Towards Legal Compliance by Correlating Standards and Laws with a Semi-automated Methodology

Cesare Bartolini[1(✉)], Andra Giurgiu[1], Gabriele Lenzini[1], and Livio Robaldo[1,2]

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg
{cesare.bartolini,andra.giurgiu,gabriele.lenzini,livio.robaldo}@uni.lu
[2] Computer Science and Communications Research Unit (CSC), University of Luxembourg, Luxembourg, Luxembourg

Abstract. Since generally legal regulations do not provide clear parameters to determine when their requirements are met, achieving legal compliance is not trivial. The adoption of standards could help create an argument of compliance in favour of the implementing party, provided there is a clear correspondence between the provisions of a specific standard and the regulation's requirements. However, identifying such correspondences is a complex process which is complicated further by the fact that the established correlations may be overridden in time *e.g.,* because newer court decisions change the interpretation of certain legal provisions. To help solve these problems, we present a framework that supports legal experts in recognizing correlations between provisions in a standard and requirements in a given law. The framework relies on state-of-the-art Natural Language Semantics techniques to process the linguistic terms of the two documents, and maintains a knowledge base of the logic representations of the terms, together with their defeasible correlations, both formal and substantive. An application of the framework is shown by comparing a provision of the European General Data Protection Regulation with the ISO/IEC 27018:2014 standard.

Keywords: Legal compliance · Legal requirements · Security standards · General data protection regulation

1 Introduction

As it happens with the European Union (EU) harmonized standards used to demonstrate that products, services, or processes comply with relevant EU legislation [11], when a standard published by a standardization body is endorsed by the law, then implementing the standard also gives a *legal presumption of compliance*. However, harmonized standards are a fortunate but uncommon case. More often, standards do not have such a direct effect on legal compliance. By adopting a standard however, an organisation can demonstrate a proactive attitude and best efforts to be compliant according to the state of the art in that

T. Bosse and B. Bredeweg (Eds.): BNAIC 2016, CCIS 765, pp. 47–62, 2017.
DOI: 10.1007/978-3-319-67468-1_4

specific domain. Standards can thus provide the organisation with an argument of compliance.

Such an argument of compliance, would rely on proving a clear correspondence between the provisions of a specific standard and the law's requirements. But identifying such correspondences is not easy. It is also a dynamic process where the established correlations can be further overridden in time, for instance because newer court decisions change the interpretation of certain legal provisions.

In this paper, we propose a way to ease, and partly to automate, the process of checking for such document-to-document correspondences. We discuss a software framework that aids in determining the formal and substantive correlations between the provisions in a standard and those in a law. The framework's core is a logic-based methodology to represent, in a machine-processable format, (*a*) the relevant syntactic concepts in the provisions, and (*b*) the relevant correlations between them. In this paper, we describe this logic-based methodology and exemplify how it works using provisions from two specific and relevant documents. One is the standard ISO/IEC 27018:2014 (ISO 27018, in short), which concerns public clouds acting as personal data processors. This standard can be regarded as a building block [11] that helps data-processing organizations comply with the principle of data protection accountability. The second document is the General Data Protection Regulation (GDPR), which is the new law on data protection in the EU (see Sect. 2).

The framework depends on two auxiliary functional blocks (see Sect. 4): *i* a *logic knowledge base*, which can be populated, corrected and extended by legal experts and that stores the machine-processable logic correlations; *ii* a *set of Natural Language Semantics (NLS) and Natural Language Processing (NLP) techniques*, which allow a user to browse a XML representation of the documents and to search and retrieve the words, the terms, and the sentences that have been found relevant for correlation. The NLS and NLP techniques help users to efficiently and precisely find the established correlations within the knowledge base, which any expert user can successively reinforce, correct, justify, and expand. The selection of relevant terms and the definition of correlations, requiring human reading, processing and decision-making, is therefore semi-automatic.

The correlations are expressed formally in a deontic and defeasible logic for legal semantics called *Reified Input/Output Logic* (see Sect. 3). Defeasibility is required since, due to differences in legal interpretation, some of the correlations could be in contradiction: interpretations from more authoritative sources (such as high courts) are thus required to eventually resolve the conflict.

2 The Data Protection Reform

The GDPR, which will apply from 25 May 2018, replaces the current Directive 95/46/EC[1] with more modern rules, better adapted to the data processing real-

[1] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

ities of today. Unlike Directive 95/46/EC, which required implementation by the Member States, the GDPR will be directly applicable and does not require transposition into national law.

The purposes of the GDPR [30] are to align data protection rules with the most recent developments in data-processing technologies while still providing a legislation that is flexible enough not to become outdated over the course of a few years. The Regulation enhances the responsibility of data controllers and strengthens the rights of the data subject. Controllers will face heavy administrative fines in case of non-compliance with its provisions [13], which go as high as four percent of the total worldwide annual turnover of an undertaking[2].

As many of the provisions of the GDPR are quite general and potentially applicable in diverse ways, its interpretation through legal doctrine and jurisprudence will be of essence. The Regulation won't be applicable before 25 May 2018 and therefore no decisions based on its provisions can exist until then. When it will be applied, relevant decisions on its interpretations are expected to be issued by the Data Protection Authorities (DPAs) of Member States, by national courts, and by the Court of Justice of the European Union (CJEU).

3 Related Work

The application of NLP and NLS to the legal domain is a research trend that has received a lot of attention and investments in recent years, as shown by several acU-funded projects on the topic such as `Openlaws`[3], `ProLeMAS`[4], and `MIREL`[5]. Modern NLP technologies [6] are able to classify and discover inter-links between legal documents thanks to parsers [1], statistical algorithms [8], and legal terminological databases or legal ontologies [8,40]. This is often done by transforming the legal documents into XML standards, such as *Akoma Ntoso*[6], where relevant information are tagged. An example of commercial legal document management system employing these technologies is *Eurocases*[7], which collects EU case law and uses NLP techniques to classify the documents on the basis of their topic [7].

Although these systems help navigate legislative documents and retrieve information, their overall usefulness is limited because they process words disregarding their possible different semantic interpretations. The latter would allow for legal reasoning, *e.g.,* correlating laws among them and determining whether they lead to inconsistencies. Semantic processing of documents like laws/regulations and security standards is what we are going to propose as a component of the methodology we present in Sect. 4.

The underlying logical framework we will use in our methodology is reified Input/Output logic [36], a recent approach designed as an attempt to investigate the *logical architecture* of the provisions in natural language. Reified

[2] GDPR, Article 83.
[3] https://info.openlaws.com/openlaws-eu/.
[4] http://www.liviorobaldo.com/Prolemas.htm.
[5] http://www.mirelproject.eu/.
[6] http://www.akomantoso.org/.
[7] http://eurocases.eu/.

Input/Output logic merges Input/Output logic [24], a well-known formalism in Deontic Logic (*i.e.,* a logic that expresses concepts like permissions, obligations, prohibitions), with the First Order Logic (FOL) for NLS proposed by [18], which is grounded on the concept of *reification*.

3.1 Reification and Input/Output Logic

Reification [10] is a concept that allows to move from standard notation in FOL such as "(give a b c)", asserting that "a" gives "b" to "c", to another notation in FOL "(give′ e a b c)", where "e" is the *reification* of the giving action. "e" is a FOL term denoting the giving event of "b" by "a" to "c". Reification is able to express a wide range of phenomena in NLS via simple *flat* FOL formulæ, which are basically conjunctions of atomic first-order predicates.

It has been argued [19,31,32] that flat logical formulæ for NLS have a twofold advantage. First, they allow to properly represent the semantics of several linguistic constructions which are hard to represent in other popular formalisms for NLS, such as Discourse Representation Theory (DRT) [23] or Minimal Recursion Semantics (MRS) [9]. Those formalisms introduce complex operators, *e.g.,* modal or causal operators, which take subformulæ as an argument. Nesting sub-formulæ within other (sub-)formulæ prevents several readings indeed available in NLS, such as cumulative readings (see [33]) or causality and concession (see [17,35]).

Secondly, flat formulæ enhance human readability and comprehension as well as the ease of controlling computational complexity. These two features are of course pivotal in the construction and debugging of large knowledge bases of formulæ, to be used in practical applications, as advocated in our methodology.

Let us consider a simple example: the representation of the sentence "Jack wants to eat an ice cream". In standard NLS formalisms, *e.g.,* DRT and MRS, this sentence is formalized by introducing a modal operator for representing the verb "want", *e.g.,* "want(... , ...)", which applies to a sub-formula representing the sentence "Exists an ice cream that Jack eats". For instance:

$$\mathsf{want}(\mathsf{Jack}, \exists_{ic}(\mathsf{eat}(\mathsf{Jack}, \mathsf{ic}))$$

In Hobbs's, the eating hypothetical action is reified into an eventuality that is inserted as argument of a FOL predicate *want′*:

$$\exists_e \exists_{e1} \exists_{ic} [\,(\mathsf{Rexist}\ e) \wedge (\mathsf{want}'\ e\ \mathtt{Jack}\ e_1) \wedge (\mathsf{eat}'\ e_1\ \mathtt{Jack}\ ic)\,]$$

Rexist is a special predicate used to assert which eventualities really exist in the context; in the example above, the wanting event really exists while the eating event does not. In the formulæ below, we will omit the Rexist predicates for space constraints; we will deem easy to understand, in those formulæ, which eventualities really exist and which do not.

On the other hand, Input/Output logic [24] is a well-known formalism in deontic logic, grounded on norm-based semantics. Norm-based semantics has been proposed as an alternative to deontic frameworks based on possible-world

semantics, such as STIT logic [20] and dynamic deontic logic [27]. It has been argued that norm-based semantics provides: (1) a more flexible handling of the well-known Jørgensen's dilemma [22], stating that, contrary to declarative statements, norms do not corresponds to truth values, *i.e.,* they cannot be described as true or false; (2) a straightforward and simple way to deal with moral conflicts and different kinds of permissions (see [25,29] to see how to deal with these in input/output logic); (3) a simpler and easy-to-control complexity; it has been argued [39] that compliance checking in Input/Output logic are coNP/NP hard and in the second level of the polynomial hierarchy, while STIT and dynamic deontic logic are respectively undecidable and EXPTIME-complete.

Input/output logic is not the only deontic framework that is not based on possible-world semantics; alternatives are imperative logic [16], prioritized default logic [21] and defeasible deontic logic [14]. A fine-grained comparison between the mentioned formalisms on the basis of norm-based semantics goes beyond the scope of this paper, in that it mostly lies on a theoretical level. Therefore, we address the interested reader to the relevant literature.

Input/output normative systems are triples $N = (O, P, C)$ of three sets of pairs: obligations (O), permissions (P), and constituency rules (C). A pair (a, b) corresponds to an if-then rule. The expression $(a, b) \in O$ reads "if a holds, then b is obligatory"; $(a, b) \in P$ reads "if a holds, then b is permitted"; and $(a, b) \in C$ corresponds to the standard FOL implication "$a \rightarrow b$". For space constraints, below we will not consider permissions (see [25]).

So far, Input/Output logic has been mostly studied from a theoretical point of view, with the elements a and b of the pairs being formulæ in propositional logic. Reified Input/Output logic is the first attempt to make Input/Output logic usable in practical applications in legal informatics, and to be used for representing norms from existing legislation available in natural language only.

3.2 Reified Input/Output Logic

As said above, reified Input/Output logic combines the advantages of the reified and of the Input/Output logic, first of all their respective formal simplicity. As argued in [36], simplicity appears to be a necessary feature for a logical formalism designed for application in legal informatics, in order to foster active collaboration of legal practitioners, usually having little expertise in logic, who can contribute to the building of large knowledge bases of formulæ. The methodology illustrated here represent the first attempt to build such a knowledge base, with respect to the data protection domain.

In reified Input/Output logic, a and b are formulæ as defined in [18]. Thus a sentence like "every bird who wants to eat is obliged to tweet" is represented as:

$$\forall_x(\exists_e \exists_{e_1}[\,(\mathsf{bird}\ x) \wedge (\mathsf{want}'\ e\ x\ e_1) \wedge (\mathsf{eat}'\ e_1\ x)\,],\ \exists_{e_2}[\,(\mathsf{tweet}'\ e_2\ x)\,]\,) \in O$$

Obligations are intended to populate the ABox of the knowledge base, *i.e.,* the set of assertive contextual statements. On the other hand, the set of definitions, axioms, and constraints on the predicates used in the ABox are part

of the TBox of the knowledge base, *i.e.,* the set of terminological declarative statements, also known as constitutive rules.

As said above, constitutive rules are expressed as standard FOL implications. For the sake of readability, below we will assert them as such (*i.e.,* in the form "$a \rightarrow b$"), rather than as pairs in the form (a, b). For instance, the TBox could contain a FOL implication specifying that all birds fly:

$$\forall_x(\,(\mathsf{bird}\ x) \rightarrow \exists_e[\,(\mathsf{fly}'\ e\ x)\,]\,) \in C$$

Finally, following [18], in reified Input/Output logic the implications populating the TBox can be made defeasible via a mechanism drawn from Circumscriptive Logic [26]. The antecedent of the implications can include an extra predicate that can be *assumed* or not. For instance, the previous implication can be made defeasible by adding a predicate "$assumption_1$":

$$\forall_x(\,(\,(\mathsf{bird}\ x)\ \wedge\ (\mathsf{assumption}_1\ x)\,) \rightarrow \exists_e[\,(\mathsf{fly}'\ e\ x)\,]\,) \in C$$

This formula reads as: "if x is a bird and it can be assumed that x is a 'normal' bird, then x flies". Not for all birds the assumption can be made. For instance, penguins are a special type of birds that do not fly. This is codified as:

$$\forall_x(\,(\mathsf{penguin}\ x) \rightarrow (\,(\mathsf{bird}\ x)\ \wedge\ \neg(\mathsf{assumption}_1\ x)\,)\,) \in C$$

If x is penguin, we derive that x is a bird but we cannot derive that x flies.

In our knowledge base, assumptions are used to model legal interpretations. Handling legal interpretations via defeasible mechanisms is a common practice in logical frameworks for legal informatics, such as [15]. A separate part of our knowledge base will specify which legal authorities adopt a certain assumption, which ones do not, and which ones either adopt it or not under certain conditions. By selecting certain legal interpretations, it is then possible to derive different conclusions from the knowledge base.

4 Methodology

The framework we propose offers a computer-aided methodology to analyze standards to make an argument of compliance with respect to a specific piece of legal text, and is schematically summarized in Fig. 1. Users (who may be lawyers, regulators, auditors, or other legal experts) access a digital and annotated XML representation of the normative texts (laws and standards). While browsing a document and selecting the relevant concepts, NLP and NLS tools help traverse the rest of the documents, find related terms, and recall previous correlations between them. Correlations from different sources have different degrees of importance, which need to be tracked using specific metadata. The framework implements a collaborative strategy to evaluate the stored correlations. The user's decisions are stored in the knowledge base, after being appropriately represented in a logic for legal semantics.

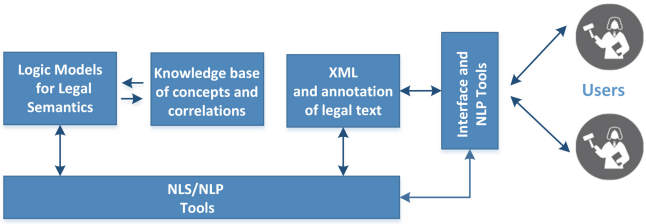

Fig. 1. The framework at a glance.

The framework, and in particular its knowledge base, does not pretend to be complete. It rather provides expert users with an updated knowledge that helps take autonomous and informed decisions, both when confirming the correlations the tool suggests and when choosing to define new correlations. The knowledge base is designed to support defeasible reasoning, *i.e.,* to tolerate (apparent) inconsistencies of different interpretations of terms, by overriding general assertions into more contextually-specific ones. Conflicts are especially frequent in legal interpretation, but they can generally be solved considering that the interpretation by higher-instance courts, such as the CJEU, prevails over lower ones. In order to cope with interpretations of different legal weights, which may supersede one another, the logic formalism that the framework embeds is defeasible: correlations can be updated, modified, rewritten and weighed. If conflicts do remain, the framework still embeds strategies that help the user take a decision.

As pointed out in Sect. 1, correlations can be further divided into two different categories: *formal* and *substantive* correlations.

Formal correlations entail a mere textual overlap between concepts. For example, formal correlations would allow us to observe that both the GDPR and the ISO 27018 standard use the term "notify" (see Sect. 5). Substantive correlations are more complex and entail the analysis of the actual meaning of terms. To assert a correlation of this kind, requirements must be met in a concrete way. Following the previous example, to assert a correlation between a provision of the GDPR and one of the standard concerning notification, it is necessary to verify the exact meaning of the term "notify" in the two texts.

The methodology follows three steps to build the correlations, involving both a legal and a technical approach. The legal approach is focused on the interpretation of the provisions of laws (the GDPR in our example) and standards (ISO 27018), whereas the technical approach consists of modelling those provisions into an ontology, and expressing the interpretation by means of logical formulæ.

Step 1: analysis of the provisions. The provisions of the law and of the security standard are analyzed by legal experts, who provide an interpretation of the terms used in the provisions and compare them in search of semantic correlations. There is no need for this interpretation to be final, as more interpretations can be added later, and old interpretations can be overridden by newer ones, but this start requires a significant manual activity.

This step entails two sub-steps. First, the legal documents need to be expressed in a machine-processable format. For the scope of this work, we have selected the `Akoma Ntoso` language[8]. This is an XML-based format that allows to easily navigate the documents and identify the relevant provisions in the legal text. The legal interpretations of the documents, on the other hand, are stored in separate documents, and contain a reference to their source, and another reference to the provision, or set of provisions, to which they apply. The use of unique namespaces and identifiers in `Akoma Ntoso` allows for a fine-grained model of a legal text and its interpretations.

To support the execution of this step, legal experts are assisted by external NLP procedures that suggest (semi-automatically and during the browsing of the documents) previous translations and correlations on the basis of the information currently stored in the knowledge base. Ultimately, it is the legal expert who must decide whether and how correlations need to be overridden. In that case, new correlations are added and annotated with the source which contributed to define it (*e.g.,* Court of Justice of the European Union, *Dapreco and Copreda Corp.*, C-XYZ/16). Implications by more authoritative sources override defeasible implications by less authoritative ones.

The interaction between the legal expert and the knowledge base will have to take into account that the former are not expected to have deep technical knowledge. This will be easily managed by means of appropriate front-ends and user interfaces that can be serve the purpose of viewing and modifying existing interpretations and their connections with specific provisions, and that of adding new interpretations, specifying metadata such as their nature, source, and validity in space and time.

Step 2: creation of legal ontologies. The legal interpretations are mapped onto legal ontologies of the law and the security standard. Legal ontologies [5] model the legal concepts, parties and stakeholders affected by the law, the duties and rights of each stakeholder, and the sanctions for violating the duties. As per ontologies in general, legal ontologies are expressed in a knowledge representation language. For this work, we chose the popular abstract language OWL, which can be serialized using various XML notations. For example, the OWL representation of the data protection ontology will contain concepts such as "controller", "data subject", "personal data", "processing" and so on.

The ontologies operate as the semantic base for the formal representation of the legal documents and their interpretation expressed in the form of logic formulæ. In other words, the ontologies represent the pivot of the methodology, as both normative documents and the objects of the logic (predicates and terms) are connected to the concepts expressed in the ontologies. In our final knowledge base, the connection will be implemented via `LegalRuleML` OASIS standard[9], along the same lines illustrated in [12].

[8] http://www.akomantoso.org/.
[9] https://www.oasis-open.org/committees/legalruleml.

The use of the OWL language guarantees that every concept in the ontologies is uniquely identified, thus creating an unambiguous connection between a logic formula, on the one side, and the concept or concepts to which it relates, on the other. Using this connection, it will be easy to perform searches based not solely on textual content, but also on semantic concepts and the relations between them. Consequently, a search for legal interpretations can overcome linguistic barriers such as typos or synonyms, and even be language-independent.

Work has started towards the creation of an improved ontology. In the interim, a preliminary version of a legal ontology for the GDPR has been defined already [4]. Albeit partial and based on an older version of the GDPR, it was designed to express the duties of the controller. As such, it can be used to find the correspondences between the requirements expressed in the GDPR and in security standards.

Step 3: generation of logic formulæ. The third and final step of the methodology consists of generating the logical formulæ representing the set of provisions in the law and the set of provisions in the security standard, as well as the implications between them. These formulæ are expressed in reified Input/Output logic [34]. An example is shown in the next section.

Associating textual provisions to logical formulæ amounts to converting ambiguous and vague terms into non-ambiguous items (predicates and terms). Words in the provisions are represented via predicates reflecting their vagueness. For example, the word "notify", included in the sample provisions used in Sect. 5, will be represented via the homonym predicate "notify". These "vague" predicates may be defined by adding implications and further constraints (axioms). Those implications will be *defeasible*, so that they can account for different legal interpretations. Predicates are associated with classes of the ontologies developed in Step 2 or with standard general-purpose ontologies/repositories belonging to the NLS literature, *e.g.,* Verbnet [38].

5 Generation of Logic Formulæ: Example

We exemplify step 3 of our methodology. This step, which lies at the core of the methodology, is the most technical, and more innovative than steps 1 and 2 which instead rely upon existing techniques. We use a provision from the GDPR and an article of the ISO 27018 security standard:

(*a*) GDPR, Article 33.2: *The processor shall notify the controller without undue delay after becoming aware of a personal data breach.*
(*b*) ISO 27018, Article A9.1: *The public cloud PII processor should promptly notify the relevant cloud service customer in the event of any unauthorized access to PII.*

The formulæ will include predicates reflecting the vagueness of the terms occurring in the sentences. Thus, for instance, the verb "notify", which occurs in both provisions, is formalized into an homonymous predicate "notify".

On the other hand, the provisions in the ISO 27018 use the term "PII" (Personally Identifiable Information) while the ones in the GDPR use the term "personal data". Although one might simply consider the two terms as synonyms (as suggested in ISO 27018, Article 0.1), and thus associate them with the same predicate, our methodology keeps them distinct, *i.e.,* it formalizes them via two different predicates "personalData" and "PII". An additional axiom is then added to the TBox, in order to correlate the two predicates:

$$\forall_x [\, (\mathsf{PII}\ x) \rightarrow (\mathsf{personalData}\ x) \,] \tag{1}$$

The implication can be made defeasible by adding an assumption as shown in Sect. 3. In that case, the normal rule is that PII is also considered personal data, unless there is a special exception which overrides the general rule.

In light of this, the GDPR provision in (a) is formalized as follows:

$$\begin{aligned}
&\forall_{e_b} \forall_x \forall_y (\\
&\quad \exists_{e_p} \exists_{e_c} \exists_{e_a} \exists_z [\, (\mathsf{dataProcessor}\ x) \wedge (\mathsf{dataController}\ y) \wedge (\mathsf{personalData}\ z) \wedge \\
&\quad (\mathsf{process}'\ e_p\ x\ z) \wedge (\mathsf{control}'\ e_c\ y\ z) \wedge (\mathsf{awareOf}'\ e_a\ x\ e_b) \wedge (\mathsf{dataBreach}\ e_b\ z) \,], \\
&\quad \exists_{e_n} [\, (\mathsf{notify}'\ e_n\ x\ y\ e_b) \wedge (\mathsf{nonDelayed}\ e_{nd}\ e_n) \,] \,)
\end{aligned} \tag{2}$$

In (2), "e_p", "e_c", "e_a", 'e_b", and "e_n" are variables referring to events. "e_p" is the event of processing the personal data "z" (patient) performed by the data processor "x" (agent). "e_c" is the event[10] of controlling the personal data "z" (patient) performed by the data controller "y" (agent). If x become aware ("e_a") of an event of data breach ("e_b") of the personal data "z", then x is obliged to notify it ("e_n") to the data controller and that event must be done with undue delay (predicate "nonDelayed", applied to the eventuality "e_n").

In (2), "notify" and "nonDelayed" are predicates whose meaning is subject to different legal interpretations. Recalling the difference between *formal* and *substantive* compliance outlined in Sect. 1, we note that the formalization in (2) only enforces formal compliance. The formula in (2) simply requires the data processor to notify data breaches without undue delay, but it does not specify *how* notifications should be performed for being legitimate.

For instance, the data processor could require the data controller to acknowledge the notification, in order to make sure it was received. Similarly, the processor could be required to avoid sending notifications of data breaches via standard paper mail, in that the time needed by the postal service to deliver the mail could be considered as an undue delay. It is up to judicial authorities to establish the substantive compliance of the obligation in (2).

Of course, we do not have the authority to decide whether (2) is performed in the proper way. In our work, we only aim at providing a methodology to keep track of all legal interpretations of the provisions. From a formal point of view,

[10] In this context, an event must not be considered as a specific occurrence happening at a given time, but as a wider concept encompassing the whole of the controller's activity.

the TBox must be enriched with axioms defining the conditions under which the predicates in the formula are true. Those axioms are defeasible, therefore they will contain *assumptions* that may be taken or not. And, it is possible to separately assert that certain assumptions are taken to be either true or false by certain legal authorities, possibly under certain further conditions (see below).

For instance, by assuming that email with electronic signature is a proper and prompt means to notify the data controller, the following (defeasible) axiom is added to the TBOX.

$$\forall_x \forall_y \forall_{e_1} \forall_{e_2} [\,(\,(\mathsf{sendEmailWithES}\ e_1\ x\ y\ e_2) \wedge (\mathsf{assumption}_2\ e_1)\,) \rightarrow \exists_{e_n}(\,(\mathsf{notify}'\ e_n\ x\ y\ e_2) \wedge (\mathsf{nonDelayed}'\ e_{nd}\ e_n)\,) \tag{3}$$

Formula (4) models the ISO 27018 provision in (*b*):

$$\forall_e \forall_x \forall_y \forall_z \forall_{e_p} \forall_{e_c} \forall_{e_b} (\,(\,(\mathsf{PIIProcessor}\ x) \wedge (\mathsf{PIIController}\ y) \wedge (\mathsf{PII}\ z) \wedge (\mathsf{process}'\ e_p\ x\ z) \wedge (\mathsf{control}'\ e_c\ y\ z) \wedge (\mathsf{access}'\ e_a\ z)\,) \wedge (\mathsf{unauthorized}\ e_a)\,), \exists_{e_n} [\,(\mathsf{notify}'\ e_n\ x\ y\ e_a) \wedge (\mathsf{promptly}'\ e_{np}\ e_n)\,]\,) \tag{4}$$

As it was done for formalizing (*a*) into (2), the formula introduces predicates that reflect the generic terms used in the text. With an important exception: we formalized "the relevant cloud service customer" via the predicate "PIIController". According to ISO 27018 the cloud service customer can be either a natural person, "PII principal" or a "PII controller", which processes the PII relating to PII principals.[11]

Therefore, building our example on the specific provision of the GDPR, the cloud service provider ("PII Processor" - "data processor") handling data of natural persons ("PII Principals" - "data subjects") will have to notify the organization on behalf of which it processes the data ("PII controller" - "data controller") of occurring incidents ("any unauthorized access to PII" - "personal data breach") in a specific time frame ("without undue delay after becoming aware" - "promptly").

The final ingredient needed for (4) and (2) are axioms relating to the predicates occurring in both, similar to the axiom in (1), which state that PII is by default considered as personal data:

$$\begin{aligned} &\forall_x [\,(\mathsf{PIIProcessor}\ x) \rightarrow (\mathsf{dataProcessor}\ x)\,], \\ &\forall_x [\,(\mathsf{promptly}'\ x\ y) \rightarrow (\mathsf{nonDelayed}'\ x\ y)\,], \\ &\forall_x [\,(\mathsf{PIIController}\ x) \rightarrow (\mathsf{dataController}\ x)\,], \\ &\forall_e \forall_z [\,(\,(\mathsf{access}'\ e\ z) \wedge (\mathsf{unauthorized}\ e)\,) \rightarrow (\mathsf{dataBreach}\ e\ z)\,] \end{aligned} \tag{5}$$

The axioms in (5) are quite intuitive. For instance, the first one states that any entity that is considered a PII processor according to ISO 27018 is also a data processor with respect to the GDPR.

[11] ISO 27018, Article 0.1: "The cloud service customer, who has the contractual relationship with the public cloud PII processor, can range from a natural person, a 'PII principal', processing his or her own PII in the cloud, to an organization, a 'PII controller', processing PII relating to many PII principals".

It is easy to verify that every tuple of variables "e", "x", "y", "z", "e_p", "e_c", and "e_a" that satisfies formula (4) also satisfies formula (2).

Again, the correlations in (5) can be made defeasible, in order to encompass different legal interpretations of the provisions. For instance, the fact that an unauthorized access *is* a data breach depends on the legal interpretation (which can vary according to the court or authority). According to Article 4(12) of the GDPR, which is essentially equivalent to Article 3.1 of ISO 27018, a data breach is defined as follows:

GDPR, Article 4(12): *'personal data breach' means a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to personal data transmitted, stored or otherwise processed.*

To encompass the different legal interpretations of Article 4(12), we enrich the last implication in (5) via a predicate stating that the eventuality e is both an unauthorized access and a data breach under general conditions, but there might be exceptions where it is an unauthorized access, but not a data breach:

$$\forall_e \forall_z [\,(\,(\mathsf{access}'\ e\ z)\ \wedge\ (\mathsf{unauthorized}\ e)\ \wedge\ (\mathsf{assumption}_\mathsf{e}\ e)\,) \rightarrow (\mathsf{dataBreach}\ e\ z)\,] \tag{6}$$

Then, in the knowledge base we separately assert that there are several possible parallel interpretations concerning the assumption. In particular, a court or authority might decide that:

- the assumption holds as true;
- the assumption does not hold;
- the assumption can hold or not, depending on the conditions.

As an example, we can assume fictitious case law, *not* pertaining to actual legal decisions, for the sole purpose of illustrating the methodology. We make up three decisions as follows:

- Italian *Corte di Cassazione, sezione civile, 12530/2012*;
- Spanish *Audiencia provincial de Toledo, n. 57/2016, 2/12/2016*;
- French *Tribunal de Grande Instance d'Avignon, décision du 17/04/2016*. In this case, we assume that the specific conditions examined by the Tribunal consisted in the company *Alpha* performing a security test on an IT system; even if unauthorized accesses indeed took place, those cannot be taken as data breaches in that they were part of the security test.

Table 1 displays the interpretations by the three sources, and the way they are represented by means of a formula in Reified Input/Output logic. Depending on the legal interpretation of Article 4(12) that is selected, different inferences are enabled on the knowledge base.

`LegalRuleML` provides tags to represent different legal interpretations of the logical items (see [3]). In our future work, we plan to enrich the knowledge base in `LegalRuleML` with legal interpretations, as soon as they come available.

Table 1. Samples of legal interpretations and their translations.

Source	Interpretation	Formula
Cassazione civile	An unauthorized access is a data breach	$(\mathsf{assumption_e}\ e)$
Audiencia de Toledo	A data breach requires not only an unauthorized access, but also a breach of security and a causal connection between them	$\neg(\mathsf{assumption_e}\ e)$
Tribunal d'Avignon	When a breach of security is part of security tests, such as the ones performed by the company *Alpha*, leading to unauthorized access, it is not considered as a data breach	$\forall_e(\exists_z\exists_{e_t}[\,((\mathsf{access}\ e\ z) \wedge (\mathsf{unauthorized}\ e) \wedge (\mathsf{partOf}\ e\ e_t) \wedge \mathsf{securityTest}\ e_t))\,] \rightarrow \neg(\mathsf{assumption_e}\ e))$

6 Discussion and Conclusion

This paper deals with the complex problematic of complying with abstract legal rules that usually give little guidance as to the implementation of technical measures to address the requirements therein. At the same time, it intends to show the benefits of standards as regards the argument of compliance they can create in favour on the implementing party once a bridge between the law and the standards has been created.

This paper advances a logic formalism whereby correlations between provisions of the law and those of a standard can be expressed, and then introduces a methodology to help build such a bridge in a semi-automated way. The paper exemplifies the methodology in the context of data protection laws (in particular, the GDPR) and security standards (ISO 27018), two domains that significantly overlap given that security is an inherent part of data protection.

By following the illustrated methodology, one can build a machine-processable *knowledge base* of logic formulæ that model and store relevant concepts from a law and a standard together with their possible formal correlations. The knowledge base, which will be collaboratively accessible, will have its records updated and labelled considering the outcomes of specific auditing processes or decision of the courts; in time, it will embed substantive correlations, which can serve as a base for a legal argument for compliance. The logic into which we translate the correlations is defeasible, allowing them be overridden.

The methodology herein is currently a work in progress and not fully implemented yet: this will require the definition of a detailed taxonomy of concepts extracted from the law and the security standard. As the work presented herein is part of a larger research project, its extension is envisioned along various research directions.

Several technical challenges related to building and updating the knowledge base are raised. The translations from natural language to logical formulæ must be uniform for similar text excerpts. To achieve this, we must overcome the

limitations of a manual translation, which would be time-consuming and error-prone. For this reason, our work must rely on NLP technologies. However, even at the best of their performances, current NLP algorithms are still unable to automatically carry out the translation with a reasonable level of accuracy, so we advocate a *semi-automatic* translation of the provisions. Similar approaches are applied to translations in general, where translators are helped by collaborative tools such as the "SDL Trados Studio"[12], which suggests, via text-similarity or pattern-matching NLP techniques (see [28,37]), how to translate a sentence on the basis of the translations of similar sentences that the translators have previously stored in the tool. Inspired by that approach, we will develop an enhanced text editor to assist the manual translation of provisions into formulæ. For each provision, the editor will display the translations of similar provisions found via NLP procedures applied to the provisions already stored in the knowledge base, in order to induce uniform translations for similar provisions.

In its current structure, the knowledge base does not rely on the connection between the legal interpretations and the ontology of concepts. Without such a connection, the knowledge base is not yet mature to retrieve legal interpretations. To achieve this goal an additional layer will be needed, some formalism that creates a link between the interpretations and their related concepts. A potential candidate for this purpose has been identified in the XML language LegalRuleML [2].

Additionally, we are aware that the knowledge base must be consistent, *i.e.*, without contradictions, even after having applied the defeasibility measures. To check for consistency, we plan to store formulæ using a XML-based data model, and employ/extend reasoners to monitor the consistency of the knowledge base, whenever new formulæ are added to it.

Finally, for a solid population of the knowledge base, the methodology would greatly benefit from a close interaction with legal authorities. We are envisioning such an interaction in the near future.

Acknowledgments. This work is financed by the Luxembourg National Research Fund (FNR) CORE project C16/IS/11333956 "DAPRECO: DAta Protection REgulation COmpliance". Robaldo has received funding from the EU Horizon 2020 Programme for Research and Innovation under the Marie Skłodowska-Curie grant agreement No. 690974 for the project "MIREL: MIning and REasoning with Legal texts".

References

1. Arora, C., Sabetzadeh, M., Briand, L.C., Zimmer, F.: Automated checking of conformance to requirements templates using natural language processing. IEEE Trans. Software Eng. **41**(10), 944–968 (2015)
2. 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 (ICAIL), pp. 3–12. Association for Computing Machinery (ACM), June 2013

[12] http://www.translationzone.com/products/trados-studio/.

3. Athan, T., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: LegalRuleML: design principles and foundations. In: Faber, W., Paschke, A. (eds.) Reasoning Web 2015. LNCS, vol. 9203, pp. 151–188. Springer, Cham (2015). doi:10.1007/978-3-319-21768-0_6
4. Bartolini, C., Muthuri, R., Santos, C.: Using ontologies to model data protection requirements in workflows. In: Proceedings of the 9th International Working on Juris-informatics (JURISIN). pp. 27–40, extended version to be published in LNAI book, November 2015
5. Benjamins, V.R., Casanovas, P., Breuker, J., Gangemi, A. (eds.): Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications. LNCS (LNAI), vol. 3369. Springer, Heidelberg (2005)
6. Boella, G., Di Caro, L., Humphreys, L., Robaldo, L., Rossi, R., van der Torre, L.: Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. Artificial Intelligence and Law to appear (2016)
7. Boella, G., Di Caro, L., Graziadei, M., Cupi, L., Salaroglio, C.E., Humphreys, L., Konstantinov, H., Marko, K., Robaldo, L., Ruffini, C., Simov, K., Violato, A., Stroetmann, V.: Linking legal open data: breaking the accessibility and language barrier in European legislation and case law. In: Proceedings of the 15th International Conference on Artificial Intelligence and Law. ICAIL 2015, pp. 171–175. ACM, New York (2015)
8. Boella, G., Di Caro, L., Rispoli, D., Robaldo, L.: A system for classifying multi-label text into eurovoc. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. ICAIL 2013, pp. 239–240. ACM, New York (2013)
9. Copestake, A., Flickinger, D., Pollard, C., Sag, I.A.: Minimal recursion semantics: an introduction. Res. Lang. Comput. **3**(2), 281–332 (2005)
10. Davidson, D.: The logical form of action sentences. In: Rescher, N. (ed.) The Logic of Decision and Action. University of Pittsburgh Press, Pittsburgh (1967)
11. De Hert, P., Papakonstantinou, V., Kamara, I.: The cloud computing standard ISO/IEC 27018 through the lens of the EU legislation on data protection. Comput. Law Secur. Rev. **32**(1), 16–30 (2016)
12. Dimyadi, J., Governatori, G., Amor, R.: Evaluating legaldocml and legalruleml as a standard for sharing normative information in the AEC/FM domain. In: Proceedings of the Lean and Computing in Construction Congress (LC3) (to appear, 2017)
13. Giurgiu, A., Lommel, G.: A new approach to EU data protection. Crit. Q. Legislation Law **97**(1), 10–27 (2014)
14. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Computing strong and weak permissions in defeasible logic. J. Philos. Logic **42**(6), 799–829 (2013). http://dx.doi.org/10.1007/s10992-013-9295-1
15. Governatori, G., Rotolo, A., Sartor, G.: Deontic defeasible reasoning in legal interpretation. In: Atkinson, K. (ed.) The 15th International Conference on Artificial Intelligence & Law, San Diego, USA (2015)
16. Hansen, J.: Prioritized conditional imperatives: problems and a new proposal. Auton. Agent. Multi-Agent Syst. **17**(1), 11–35 (2008)
17. Hobbs, J.R.: Toward a useful notion of causality for lexical semantics. J. Semant. **22**, 181–209 (2005)
18. Hobbs, J.R.: Deep lexical semantics. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 183–193. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78135-6_16

19. Hobbs, J.: The logical notation: ontological promiscuity. In: Chapter 2 of Discourse and Inference (1998). http://www.isi.edu/~hobbs/disinf-tc.html
20. Horty, J.: Agency and Deontic Logic. Oxford University Press, New York (2001)
21. Horty, J.: Reasons as Defaults. Oxford University Press, New York (2012)
22. Jørgensen, J.: Imperatives and logic. Erkenntnis **7**, 288–296 (1937)
23. Kamp, H., Reyle, U.: From Discourse to Logic: An Introduction to Model-Theoretic Semantics, Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers, Dordrecht (1993)
24. Makinson, D., van der Torre, L.W.N.: Input/output logics. J. Philos. Logic **29**(4), 383–408 (2000)
25. Makinson, D., van der Torre, L.: Permission from an input/output perspective. J. Philos. Logic **32**, 391–416 (2003)
26. McCarthy, J.: Circumscription: A form of nonmonotonic reasoning. Artif. Intell. **13**, 27–39 (1980)
27. van der Meyden, R.: The dynamic logic of permission. J. Logic Comput. **6**, 465–479 (1996)
28. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence. AAAI 2006, vol. 1, pp. 775–780. AAAI Press (2006). http://dl.acm.org/citation.cfm?id=1597538.1597662
29. Parent, X.: Moral particularism in the light of deontic logic. Artif. Intell. Law **19**(2–3), 75–98 (2011)
30. Reding, V.: The upcoming data protection reform for the European Union. Int. Data Priv. Law **1**(1), 3–5 (2011)
31. Robaldo, L.: Independent set readings and generalized quantifiers. J. Philos. Logic **39**(1), 23–58 (2010)
32. Robaldo, L.: Interpretation and inference with maximal referential terms. J. Comput. Syst. Sci. **76**(5), 373–388 (2010)
33. Robaldo, L.: Distributivity, collectivity, and cumulativity in terms of (in)dependence and maximality. J. Logic, Lang. Inf. **20**(2), 233–271 (2011)
34. Robaldo, L., Humphreys, L., Sun, L., Cupi, L., Santos, C., Muthuri, R.: Combining input/output logic and reification for representing real-world obligations. In: Post-proceedings of the 9th International Workiung on Juris-informatics. Lecture Notes in Artificial Intelligence (2016)
35. Robaldo, L., Miltsakaki, E.: Corpus-driven semantics of concession: where do expectations come from? Dialogue Discourse **5**(1), 1–36 (2014)
36. Robaldo, L., Sun, X.: Reified input/output logic: Combining input/output logic and reification to represent norms coming from existing legislation. J. Logic Comput. (to appear, 2017)
37. Robaldo, L., Caselli, T., Russo, I., Grella, M.: From Italian text to TimeML document via dependency parsing. In: Gelbukh, A. (ed.) CICLing 2011. LNCS, vol. 6609, pp. 177–187. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19437-5_14
38. Schuler, K.K.: Verbnet: a broad-coverage, comprehensive verb lexicon. Ph.D. thesis, Philadelphia, PA, USA, aAI3179808(2005)
39. Sun, X., Robaldo, L.: On the complexity of input/output logic. J. Appl. Logic (to appear, 2017)
40. Vibert, H., Jouvelot, P., Pin, B.: Legivoc - connectings laws in a changing world. J. Open Access Law **1**(1), 165–174 (2013)