A Toolset for Normative Interpretations in FLINT

Thom van Gessel*, Giulia Biagioni, Jeroen Breteler, Ioannis Tolios and Erik Boertjes

TNO, Anna van Buerenplein 1, The Hague, The Netherlands

Abstract

In light of the challenges in tracking and comprehending diverse interpretations of legal texts, this paper introduces a method for norm interpretation using ontologies and digital tools. We propose the FLINT ontology, a language for expressing interpretations of normative sources, as well as a set of software tools to support its utilization.

The FLINT ontology represents norms as normative acts with pre- and postconditions. Each of these elements are linked to its source, ensuring traceability and explainability. The software tools we introduce in this paper include a source decomposition application that converts normative documents into RDF-based representations, a norm editor for creating interpretations of norms based on FLINT, and the integration of an automated assistance module that generates annotation suggestions to assist users in encoding normative texts.

Keywords

Norms, Normative systems, Legal interpretation, FLINT

1. Introduction

Norms can be subjected to multiple interpretations. Not having a reference model allowing to indicate which interpretation has been applied to a given normative source, such as a legal text or regulation, makes it challenging to track from which perspective it has been interpreted. This issue becomes even more problematic when trying to keep track of decision-making processes, as the interpretation of rules directly influences the outcomes.

Ontologies can help to record information regarding the interpretation of norms. When they are integrated within digital tools, they facilitate normative cooperation and foster further reflection on how a certain regulation has been interpreted by different users. Moreover, they may allow stakeholders to reach deeper insights into the underlying logic of norms, leading them to further discuss on the interpretation that should be applied to norms.

To enable efficient tracking of the interpretation that has been employed to structure data of rules and regulations, we worked on solutions fostering the representation of normative knowledge based on a sound theoretical foundation. As part of this endeavour, we developed an ontology for the FLINT language [1]. Complementing this ontology, we have designed a comprehensive set of tools to support its utilization. They allow to A) create semantic representations of the structure of law documents, to relate interpretations back to their source, B) create interpretations of law documents by tagging and structuring data based on the

SEMANTiCS '23 Posters & Demos Track, Leipzig

*Corresponding author. E-mail: thom.vangessel@tno.nl.

expressiveness of the FLINT ontology, and C) generate suggestions for the ontological categories under which pieces of textual information regarding norms shall be structured.

The rest of the paper is set up as follows. We start by giving a brief overview of the state of the art of normative ontologies and the tooling that is available to utilize them in Section 2. We then introduce the FLINT ontology in Section 3 and describe the tooling in Section 4. We conclude in Section 5.

2. State of the art

Numerous vocabularies and ontologies have been developed to structure and organize data relating to norms [2]. We focus on representing the full normative action space available to agents across all relevant contexts. Therefore, our work is at some distance from solutions focusing on obligations and prohibitions or evaluating violations of situations rather than actions, such as LKIF [3], LegalRuleML [4] and recent work by Francesconi & Governatori [5].

Arguably the most relevant comparison for our work is to the UFO-L Legal Core Ontology [6]. UFO-L provides a formal description of legal relations. Similar to FLINT, UFO-L has a rich representation for classes of power—liability relations, including ways of identifying what classes of actors can occupy which legal positions within the relation (power holder, liability holder). As far as we are aware, no tooling exists to generate interpretations of normative sources in UFO-L, or any of the other ontologies mentioned above.

A line of work focused on mapping normative texts to formal models by means of an annotation tool is that of Dynamic Condition Response (DCR, [7]). This framework allows to map textual segments of norms to DCR graphs. Its primary aim is to enable mapping between law texts and code executing the law. By doing so, it enhances transparency on decision rules that are written in a procedural programming language. The editor however does not allow to structure data against the expressiveness of RDF-based ontologies, nor is it intended to do so. A similar initiative, focused specifically on obligations in contracts, extracts knowledge from natural language contracts in the form of Obligation Logic Graphs, and in turn maps these graphs to executable code [8]. Due to space constraints, a thorough comparative analysis of the aforementioned related works will be reserved for a future publication.

3. FLINT ontology

FLINT is a language that can be used to express interpretations of normative sources, while also serving as a basis for technical implementation. It does so by representing norms in terms of normative acts and their pre- and postconditions. The FLINT ontology is developed in RDF/OWL and is publicly available under an open-source license.[1]

FLINT's central concept is the *frame*: a container that bundles several pieces of information [9]. At the highest level, we distinguish two types of frames. *Fact frames* describe matters that characterize the state of the normative system. *Act frames* describe actions that agents might take, which affect the state. An act frame is connected to fact frames that describe

[1]The repository can be found at https://gitlab.com/normativesystems/knowledge-modeling/flint-ontology. For documentation, see https://normativesystems.gitlab.io/knowledge-modeling/documentation-website/docs/.

who can perform the act, who can undergo the act and what objects can be affected by it. Its preconditions describe the circumstances in which the act can be performed legally, while its postconditions are given in terms of the facts that become true and false by means of the act.

An important characteristic of FLINT, which is reflected in our ontology, is that each frame contains a reference to the source that this frame is an interpretation of (e.g. one specific sentence in a law text). These references ensure that the interpretation is explainable and traceable. To support fine-grained referencing, we developed a source ontology module that describes the structure and content of a source of norms.

4. Tooling

The software modules that we designed aim to support data-structuring activities related to norms while providing the possibility to everyone to specify and inspect information regarding rules and regulations. The target users for our tooling are legal experts, public administrators, non-profit organizations, businesses, and citizens. In fact, we envision a world where anyone can contribute to the clarification of the normative systems they participate in.

The tools are meant to exchange data in a standardized way, following the previously mentioned ontology modules. This ensures that the tools can also be used independently and in combination with other software. We describe the tools in detail in the following subsections.

4.1. Source Decomposition

The first module of our tooling is an application for preparing any source of norms into a format that allows fine-grained referencing. This application, which we named *Choppr*, converts a document into an RDF-based representation of the structure and contents of that document.[2] The process is semi-automatic: our source decomposition tool takes as input any text file, and allows the user to develop and test a set of instructions for the decomposition of the document. The instructions are based on scoped, sequential application of regular expressions. The tool can export the decomposition result as RDF-compliant data, using the concepts of the source ontology module.

4.2. Norm Editor

The second module of our tooling is the norm editor.[3] It is an application, built using web-based technologies, which allows a user to create interpretations of sources of norms in FLINT in a user-friendly and interactive way. The code of the editor is open source.[4]

The editor consists of three main panes. In the *source view* pane, a decomposed source as it is generated by the Choppr tool can be imported. The decomposed source is rendered and allows the user to *annotate* pieces of textual information according to the expressiveness of the FLINT ontology (acts, facts and their subtypes). It provides the capability to continuously observe the relationship between each FLINT frame and the corresponding source of the regulation. In the

[2]A prototype, developed in collaboration with software company Tick, is available at https://choppr.app/.

[3]A prototype of the norm editor can be accessed at https://norm-editor.tnodatalab.nl/.

[4]The repository can be accessed at https://gitlab.com/normativesystems/ui/interpretation-editor.

interpretation actions pane, an overview of the already annotated parts is given. Additionally, the user can select an annotation and include it in the supported interpretations actions (creation of facts and acts) of the right pane. The final step, exporting the interpretation in RDF, is under development.

4.3. Automated Assistance

In the process of encoding textual information into the FLINT ontology, users are supported by automated suggestions. For this we use a software module that can automatically generate tag suggestions for stakeholders, developed by Bakker et al. [10, 11]. This module uses a transformer-based model, fine-tuned on annotated law texts. It adds semantic role labels to parts of sentences in source texts, which map to FLINT concepts (actor, recipient, action, object). By incorporating this module into the editor, we enable users to utilize these suggestions, to save time during the manual encoding of normative texts. As AI capabilities improve, the tool can assist users in spotting human errors and guarantee comprehensiveness of annotations by suggesting tags for text segments that the stakeholders might have overlooked.

4.4. Software Ecosystem

The information exchanged between the components of our software tooling follows the path that users undertake to structure normative textual information. These steps include decomposing a source, manually or automatically encoding the segmented texts into the expressiveness of the FLINT ontology, and saving FLINT encoded information into the repository of choice. By following this process, the generation of structured data regarding norms becomes a streamlined process, allowing for easy sharing and dissemination of information. It ultimately supports the creation of machine-understandable data that can feed different sectoral ecosystems while contributing to interoperability of information.

5. Conclusion

In this paper, we reported on our developments of tooling for legal interpretation. We presented OWL ontologies formalizing relevant concepts, including a language to describe normative systems, and a representation of the content and structure of normative sources. We presented software tooling to support decomposition of normative sources, and manual and computer-assisted interpretation of those sources.

The tooling presented in this paper is still in active development. Usability tests are planned in the fall of 2023. Although FLINT has been tested in case studies, we have so far worked with manually created models without the use of tooling. We are now working on case studies to also test the maturity of the tools. For example, we are creating an interpretation of the Dutch budget laws in order to see whether we can accurately model the national budget cycle.

In addition, we are pursuing computational implementations of FLINT interpretations in order to automate normative reasoning. A norm engine based on the FLINT ontology is under development, which uses the ontology in combination with SHACL constraints and inference rules to reason about the lawfulness of acts in concrete scenarios.

As part of our future work, we intend to conduct a comprehensive comparison with the other normative ontologies and tooling mentioned in this paper, aiming to gain valuable insights into their similarities, differences, and overall efficacy.

Acknowledgments

We thank the Dutch Ministry of the Interior and Kingdom Relations for financially supporting our research. We are also grateful to Kamiel Naaktgeboren and Tinus Koorn for the development of Choppr and to Rutger Dik and Robert van Doesburg for their work on Choppr and the norm editor.

References

[1] R. Van Doesburg, T. Van Der Storm, T. Van Engers, Calculemus: Towards a formal language for the interpretation of normative systems, AI4J workshop at ECAI 2016, The Hague, Netherlands (2016).

[2] V. Leone, Legal knowledge extraction in the data protection domain based on ontology design patterns, Ph.D. thesis, University of Luxembourg, Luxembourg City, Luxembourg, 2021. URL: http://orbilu.uni.lu/handle/10993/47854.

[3] R. Hoekstra, J. Breuker, M. Di Bello, A. Boer, The lkif core ontology of basic legal concepts., in: LOAIT 07: II Workshop on Legal Ontologies and Artificial Intelligence Techniques, 2007, pp. 43–63.

[4] T. Athan, G. Governatori, M. Palmirani, A. Paschke, A. Wyner, LegalRuleML: Design principles and foundations, Reasoning Web. Web Logic Rules: 11th International Summer School 2015, Berlin, Germany, July 31-August 4, 2015, Tutorial Lectures. 11 (2015) 151–188.

[5] E. Francesconi, G. Governatori, Patterns for legal compliance checking in a decidable framework of linked open data, Artificial Intelligence and Law (2022) 1–20.

[6] C. Griffo, J. P. A. Almeida, G. Guizzardi, Conceptual modeling of legal relations, in: Conceptual Modeling: 37th International Conference, ER 2018, Xi'an, China, October 22–25, 2018, Proceedings, Springer, 2018, pp. 169–183.

[7] H.-A. López-Acosta, T. Hildebrandt, S. Debois, M. Marquard, The process highlighter: From texts to declarative processes and back, in: CEUR Workshop Proceedings, 2018, pp. 66–70.

[8] S. Servantez, N. Lipka, A. Siu, M. Aggarwal, B. Krishnamurthy, A. Garimella, K. Hammond, R. Jain, Computable contracts by extracting obligation logic graphs, in: Proceedings of ICAIL23, 2023, pp. 267–276.

[9] M. Minsky, A framework for representing knowledge, 1974.

[10] R. Bakker, R. van Drie, M. de Boer, R. van Doesburg, T. van Engers, Semantic role labelling for dutch law texts, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 448–457.

[11] R. Bakker, M. de Boer, R. van Drie, D. Vos, Extracting structured knowledge from dutch legal texts: A rule-based approach, in: Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management, 2022.