligned one-to-one to a formal language such as SPARQL.

6.2. Misalignments Between Legal Statements and their Representation

Regarding the case in which the norms expressed by the legal statements cannot be represented in the model or in the ruleset in a way that allows automated compliance checking, we have identified two similar yet distinct situations related to legal interpretations and data observability, respectively.

Legal Interpretation. No effort on legal knowledge representation can ignore the subject of legal interpretation. The law being an inter-subjective reality [7], its statements do not provide information about factual reality, but rather describe a desired reality using generic and abstract terminology. The application of the law to the case in point is always subject to legal interpretation, an intellectual process which takes into account the semantics of the text(s) but also the underlying principles and assumptions on the intentions of the regulator. The challenges posed by legal interpretation are strictly related to the *representational gap* (see Sect. 2).

As an example, let's consider the following: on 3 November 2021 CSSF, the Luxembourg financial market supervisor, published an updated version of its document providing clarifications on the holding of assets by UCITS (the "UCITS FAQ" [15]). Regarding Article 43, the document specifies that the 20% limit on deposits applies to ancillary liquid assets and does not apply to margin accounts. This specification impacts the definition of "deposit"; we can easily imagine how, before this clarification, several alternative versions of the norm would have been possible, either including or excluding ancillary liquid assets or margin accounts. Another example is in Art. 43(5), where, after stating that certain limits cannot be combined, the successive statement introduces a prohibition with the word "thus", which may qualify the prohibition as either an example or a concretisation of the previous legal statement. In this work, we opted for the second interpretation, which resulted in R10; however, different interpretations might result in different rules.

In order to handle legal interpretation, during the process of rule (and model) creation the legal expert should be in control of the alternative interpretations of a legal statement, as the choice on which version of the norm to represent in the ruleset carries important consequences in the compliance checking.

Data Observability. We note that not all elements expressed in Art. 43 were observed in our dataset. An example is an asset being "*guaranteed by a Member State*" (Art. 43(3), implemented in R1 via inter-rule relationship as shown in Table 1) or a company being "*included in the same group for the purposes of consolidated accounts*" (Art. 43(5)). As opposed to legal interpretation, which results in alternative – and possibly conflicting – versions of the model and ruleset, non-observable properties can be unambiguously represented in the ruleset; however, the related model category cannot be automatically instantiated, i.e., the values for those properties cannot be automatically determined based on available data, which can prevent a checker from effectively verifying the rules. Despite missing data being a widespread problem in the financial industry [40], the issue of data observability is – to the best of our knowledge – not taken into account in the literature of rule representation, possibly because it becomes obvious only when complex models have to be implemented against real data.

7. Outlook

In this section, we discuss possible research directions in the multidisciplinary field of legal knowledge representation to handle the aforementioned challenges.

7.1. Handling Representational Complexity: Calling for a Splitting of Legal Rules

In Sect. 6, we introduced inter-relations among norms as elements of complexity, whose solution affect the logical structure of the rules to be executed on the data. However, inter-relations are not the only elements that may prejudice the self-containment of norms as individual rule artifacts. In fact, we note that certain rules may share common constraints. Repeating the same constraint checking for each rule (i.e., repeated subqueries) may be inefficient and may hinder the readability of the rule. An alternative consists in introducing new concepts (as variables if the language permits, otherwise as additional model elements). For example, identifying “*certain bonds issued after 8 July 2022*” can be eased by representing these bonds as a specific subclass of “bond” in the model (“*bond issued after 8-7-22*”) or as a Boolean class attribute specifying these characteristics, instead of checking the date of issue for every rule that includes that constraint. These concepts are called **intermediate legal concepts (ILCs)** in the legal literature [41].

The choice of whether to represent ILCs impacts both the knowledge generation phase (i.e., when building the model and the rules) and the checking phase (i.e., when instantiating the model and running the rules on it):

- *Ignoring ILCs in the representation* implies that, at runtime, the related restrictions are explicitly calculated for each rule. This approach has the main downside of resulting in a considerable increase in the length of the ruleset, especially when the ILC is relevant to multiple rules.
- Alternatively, ILCs can be *represented as additional properties (subclasses or attributes) in the model*, with a specific rule devoted to their instantiation. Technically this could be done either by creating *views* over the underlying data or by using named subqueries (known as *Common Table Expressions* in SQL). This approach is in line with the legal theory’s view of ILCs [41]; while it has the downside of splitting the norm into several rules, these end up being shorter and simpler.

Further research is required to determine the optimal balance to maximise the readability and the usability of the ruleset for legal experts. A possible solution consists in defining a set of patterns for the rule/query language, similar to the Ontology Design Patterns [42].

Inter-relations among norms and ILCs suggest a need for a **more articulated structuring of legal rules**, representing the *antecedent* and *consequent* parts [31] as two distinct rules, where the *antecedent* rule describes the applicability conditions, and the *consequent* rule states the deontic statements that apply to the relevant entities. Entities that fulfill the first rule would thus be reclassified as *relevant to* (i.e., *fulfilling the default precondition for*) the second rule(s). This structure would allow exceptions to apply to either of these parts, while also providing a template for ILCs. It also allows for representing definitions with a limited application context, such as R11 (see the online annex).

Using SHACL to Address Representational Complexity

Some of the complexity issues highlighted in the previous subsection can be overcome by SHACL, a recommendation of the W3C (World Wide Web Consortium) for describing and validating RDF graphs [43]. In SHACL, validation is based on *shapes*, which define particular constraints for specific nodes in a graph. SHACL has been considered a viable solution for representing legal rules [10]; also, the approach of dealing with (non-)compliance as a *validation shape* is in line with the intrinsic nature of compliance, as shown in ontological representations where the (non-)compliant behaviour is seen as a subset of the possible behaviour [16]. Furthermore, SHACL provides a declarative way (target declarations) to specify the focus nodes, i.e., which nodes in the RDF graph should be validated, unlike SPARQL which requires filtering and grouping. Where SPARQL lacks modularity and reusability, SHACL in fact provides ways to handle inter-rule relations and layered definitions: for example, it supports the creation of so-called **intermediate shapes**, which allow to handle nested constraints for a more intuitive and modular rule definition. Intermediate shapes can help in handling inter-relations between norms, whose representation in SPARQL requires repeated subqueries. Following is an example of intermediate shapes handling the exceptions to R1:

```

ex:CoveredBondExceptionShape
  a sh:NodeShape ;
  sh:property [
    sh:path rdf:type ;
    sh:hasValue rumofa:CoveredBond ;
  ] .

ex:IssueDateExceptionShape
  a sh:NodeShape ;
  sh:property [
    sh:path rumofa:issue_date ;
    sh:lessThan "2022-07-08"^^xsd:date ;
  ] ;
  sh:property [
    sh:path rumofa:issuedBy ;
    sh:class rumofa:CreditInstitution ;
  ] .

```

R1 can then be rewritten, referring to the nested constraints introduced above for representing the exceptions, as follows:

```

ex:InvestmentLimitShape
  a sh:NodeShape ;
  sh:targetNode rumofa:Investment ;
  sh:not ex:CoveredBondExceptionShape ;
  sh:not ex:IssueDateExceptionShape ;
  sh:sparql [
    a sh:SPARQLConstraint ;
    [SPARQL query of Figure 2(c)]
  ]

```

While it successfully modularizes exceptions, this SHACL shape does not get rid of SPARQL: in order to handle aggregate functions (grouping investment value by issuer), it is in fact necessary to embed the same SPARQL query from Figure 2(c) inside the shape, using the SHACL-SPARQL extension, since SHACL-Core can only validate constraints on individual nodes or properties.

In addition to intermediate shapes, we note that SHACL provides a mechanism to declare the severity of a shape by attaching different semantic labels to its validation result (i.e., info, warning, violation), and allows for the definition of shapes that refer to other shapes, enabling recursive validation (the latter being particularly useful for modeling legal norms that involve hierarchical or nested conditions, e.g., group membership). However, some of the features of SHACL (such as intermediate shapes) are not mentioned in the formal documentation; their usage rather arises from practice. Therefore, these features are not standardized and may differ in practical implementations. This further complicates the understanding of the formal representation for a legal expert, prompting the need for reusable patterns (as mentioned previously) and/or an intermediate language (see the next subsection).

In conclusion, we note that SHACL seems promising to address some of the challenges related to the complexity of the query; however it is not expressive and standardized enough to bridge the human understanding of the law and its formal representation.

7.2. Handling Representation Misalignment: the Need for Human Intervention

Overall, the challenges posed by legal interpretation and data observability impact the representation of norms either by making the rules more complex or by preventing the automation of rule checking. The issue of **interpretation** is well known in the literature [44], being widely recognized as a reason for the need of explainability of the reasoner's results [10, 18] and for the need of argumentation frameworks to handle mutually exclusive arguments [12]; however, over the last 50 years the most common approach to automating compliance in practical scenarios has been to embed an interpretation of the legal statements into a black-box compliance checking system [2]. We recall that no legal analysis can be considered accurate if it does not take into account alternative interpretations. In the cases where

the ambiguity does not allow for establishing concrete checks, the process can still be semi-automated by providing the relevant information and possible alternative representations to the domain expert in charge of the assessment (the *human intervention*, see below).

Regarding **data observability**, compliance checking for non-observable variables can be semi-automated: in this scenario, the legal expert or third-party data may provide the information that is missing from the dataset to instantiate the model. For instance, whether a “[company is included in group] in accordance with internationally recognized accounting rules” (Art. 43(5)) is typical additional knowledge that can be instantiated by the domain expert. This can be done either before the rules are applied (e.g., by specifying which types of accounting rules are to be considered as internationally recognized) or as part of the check (e.g., by prompting the expert who will manually instantiate the attribute for each company). Non-observable data could be detected in SHACL by using intermediate shapes with specific severity.

We note that both the above-presented challenges of legal interpretation and data observability imply the **need for human intervention**: the check cannot be fully automated, because in certain cases the reasoner needs external input to complete the reasoning. This implies the need to present modelling cues (e.g., a reference to non-observable data) in a manageable way. Compliance checking is then a process where facts are gathered, automatic (and possibly conflicting) classifications are made, and then a legal expert provides contextualized information that could not be automatically determined. Note that this contextualized information might also depend on the preferences or values of the human expert among the alternative interpretations of the legal statement, and thus advocates for an argumentation structure.

The complexity of operational languages such as SPARQL and SHACL [28] prompts the **need for an intermediate language** to facilitate the legal expert’s understanding of the implications of alternative formal representations. According to state-of-the-art in legal knowledge representation, this interaction can be ensured via an argumentation framework [32, 33], representing alternatives (e.g., a non-observable variable, like, in our example, a ratio being reasonable or not) as arguments, which the expert can manually accept or reject according to the foreseeable legal implications. Such a framework has the potential to represent the application of complex rules to real data sets in a human-readable way, especially if supported by a user interface (but unfortunately, as already noted, the state of the art in the visualization of legal reasoning is lacking [39]).

8. Conclusion

In this paper, we have reported on our experience on representing complex financial rules in a machine-readable language for compliance checking. We have presented the approach, consisting of a model and a rule set, and shown an example. We have then discussed the challenges posed by legal texts to their abstract representation for automated application. As mentioned in Sect. 7, it seems that the best way to address the challenges presented in this paper is to increase the number of rules, reducing their length, and allow human intervention in the choice among alternative interpretations and for non-observable variables. In that regard, SHACL is a promising candidate to address some of the problems related to the sheer length of the rules and the repeated subqueries, even though it still has to rely on SPARQL to handle complex operations such as aggregate functions. The complexity related to the increased number of rules can be handled through the use of intermediate legal concepts and an intermediate language, possibly representing the entire compliance process as an argumentation framework.

As part of future work, we plan to define complexity measures for the resulting rules, with an augmented dataset that would allow to measure execution times and more generally to evaluate the performance of SPARQL and SHACL in a practical scenario. Next steps in the representation of complex legal norms include the investigation of argumentation frameworks and of intermediate representations that are more lawyer-friendly, in a strive to achieve a modeling of legal knowledge that allows the integration of a legal expert into the automation of the compliance checking process.

Acknowledgments

This research was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), under grant numbers NCER22/IS/16570468/NCER-FT and C24/IS/18894115/AGLAI.A.

Declaration on Generative AI

The authors did not use any Generative AI during the preparation of this work.

References

- [1] M. Ceci, N. Sannier, S. Abualhaija, D. Shin, D. Bianculli, M. Halling, Toward automated compliance checking of fund activities using runtime verification techniques, in: *FinanSE 2024*, 2024.
- [2] R. Amor, J. Dimyadi, The promise of automated compliance checking, *Developments in the Built Environment 5* (2021) 100039.
- [3] D. S. Chittoor, Implementing data lineage frameworks in financial institutions: A systematic analysis of compliance, efficiency, and risk management, *Int. J. of Science and Research Archive 14* (2025) 353–361.
- [4] B. Bogaerts, M. Jakubowski, J. Van den Bussche, SHACL: A description logic in disguise, in: *Logic Programming and Nonmonotonic Reasoning*, 2022, pp. 75–88.
- [5] N. Sannier, M. Ceci, S. Abualhaija, D. Bianculli, M. Halling, A model toward formalizing and monitoring compliance of investment funds activities, in: *MoDRE 2024*, 2024, pp. 272–280.
- [6] L. Robaldo, S. Batsakis, R. Calegari, F. Calimeri, M. Fujita, G. Governatori, M. Morelli, G. Pisano, K. Satoh, I. Tachmazidis, Taking stock of available technologies for compliance checking on first-order knowledge, *CEUR Workshop Proceedings 3204* (2022) 1–16.
- [7] Y. N. Harari, *Nexus: A Brief History of Information Networks from the Stone Age to AI*, Diversified Publishing, 2024.
- [8] M. Ceci, D. Bianculli, L. C. Briand, Defining a model for content requirements from the law: An experience report, in: *RE 2024*, 2024, pp. 18–30.
- [9] M. Ceci, F. Al Khalil, L. O’Brien, T. Butler, Requirements for an intermediate language bridging legal text and rules, in: *MIREL@JURIX*, 2016.
- [10] L. Robaldo, S. Batsakis, R. Calegari, F. Calimeri, M. Fujita, G. Governatori, M. Morelli, F. Pacenza, G. Pisano, K. Satoh, I. Tachmazidis, J. Zangari, Compliance checking on first-order knowledge with conflicting and compensatory norms: a comparison among currently available technologies, *Artif. Int. and Law 32* (2024) 505–555.
- [11] S. Abualhaija, M. Ceci, N. Sannier, D. Bianculli, L. C. Briand, D. Zetsche, M. Bodellini, AI-enabled regulatory change analysis of legal requirements, in: *RE 2024*, 2024, pp. 5–17.
- [12] A. Rotolo, G. Governatori, G. Sartor, Deontic defeasible reasoning in legal interpretation: two options for modelling interpretive arguments, in: *ICAIL 2015*, 2015, pp. 99–108.
- [13] C. Guitton, A. Tamò-Larrieux, S. Mayer, G. van Dijck, The challenge of open-texture in law, *Artificial Intelligence and Law* (2024) 1–31.
- [14] P. Leith, Logic, formal models and legal reasoning, *Jurimetrics J.* 24 (1983) 334.
- [15] CSSF, FAQ concerning the Law of 17 December 2010 relating to UCITS, https://www.cssf.lu/wp-content/uploads/FAQ_Law_17_December_2010.pdf, 2025. Accessed: 2025-08-20.
- [16] E. Francesconi, G. Governatori, Patterns for legal compliance checking in a decidable framework of linked open data, *Artificial Intelligence and Law 31* (2022) 1–20.
- [17] S. Abualhaija, M. Ceci, N. Sannier, D. Bianculli, S. Lannier, M. Siclari, O. Voordeckers, S. Tosza, LLM-assisted elicitation of regulatory requirements: A case study on the GDPR, in: *RE 2025* (to appear), 2025, pp. 5–17.
- [18] J. Lai, W. Gan, J. Wu, Z. Qi, P. S. Yu, Large language models in law: A survey, *AI Open 5* (2024).
- [19] C. Panigutti, R. Hamon, I. Hupont, D. Fernandez Llorca, D. Fano Yela, H. Junklewitz, S. Scalzo,

- G. Mazzini, I. Sanchez, J. Soler Garrido, E. Gomez, The role of explainable AI in the context of the AI Act, in: *FACCT 2023*, 2023, p. 1139–1150.
- [20] J. Anim, L. Robaldo, A. Z. Wyner, A SHACL-based approach for enhancing automated compliance checking with RDF data, *Information* 15 (2024).
- [21] B. Fawei, A. Wyner, J. Z. Pan, M. J. Kollingbaum, Using legal ontologies with rules for legal textual entailment, in: *AI Approaches to the Complexity of Legal Systems*, 2017, pp. 317–324.
- [22] T. D. Breaux, A. I. Antón, J. Doyle, Semantic parameterization: A process for modeling domain descriptions, *ACM Trans. Softw. Eng. Methodol.* 18 (2008) 5:1–5:27.
- [23] F. Al Khalil, M. Ceci, K. Yapa Bandara, L. O’Brien, SBVR to OWL2 mapping in the domain of legal rules, in: *RuleML 2016*, 2016, pp. 258–266.
- [24] G. Antoniou, G. Baryannis, S. Batsakis, G. Governatori, M. B. Islam, Q. Liu, L. Robaldo, G. Siragusa, Large-scale legal reasoning with rules and databases, *Journal of Applied Logic* 8 (2021) 911–939.
- [25] A. Sleimi, N. Sannier, M. Sabetzadeh, L. C. Briand, M. Ceci, J. Dann, An automated framework for the extraction of semantic legal metadata from legal texts, *Empir. Softw. Eng.* 26 (2021) 43.
- [26] O. Kosenkov, M. Unterkalmsteiner, D. Méndez, D. Fucci, T. Gorschek, J. Fischbach, On developing an artifact-based approach to regulatory requirements engineering, in: *MoDRE 2024*, 2024.
- [27] T. Athan, G. Governatori, M. Palmirani, A. Paschke, A. Wyner, LegalRuleML: Design principles and foundations, *Reasoning Web. Web Logic Rules 2015* (2015) 151–188.
- [28] D. Merigoux, N. Chataing, J. Protzenko, Catala: a programming language for the law, *Proc. ACM Program. Lang.* 5 (2021) 1–29.
- [29] G. Boella, L. Humphreys, R. Muthuri, P. Rossi, L. van der Torre, A critical analysis of legal requirements engineering from the perspective of legal practice, in: *RELAW 2014*, 2014, pp. 14–21.
- [30] S. Ghanavati, D. Amyot, A. Rifaut, Legal goal-oriented requirement language (legal GRL) for modeling regulations, in: *MiSE 2014*, 2014, pp. 1–6.
- [31] S. Ingolfo, A. Siena, A. Susi, A. Perini, J. Mylopoulos, Modeling laws with nomos 2, in: *RELAW 2013*, 2013, pp. 69–71.
- [32] L. Longo, Argumentation for knowledge representation, conflict resolution, defeasible inference and its integration with machine learning, *Machine Learning for Health Informatics* (2016) 183–208.
- [33] M. Billi, R. Calegari, G. Contissa, F. Lagioia, G. Pisano, G. Sartor, G. Sartor, Argumentation and defeasible reasoning in the law, *J – Multidisciplinary Scientific Journal* 4 (2021) 897–914.
- [34] G. Pisano, Argumentation for Legal Reasoning: Meta-models, Technology and Beyond, Ph.D. thesis, UNIBO - Università di Bologna, Italy, 2024.
- [35] C. Buil-Aranda, M. Ugarte, M. Arenas, M. Dumontier, A preliminary investigation into SPARQL query complexity and federation in bio2rdf, in: *AMW 2015*, 2015.
- [36] M. Arias, J. D. Fernández, M. A. Martínez-Prieto, P. de la Fuente, An empirical study of real-world SPARQL queries, *CoRR abs/1103.5043* (2011).
- [37] G. Governatori, Defeasible description logics, in: *Rules and Rule Markup Languages for the Semantic Web*, 2004, pp. 98–112.
- [38] F. Haag, S. Lohmann, S. Siek, T. Ertl, QueryVOWL: Visual composition of SPARQL queries, in: *ESWC 2015 Satellite Events*, 2015, pp. 62–66.
- [39] S. McLachlan, L. C. Webley, Visualisation of law and legal process: An opportunity missed, *Information Visualization* 20 (2021) 192–204.
- [40] S. Bryzgalova, S. Lerner, M. Lettau, M. Pelger, Missing financial data, *The Review of Financial Studies* 38 (2024) 803–882.
- [41] G. Sartor, Legal concepts as inferential nodes and ontological categories, *Artificial Intelligence and Law* 17 (2009) 217–251.
- [42] A. Gangemi, V. Presutti, *Ontology Design Patterns*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 221–243.
- [43] P. Pareti, G. Konstantinidis, *A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs*, Springer International Publishing, Cham, 2022, pp. 115–144.
- [44] G. Boella, M. Janssen, J. Hulstijn, L. Humphreys, L. van der Torre, Managing legal interpretation in regulatory compliance, in: *ICAIL 2013*, 2013, p. 23–32.