Legal Knowledge and Information Systems
E. Schweighofer (Ed.)

doi:10.3233/FAIA210331

Identification of Contradictions in Regulation

Michał ARASZKIEWICZ [a,1], Enrico FRANCESCONI [b] and Tomasz ZUREK [c]

[a] *Department of Legal Theory, Faculty of Law and Administration, Jagiellonian University, Ul. Gołebia 24, 31-007 Krakow, Poland*

[b] *Italian National Research Council (Florence, Italy) and European Parliament (Luxembourg)*

[c] *Institute of Computer Science, Maria Curie Sklodowska University, Ul. Akademicka 9, 20-033 Lublin, Poland*

Abstract. This paper presents a Semantic Web–based model for detecting contradictions in regulations. We introduce a conceptual model of contradictions and, on the basis of this model, a knowledge representation–based model is used, which is able to represent the semantics of provision types and related properties. The usefulness of the model is shown through an example.

Keywords. Frames, Legal Interpretation, Legislative errors, Knowledge Representation

1. Introduction[2]

Legislative drafting is a complex task. It requires not only linguistic competence, but also thorough knowledge of the regulated domain, as well as of the legal/theoretical assumptions concerning the legal system, including its completeness, consistency, and lack of redundancy [1,2]. One type of error occurs when the regulation is encumbered with contradictions. Legislators should ensure not only the internal consistency of the regulation but also a lack of contradiction with hierarchically higher regulations. While some contradictions may be eliminated through legal reasoning, this is not always possible or desirable. The lack of consistency in regulation is typically not straightforward. Therefore, it is worthwhile to provide a legislative tool representing the structure and semantic content of the drafted provisions - to ascertain whether an inconsistency actually exists. Knowledge representation tools have been fruitfully used to represent legislation ([3,4,5]). However, the development of such a model is a complex task, also because it requires expertise in the regulated domain of law. In this paper, we do not develop a formal representation of legal regulation but, rather, a conceptual framework that may be used by a legislator to formalize a drafted text and analyze its features, in particular to detect legislative errors. Our proposal fits well with the idea of designing systems that

[1] Corresponding Author: michal.araszkiewicz@uj.edu.pl

[2] The paper presents the results of the project UMO-2018/29/B/HS5/01433 entitled Legislative Errors and Comprehensibility of Legal Text.

assume human–machine interaction, taking into account the strengths and weaknesses of each party.

We present an ontology-based model to describe the structure and the semantic content of legal provisions. The purpose of this model is to support the process of legislative drafting. Let us assume that a legislator has already prepared a draft of a set of legal provisions in natural language. The tool enables the legislator to:

- Represent the content of legal provisions in a systematized manner, taking into account the defined categories of entities, such as legal subjects, legal objects, different types of deontic relations, etc.
- Use ontological reasoning facilities to detect errors, in particular contradictions, in regulation;
- Evaluate whether the regulation under analysis is acceptable with regard to the assumed criteria or if it needs amending.

We argue that this approach enables us to identify certain non-trivial types of legislative errors, including potential and actual contradictions.

2. Conceptual Framework

2.1. Introductory Remarks

The main part of any normative piece of legislation is composed of legal provisions, namely elementary, sentence-like linguistic expressions systematized as sections, articles, or points. We focus on the representation of the structure and semantic content of legal provisions. Each representation of a legal provision in a knowledge-based model is an interpretation of this provision. For the purposes of our project, we understand the interpretation as a result of the reasoning process performed by a human agent.

The model, based on our earlier work ([6,7]) enables the representation of the structure and semantic content of legal text for the purpose of identifying legislative errors. A legislative error may be understood as a violation of one or more criteria of rational lawmaking, such as aiming at consistency, completeness, lack of redundancy or axiological coherence of the created regulation. The legislator should detect potential violations of these criteria during the process of legislative drafting and consider whether these violations can be eliminated through legal reasoning in the first place. The application of different interpretive arguments, including linguistic, systemic, and functional ones, [8] may lead to the resolution of apparent problems and, therefore, the drafted text may not need amending. However, not all the results of such remedial interpretation are acceptable, as there exist some interpretive rules that constrain the space of acceptable interpretations, discussed in [6]. In some cases, the identification of a legislative error does not involve complex interpretive reasoning because some violations are ascertainable through a literal interpretation of the considered provisions, and no other interpretation is conceivable. This does not mean that detecting a violation is always a trivial task. In some cases, the structure of the drafted provision needs to be clearly represented for a violation to become apparent. In this paper, we focus on one particular type of legislative error: contradictions in regulations.

2.2. The notion of contradiction between provisions

We argue that the nature of contradiction between legal provisions requires in-depth analysis of the features and scope of the provisions because it is relatively seldom that two provisions lead to strictly opposite conclusions (p vs $\neg p$ or $perm(p)$ vs $proh(p)$).

First, following [6], we assume that every regulation creates a certain legal relation between particular entities. For example, by *Bearer*, we denote an agent which is an addressee of the regulation, by $Object$, we denote the object of a regulation, by $Action$, we denote the action regulated by the provision, and by $Counterpart$, we denote the agent functioning as a counterpart of the regulation. Second, we have to notice that every concept has its own scope and that scopes of different concepts can be in different relations, e.g., inclusion. We have to assume also that the analyzed sets of provisions can contradict an existing regulation not only in their whole scope but also in a subset of the regulated entities. To represent the idea of the broader and narrower scopes of two concepts, we introduce operator $\sqsubseteq$, where $X \sqsubseteq Y$ denotes that every entity represented by X is also represented by Y.

To represent the relation defined by a legal provision (or set of legal provisions), we introduce a predicate $rel()$ representing such a relation. The first argument of this predicate will be a type of legal relation created by the analyzed provision; other arguments will represent the concepts discussed above used in the wording of a legal provision (for example, the second argument will represent *Bearer*, the third $Object$, etc.).
If we introduce:

- a set of types of legal provisions ($Type = \{perm, proh, obl, ...\}$);
- a relation of opposition represented by pairs of opposite types of provisions $Opposite = \{< perm, proh >, < obl, proh >, ...\}$ (relations containing those pairs are potentially contradictory);
- a predicate $rel(type, X_i, X_j, ...)$, where $X_i, X_j, ...$ represent various concepts;

then the conflict will appear if:

- one set of provisions defines $rel(type_l, X_i, X_j, ...)$,
- second set of provisions defines $rel(type_k, X_m, X_n, ...)$ [3]
- there will be a set of concepts $x_i, x_j, ...$ s.t. $x_i \sqsubseteq X_i$, $x_i \sqsubseteq X_m$, $x_j \sqsubseteq X_j$, $x_j \sqsubseteq X_n$, ... and
- relations $(type_l, type_k)$ are opposite i.e. $< type_l, type_k > \in Opposite$ (for example $type_l = perm$, $type_k = proh$)

The above model can be illustrated by a very simple example. Suppose we have two provisions: (1) vehicles are forbidden to enter the park and (2) ambulances are allowed to enter the park. Additionally, we know that *vehicles* is a broader concept than *ambulance* (i.e., every ambulance is vehicle). To represent the example, we assume that predicate $rel()$ has the following arguments: 1) type, 2) *Bearer*, 3) $Object$, and 4) *Action*. The example can be modelled as follows: $rel(proh, vehicles, park, enter), rel(perm, ambulance, park, enter), ambulance \sqsubseteq vehicle$ Since *ambulance* is a narrower concept than *vehicles* and $< proh, perm > \in Opposite$, then ambulances are simultaneously prohibited (on the

[3] Note that in order to correctly detect contradictions, the arguments in both relations should be in the same order, i.e. *Bearer* should be the second argument in both relations

basis of provision 1) and permitted (on the basis of provision 2), which is an obvious contradiction[4].

3. The description of the ontology and tools

The presented conceptual model for detecting errors and inconsistencies in legislative draft bills can be implemented in the Semantic Web using the Provision Model, introduced by [9]. The Provision Model aims to provide a formal representation of textual objects, subject to a given interpretation. For this reason, it represents a model of legal rules that can be effectively used to detect legislative errors. The Provision Model has been used in the literature to provide advanced legal information retrieval and reasoning services based on the semantics of legal rules [10], but primarily, it has been targeted at implementing model-driven legislative drafting facilities [11]. The aim of this approach is to improve the quality of legal texts and ensure the maintenance of legal information by monitoring the impacts of new regulations on the legal system (including the consistency and completeness of new provisions within the same text or of different texts within the same legal order as well as between different legal orders, as in the case of a European directive and its transposition in national legislation).

3.1. The Provision Model

The Provision Model represents a knowledge model of the rules conveyed in legislative texts. It is organized into provision types and properties. It describes constitutive and regulative rules independently of the domain in which they operate, as well as rules on rules, namely, amendments (which may be related to the textual content, the timing of the enactment of the rule, or the scope within which the rules operate).

Examples of regulative rules, in terms of provision types, are shown in Fig. 1; they are represented as classes of the Provision Model ontology. In particular, the provision types prv:Obligation and prv:Permission are reported. They are associated with specific properties; for example, for prv:Obligation, the specific properties are prv:hasObligationBearer, prv:hasObligationAction, prv:hasObligationCondition, and prv:hasObligationCounterpart (see Fig. 1). This model can be used to provide semantic annotation of legal texts and is able to reflect the lawmakers' directions in a machine-readable format.

Note that the properties of the provision types represent the arguments of relation $rel()$ described in section 2.4. In particular, prv:hasObligationBearer and prv:hasPermissionBearer represent the argument $Bearer$; prv:hasObligationAction, and prv:hasPermissionAction represent the argument $Action$, etc.

As the Provision Model describes legislative rules independently of the domain in which they operate, a complete representation of a legal rule instance typically contained in a textual paragraph can be obtained by combining the Provision Model with the controlled vocabularies represented in knowledge organization systems capable of providing additional information on the entities of the regulated domain [12,13]. Controlled vocabulary terms can be used as provision property values to be used for semantic annotation of legal provisions. This can be useful to disambiguate and harmonize different possible literal interpretations of textual wording in the law and minimize the risk of interpretive doubts.

[4] This example can be easily solved by using *lex specialis...*, but this is outside of the scope of this paper

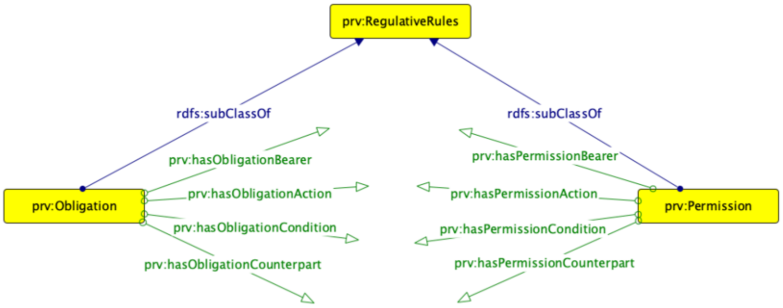

Figure 1. Provision properties in Permission and Obligation

3.2. *Logical and technical relations in the Provision Model*

The Provision Model is also endowed with axioms that are able to describe logical relations between provisions. *Logical relations* are relations between provisions that are necessary from a logical point of view, such as the classical Hohfeldian relations between Right and Duty as well as No-right and Privilege and the opposite relations between Right and No-right as well as Duty and Privilege [10].

In our conceptual model of contradiction (see section 2.4), logical relations are represented by pairs from set $Opposite$, which provides a set of relations that may constitute potential contradictions. *Technical relations* between provisions, on the other hand, are relations that are not necessary from a logical point of view, but they derive from considerations of legislative techniques. This means that they are possible and can be identified in legislative texts, provided that the legislative drafter follows specific legislative techniques in expressing these provisions. An example of such relations is the one existing between a *Definition*, introducing a concept identified by the attribute *Definiendum*, and all the other provisions having, as an attribute value, the value of such *Definiendum*. Another example is the relation between the *Duty* of a specific *Bearer* to accomplish a specific *Action* toward a given *Counterpart* as well as the *Procedure* to fulfill it. In the conceptual model of contradiction, technical relations are represented by arguments of predicate $rel()$. While *logical relations* can be described by axioms at the level of the Provision Model and are inherited by all the related instances (see [10] and [7]), *technical relations* can be identified and described at the level of specific provision instances only, because they are linked to their content. As reported in [9], technical relations between provisions can be established directly by the legislator through references or can be deduced by reasoning over provisions content, expressed by provisions attribute values.

Therefore, contradictions can be detected by checking *technical relations* between provisions, namely, relations that can be revealed only by reasoning about the semantics and content of the provision instances (including axioms if needed) and not by reasoning about provisions semantics and related axioms only.

4. Example of provision semantic representation

We apply the presented model to an example involving a contradiction between a European Union directive, on the one hand, and a Member State law, on the other hand.

Directives are instruments which bind the Member State as to the result to be achieved but leave to the national authorities the choice of form and methods (TFEU Art. 288)[5]. Therefore, directives need to be properly transposed into the legal system of the Member State. Incorrect transposition of a directive may lead to diverse legal consequences. Due to the general wording of directives, it is sometimes difficult to determine whether a particular Member State regulation is a proper transposition. The model developed here may be fruitfully used to detect contradictions between national and EU regulations by taking into account *technical relations*. The following example is based on the Judgment of the Court of 4 May 2006, case C-508/03. Let us consider Art. 2 Par. 1 in the Council Directive of 27 June 1985:

> *Member States shall adopt all measures necessary to ensure that, before a planning permission is given, urban development projects likely to have effects on the environment are made subject to an assessment with regard to their effects.*

From the Provision Model viewpoint, this paragraph can be classified as an prv:Obligation for Member States "to adopt all measures necessary to the purpose that before a planning permission is granted, environmental impact assessment of a project is performed." The prv:Obligation to be implemented can be expressed in an attribute–value pair notation, according to the Provision Model, as follows:

prv:Obligation

prv:hasObligationBearer =	'Competent authority'
prv:hasObligationAction =	'to assess effects on environment'
prv:hasObligationCondition =	'before planning permission'
prv:hasObligationObject =	'project'

Note that, in this example, property values are reported as literals for the sake of readability. However, as previously discussed, they are typically terms in controlled vocabularies[6]. Domain terms from a controlled vocabulary are important in the context of detecting technical relations between provision instances as they contribute to disambiguation and to comparison between domain entities.

To provide a Provision Model representation of the above EU directive paragraph, let's consider that the relevant concepts introduced by this directive are organized in a controlled vocabulary called ex:EnvironmentalImpactAssessementVoc, described in SKOS[7], with a namespace ex: (which just stands for "example") that includes concepts like ex:MemberState, ex:PlanningPermission, ex:Project, and related skos:prefLabel.The model graph is provided in Fig. 2; the graph also shows the relations skos:inScheme between the vocabulary terms and the related vocabulary top-concept ex:EnvironmentalImpactAssessementVoc (instance of a skos:ConceptScheme).

The RDF representation of such a graph related to the obligation in the cited directive, having [docURI] as URI, is therefore the following:

```
<rdf:Description rdf:about="[docURI]">
   <rdf:type rdf:resource="prv:Obligation"/>
   <prv:hasObligationBearer rdf:resource="ex:CompetentAuthority"/>
   <prv:hasObligationAction rdf:resource=
```

[5] Treaty on the Functioning of the European Union (https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=LEGISSUM:4301854)

[6] Examples of controlled vocabularies can be found at https://op.europa.eu/en/web/eu-vocabularies/controlled-vocabularies

[7] Simple Knowledge Organization System (https://www.w3.org/2004/02/skos/)

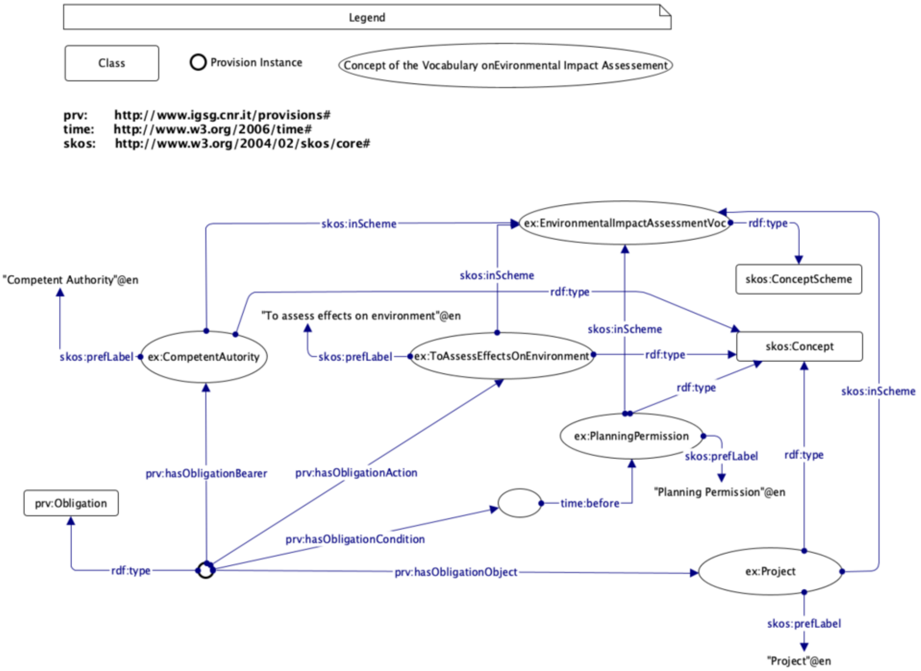

Figure 2. Semantic representation of EU Directive 27 June 1985 art. 2 par. 1

```
                                          "ex:ToAssessEffectsOnEnvironment"/>
   <prv:hasObligationCondition>
                 <time:before rdf:resource="ex:PlanningPermission"/>
   </prv:hasObligationCondition>
   <prv:hasObligationObject rdf:resource="ex:Project"/>
</rdf:Description>
```

Note that:

- the Provision Model imports the W3C Time Ontology[8] allowing the expression of time-wise relations between domain entities.
- this formal representation of legal provisions is used in our approach to identify errors, particularly contradictions/inconsistencies in a set of legal rules. Errors may be revealed according to the interpretative canon used to describe provision semantics, in this case, a literal interpretative canon.

5. Approach for detecting contradictions

To show the approach to detecting errors/inconsistences in legal rules, let's consider the case of a set of provisions in the U.K. legal order, aimed to implement the provision of the EU Directive of 27 June 1985 art. 2 par 1, introduced in Section 4. The UK provisions are as follows:

[8] http://www.w3.org/2006/time#

(1) "outline planning permission" means planning permission granted, in accordance with the provisions of a development order, with the reservation for subsequent approval by the local planning authority or the Secretary of State of matters not particularized in the application ("reserved matters").

(2) the competent authority cannot grant planning permission unless it has first taken the environmental information into consideration and states in its decision that it has done so.

(3) In the case of outline planning permission, an environmental impact assessment can be carried out only at the initial stage of granting such permission and not at the later stage of approval of the reserved matters.

Paragraph (1) introduces the expression "outline planning permission," which represents a narrower concept than the concept "planning permission" that is introduced by the transposed EU directive. In paragraph (3), the implementing measure reproduces the obligation to assess the effects on the environment before planning permission is given but makes reference to the concept "outline planning permission," opening the possibility of not assessing the environmental impact at a later stage when "reserved matters" are approved.

This represents a violation of the EU directive, which we intend to detect automatically. To do that, first, we provide a semantic representation of the implementing provisions. Here, just the representation of paragraph (3) is reported because it is sufficient to detect the inconsistency. The attribute–value pair notation of the paragraph is as follows:

prv:Obligation	
prv:hasObligationBearer =	'Competent authority'
prv:hasObligationAction =	'to assess effects on environment'
prv:hasObligationCondition =	'before outline planning permission'
prv:hasObligationObject =	'Project'

The knowledge graph of both legal provisions (the provision of the EU directive described in Section 4 and the implementing U.K. provision) are reported in Fig. 3, including the concept ex:OutlinePlanningPermission in the vocabulary of the domain concepts. The RDF representation of paragraph (3) of the U.K. implementing measure, having [docURI-UK] as URI is, therefore, the following:

```
<rdf:Description rdf:about="[docURI-UK]#par3">
   <rdf:type rdf:resource="prv:Obligation"/>
   <prv:hasObligationBearer rdf:resource="ex:CompetentAuthority"/>
   <prv:hasObligationAction rdf:resource=
                  "ex:ToAssessEffectsOnEnvironment"/>
   <prv:hasObligationCondition>
        <time:before rdf:resource="ex:OutlinePlanningPermission"/>
   </prv:hasObligationCondition>
   <prv:hasObligationObject rdf:resource="ex:Project"/>
</rdf:Description>
```

Given the U.K. provision semantic representation, it is possible to select the provisions concerning the obligation of a competent authority to assess the effect on the environment for urban development project. This leads to a comparison of the provisions reported in Fig. 3, which spots the existence of two different conditions to carry out such an assessment: one before "planning permission" (ex:PlanningPermission) is given and the other before "outline planning permission" (ex:OutlinePlanningPermission) is given.

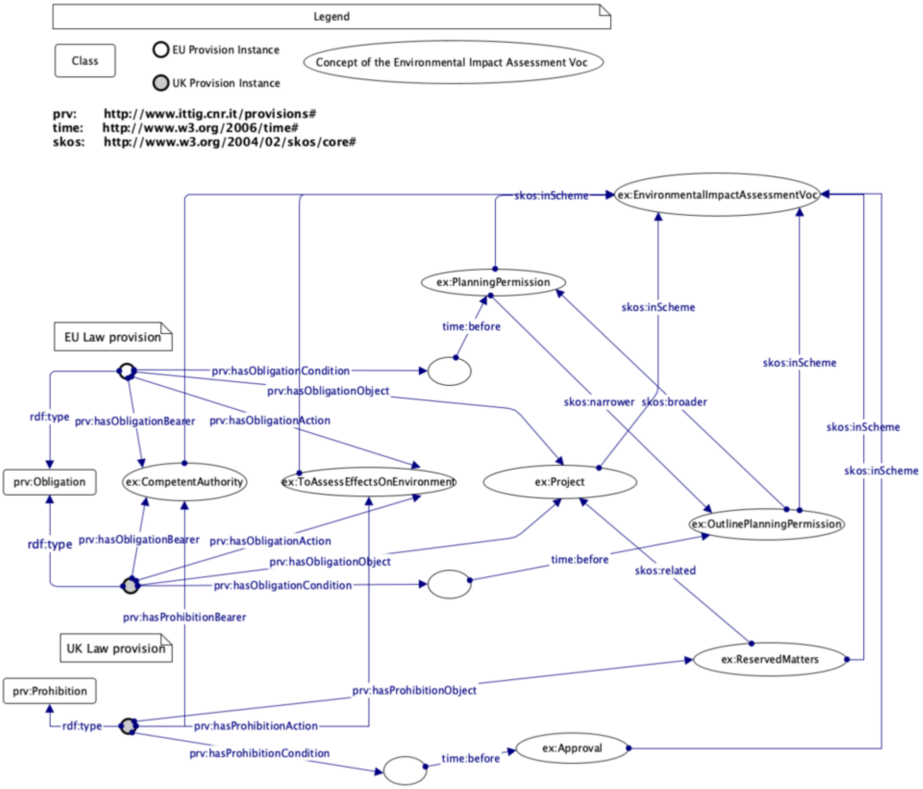

Figure 3. Semantic representation of EU and UK legal provisions

Given that these two concepts are in a skos:narrower relation, the inconsistency at the level of the condition of the obligation in the U.K. implemented provision with respect to the EU provision can be automatically detected. Moreover, the relation (skos:related[9]) between the concept of "urban development project" (ex:Project) and the "reserved matters" (ex:ReservedMatters) is able to automatically reveal the existence of a prohibition, introduced in the U.K. implemented legislation, to "assess the effect on the environment" (ex:AssessEffectsOnEnvironment) on "reserved matters," which represents an inconsistent exception with respect to what is prescribed by the EU directive.

It is worth emphasizing that the contradictions revealed are detected at the level of provision content by the relation between provision property values, as discussed in Section 3.2.

[9] Note that the relation between ex:ReservedMatters and ex:Project is rather a part-of relation (the specialization of the skos:related or skos:broader/narrower relation to capture partitive relations has been discussed but not approved by W3C yet)

6. Conclusions

In this paper, we introduced a model of contradictions between legal provisions and a practical implementation with the use of knowledge representation tools. The analysis of legal provisions requires a consideration not only of logical contradictions ($proh(p)$ vs $perm(p)$) or (p vs $\neg p$) but also of the scope of the terms used. We argue that, to detect a contradiction in sets of provisions, two conditions must be satisfied: 1) the analyzed provisions regulate at least a partially overlapping set of entities and 2) the analyzed provisions impose conflicting modalities on those sets.

On the basis of this assumption, we introduced a conceptual model of contradictions in legal text, as well as a Semantic Web-based implementation of such model, allowing a mechanism for automated detection of contradictions. This model enables the human expert to evaluate the drafted regulation and detect the contradictions therein, thus enabling the correction of the draft regulation and avoiding the necessity of amending the regulation after it enters into force. The presented model assumes human–computer interaction.

References

[1] Alchourron CE, Bulygin E. Normative Systems. Wien-New York; 1971.

[2] Araszkiewicz M, Pleszka K. The Concept of Normative Consequence and Legislative Discourse. In: Araszkiewicz M, Pleszka K, editors. Logic in the Theory and Practice of Lawmaking. vol. 2 of Legisprudence Library. Springer; 2015. p. 253-97.

[3] Van Kralingen R. Frame-based Conceptual Models of Statute Law. Kluwer; 1995.

[4] Visser PRS, van Kralingen RW, Bench-Capon TJM. A Method for the Development of Legal Knowledge Systems. In: Proceedings of ICAIL 1997. ACM; 1997. p. 151–160.

[5] van Doesburg R, van Engers T. The False, the Former, and the Parish Priest. In: Proceedings of ICAIL 2019. ACM; 2019. p. 194–198.

[6] Araszkiewicz M, Zurek T. Identification of Legislative Errors through Knowledge Representation and Interpretive Argumentation. In: Rodríguez-Doncel V, Palmirani M, Araszkiewicz M, Casanovas P, Pagallo U, Sartor G, editors. AI Approaches to the Complexity of Legal Systems. Cham: Springer International Publishing; 2021. p. 1-16.

[7] Francesconi E. A Description Logic Framework for Advanced Accessing and Reasoning over Normative Provisions. International Journal on Artificial Intelligence and Law. 2014;22(3):291-311.

[8] MacCormick N, Summers R. Interpreting Statutes. A Comparative Study. Dartmouth: Ashgate; 1991.

[9] Biagioli C. Modelli Funzionali delle Leggi. Verso testi legislativi autoesplicativi.. vol. 6 of Legal Information and Communications Technologies Series. Florence, Italy: European Press Academic Publishing; 2009.

[10] Francesconi E. Semantic Model for Legal Resources: Annotation and Reasoning over Normative Provisions. Semantic Web journal: Special Issue on Semantic Web for the legal domain. 2016;7(3):255-65.

[11] Biagioli C, Cappelli A, Francesconi E, Turchi F. Law Making Environment: perspectives. In: Proceedings of the V Legislative XML Workshop. European Press Academic Publishing; 2007. p. 267-81.

[12] Antoniou G, Billington D, Governatori G, Maher MJ. On the modeling and analysis of regulations. In: Proceedings of the Australian Conference Information Systems. Victoria University of Wellington, New Zealand; 1999. p. 20-9.

[13] Hoekstra R, Breuker J, di Bello M, Boer A. Lkif core: Principled ontology development for the legal domain. In: Breuker J, Casanovas P, Klein M, Francesconi E, editors. Law, Ontologies and the Semantic Web. vol. 188 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2009. p. 21-52.