Linked Open Data Approach to Publishing Legal Information

Marko Vuković*, Filip Jakovljević*, Marko Marković*, Stevan Gostojić*
* University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
{vukovic.marko171, fillix96}@gmail.com, {markic, gostojic}@uns.ac.rs

***Abstract*—Knowledge of legislation and regulation is of great importance, especially to legal workers. However, finding a document that contains the relevant information is not a trivial task. Existing services that enable storage, retrieval, and browsing of legislation and regulation usually offer simple search features. They do not utilize metadata to the fullest possible extent and do not link metadata with other data published on the (semantic) web. In this paper, we present a software solution based on linked data principles and technologies that offers key features that the existing legislation and regulation retrieval and browsing services have but also introduces advanced features such as advanced search using document metadata and better handling of document references and links. The content is semantically enriched by using data from DBpedia, a project aiming to extract structured knowledge from the content created in various Wikimedia projects, which makes it more usable by other systems. However, the solution has its constraints because it requires documents to follow a specific legal document format.**

Keywords—linked data, open data, legal information, legislation, regulation

I. INTRODUCTION

Legislation and regulation contain information laid out in a way that follows a certain structure and predefined set of rules. They are published by multiple entities referencing each other to make a single system. Most existing services for storage, retrieval, and browsing of legislation and regulation do not utilize metadata to the fullest possible extent to offer advanced retrieval and browsing features to its users. Thus, finding relevant documents is usually a challenging task. Those problems particularly concern legal workers including lawyers, judges, paralegals, but also law students and other interested parties.

For example, Pravno-informacioni sistem [1] and Paragraf Lex [2], services used for storage, retrieval, and browsing of Serbian legislation and regulation, do not utilize such metadata and therefore cannot offer features such as advanced document retrieval and browsing.

We addressed two research questions. The first question is how to publish legislation and regulation in a distributed manner without impairing the possibility to use a single point for its retrieval and browsing. The second question is how to improve the retrieval and browsing of legislation and regulation by linking it with other data published on the (semantic) web.

To be able to answer those questions, we represented legal documents via Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies (Akoma Ntoso) [3] – an international technical standard for representing parliamentary, legislative, and judiciary documents in a structured manner using XML vocabulary, which we adapted to Serbian legal system. Additionally, we represented metadata corresponding to legal documents via ontology we created using Web Ontology Language (OWL) [4] – a semantic web language for representing knowledge about things, groups of things, and relations between things. For querying documents, we used SPARQL Protocol and RDF Query Language (SPARQL) [5] – a query language and protocol for linked open data and RDF databases. Lastly, to adhere to semantic web principles, we utilized available content from DBpedia [6] – a project aiming to extract structured knowledge from the content created in various Wikimedia projects – to make our solution machine-readable and to provide better connectivity to other services.

In this paper, we present a software prototype that offers features that are provided by currently available services – storing legal documents and offering basic retrieval and browsing functionality, as well as offers advanced features such as advanced retrieval and browsing by utilizing advanced metadata.

Other than the Introduction, this paper contains the following sections. Section II. Methodology explains the approach to modeling legislation and regulation and the important aspects of the design and implementation of the publishing system. The ontology and the publishing system are presented in Section III. Results. Section IV. Conclusion concludes the paper by summarizing the problem, solution, advantages, and limitations of the system.

II. METHODOLOGY

In this approach, legislation and regulation are modeled via a modified Akoma Ntoso XML schema which we adapted to the Serbian legal system.

The metadata that describes those documents is modeled via custom ontology. The ontology is based on the OWL standard. The base of the ontology is created by combining FRBR [7] and SKOS [8] ontologies. Then, classes, data properties, and object properties are created to model Serbian legal documents per Akoma Ntoso ontology guidelines. Akoma Ntoso suggests modeling legal document metadata by using two different types of classes. The first group corresponds to FRBR entities work, expression, manifestation, and item. The second group is called top-level classes (TLC). TLC are focused on who is responsible for the production of the content and what the content is about. Lastly, to adhere to linked data principles, we utilized available content from DBpedia to provide better connectivity to other solutions.

The system architecture is shown in Figure 1.

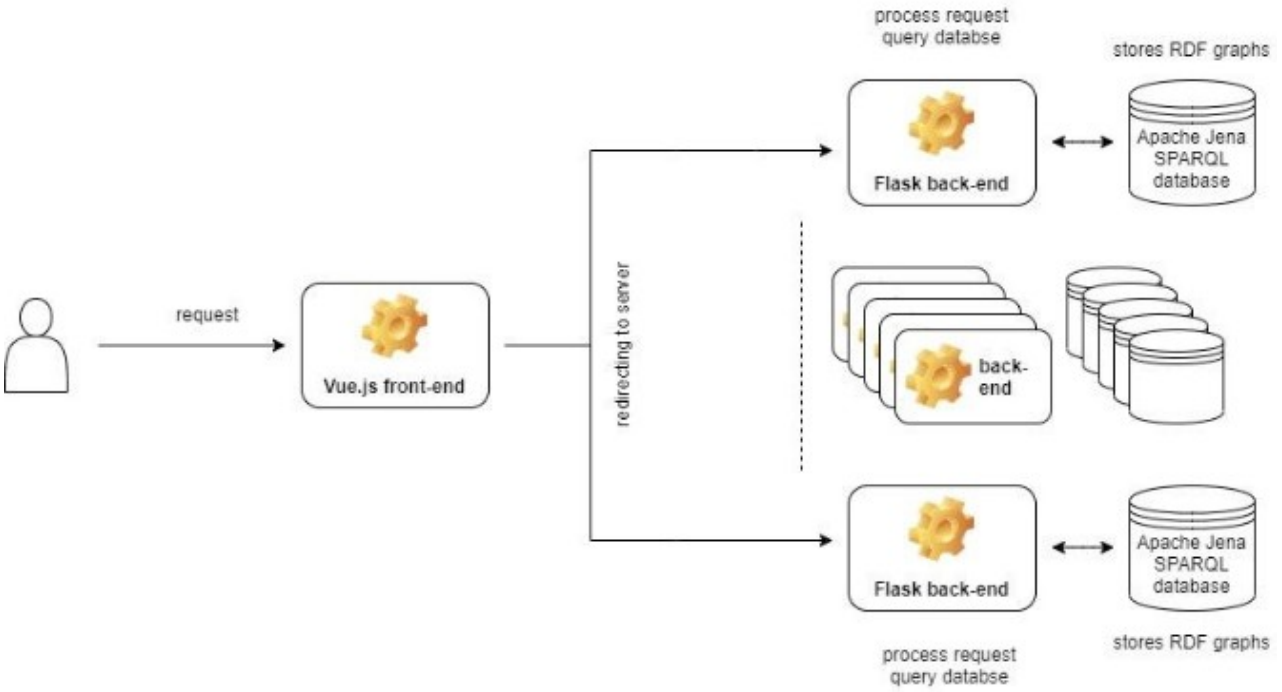

Figure 1: System Architecture.

It is envisioned that the system has multiple server instances with multiple instances of triple stores. This kind of architecture provides a possibility for every entity that publishes legislation and regulations in the electronic form to have its instance of servers and database while providing access to all published legislation and regulations within a single user interface regardless of their location. For example, the Official Gazette of the Republic of Serbia, the Official Gazette of the Autonomous Province of Vojvodina, and the Official Gazette of the City of Novi Sad would have separate server and databases instances, and end-users could search for legislation and regulation through a single front-end interface as if they were all in one place, without knowing their location.

The Quasar Framework [9], a working environment for Vue.js [10] JavaScript environment, was used for developing the web application and user interface implementing the front-end of the application through which end-users use the application.

The Flask web framework [11] in Python and Apache Jena Fuseki [12] were used to implement the back end.

Flask is a lightweight WSGI (Web Server Gateway Interface) web framework and it is treated as a micro framework. It has a very simple and nominalistic design but can be easily extended and upgraded, where the programmer chooses the libraries and tools that will append to it. It is currently one of the most popular Python web frameworks.

Apache Jena Fuseki is a standalone SPARQL server that comes with a tightly integrated high-performance database for persistent storage of RDF graphs. Besides persistent graph storage Apache Jena Fuseki provides SPARQL 1.1 Query Protocol and Update Protocol and its alternative Graph Store HTTP Protocols for managing collections of stored RDF graphs. Graph Store HTTP Protocols follow the RESTful architectural style. One big advantage of Jena is its support for OWL (Web Ontology Language), which provides greater semantics of models. Apache Jena Fuseki is an open-source project written in the Java programming language. It is a reliable and easy-to-use solution, which meets all demands for storing RDF graph data. Since the database is on a separate server and stores RDF graphs, there was no need for an ORM (object-relational mapper) part offered by various web frameworks therefore the extremely simple Flask web framework in Python programming language was chosen.

III. RESULTS

The ontology that is the main result of the research is shown in Figure 2 and the full specification of the ontology is available at [13].

The ontology contains two groups of classes. The first group corresponds to FRBR Group 1 entities (WEMI), which includes Work, Expression, Manifestation, and Item entity.

Work depicts a legal document at the highest level of abstraction. This ontology is represented via the FRBRWork class and contains information about the title and identification number. It is connected via an object property to a class that models the Expression entity.

The FRBRExpression class represents the version and language variant of the document. It contains metadata about the date of the version of the document and the language of the document. Although countries and languages are modeled via FRBR classes, they are connected to data from the DBpedia project. The FRBRExpression is connected via an object property to the class that models the Manifestation entity.

The FRBRManifestation is connected via an object property to a class that models different file formats of a document.

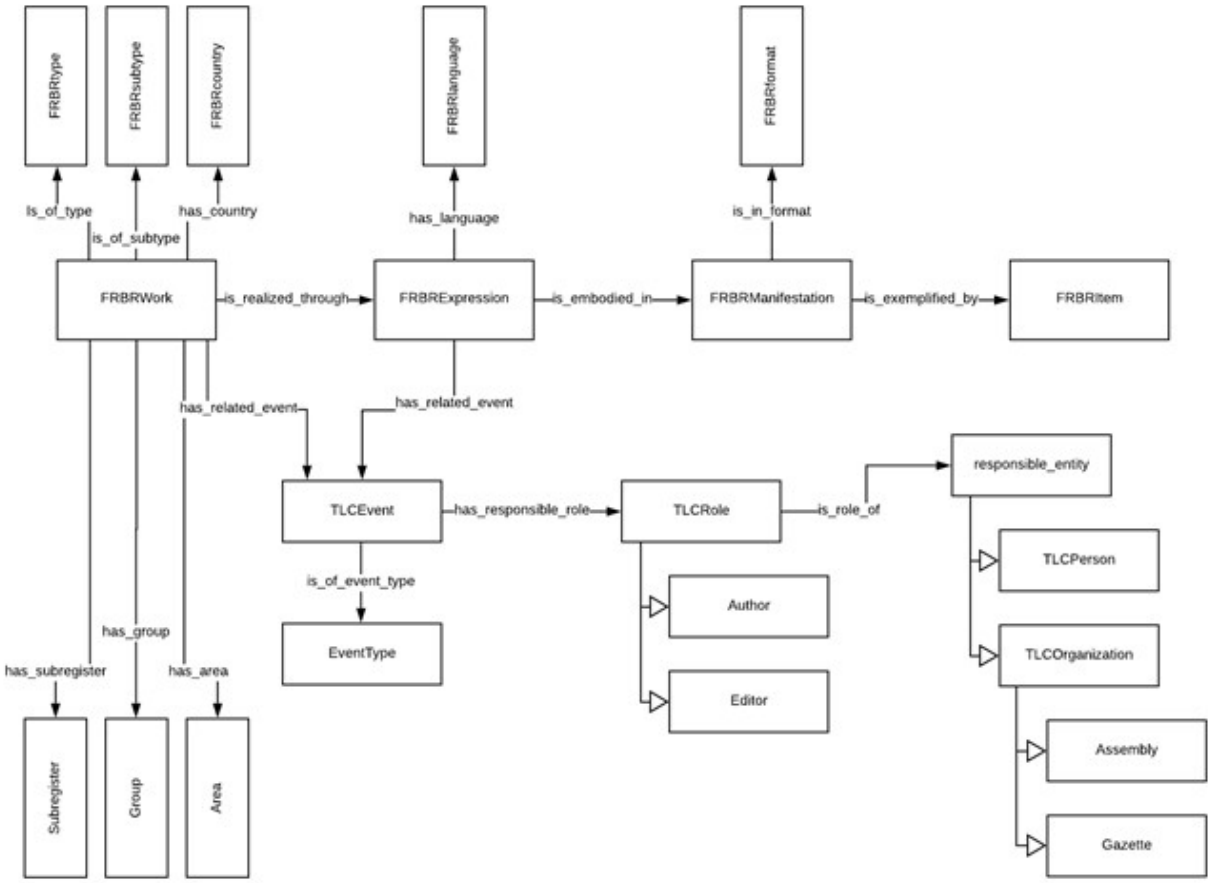

Figure 2: Ontology.

The second group of classes is called Top Level Classes (TLC) and they are focused on two different aspects, namely who is responsible for the production of the content and what the content is about. TLC classes are used to model lifecycle events. Events are connected to Work and Expression entities via an object property. The event is described by information of a responsible entity (person or organization) in a specific role (e.g author, editor, etc.) and the date on which the event occurred. We also used TLC classes to model sub-registers, groups, and areas that the documents can be affiliated with, as well as type, subtype, and keywords which are all connected to Work entity via object properties. Sub-registers, groups, and areas, as well as Assembly and Gazette (subclasses of Organization class mentioned earlier), are modeled as SKOS concepts to utilize object properties denoting hierarchical relations between them.

The source code of the implemented application is available at [14] and the home screen of its user interface is shown in Figure 3. It offers three options for searching data in the database: simple search, search through forms and search through SPARQL queries, which is the most flexible search mode.

In the last search mode, the query is passed directly to the database, which returns an RDF graph that satisfies it. This type of search is very powerful and it is intended for users who are familiar with semantic web techniques and SPARQL queries, allowing them to make very useful and complex searches. Users can specify return format of data (e.g. PDF, XHTML, XML, etc.). Data matching the search will be returned in the desired format. That being said, our database stores data in RDF format only. The desired formats are obtained by transforming retrieved data from RDF.

The back-end also enables data manipulation through a web API. By knowing the URI of an individual belonging to one of the FRBR classes (i.e. Work, Expression, and Manifestation) it is possible to perform CRUD operations on them.

HTTP GET method is used for reading data from the database. It is the simplest request that returns the wanted individual (if it exists).

HTTP DELETE method finds an individual with a given URI (if it exists) and deletes all triplets in which the individual participates directly or indirectly.

HTTP POST method is used for inserting new data into a database. First, we check if the individual with that URI already exists and validate if the data is sent in proper Akoma Ntoso XML format. After this check, we construct the RDF graph from the given data and validate that graph with the RDF schema. If all validations are passed the new graph data is inserted into a database.

The last and the most complex method is HTTP UPDATE. For a given URI, we conduct validation of data if we find that the individual exists (similarly to the HTTP POST method). If validations are passed, we first delete existing data similarly to the HTTP DELETE method and then insert a new graph for that individual.

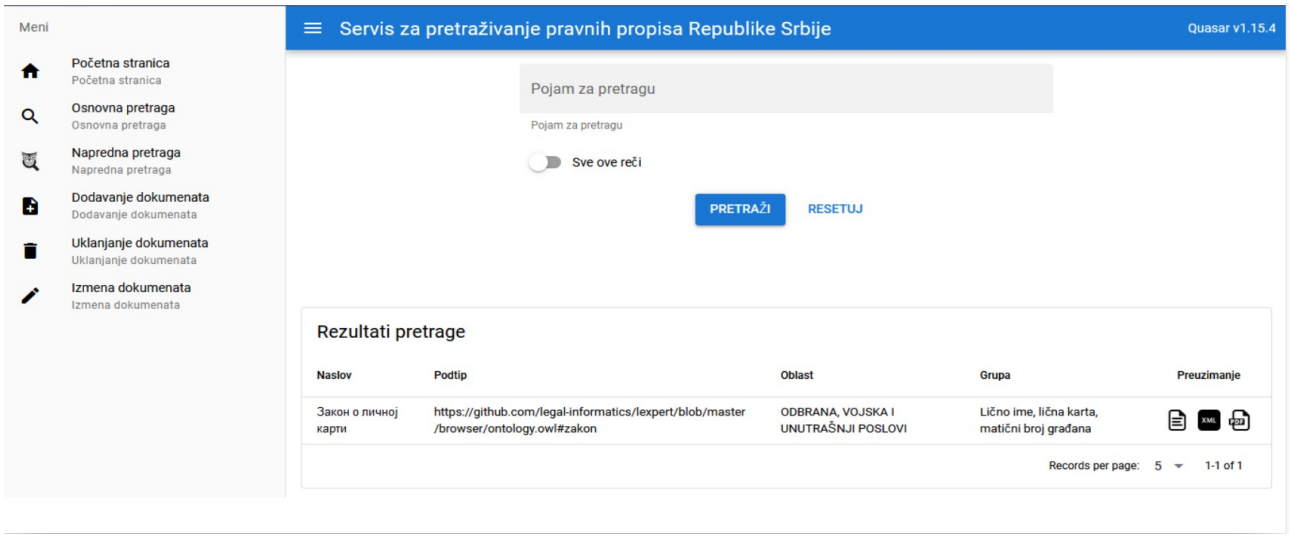

Figure 3: Application's UI.

IV. Conclusion

This paper presents a service that enables distributed publishing of legislation and regulation and offers three modes for retrieval: a simple search, a search through forms, and a search through SPARQL queries (which is the most flexible type of search).

The service described in this paper differs from the existing services in that it makes legislation and regulation available in a standardized legal document format and makes its metadata available as linked data through a SPARQL endpoint. The service offers key functions offered by other services of this type and it adds more advanced retrieval and browsing functions. Also, integrating metadata with the DBpedia goes a step further in striving for machine readability of documents, which provides a richer context for the legislation and regulations it stores and creates the basis for connecting this system with others.

The ontology was implemented by composing other ontologies and by extending them with classes and properties that are specific to the legal system of the Republic of Serbia. By combining semantic web technologies with the widely used RESTful web services technology, the benefits of ontologies come into contact with the end-users.

The disadvantage of the system is its dependence on documents in a specific document format and metadata expressed according to a specific metadata schema. It is not realistic to expect that a large quantity of legislation and regulation published in Serbia will be marked up manually. Another software solution that complements the service described, which enables automated conversion of regulations from textual to a standardized format, overcomes this shortcoming. Another possibility for improving the system is describing the documents in a semantically richer manner. This would even further improve the described functionalities of retrieval and browsing legislation and regulation.

Acknowledgment

This research was supported by the Science Fund of the Republic of Serbia, Program DIASPORA, #6497472, AI4Law.

References

[1] Pravno-informacioni system RS. 2020. Pravno-informacioni system RS. [ONLINE] Available at: http://www.pravno-informacioni-sistem.rs/. [Accessed 01 December 2020].

[2] Paragraf Lex. 2020. Paragraf Lex | Elektronska pravna baza | Online baza propisa. [ONLINE] Available at: https://www.paragraf.rs/. [Accessed 01 December 2020].

[3] Akoma Ntoso | Akoma Ntoso Site. 2020. Akoma Ntoso | Akoma Ntoso Site. [ONLINE] Available at: http://www.akomantoso.org/. [Accessed 01 December 2020].

[4] W3.org. 2020. *OWL - Semantic Web Standards*. [online] Available at: https://www.w3.org/OWL/. [Accessed 01 December 2020].

[5] W3.org. 2020. *SPARQL Query Language For RDF*. [online] Available at: https://www.w3.org/TR/rdf-sparql-query/. [Accessed 01 December 2020].

[6] DBpedia. 2020. DBpedia. [ONLINE] Available at: https://wiki.dbpedia.org/. [Accessed 01 December 2020].

[7] vocab.org. 2020. Expression of Core FRBR Concepts in RDF. [ONLINE] Available at: https://vocab.org/frbr/core. [Accessed 01 December 2020].

[8] World Wide Web Consortium (W3C). 2020. SKOS Simple Knowledge Organization System Namespace Document 30 July 2008 "Last Call" Edition. [ONLINE] Available at: https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html. [Accessed 01 December 2020].

[9] Quasar Framework. 2020. Quasar Framework. [online] Available at: https://quasar.dev/. [Accessed 1 December 2020].

[10] Vuejs.org. 2020. *Vue.Js*. [online] Available at: https://vuejs.org/. [Accessed 01 December 2020].

[11] Flask. 2020. Flask Documentation (1.1.x). [ONLINE] Available at: https://flask.palletsprojects.com/en/1.1.x/. [Accessed 01 December 2020].

[12] Apache Jena. 2020. Jena documentation overview. [ONLINE] Available at: https://jena.apache.org/documentation/. [Accessed 01 December 2020].

[13] Vuković, M., Jakovljević, F., Marković, M., and Gostojić, S. 2020. Legal Document Ontology. [ONLINE] https://github.com/legal-informatics/lexpert/blob/master/browser/ontology.owl. [Accessed 01 December 2020].

[14] Vuković, M. and Jakovljević, F. 2020. Legal Document Browser. [ONLINE] https://github.com/legal-informatics/lexpert/blob/master/browser/. [Accessed 01 December 2020].