D3.1
Ontology-based access to normative knowledge

<table>
<thead>
<tr>
<th>Grant Agreement nº:</th>
<th>690974</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Acronym:</td>
<td>MIREL</td>
</tr>
<tr>
<td>Project Title:</td>
<td>Mining and REasoning with Legal texts</td>
</tr>
<tr>
<td>Website:</td>
<td><a href="http://www.mirelproject.eu/">http://www.mirelproject.eu/</a></td>
</tr>
<tr>
<td>Contractual delivery date:</td>
<td>31/12/2017</td>
</tr>
<tr>
<td>Actual delivery date:</td>
<td>31/12/2017</td>
</tr>
<tr>
<td>Contributing WP</td>
<td>WP3</td>
</tr>
<tr>
<td>Dissemination level:</td>
<td>Public</td>
</tr>
<tr>
<td>Deliverable leader:</td>
<td>HUD</td>
</tr>
<tr>
<td>Contributors:</td>
<td>HUD</td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690974
### Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Partner</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>27/07/2017</td>
<td>Sotirios Batsakis</td>
<td>HUD</td>
<td>Initial draft</td>
</tr>
<tr>
<td>1.0</td>
<td>31/12/2017</td>
<td>Sotirios Batsakis</td>
<td>HUD</td>
<td>Final Version</td>
</tr>
</tbody>
</table>

### Contributors

<table>
<thead>
<tr>
<th>Partner</th>
<th>Name</th>
<th>Role</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUD</td>
<td>Sotirios Batsakis</td>
<td>Editor</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>HUD</td>
<td>George Baryannis</td>
<td>Editor</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>HUD</td>
<td>Ilias Tachmazidis</td>
<td>Editor</td>
<td>Author of the deliverable</td>
</tr>
<tr>
<td>HUD</td>
<td>Grigoris Antoniou</td>
<td>Reviewer/WP leader</td>
<td>Reviewer</td>
</tr>
</tbody>
</table>

**Disclaimer:** The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. MIREL consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.
# Table of Contents

List of Figures ..................................................................................................................... 4  
List of Acronyms .................................................................................................................. 4  
Executive Summary .............................................................................................................. 5  
1  Introduction ....................................................................................................................... 6  
2  Background and related work ......................................................................................... 6  
3  Representation of Legal Norms ....................................................................................... 9  
  3.1  The LKIF ontology ....................................................................................................... 15  
  3.2  The SPARQL query language ...................................................................................... 17  
4  Examples of representation of norms .............................................................................. 18  
  4.1  Legal Norm Representation using OWL axioms ......................................................... 18  
  4.2  Legal norm representation using rules ....................................................................... 21  
  4.3  Legal norm representation using Answer Set Programming (ASP) ......................... 25  
5  Conclusions and Research Challenges .......................................................................... 27  
References ............................................................................................................................ 28
List of Figures

Figure 3-1: Wordnet synsets of term “law”

Figure 3-2: Wordnet first synset of term “law” and direct hyponyms.

Figure 3-3: Top concepts of LKIF-core ontology

List of Acronyms

LKIF: Legal Knowledge Interchange Format
DL: Description Logics
OWL: Web Ontology Language
SWRL: Semantic Web Rule Language
AF: argumentation framework
ASP: Answer Set Programming
NLP: Natural Language Processing
KOS: Knowledge Organization System
SKOS: Simple Knowledge Organization System
Executive Summary

The main objective of deliverable 3.1 is to provide a description of ontology based representation of legal norms and querying over represented knowledge. Such a representation will allow for formal, machine readable semantics combined with a reasoning mechanism to be used for applying legal norms.

Introduction, and background and related work on legal norm representation and Semantic Web standards are the first sections of this deliverable. Several approaches for legal norm representation are presented in the following. Specifically the LKIF ontology and related approaches based on OWL axioms and SWRL rules are presented. An alternative representation of legal norms based on answer set programming is presented as well. The above mentioned approaches are applied in two use cases in conjunction with reasoning and querying. Finally conclusions and research challenges identified are presented.
1 Introduction

Regulations are a widespread and important part of governments and businesses. They encode how products are manufactured, and how the processes are to be performed. Such regulations, in general, are difficult to understand and apply. Undoubtedly, the law, for example, as the reflection of human society, presents the broadest range of expression and interpretation, since the interpretation of even the most common words becomes problematic. Even individual regulations may be self-contradictory as a result of their gradual development process, as well as the lack of a formal (formal) drafting process. The problem becomes more and more difficult when independent regulations are applied in the same circumstances. For example, when two regulations overlap, it is not clear whether a regulation will override or will both apply. Even though regulations are typically drawn up, as is often the case in the legal field, problems such as consistency, interpretation and use remain. In an increasingly complicated environment, as well as regulatory review, an automated reasoning process becomes more and more necessary. This deliverable is the first step towards the development of such a system.

The ontology based representation of legal norms will be examined in the first part of this deliverable. Related work and state of the art in the area of legal norm representation, ontologies and querying are presented. This is followed by representation of example use cases and querying. Alternative approaches for the representation and the corresponding reasoning mechanism are examined. Conclusions and research questions are presented in the final part of the deliverable.

2 Background and related work

Research on the confluence of AI and Law has been active for more than four decades. We refer the interested reader to detailed accounts of such research in [1], [2], [3]. In this section, the focus is only on the various approaches for ontology-based access to normative knowledge.

Early research in legal theory significantly contributed to the conceptualization of the legal domain and led to explicit representations using formal languages. The earliest such representations are, to the best of our knowledge, McCarty’s Language for Legal Discourse (LLD) [4] and Stamper’s NORMA formalism [5], [6]. LLD relies on sets of terms that essentially represent the legal domain itself: count terms to express tangible objects and mass terms to express intangible objects. Using such terms, legal discourse is modeled in the form of first-order legal rule statements and second-order modal-based statements, supporting time, events, actions and deontic expressions. On the other hand, Stamper argued that expressing legal knowledge in the form of rules is oversimplifying and confining. Instead, he proposed a logic of forms and affordances called NORMA, describing entities based on their behavior rather than truth values. NORMA includes concepts that model agents, their behavioral invariants and their actions.

The aforementioned conceptualizations relied on semi-formal structures and were never intended to be proposed as ontologies of the legal domain [7]. The first such example is the Functional Ontology of Law (FOLaw) [8], which organizes legal knowledge under a number of primitive types.
Ontology-based access to normative knowledge

FOLaw’s primitive types include: normative knowledge, expressing desirable and undesirable behavior; meta-legal knowledge, used to solve conflicts between applicable norms; world knowledge, acting as an interface between the legal system and social reality; responsibility knowledge, to establish links between norm violation and the agent accountable, guilty or liable; reactive knowledge, to represent the relevant sanctions or rewards for violating or abiding by the law; and creative knowledge, involved in the creation of social institutions and legal entities. FOLaw allows for a formalized representation of legal information aiming to increase usability and reusability of legal information and to assist in automated legal services.

Frame-Based Ontology [9], [10] proposes a different, broader distinction between a generic legal ontology and various statute-specific ontologies. The legal ontology is reusable and can model norms, acts and legal concepts that are present in any legal domain. On the other hand, statute-specific ontologies includes additional predicate relations that are used to complement the legal ontology in order to introduce specializations on norms, acts and legal concepts that are only applicable to a single statute (or a group of related ones).

The need for an ontological approach in normative knowledge access is also argued by Bench-Capon and Visser [11], highlighting its benefits in knowledge acquisition, sharing and reuse, as well as domain theory development. They also stressed the fact that the inability of handling contradictory or unspecified rules, which plagued traditional rule-based normative approaches, could be addressed through an ontological approach.

One of the first attempts to implement a legal information management system that relies on an ontology of the legal domain is ON-LINE [12]. The authors argue that realizing a combination of normative knowledge requirements that range from storage to retrieval and reasoning can only be achievable based on a strong theoretical foundation underpinned by a legal ontology expressing legal norms. Their work is the first to shift legal assessment from a rules-oriented formalism, probably the most prominent approach at that time, to knowledge representation, emphasizing what law is made of rather than merely representing legal consequences [13]. The final version of the ON-LINE architecture [14] is able to create knowledge bases and convert them from text to a Description Logic (DL) formalism and supports legal information retrieval and automated legal analysis by relying on LOOM [15], an early DL reasoner.

The advent of the Semantic Web and the introduction of the OWL family of languages led to a renewed interest in the use of ontologies in relation to normative knowledge, with a focus not on taxonomy-style, conceptual ontologies but more sophisticated, computational ones, targeted at applications such as information retrieval, reasoning and inter-language understanding [16]. The MetaLex initiative [17] is an attempt to exploit Semantic Web technologies to represent legislation, also supporting comparison and harmonization facilities. XML documents are created to describe the semantics of legal resources while OWL ontologies represent legal knowledge independent of specific jurisdictions. These ontologies are used to resolve conflicts between norms and work towards norm harmonization. Another example is Core Legal Ontology [18],
Ontology-based access to normative knowledge

which is intended as a high-level ontology to support the definition of specific legal domain ontologies, juridical wordnets and the design of legal decision support systems.

The ESTRELLA project (European project for Standardized Transparent Representations in order to Extend Legal Accessibility)\(^1\) led, among others, to several contributions in ontology-based access to normative reasoning. One of them is the creation of the Legal Knowledge Interchange Format (LKIF)\(^1\) [19], an XML Schema for representing legal theories and arguments constructed from these theories. LKIF relies on an OWL ontology, named LKIF-Core [20], to represent legal concepts. LKIF-Core is influenced from and inspired by many preceding ontologies for the legal domain, including LRI-Core [21]. A more detailed analysis of LKIF-Core can be found in Section 3.1.

ESTRELLA also advanced research on ontology-based normative reasoning, producing HARNESS [22] (also known as OWL Judge [23]), a tool that exploits well established sound and decidable DL reasoners such as Pellet. However, this demands a significant compromise in terms of expressiveness. The most important issue is that relationships can only be expressed between concepts and not between individuals, e.g. using the example in [22], if we have statements expressing the facts that a donor owns a copyright donation and that a donor retains some rights, there is no way to express (in pure OWL) that the donor in both cases is the same individual. This can be expressed via rules (e.g. written in SWRL); however, to retain decidability these rules must be restricted to a so-called DL-safe subset [24].

Building on aforementioned standards and initiatives, such as MetaLex and LKIF, Emerald [25] is an integrated framework that supports the complete lifecycle of legal modeling. It produces standardized XML documents from legal source texts and extends them with metadata, annotations and resource links. These can be further elaborated into formal conceptual and logical models, stored in RDF databases, which can then be queried to fulfill advanced reasoning services, such as change management of formal resources.

It should be noted that the typical forms of reasoning that are enabled by ontological formalisms relying on description logics are limited by the fact that they are monotonic: logical consequences cannot be retracted, once entailed. This is in contrast to the nature of law, where legal consequences have to adapt in light of new evidence and conflicts between different regulations must be accounted for and resolved. Therefore, it is natural to employ non-monotonic logic for the purposes of normative reasoning, such as the Defeasible Logic framework [26]. On the other hand, if conflicts arise not from norms but from interpretation, case-based normative reasoning may be more suitable, as evidenced by tools such as HYPO [27], CATO [28] and GREBE [29]. A more detailed analysis of the various approaches to normative reasoning can be found in the MIREL deliverable D3.2.

The aforementioned limitations that ontologies and description logics impose on normative reasoning have also led the Semantic Web research community to propose a new rule language designed to facilitate ontology-based access to normative knowledge. This language, known as

\(^{1}\) http://www.estrellaproject.org/
LegalRuleML [30], does not propose or assume the use of a specific core or domain legal ontology, focusing mainly on increasing expressiveness by including constructs that are native to normative reasoning but are not provided by traditional rule languages. These include: support for both constitutive and prescriptive rules; representation of obligations, permissions, prohibitions and other normative effects; implementation of defeasibility; implementation of isomorphism, i.e. the explicit association of a rule to its legal sources; and support for alternative interpretations.

3 Representation of Legal Norms

The objective of this deliverable is to analyse ontology based representation of legal information and norms. Ontology can be defined as “A formal specification of a shared conceptualization of a domain of interest”. Being formal means that the ontology should be machine-readable, so as to allow for automatic processing. It must also be shared i.e., accepted by a community of users. Typically an ontology is restricted to a given domain of interest and model concepts and relations that are relevant to a particular application domain or a particular task. Using formal axioms allows for reasoning, and typically a concept rather than a term based representation is language independent.

Semantic representations that can be used in the legal domain are lexicons, thesauri or lightweight ontologies such as taxonomies offering a simple representation based on lexical terms but without complex reasoning capabilities. Sources are legal texts in natural language. In legal language the meaning of terms in a legal concept often differs from that in everyday language. In addition, common problems in Natural Language Processing (NLP) such as polysemy of terms, also appear. By providing formal definitions of concepts a semantic representation can be useful for tasks such as data access and information retrieval, publication exchange interoperability and harmonization. Interoperability in particular is very important in case of multilingual corpora, and comparison/integration of sources, especially from different legal systems.

Structured vocabularies such as taxonomies and thesauri are lists of terms organized in hierarchies of broader and narrower terms and also associated terms using related term relations. The definition of terms is restricted to the relationships with other terms into the taxonomy, without complex concept constructs and semantic constraints. Thus complex reasoning tasks are not supported, but tasks such as document tagging and classification as part of retrieval of information are. Controlled vocabularies, taxonomies and thesauri, referred to as Knowledge Organization Systems (KOS), have been used in digital libraries among others and a recent development is the introduction of standards for their representation and exchange on the web.

The Simple Knowledge Organization System (SKOS) is a W3C initiative “for developing specifications and standards to support the use of knowledge organization systems (KOS) such as
Ontology-based access to normative knowledge

thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web” 2.

Following SKOS recommendations a KOS can be distributed and interlinked and additionally interoperability is achieved. SKOS is based on Semantic Web standards and specifications to achieve the creation of modular, reusable and interlinked controlled vocabularies. Specifically representation of concepts and individuals in SKOS is based on Semantic Web standards RDF3 and RDFS4 for the creation of distributed and interlinked datasets. Since concepts, their properties and individuals in SKOS are represented using RDF, they have unique identifiers- uniform resource identifier (URI). Using URIs each concept, property or individual, is identified as a resource that can be referred on the web.

SKOS is used in several resources; The British Integrated Public Sector Vocabulary (IPSV) thesaurus provides a vocabulary for encoding public sector metadata over several subjects5 including the legal domain. The SKOS based STW Thesaurus for Economics6 contains almost 6,000 standardized subject headings and about 20,000 additional entry terms to support individual keywords including technical terms used in law. The European Union's EuroVoc7 is a multilingual, multidisciplinary thesaurus covering the activities of the EU. It contains terms in 23 EU languages over several domains including the legal domain. EuroVoc adopts ontology-based thesaurus management and semantic web technologies conformant to W3C recommendations. In addition the EU N-Lex portal8 offers a single entry point to national law databases in individual EU countries. The specialized European Union’s ECRIS system9 was established to achieve an efficient exchange of information on criminal convictions between EU countries based on an electronic standardised format of transmission. In US, The Library of Congress has released a machine-readable version of the Library of Congress Subject Headings (LCSH) using SKOS as a data model, which includes a Thesaurus of Law10. A SKOS Vocabulary for representing the US Code of Federal Regulations is presented in [31].

Manual construction of resources such the aforementioned requires huge effort and is costly. SKOS-XL 11 defines an extension for the Simple Knowledge Organization System, providing additional support for describing and linking lexical entities. Tools for collaborative development and integration of OWL ontologies, SKOS(XL) thesauri and generic RDF datasets include VocBench12 and HIVE13. Alternatively, crowdsourcing can be used for definition of concepts,

---

2 https://www.w3.org/2004/02/skos/intro
3 https://www.w3.org/RDF/
4 https://www.w3.org/TR/rdf-schema/
5 http://id.esd.org.uk/list/subjects
6 http://zbw.eu/stw/version/latest/about
7 http://eurovoc.europa.eu/
8 http://eur-lex.europa.eu/n-lex/
10 http://id.loc.gov/vocabulary/subjectSchemes/noubojur.html
11 https://www.w3.org/TR/skos-reference/skos-xl.html
12 http://vocbench.uniroma2.it/
13 http://cci.drexel.edu/mrc/projects/hive/
categories and for dealing with multilinguality. Wikipedia\textsuperscript{14} and Wiktionary\textsuperscript{15} are important examples of crowdsourcing providing wide coverage and low formality, including the legal domain\textsuperscript{16}. DBpedia\textsuperscript{17} is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web in RDF format. DBpedia allows users to submit queries against Wikipedia, e.g. using SPARQL\textsuperscript{18}, and to link the different data sets on the Web, to Wikipedia data.

An important task of MIREL project is the mapping between lexical resources and controlled vocabularies. Natural language is characterized by ambiguity and lexical variability, which complicates tasks such as extraction of lexical meaning from texts (which in turn involves lexical similarity and automatic word sense disambiguation), text categorization and multilinguality/translation. NLP algorithms need data to operate, specifically big textual collections (corpora). Often multilingual parallel corpora and linguistic resources like lexicons and grammars. Lexical Ontologies or lexical knowledge bases are databases representing lexical meaning to be accessed by systems for text analysis. They provide formal and explicit knowledge background to machines to deal with natural language, by representing concepts encoded by natural language expressions (lexical units, terms, etc.). Additionally they specify semantic classes grouping terms at the semantic level. Lexical Ontologies can be general purpose or domain specific, monolingual or multilingual, and contain terminological, syntactic or semantic information, or a combination of those. Semantic computational lexicons in particular represent the meaning of words. They distinguish different senses of a word and represent similarity and relatedness.

WordNet\textsuperscript{19} is an important lexical/semantic resource developed for English at the cognitive science laboratory of Princeton University and it is based on psycholinguistic theories of human lexical memory. Specifically WordNet maps words to meanings/senses and is organized around word meaning (not word forms as with traditional lexicons). It can be used as a database (DB) as an application programming interface (API), or using a web interface. Wordnet consists of nouns, verbs, adjectives and adverbs each organized in sets of cognitively equivalent synonyms or synsets.

\begin{itemize}
  \item \textsuperscript{14} https://www.wikipedia.org/
  \item \textsuperscript{15} https://www.wiktionary.org/
  \item \textsuperscript{16} https://en.wikipedia.org/wiki/Portal:Law
  \item \textsuperscript{17} http://wiki.dbpedia.org/
  \item \textsuperscript{18} https://www.w3.org/TR/rdf-sparql-query/
  \item \textsuperscript{19} http://wordnet.princeton.edu/
\end{itemize}
Ontology-based access to normative knowledge

Figure 3-1: Wordnet synsets of term “law”

Each set of synonyms represent a lexical concept and then a sense, and all members of a synset represent the same concept. In addition all concepts are associated with a particular part of speech (noun, verb, adjective, and adverb). Synsets represent a word meaning and words that occur in several synsets have a corresponding number of meanings (polysemy). The meaning of each concept is explained by a gloss and the concept can be referred using one or more lexical forms. Every WordSense is associated exactly to a single synset and every WordSense is referred to a single lexical form. Every lexical form can belong to one or more Wordsense and thus can be associated to one or more synsets (polysemy). Synsets are organized in hierarchies with generalization relations (hypernymy) and specialization relations (hyponymy). Synsets are associated using semantic intra-lingual relations (e.g. hyponymy, hypernymy, meronymy) and by inter-lingual equivalence relations, using an interlingual index (ILI).
Ontology-based access to normative knowledge

Figure 3-2: Wordnet first synset of term “law” and direct hyponyms.

EuroWordNet\(^\text{20}\) is a system of semantic networks for European languages, based on WordNet. Each language develops its own Wordnet but they are interconnected with interlingual links stored in the Interlingual Index. Concepts (synset of a monolingual WN) connected to the same ILI record are considered equivalent concepts. OntoWordNet [32] is a formal specification of WordNet through extension and axiomatization of its conceptual relations in OWL format. In the legal domain terms have a precise meaning, sometimes different than that of everyday language but terms can still be polysemous and assigned to more than one concept. JurWordNet [33] is a WordNet based semantic network for the legal domain.

Linguistic ontologies support semantic annotation and improve conceptual and cross-lingual retrieval in more complex applications, as consistency checking and concept comparison linguistic ontologies must be linked to complex formal ontologies. In LOIS project [34] links to Core and

\(^{20}\) \url{http://projects.illc.uva.nl/EuroWordNet/}
Foundational Ontologies are used for term disambiguation. In DALOS project\textsuperscript{21} the legislation drafting is supported by ontologies, both complex/formal and lexical. The DALOS lexicon is automatically extracted from EU texts (Directives and Judgements) on Consumer Law, using NLP tools and it is linked formal ontologies. LemOn\textsuperscript{22} (Lexical Model for Ontologies) is a general model for formally defining lexical features in relation to independently defined ontological semantics and connecting the lexical features to the ontological semantics.

The aforementioned lexical resources can be used for tagging of legal text and legal information is often available as Linked Open Data, using Semantic Web standards, on the national (e.g. UK\textsuperscript{23}, Netherlands\textsuperscript{24}), European\textsuperscript{25} and global\textsuperscript{26} level. Reasoning over the resulting representations requires axiomatic definitions of concepts their properties and their relationships. Thus formal ontologies (linked with the lexical ones) are used for representation of legal information and legal norms. In case legal information is in a structured database format and the correspondence of databases fields with ontology concepts is known then legal data can be mapped to the ontology concept using tools for this mapping. Ontop \textsuperscript{[35]} is such a widely used tool for defining mapping between databases and ontology concepts.

By using formal, language independent ontologies, interoperability of systems can be achieved along with knowledge acquisition, reasoning and information retrieval. Querying information when using a W3C compliant ontology based representation can be achieved using the SPARQL query language, which will be presented in a section 3.2.

Ontologies can be generic/upper ontologies, core (main concepts), or domain specific. They can be developed using top-down or bottom-up approaches. Reusing existing ontologies along and developing design patterns are common practices. Examples of legal ontologies are: the Free Legal Ontology\textsuperscript{27}, the Copyright Ontology\textsuperscript{28}, the Legal Cases Ontology\textsuperscript{29}, the Core Legal Ontology\textsuperscript{30} and the LKIF-Core Ontology\textsuperscript{31}. The Core Legal Ontology provides definitions for the generic concepts of the legal domain that are in principle found in all the legal systems and specific domains, like: law, legal norm, regulation, legal agent and legal role and it depends on DOLCE\textsuperscript{32} ontology. The LKIF-Core ontology contains definitions of basic legal concepts and it will be presented in detail in the following.

\begin{footnotesize}
\begin{itemize}
    \item \textsuperscript{21} http://www.dalosproject.eu/
    \item \textsuperscript{22} http://www.lemon-model.net/
    \item \textsuperscript{23} http://www.legislation.gov.uk/
    \item \textsuperscript{24} http://doc.metalex.eu/
    \item \textsuperscript{25} http://api.epdb.eu/
    \item \textsuperscript{26} https://openspending.org/
    \item \textsuperscript{27} http://derecho-internet.org/olj/vocabulario/index.html
    \item \textsuperscript{28} http://rhizomik.net/ontologies/2006/01/copyrightonto.owl
    \item \textsuperscript{29} http://wyner.info/research/ontologies/LegalCaseOntology_v9.owl
    \item \textsuperscript{30} http://www.loa.istc.cnr.it/ontologies/CLO/CoreLegal.owl
    \item \textsuperscript{31} https://github.com/RinkeHoekstra/lkif-core
    \item \textsuperscript{32} http://www.loa.istc.cnr.it/old/DOLCE.html
\end{itemize}
\end{footnotesize}
3.1 The LKIF ontology

The LKIF-Core ontology was developed within the ESTRELLA project\(^\text{33}\) for defining basic legal concepts. The LKIF core legal ontology\(^\text{34}\) “consists of 13 modules, each of which describes a set of closely related concepts from both legal and common-sense domains”. Thus the LKIF core ontology is a library of ontologies relevant for the legal domain. The most abstract concepts are defined in modules: top\(^\text{35}\), place\(^\text{36}\), mereology\(^\text{37}\), time/spacetime\(^\text{38}\). Basic-level concepts are distributed across four modules: process\(^\text{39}\), role\(^\text{40}\), action\(^\text{41}\) and expression\(^\text{42}\). These modules are extended by three modules that form the legal ontology: legal action\(^\text{43}\), legal role\(^\text{44}\) and norm\(^\text{45}\). In addition to these legal modules, two modules are provided that cover the basic vocabulary of two frameworks: modification\(^\text{46}\) and rules\(^\text{47}\). Finally, the modules of the abstract, basic and legal level are integrated in the LKIF Core ontology module\(^\text{48}\). The two framework modules are accessible through the LKIF Extended ontology module\(^\text{49}\) which also imports the LKIF Core module.

\(^{33}\) \url{http://www.estrellaproject.org/}
\(^{34}\) \url{http://www.estrellaproject.org/lkif-core/}
\(^{35}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/lkif-top.owl}
\(^{36}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/relative-places.owl}
\(^{37}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/mereology.owl}
\(^{38}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/time.owl}
\(^{39}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/process.owl}
\(^{40}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/role.owl}
\(^{41}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/action.owl}
\(^{42}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/expression.owl}
\(^{43}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/legal-action.owl}
\(^{44}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/legal-role.owl}
\(^{45}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/norm.owl}
\(^{46}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/time-modification.owl}
\(^{47}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/lkif-rules.owl}
\(^{48}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/lkif-core.owl}
\(^{49}\) \url{https://github.com/RinkeHoekstra/lkif-core/blob/master/lkif-extended.owl}
Ontology-based access to normative knowledge

Figure 3-3: Top concepts of LKIF-core ontology

LKIF Core contains OWL axioms that allow for reasoning over represented information. For example the definition of concept Allowed in LKIF consists of the following axioms:

Allowed SubClassOf normatively_comparable some Normatively_Qualified
Allowed SubClassOf qualified_by some Norm
Allowed SubClassOf qualitatively_comparable some Qualified
Allowed SubClassOf qualitatively_comparable only Qualified
Allowed SubClassOf qualified_by some Qualification

The LKIF ontology is used in conjunction with the YAGO upper ontology\(^{50}\) in [36] for ontology based representation of legal texts. Specifically in [36] a mapping between LKIF and YAGO concepts is defined and machine learning is applied on legal texts in order to tag concepts appearing into the text to the corresponding ontology concepts. Such a tagging is the first step towards efficient representation and the subsequent querying and reasoning of legal information.

\(^{50}\) https://github.com/yago-naga/yago3
In the following section two examples will be used to demonstrate the representation of legal norms using OWL axioms and alternatively using SWRL rules. Querying information is achieved using SPARQL. A different reasoning approach using Answer Set Programming (ASP) is also presented.

### 3.2 The SPARQL query language

Querying legal information represented using can be achieved using the SPARQL query language, which is a W3C standard. The current version of SPARQL is SPARQL 1.1. There are several SPARQL query engines available and compliance with W3C standards ensures that these query engines will be available as well. Furthermore reasoning capabilities are supported by several query engines. A SPARQL query has the generic form SELECT ... WHERE, where the select keyword is followed by a list of variables and the where statement is followed by a condition that these variables must satisfy. For example the following query returns type of normative qualifications over the LKIF ontology:

```sparql
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?subject
WHERE
{?subject rdfs:subClassOf <http://www.estrellaproject.org/lkif-core/norm.owl#Normatively_Qualified> }
```

The result of this query is: Allowed and Disallowed.

The select statement can be replaced by the CONSTRUCT keyword and instead of retrieving results, when the condition in WHERE clause is satisfied then the triples defined in the construct clause are asserted. Using COSTRUCT keyword the knowledge base can be updated. Another important operator in SPARQL is the OPTIONAL operator in WHERE clause, that returns variables if a specific type of triple exists (e.g. a query like “return the names of courts related to these cases and their locations if these locations are available”). Negation is also supported in SPARQL (using

---

51 [https://www.w3.org/Submission/SWRL/](https://www.w3.org/Submission/SWRL/)
52 [https://www.w3.org/TR/rdf-sparql-query/](https://www.w3.org/TR/rdf-sparql-query/)
53 [https://www.w3.org/TR/sparql11-overview/](https://www.w3.org/TR/sparql11-overview/)
54 [https://www.w3.org/wiki/SparqlImplementations](https://www.w3.org/wiki/SparqlImplementations)
55 [https://www.w3.org/TR/sparql11-entailment/](https://www.w3.org/TR/sparql11-entailment/)
the NOT BOUND statement) and SPARQL 1.1 offers support for various reasoning regimes. In addition SPARQL 1.1 allows for federated queries that span over different SPARQL endpoints, thus enriching existing information with information from various sources, which in turn are further enriched by applying reasoning. A detailed description of SPARQL can be found in [37].

4 Examples of representation of norms

In the following examples of representations of norms will be presented. In the first example norms of a simple use case are represented using the LKIF ontology and OWL axioms. In the second example rules are used instead of OWL axioms. In the third example an approach based on Answer Set programming is presented.

4.1 Legal Norm Representation using OWL axioms

LKIF can be used both for representation of legal cases and norms and reasoning. This can be achieved by applying design patterns in the representation introduced in [23], the example case is also presented in [23]. The case to represent is related to university library regulations:

“1a) Students registered at this university are allowed to check out a book from this library

1b) Students registered at other universities are allowed to check out a book from this library provided that they are enrolled in at least one course given at this university.

1c) Students who have checked out more than five books are not allowed to check out another book.

One of the students is Amy. She is registered at the University and she has checked out 6 books on some general topic”

According to the regulations, Amy’s situation is allowed by article 1a, but disallowed by article 1c. However, these two articles contain an intuitive hierarchic pattern: the case described by article 1a subsumes that of article 1c. In fact, article 1c is an exception to 1a, because 1a implicitly expresses that students registered at other universities are not allowed to check out any books. The expected verdict of the system should therefore be that Amy's situation is disallowed, as article 1c is more specific than article 1a, the lex specialis principle states that 1c trumps 1a.” [23].
For this example assertions and axioms are added manually to LKIF-core using the Protégé editor\(^{56}\). Intermediate lexical ontologies are not used. Firstly individual assertions about the student and the books are added. Then axioms representing the cases are added:

\begin{verbatim}
Default_GC EquivalentTo Registered_Student
    and (checks_out some Library_book)

Default_GC SubClassOf Generic_Case
    and (allowed_by value art1a)

Art1a_GC EquivalentTo checks_out some Library_book

Art1a_GC SubClassOf Generic_Case
    and (disallowed_by value defaultnorm)

Art1c_GC_P SubClassOf Student
    and (checks_out some Library_book)
    and (checks_out max 5 Library_book)

Art1c_GC_F SubClassOf Student
    and (checks_out min 6 Library_book)
\end{verbatim}

Axioms representing norms are:

\begin{verbatim}
Default_Norm EquivalentTo {defaultnorm}

Default_Norm SubClassOf Prohibition and (disallows only Default_GC)
\end{verbatim}

\(^{56}\) http://protege.stanford.edu/
Art1c_Prohibition EquivalentTo \{art1c\}

Art1c_Prohibition SubClassOf Norm

Art1a_Permission EquivalentTo \{art1a\}

Art1a_Permission SubClassOf Permission
    and (allows only Art1a_GC)

Axioms about the exception relation between two norms are:

\[ \text{disallows}^{o} \text{ allowed_by} \text{ SubPropertOf} \text{ exception} \]
\[ \text{allows}^{o} \text{ disallowed_by} \text{ SubPropertOf} \text{ exception} \]

Using the abovementioned axioms and assertions about individuals, reasoning can be applied and norms applicable to the specific case and involved persons are automatically derived. In this case Art1c_GC_F and Default_GC apply to Amy, with the first being an exception to the second. The following SPARQL query extracts classes that Amy belongs

```sparql
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?c
WHERE {<http://www.estrellaproject.org/lkif-core/lkif-core.owl#Amy> a ?c }
```

Every OWL compatible reasoner such as FaCT++\(^{57}\), Jfact\(^{58}\), HermiT\(^{59}\), Pellet\(^{60}\) and

\(^{57}\) https://bitbucket.org/dtsarkov/factplusplus
^{58} http://jfact.sourceforge.net/
RacerPro\textsuperscript{61} can be used. In this example HermiT 1.3.8 as a Protégé plug-in was used and reasoning time was 73595 ms on a PC with 64-bit operating system, 6GB RAM and AMD A6-6310 CPU. The advantage of this approach based on OWL-axioms is that reasoning and representation are integrated and fully compliant with W3C standards. The disadvantages are slow reasoning and the need to define complex axioms. Also, since OWL reasoning is monotonic, inferences that are based on norms that have exceptions are not retracted. Finally OWL axioms cannot represent all types of rules, as is the case of the following example.

### 4.2 Legal norm representation using rules

The second example is a US bar exam question [38]. Specifically the description of the case and the corresponding questions are:

"An entrepreneur from State A decided to sell hot sauce to the public, labelling it “Best Hot Sauce.” A company incorporated in State B and headquartered in State C sued the entrepreneur in federal court in State C. The complaint sought $50,000 in damages and alleged that the entrepreneur’s use of the name “Best Hot Sauce” infringed the company’s federal trademark. The entrepreneur filed an answer denying the allegations, and the parties began discovery. Six months later, the entrepreneur moved to dismiss for lack of subject-matter jurisdiction.

Should the court grant the entrepreneur’s motion? No, because the company’s claim arises under federal law. The claim asserts federal trademark infringement, and therefore it arises under federal law. Subject-matter jurisdiction is proper under 28 U.S.C. § 1331 as a general federal-question action. That statute requires no minimum amount in controversy, so the amount the company seeks is irrelevant.

Should the court grant the entrepreneur’s motion? No, because the entrepreneur waived the right to challenge subject-matter jurisdiction by not raising the issue initially by motion or in the answer. Under Federal Rule 12(h)(3), subject-matter jurisdiction cannot be waived and the court can determine at any time that it lacks subject-matter jurisdiction. Therefore, the fact that the entrepreneur delayed six months before raising the

\textsuperscript{59} http://www.hermi-reasoner.com/
\textsuperscript{60} https://github.com/stardog-union/pellet
\textsuperscript{61} http://www.ifis.uni-luebeck.de/~moeller/racer/
lack of subject-matter jurisdiction is immaterial and the court will not deny his motion on that basis.

Should the court grant the entrepreneur’s motion? Yes, because although the claim arises under federal law, the amount in controversy is not satisfied. There is no amount-in-controversy requirement for actions that arise under federal law. Although diversity jurisdiction requires an amount in controversy of $75,000 or more, when diverse parties are litigating a federal claim, the action is treated for jurisdictional purposes as a federal-question action, not a diversity action. The claim here asserts federal trademark infringement and therefore it arises under federal law. The fact that the action does not meet all the requirements for diversity jurisdiction is irrelevant.

Should the court grant the entrepreneur’s motion? Yes, because although there is diversity, the amount in controversy is not satisfied.”

As in the previous example the facts of the case and related entities are manually added into LKIF.

These include definitions of classes such as States and Trademark and assertions about individuals such entrepreneur, company, trademark, federal court and related norms.

Using OWL axioms as in the previous example is problematic since complex chains of properties are involved combined with restrictions over them. Furthermore rules are a natural way to express norms and in this example a rule based approach is applied. The representation in this example does not include complex representation of norms as the previous example, thus reasoning using the SWRLAPI Drools Engine took 828 milliseconds.

After asserting facts about the case and reason over them, the norms that apply to the case are defined manually. Rules are defined using SWRL and in particular the Protégé SWRL tab. The rules are the following:

Rule 1 (Diverse individuals) : State(?p1) ^ State(?p2) ^ located(?x, ?p1) ^ located(?y, ?p2) ^ differentFrom(?p1, ?p2) ^ differentFrom(?x, ?y)

-> diverse(?x, ?y)

Rule 2 (Diversity case): LegalCase(?c) ^ defendant(?c, ?d) ^ accuser(?c, ?a) ^
diverse(?a, ?d) ^ swrlb:greaterThan(?r, "75000"^^xsd:int)

-> DiversityCase(?c)

Rule 3 (FR 12 h 3) : LegalCase(?c) ^ defendant(?c, ?d) ^ court(?c, ?cr)

-> challenge_jurisdiction(?d, ?cr)

Rule 4 (Copyright infringement) : LegalCase(?c) ^ accuser(?c, ?a) ^ defendant(?c, ?d) ^
uses(?d, ?t) ^ owns(?a, ?t) ^ differentFrom(?a, ?d) ^
Trademark(?t) -> CopyrightCase(?c)

Rule 5 (28 USC 1331) : LegalCase(?c) ^ accuser(?c, ?a) ^ defendant(?c, ?d) ^
uses(?d, ?t) ^ owns(?a, ?t) ^ differentFrom(?a, ?d) ^
FederalTrademark(?t) -> FederalCopyrightCase(?c)

The process of reasoning over the above rules took 438 milliseconds using the SWRLAPI Drools
Engine, thus using rules the representation is simpler, and reasoning faster than using OWL axioms
only. Facts such as that the case is federal case are inferred. Since rules are monotonic, conflicting
norms can still apply, thus leading to contradicting assertions. Dealing with, calls for non-
monotonic reasoning, and a corresponding example will be presented in the following section.

Using SPARQL queries information can be extracted and in addition to this norms can be
represented using query patterns. By issuing these queries it can be detected whether or not a
norm applies, and the case instances that it applies to. SPARQL query examples are presented in
the following.

Querying for norms that apply at cases allowing an action:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?object
WHERE {
?subject <http://www.estrellaproject.org/lkif-core/norm.owl#allowed_by> ?object
}

Detecting a federal copyright infringement:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?subject ?x ?object
}

Detecting a federal case:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT distinct ?subject ?s2 ?s1
?s1 a <http://www.estrellaproject.org/lkif-core/lkif-core.owl#State>.}
Ontology-based access to normative knowledge

\{?
s2 a <http://www.estrellaproject.org/lkif-core/lkif-core.owl#State>.
?
s1 <http://www.estrellaproject.org/lkif-core/lkif-core.owl#locationOf> ?d.
?

Filter( ?s2 != ?s1).
\}

Notice that the above SPARQL queries are actually implementing rules that retrieve results when several conditions corresponding to a rule body apply. Furthermore exceptions or combinations of rules can be also implemented using SPARQL, even cases that negation is involved (using the SPARQL NOT EXISTS or the MINUS operators\(^{63}\)). For example using a SPARQL query federal cases that are not copyright infringement cases can be retrieved.

Overall combining OWL axioms, rules and SPARQL queries, legal norms can be represented, information can be queried and reasoning can be applied with the exception of conflicting rules that require non-monotonic reasoning.

4.3 Legal norm representation using Answer Set Programming (ASP)

In this section the example case is the same US bar exam case as in the previous section. Logic programming is used for dealing with conflicting rules and the representation of facts axioms and norms. LKIF is not used, although any populated legal ontology can be used for storing and retrieving facts to be used by the reasoning mechanism.

There are several approaches for reasoning over conflicting norms as presented in the background section. In this case Answer Set Programming (ASP) with inheritance networks [39] is used for reasoning since the required expressiveness is supported and corresponding tools are available. Norms are represented as rules, and their priority as inheritance networks. Furthermore ASP supports disjunction, e.g. for rules of the form “if he is registered student or holds a specific permission he can get the books” that are not directly supported by SWRL. Also negation is supported, e.g. for rule “if he is not proven to be guilty he is innocent “, which also cannot be directly represented using SWRL. An ASP program has been created for this example case, and reasoning using the DLV system [40] has been applied. The logic program is the following:

\(^{63}\) https://www.w3.org/TR/sparql11-query/#negation
Ontology-based access to normative knowledge

copyrightInfringment(A,B,X):-sells(A,X),holdCopyright(B,X),A<>B.

isAt(A,S):- from(A,S).

isAt(A,S):- incorporated(A,S).

isAt(A,S):- headquartered(A,S).

diversityCase(A,B):-copyrightInfringment(A, B, X),accuses(B,A),diversityAmount(B,A,M).

federalCase(A,B):-copyrightInfringment(A, B, X),accuses(B,A),isAt(A,S1),isAt(B,S2),S1<>S2.

o\{acceptsClaim(C,A):-accuses(B,A),court(B,A,C),challenges(A,C),not diversityCase(A,B).\}

o1\{acceptsClaim(C,A):-accuses(B,A),court(B,A,C),challenges(A,C),not diversityCase(A,B).\}

o2:o1,o\{acceptsClaim(C,A):-accuses(B,A),court(B,A,C),federalCase(A,B),challenges(A,C).\}

diversityAmount(B,A,M):-claims(B,A,M),M>75000.

claims(b,a,50000).

sells(a,hotsouce),holdCopyright(b,hotsouce).

from(a,stateA).incorporated(b,stateB).

headquartered(b,stateC).isAt(c,stateC).

accuses(b,a).court(b,a,c).challenges(a,c).

The output produced from the program is:

DLV [build BEN/Dec 17 2012   gcc 4.6.1]

\{sells(a,hotsouce), holdCopyright(b,hotsouce), from(a,stateA), incorporated(b,stateB),
headquartered(b,stateC), accuses(b,a), court(b,a,c), challenges(a,c), claims(b,a,50000),
copyrightInfringment(a,b,hotsouce), isAt(b,stateB), isAt(b,stateC), isAt(a,stateA), isAt(c,stateC),
federalCase(a,b), -acceptsClaim(c,a)\}

The fact that copyright infringement occurs is inferred, along with the fact that it is a federal case
and that the court does not accept the defendant’s claim. The above example was executed in less
than 1 sec using DLV. Overall the logic programming approach offers all required expressiveness,
but also calls for a reasoning mechanism on top of the ontology based representation. Notice that
support for preferences is now part of main Answer Set Programming tools [41], and the
abovementioned representation based on inheritance networks is not the only one available when using ASP for legal reasoning.

5 Conclusions and Research Challenges

In this deliverable the representation of legal norms in conjunction with an ontology based representation is examined. The available options and work related both with representation of norms and reasoning are presented. Two specific examples are also presenting, demonstrating the available options.

There are several options for an RDF based representation, and the corresponding ontologies, including lexical ontologies and formal ontologies with complex axioms. Furthermore ontologies can be generic or domain specific. Information retrieval over such a representation can be achieved using the SPARQL query language. SPARQL can be combined with reasoning and both the reasoning capabilities and scalability of SPARQL engines over big data are active areas of research.

Legal norms can be represented using OWL axioms, but this approach is applicable only to small scale applications because of the complexity and the limitations in expressiveness of this approach. In case norms are not conflicting and they don't contain disjunction or negation (i.e., Horn formulas) then norms can be naturally represented using rules (e.g. using SWRL), and in addition these rules can be integrated into the ontology. In the more general case that norms are conflicting and/or contain negation or disjunction then expressive non-monotonic (for dealing with conflicts) logic programming formalisms must be used. This is the most generic approach for realistic use cases.

Creating the reasoning rules from the natural language description is a challenging task requiring legal and technical expertise. Developing auxiliary tools to help users in creating the rules is a research challenge. Furthermore in a realistic large scale use case, the number of rules and facts can be big, thus causing scalability problems. Dealing with such problems and achieving efficient large scale semantic reasoning in the main research challenge to deal with.
Ontology-based access to normative knowledge

References


Ontology-based access to normative knowledge


Ontology-based access to normative knowledge


