INTLIB – an INTelligent LIBrary[⋆]

Irena Holubová[1], Tomáš Knap[1], Vincent Kríž[2], Martin Nečaský[1], and
Barbora Vidová-Hladká[2]

[1] Department of Software Engineering
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
holubova@ksi.mff.cuni.cz
[2] Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
hladka@ufal.mff.cuni.cz

Abstract. In this paper we describe the project *INTLIB – an INTelligent LIBrary* whose aim is to provide a more sophisticated and user-friendly tool for querying textual documents than full-text search. On the input we assume a collection of documents related to a particular problem domain (e.g., legislation, medicine, environment, etc.). In the first phase we extract from the documents a *knowledge base*, i.e. a set of objects and their relationships, which is based on a particular ontology (semantics). In the second phase we deal with sophisticated and user friendly visualization and browsing (querying) of the extracted knowledge. The whole system is proposed as a general framework which can be modified and extended for particular data domains. To depict its features we use the legislation domain.

1 Introduction

Nowadays, large collections of documents form one of the main sources of information and their sophisticated browsing or querying is the key aspect in many areas of human activity. Existing solutions to the problem of searching large collections of documents typically implement two approaches. The *full-text search* allows the user to find documents with the highest frequency of occurrences of a specified set of keywords. The search is automatically optimized using a pre-generated index that keeps track of the occurrences of keywords. Other approaches enable to search, e.g., the co-occurrence of words, specify their proximity, etc. By contrast, the *metadata search* allows the user to find documents with given properties (such as, e.g., author, creation date, expiration date, list of keywords, etc.). Nevertheless, the metadata are assigned to the documents manually and, thus, inefficiently and expensively.

In general, both the common approaches do not work with the *semantics* (meaning) of the documents in the collection. For example, considering the legislation, we may need to know that the term "the High Court" means a particular institution in a particular country that has certain powers and relations to the Constitutional Court. To enable the user to access the data this way means:

[⋆] This work was supported by the TAČR grant INTLIB, no. TA02010182.

J. Pokorný, K. Richta, V. Snášel (Eds.): Dateso 2014, pp. 13–24, ISBN 978-80-01-05482-6.

1. to interpret the semantics of the documents in terms of real-world objects and the relationships between them which are described in the documents,
2. to transform the interpretation into a suitable database preferably having a standard format and standard query language, and
3. to present the interpretation to the user in a form which enables sophisticated, precise and user-friendly browsing and filtering.

In this paper we describe project *INTLIB – an INTelligent LIBrary* whose aim is to provide a more sophisticated and user-friendly tool for querying textual documents than full-text or metadata search. On the input we assume a collection of human-written documents related to a particular problem domain. INTLIB processes the data in two phases. In the *extraction phase* we extract from the documents a *knowledge base*, i.e. a set of objects and their mutual relationships, which is based on a particular ontology. The extraction phase first exploits and utilizes linguistic approaches and machine learning techniques. Then it applies algorithms for cleaning and linking of the data, and their transformation to RDF [9]. In the *presentation phase* we deal with efficient and user-friendly visualization and browsing (querying) of the extracted knowledge. The whole system is proposed as a general framework which can be modified and extended for various data domains using plug-ins. Naturally, each of the domain may require specific features; however, the general methodology we propose will remain the same. To depicts the features of the framework we use the legislation domain and we implement plug-ins that process the legislation of the Czech Republic.

The rest of the paper is structured as follows: In Section 2 we provide a larger motivating example from the area of legislation. In Section 3 we describe the current systems used for legislation processing in the Czech Republic and in Section 4 solutions used abroad. In Section 5 we propose the architecture of INTLIB and describe particular modules. In Section 6 we conclude and outline future work.

2 Motivating Example: Legislation

Sources of law are usually structured into *sections* which may contain further subsections. Moreover, a source of law may contain links to other sources which may target not only a whole source of law but also its particular section. Therefore, the structure encoded in sources of law and links between them form a complex network which the users want to browse and search for relationships between sources of law and/or their parts. Common use cases are, e.g.:

- A user is reading a particular section of an act. He would like to see what court decisions have been made in the last decade related to this particular section.
- A user is working with a particular amendment. He would like to see what sections of what acts have been corrected by this amendment or by its section.
- A user is reading a particular section of an act. He would like to find out what amendments correcting the chosen section will come to force in the next year.

A natural solution is to enable machines to search for the relationships. However, much has to be done to achieve an efficient software solution. In particular, we have to find out ways of how to:

- automatically extract the logical structure of sources of law,
- assign unique machine interpretable identifiers to the sources of law as well as to their sections and subsections so that they can be linked,
- automatically extract the links between sources of law and their parts,
- represent the extracted structure and links in a data format suitable for representing generic graph structures.

In INTLIB we concentrate on both recognizing the logical structure of source of law and recognizing references (links) between them automatically in their textual representations. We also propose a data structure which allows us to represent the recognized structure and links in a way suitable for further database processing.

Besides the logical structure and links, sources of law contain also semantic information. This is mainly the case of acts (and their amendments). Acts and other sources of law define *rights* and/or *obligations* of *natural* and *legal persons*. Different sources of law define different rights and obligations for the same kind of natural or legal person or for different persons which are, however, semantically related (e.g. one person is a special type of another person and it "inherits" the rights and obligations). Therefore, the rights and obligations of persons defined by acts and other sources of law form a complex network, similar to the described network of links among sources of law. In this case the network is defined by the semantic information encoded in the sources of law and we can therefore speak about a *semantic network* or a *knowledge graph*. Again, it would be useful for users to be able to browse and query such network. We list some sample common use cases and demonstrate them in Figure 1:

- A user wants to know what are the obligations of his employer regarding his health insurance. For example, according to the sample network depicted in Figure 1, the user can get information that his employer has an obligation to record employee's documentation, notify insurance company about changes in case of changes in employee's information, etc.
- A user wants to know what kind of information his health assurance company has to provide him. For example, according to Figure 1, the user can see that he has the right to obtain information from his insurance company about services provided and paid by the company as well as information about prices of services which are paid by him.

A software solution which enables browsing the network of the semantic concepts and relationships and automates searching and querying the network and its visualizations would be helpful for users. However, it is again necessary to solve various problems, such as to:

- automatically extract the semantic concepts and relationships between them from the textual representation of sources of law,
- assign unique machine interpretable identifiers to the concepts so that they can be linked on each other and other extending information can be linked on them,
- represent the extracted concepts and links between them in a data format which allows further database processing.

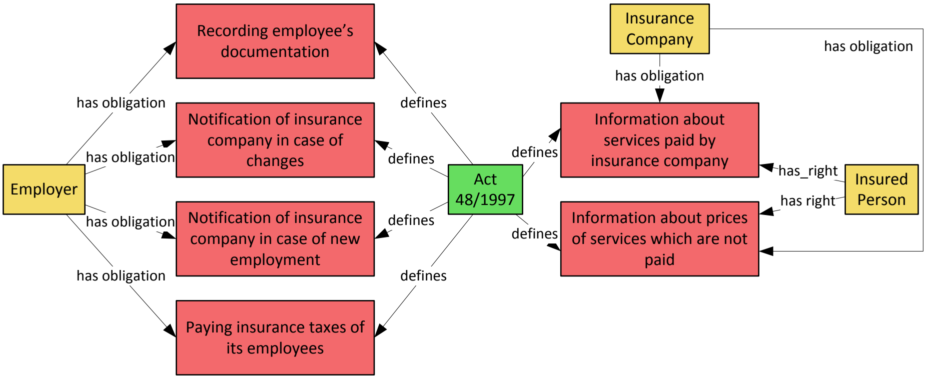
Recording employee's documentation
Notification of insurance company in case of changes
Notification of insurance company in case of new employment
Paying insurance taxes of its employees
Employer
has obligation
has obligation
has obligation
has obligation
defines
defines
defines
defines
Act 48/1997
defines
defines
Insurance Company
has obligation
has obligation
Information about services paid by insurance company
Information about prices of services which are not paid
has_right
has right
Insured Person

Fig. 1. Sample of semantic concepts extracted from Public Health Act valid in Czech Republic

3 Current Czech Legislation and Related Systems

In the Czech Republic, there currently exist several systems that provide an access to (a subset of) law, court decisions or other related information in an electronic form. Some of them even claim to provide the *consolidated versions of acts*, nevertheless none of them is an official version to be approved by the Head of the Parliament (who is responsible for it). If fact, even the members of the Czech Parliament work with these systems, i.e. with unofficial data. The solution to the problem is being provided in two closely related projects proposed by the Czech Government – *eSbirka*[3] and *eLegislativa*[4] – whose aim is (1) to provide official and approved version of the consolidated versions of acts in the electronic form and available to anyone and (2) to enable to speed up the legislature process in the Czech Republic via direct amending of these official electronic acts. The problem of these systems is currently the financial support. It is not the question of preparing an electronic version of the documents or suitable interfaces for various types of users (a citizen, a member of the Parliament, a Head of the Parliament, etc.), but it requires a tremendous effort of experts in law to solve known ambiguities, to study historical acts that are still valid and not in accordance with newer acts, etc.

In the following sections we provide a brief overview of existing systems that enable to browse and query (a part of) the Czech legislation.

3.1 ASPI

System *ASPI*[5] from the Wolters Kluwer, Czech Republic is currently one of the most popular systems that enable to browse and query electronic version of legislation and related data. In addition, being a publishing company, the vendor provides an interesting and important extension – an access to related basic *literature* where various acts are

[3] eLegislation in English
[4] eLegislature in English
[5] `http://www.systemaspi.cz/` [in Czech]

explained, commented and discussed. Considering the browsing and querying aspects, it supports full-text search supporting also all grammatical forms of Czech or Slovakian respectively. The search can cover all texts, or selected parts such as titles, content, appendices, notes, or tables of contents. The system enables to filter the documents according to meta data, such as identification (e.g. file numbers of court decisions), date of issue (valid versions or older versions), or issuing institution (e.g. the Constitutional Court, the Supreme Administrative Court etc.).

3.2 LexGalaxy

LexGalaxy[6] is another tool which enables to browse and search the legislation. It involves a selection of 100,000 documents from the constitutional order of the Czech Republic from 1918. The system includes also a selected information on law of the European Union and the European Court of Human Rights. The search in the legislation can be done in three ways – using full-text search, document identifications, or indexes. Full-text search enables to specify the searched area (e.g. paragraph, title, appendix, etc.) and supports various grammatical forms of the searched words. The indexes involve types of documents (e.g. acts, Constitutional Court decisions, etc.), date of issue, validity, issuing institution, etc. The search conditions can be combined using logical and proximity operators.

3.3 Public Administration Portal *Portal.Gov.cz*

The Public Administration Portal *Portal.Gov.cz*[7] involves various information for citizens, entrepreneurs and businessmen, foreigners living in the Czech Republic, and public authorities. The functionalities involve also a module for simple full-text search in legislation. It enables to search in texts of acts, their titles or according to their number.

4 Current Research Projects and Foreign Solutions

The first research solutions in the considered area are the techniques for automated *categorization* of documents or their parts based on *machine learning* [17]. They are able to assign each document or its part (e.g. a paragraph) a category from a given set. The methods achieve good results in case of categorization of collections of documents from a narrowly focused domain of interest.

The next step in this field is the usage of an *ontology* instead of a set of categories. An ontology formally describes the semantics of the domain of interest; however, in addition to the categories of objects they also describe possible relationships between objects. The aim is to interpret individual parts of the document just against the ontology which is actually an extension to the methods described in [17]. The current literature is currently focussed in extending these applications in the biomedical field [4]. In the area of legislative data; however, their application has not yet been sufficiently explored.

[6] `http://www.legsys.cz/` [in Czech]

[7] `http://portal.gov.cz/app/zakony/?path=/portal/obcan/` [in Czech]

The interpreted objects and relationships between them can be further effectively represented in the RDF data model. At present there are many methods for database processing of RDF data, although yet especially at the level of basic research [10].

4.1 The JURIX Conference

The *Foundation for Legal Knowledge Based Systems* (JURIX)[8] is a forum for researchers in the field of Law and Computer Science in the Netherlands and Flanders. From the point of view of INTLIB we can find several interesting papers which deal with enhancing the way legislation is searched using semantics of the data.

Paper [14] distinguishes the relevant approaches into *knowledge-engineering*, involving artificial intelligence or case-base reasoning, and *natural language processing*. The authors claim that the former class suffers from several problems, such as domain specificity or high financial cost, and they argue that natural language processing is promising. Paper [12], similarly, analyzes whether machine learning techniques or knowledge-engineering approaches are better for classification of sentences in laws. The conclusion is that both the approaches reach similar results; however, the machine learning techniques are naturally sensitive to the training set and its correspondence to the analyzed data. In paper [13] the authors apply a thesaurus-based statistical indexing technique to retrieve relevant case law from 68,000 court verdicts. It is based on classical vector space model extended with thesaurus so that only terms from a particular domain are considered. Finally, paper [8] describes the results of a study consisting of two tasks: (i) how the "obligation" Fundamental Legal Concept is differently represented in the FrameNet[9] resource, in terms of Semantic Frames, and (ii) how the concept of "public function" stemmed from the "obligation" Fundamental Legal Concept can be ontologically characterized. The *FrameNet* project is building a lexical database of English that is both human- and machine-readable, based on annotating examples of how words are used in texts.

In general, the papers prove that our aim is right and that it is crucial to create a general system that enables to work with the legislation data in a more sophisticated way provided by extending them with semantics. The experiments show that the strategy is promising; however, such a system is still missing.

4.2 Plans of the European Union

Another important related work can be found in strategies and plans of the European Union. The final report of Working group "Indexing and Search" [7] identifies and recommends best practices and technologies which highly correlate with our aim and which thus confirms that the strategy is promising a should be further extended.

5 INTLIB Architecture

As we have mentioned in the Introduction, there are two key parts of the INTLIB project – extraction and presentation, i.e. creating the knowledge base and its user-friendly

[8] `http://www.jurix.nl/`

[9] `https://framenet.icsi.berkeley.edu/fndrupal/`

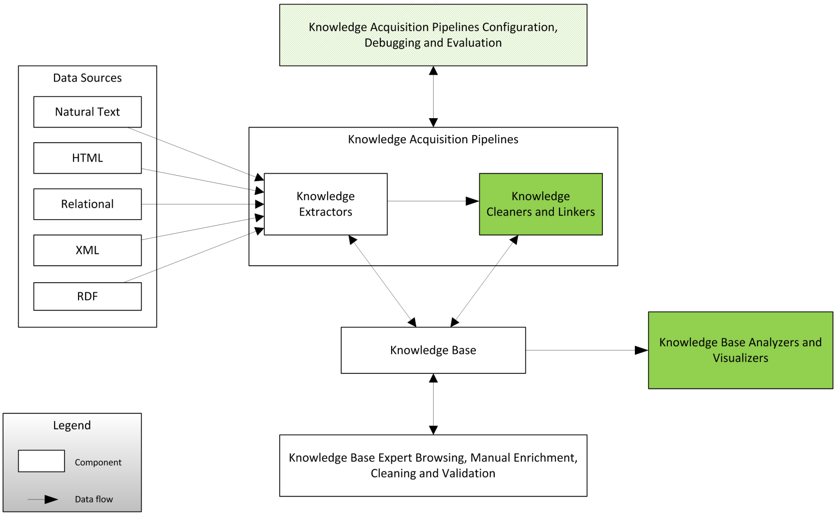
Knowledge Acquisition Pipelines Configuration, Debugging and Evaluation
Data Sources
Natural Text
HTML
Relational
XML
RDF
Knowledge Acquisition Pipelines
Knowledge Extractors
Knowledge Cleaners and Linkers
Knowledge Base
Knowledge Base Analyzers and Visualizers
Knowledge Base Expert Browsing, Manual Enrichment, Cleaning and Validation
Legend
Component
Data flow

Fig. 2. Architecture of system INTLIB

browsing. Since our aim is a general framework utilizable and extensible for particular problem domains with plug-ins, the architecture is more complex. It is based on the idea of *pipelines* which specify the selected steps of the process.

5.1 System Architecture

The architecture of the system is depicted in Figure 2. On the input we can assume various types of data, i.e. not only documents with a natural text in, e.g., PDF [3] format (i.e. in our case acts and court decisions), but also HTML [16] or XML [11] documents, data stored in a relational database, RDF [9] triples etc.

The data are first provided to the *knowledge acquisition pipelines* which extract the knowledge base, i.e. the objects described in the data, their relations and properties. The pipelines need to be first configured, i.e. particular modules need to be selected. The configured pipeline also needs to be debugged and evaluated. The pipelines consist of *knowledge extractors* and *knowledge cleaners and linkers*. The cleaners ensure, that the data extracted from different data sources of various quality are cleaned, e.g., the duplicities and false candidates are removed. The linkers map the newly extracted data to the current data in the knowledge base.

Apart from automatic knowledge base extraction, cleaning and linking, the system also involves a user interface that enables browsing the knowledge base and its manual enrichment (i.e adding new components), as well as cleaning and linking. The aim of this part is:

1. to provide a preliminary interface which enables to create at least a basic knowledge base for testing related modules,

2. to enable to add data that cannot be added manually (due to various reasons, such as, e.g., confidentiality of data sources or limitations of the automatic knowledge base extraction part), and
3. to enable to confirm or refute candidates for linking, discarding, unifying etc.

An emphasis is put on the GUI, because in this case we assume a user which is an expert in the particular domain (i.e. a lawyer), but not in the RDF representation and related technologies such as SPARQL. In preliminary stages of the implementation the module provides all possible candidates to be confirmed/refuted. Later we will add also filtering and cleaning modules that show only possible candidates for correction of discarding.

The last but not least part of the system involves the module which enables to *visualize the knowledge base* and in particular *analyze its content* in a user friendly manner. The analytical part involves an advanced query interface including smart hints, learning from previous user queries and results, etc. For the purpose of visualization of the data we will utilize the SW project *Payola* [5] which enables to visualize graph data in a user friendly manner.

5.2 System Pipelines

Systems pipelines can in general form an acyclic graph whose nodes represent particular *modules* which process the data and edges represent inputs and outputs of the modules. In general, the exchanged data can be of any format that is understood by the respective output/input module; however, for our case and for the sake of simplicity and clarity we assume RDF data in all the cases (if not stated otherwise).

The idea of pipelines results from a SW project *ODCleanStore* [6] which supports textual configuration of pipelines and respective modules. In case of INTLIB the user interface is more friendly, providing a graph visualization of the pipeline and a set of forms that enable to fill in the respective parameters of the particular modules (based on the idea of *XForms* [2] and *VAADIN* [1]).

Examples of two use cases represented by pipelines are depicted in Figure 3. In the former case we can see a set of pipelines for extraction of references among acts, in the latter case a set of pipelines which recognizes structure and terms used in particular acts. In both the cases first the *annotation pipeline* annotates “interesting” parts of the input data, i.e. substrings of acts that represent references, terms, parts etc. Next, the *extraction pipeline* processes the annotated text and select parts which truly represent particular items. The *transformation pipeline* deals with cleaning of the extracted data, whereas it can be even empty when we know that the previous steps cannot produce duplicities. Finally, the *loader* ensures loading of the extracted and cleaned data into the knowledge base, including the linking phase.

The system involves also a *scheduler* which enables to run the pipelines periodically (e.g. in case a pipeline crawles data refreshed or extended gradually), after another pipeline finishes, etc. Similarly, for the purpose of debugging the system involves also a *debugger*. It enables, e.g., to log intermediate results of particular modules of pipelines, run the modules in a debugging mode with extended reports on status, or running only a part do the pipeline (e.g. a selected path or subgraph).

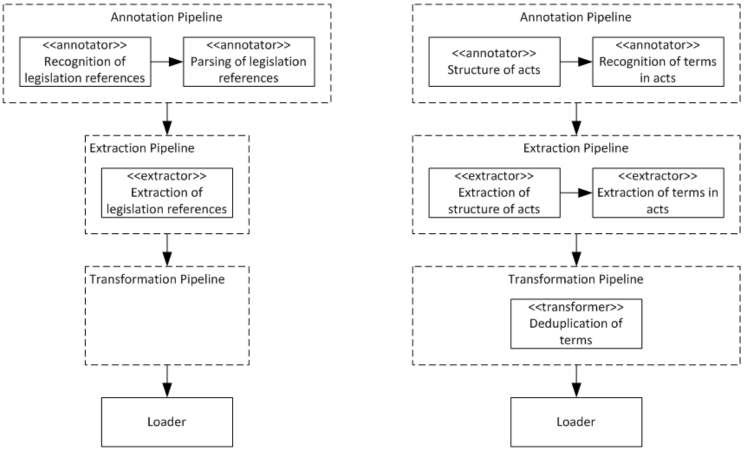
Annotation Pipeline
<<annotator>> Recognition of legislation references
<<annotator>> Parsing of legislation references
Extraction Pipeline
<<extractor>> Extraction of legislation references
Transformation Pipeline
Loader
Annotation Pipeline
<<annotator>> Structure of acts
<<annotator>> Recognition of terms in acts
Extraction Pipeline
<<extractor>> Extraction of structure of acts
<<extractor>> Extraction of terms in acts
Transformation Pipeline
<<transformer>> Deduplication of terms
Loader

Fig. 3. Sample pipelines of the system

5.3 LEX Ontology

The goal of the LEX ontology is to enable to represent the legislation (sources of law) in a machine-readable form conforming to Linked Data principles. Data from other data sources can be linked to legislation represented according to the LEX ontology and, therefore, enriched with the legislative information for further processing by machines. Legislation can also be linked to other data sources.

As we have already indicated, there are different kinds of sources of law:

- *act* is a source of law enacted by a national or regional parliament,
- *decree* is a source of law issued by a national or regional government, ministry, or another public authority
- *regulation* is a source of law issued by a national or regional government, ministry, or another public authority which complements and/or specifies an act,
- *court decision* is a source of law issued by a court as an official decision in a particular legal case,

A source of law can also change another source of law. In that case, we call the source of law *amendment*. An amendment of a given kind can change a source of law which is of the same kind. For example, an act may change another act.

The LEX ontology introduces the following classes for the kinds of sources of law mentioned above: `lex:SourceOfLaw` for sources of law of all kinds (superclass of all other classes), `lex:Act` for acts, `lex:Decree` for decrees, `lex:Regulation` for regulations, and `lex:Decision` for court decisions. (We omit the respective UML diagram for simplicity and space limitations.)

Sources of law of most kinds (except of court decisions) exist in different versions. Some versions are outdated, at most one version is currently valid, and some versions are enacted but have not come to force yet. From this viewpoint, it is reasonable to represent a source of law as an abstract notion of intellectual creation which is independent of

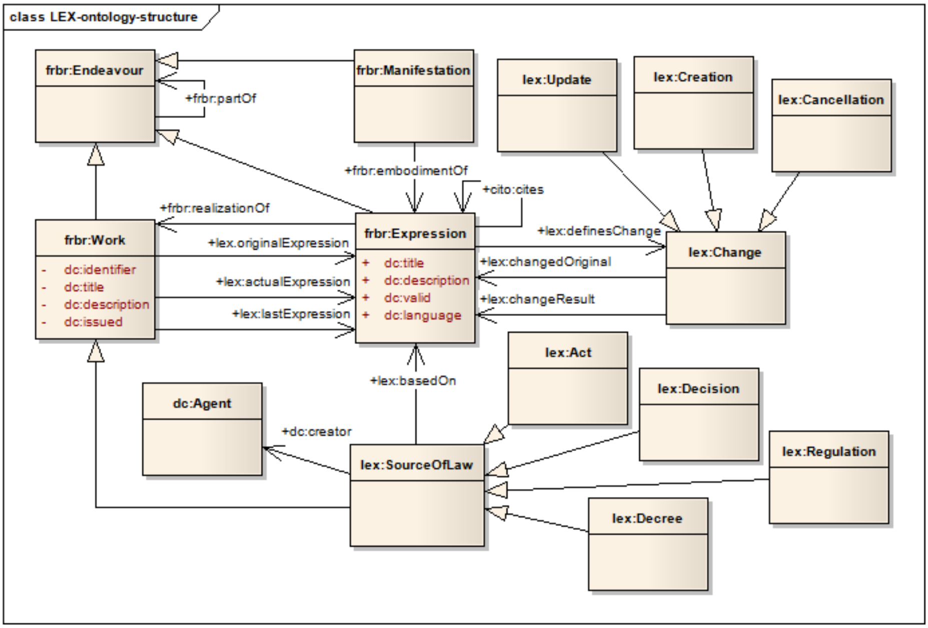

Fig. 4. Legislation Ontology LEX

particular versions of the source. Moreover, each version of the source as well as its each physical embodiment should have representation on its own. This logic is built into the LEX ontology. However, we do not introduce own ontological constructs but reuse the Functional Requirements for Bibliographic Records (FRBR)[10] ontology (as depicted in Figure 4). We reuse the following three FRBR classes: `frbr:Work` for abstract notions of an intellectual creation which are sources of law, `frbr:Expression` for particular versions of sources of law, and `frbr:Manifestation` for particular documents which are physical embodiments of particular versions of sources of law. The usage of FRBR allows us to distinguish a source of law itself, its particular versions and their physical embodiments. From the linked data point of view, it is therefore possible to link and query the source of law as an abstract entity which is independent of particular versions of the source. It is also possible to link and query its particular versions and also their amendments.

We also reuse two FRBR properties: `frbr:realizationOf` to link a version (member of `frbr:Expression`) to its source of law (member of `frbr:Work`) and `frbr:embodimentOf` to link a document (member of `frbr:Manifestation`) to a version of a source of law it is embodiment of (member of `frbr:Expression`). Because each source of law (member of `lex:SourceOfLaw`) is also a member of `frbr:Work` we set `lex:SourceOfLaw` as a subclass of `frbr:Work`.

For a given source of law we need to know its currently valid version, original version (i.e. the first version), and the last enacted version (which have not necessarily

[10] `http://www.loc.gov/cds/downloads/FRBR.PDF`

needed to come to force yet). For this, we introduce three new properties in the LEX ontology: `lex:originalExpression` to link the original (first) version to the respective source of law, `lex:actualExpression` to link the currently valid version to the respective source of law, and `lex:lastExpression` to link the last enacted version to the respective source of law.

Last but not least, we also model changes in legislation with `lex:Change` class. We distinguish three subclasses for three specific kinds of changes: `lex:Creation` to model that something new has been created, `lex:Cancellation` to model that something existing has been removed and `lex:Update` to express that something existing has been updated. More information on the LEX ontology can be found in [15].

6 Conclusion

The aim of this paper was to describe the first phase of project *INTLIB – an INTelligent LIBrary*, i.e. analysis of the problem domain, architecture of the project and related ontology for legislation documents. The target of the project is to provide a general framework for extraction of knowledge from input data (of any kind) so that more advanced querying than usual full-text is possible. Using plug-ins it can be utilized for a particular kind of data – in the first phase the legislation documents.

In the following phases of the project we will naturally focus on implementation of all parts of the system described in Section 5. The key emphasis will be first put on the user interface, configuration and evaluation parts and interfaces between the modules, so that in the next phase we can focus on implementation of all the related plug-ins for legislation processing. As a future work we plan to use the system in other applications, such as environmental reports, policies of companies, etc.

References

1. *VAADIN.* W3C Recommendation, 20 October 2009. `https://vaadin.com/`.
2. *XForms 1.1.* W3C Recommendation, 20 October 2009. `http://www.w3.org/TR/xforms/`.
3. *ISO 32000-1:2008: Document Management – Portable Document Format.* Adobe, 2008.
4. Current Issues in Biomedical Text Mining and Natural Language Processing. *Journal of Biomedical Informatics*, 42(5):757 – 759, 2009.
5. *Payola.* Student SW Project, Charles University in Prague, Czech Republic, 2012. `https://github.com/payola/`.
6. *ODCleanStore.* Student SW Project, Charles University in Prague, Czech Republic, 2013. `http://sourceforge.net/p/odcleanstore/home/Home/`.
7. A. Gola et al. *Working Group Indexing and Search – Final Report.* European Forum of Official Gazettes, Riga, Latvia, 2011.
8. T. Agnoloni, M. Fernandez, M. Sagri, D. Tiscorni, and G. Venturi. When a FrameNet-Style Knowledge Description Meets an Ontological Characterization of Fundamental Legal Concepts. In *AI Approaches to the Complexity of Legal Systems*, volume 6237 of *LNCS*, pages 93–112. Springer Berlin Heidelberg, 2010.

9. D. Beckett. *RDF/XML Syntax Specification (Revised).* W3C, February 2004. `http://www.w3.org/TR/rdf-syntax-grammar/`.
10. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data – The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
11. T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. *Extensible Markup Language (XML) 1.0 (Fifth Edition).* W3C, 2008. `http://www.w3.org/TR/xml/`.
12. E. de Maat, K. Krabben, and R. Winkels. Machine Learning versus Knowledge Based Classification of Legal Texts. In *JURIX 2010*, pages 87–96, Amsterdam, The Netherlands, 2010. IOS Press.
13. M. Klein, W. van Steenbergen, E.Uijttenbroek, Arno Lodder, and F. van Harmelen. Thesaurus-based Retrieval of Case Law. In *JURIX 2006*. IOS Press.
14. K. T. Maxwell and B. Schafer. Concept and Context in Legal Information Retrieval. In *JURIX 2008*, pages 63–72, Amsterdam, The Netherlands, 2008. IOS Press.
15. M. Necasky, T. Knap, J. Klimek, I. Holubova, and Barbora Vidova-Hladka. Linked Open Data for Legislative Domain – Ontology and Experimental Data. In *LIT 2013, Poznan, Poland*, pages 172 – 183, Poznan, Poland, 2013. Springer-Verlag.
16. D. Raggett, A. Le Hors, and I. Jacobs. *HTML 4.01 Specification.* W3C, December 1999. `http://www.w3.org/TR/html401/`.
17. F. Sebastiani. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.