Towards Automatic Structuring and Semantic Indexing of Legal Documents

Marios Koniaris
KDBS Lab, School of ECE,
National Technical University
of Athens, Greece

George Papastefanatos
Institute for the Management
of Information Systems,
Athena Research Center,
Greece

Yannis Vassiliou
KDBS Lab, School of ECE,
National Technical University
of Athens, Greece

ABSTRACT

Over the last years there has been a great increase on the number of freely available legal resources. Portals that allow users to search for legislation, using keywords are now a common place. However, in the vast majority of those portals, legal documents are not stored in a structured format with a rich set of meta data, but in presentation oriented manifestation, making impossible for the end users to inquiry semantics about the documents, such as date of enactment, date of repeal, jurisdiction, etc. or to reuse information and establish an interconnection with similar repositories. In this paper, we present an approach for extracting a machine readable semantic representation of legislation, from unstructured document formats. Our method exploits common formats of legal documents to identify blocks of structural and semantic information and models them according to a popular legal meta-schema. Our proposed method is highly extensible and achieves high accuracy for a variety of legal and para legal documents, especially legislation. Our evaluation results reveal that our methodology can be of great assistance for the automatic structuring and semantic indexing of legal resources.

CCS Concepts

•**Information systems** → *Expert systems;* •**Applied computing** → *Law; Document analysis;*

Keywords

Legislation, legal text analysis, natural language processing.

1. INTRODUCTION

Nowadays, as a consequence of many open data initiatives, a plethora of publicly available portals and datasets provide legal resources (mainly legal documents) to citizens, researchers and legislation stakeholders. Legal information, previously accessible only by a specialized audience, is now freely available on the internet. Although this open access is usually supported by services for keyword search, thematic categorization and browsing, legal resources are mostly disseminated in a semantically poor textual representation, which can not capture the structure and the legal semantics of the data or the interconnection between different resources.

Recent advancements in the Data web [5] a web of data that can be processed directly and indirectly by machines, provide a solid background for the foundation of legal semantic web [4], where (legislative) information is machine understandable, identifiable across sites, interconnected and processable according to its legal meaning. Initiatives on adoption of XML standards for the representation of legislative document structures and metadata have been recently brought on at national and international level in different countries [6],[18], [3]. A necessary precondition for effective legal document dissemination is the electronic availability of legal sources in a structured and standard format.

Moreover, the World e-Parliament Report 2010 survey [22] found that only 34% of parliaments with document systems in place (14% total) currently used XML, and even less use XML as a format for distributing public documents. Data is mainly offered to the end user in a friendly and human readable manifestation, primary presentation oriented. The use of such proprietary and unstructured format (PDF/A) makes it impossible to establish an interoperability layer among the different sources of information, to allow reuse and interconnection with repositories in the Semantic Web and to use reasoning tools in the text or semantic ontologies to manage texts.

Motivated by the aforementioned lack of availability of legal documents in a structured and standard format, in this work, we present a novel methodology that acquires a semantic representation of legislation, from unstructured formats. Our methodology exploits the common characteristics of legal documents for identifying blocks of legal information, i.e., *legal resources*, extracting and modeling resources in a semantically rich way and finally interlinking relevant resources in meaningful ways. To this end, we utilize Akoma Ntoso[1], a popular legal meta-schema, and customize it to accommodate legal information specific to Greek legislation. We then proceed with defining a domain-specific language (DSL) for legal documents; the DSL enables the implementation of a syntactic parser which processes a given text according to syntax rules described by the DSL. The parser is capable of recognizing the structure and metadata of le-

PCI '16, November 10 - 12, 2016, Patras, Greece

ISBN 978-1-4503-4789-1/16/11...$15.00

DOI: http://dx.doi.org/10.1145/3003733.3003801

[1]http://www.akomantoso.org/

gal documents from plain text and modeling them according to the meta-schema used for the semantic representation of legal resources.

We evaluated our methodology using a real dataset of Greek legal documents. Our findings reveal that our methodology achieves high success rate in the conversion of legacy and plain legal documents in a standard XML format, with a conversion speed/ memory consumption suitable for a production environment. Our method has been recently deployed in a real-world web platform[2], aiming to provide semantic access to Greek tax legislation, under the supervision of the General Secretariat of Public Revenue.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 provides an overview of our approach, while our implementation of document structure parser is explained in section 4. Section 5 describes our experimental results and discuss their significance. Finally, we draw our conclusions and discuss future work aspects in section 6.

2. RELATED WORK

We first present related work on meta models for legal resources, afterwards on legal document structuring techniquess and then on Greek legislation data.

2.1 Legal Resources Meta Models

A legal XML schema is required to structure documents, represent metadata and (cross-)links, and format legislative documents. Several legal document standards were introduced in the recent years. Among the more wide spread ones are MetaLex [6], NormeInRete [18], AkomaNtoso [3], Crown Legislation Markup Language (CLML) [3] and LexML Brasil [15]. A comparative analysis of the main standards for legislative documents, underlining specific strengths and weaknesses of each standard, at European and extra-European levels is presented in [17]. In this work we utilize the Akoma Ntoso schema as to model legal documents. Akoma Ntoso is an XML schema for modeling parliamentary, legislative, and judiciary documents. The Akoma Ntoso XML schema makes the structural and semantic components of legislative documents fully accessible to machine-driven processes, thereby supporting the creation of high-quality legislative information services and greatly improving efficiency and accountability in parliamentary, legislative, and judicial contexts.

2.2 Legal Document Structuring

Ongoing research in the legal informatics field is pursuing two approaches as to acquire a machine readable representation of legislation: creating a model at the same time that the text is created and creating a model after the text has been created. The former approach requires the use of specialized editing software, as presented in [19] and [1], where the author is required to provide information about the text at the moment of writing. Requirements for a Semantic Web regulation-drafting environment are presented in [23].

In this work we focus on the latter approach of creating a model after the text is created. It utilizes automated methods of translating legal documents, which concentrate on recurring patterns that appear in the text. Within this context, structure detection means to identify the different parts of the document, such as chapters, articles and paragraphs, and then add metadata to the document to mark those parts. A parser supporting the automatic mark-up of Italian legislative documents is presented in [2]. Their approach relies on expressing the input subtypes in hidden markov models whereas our work employs context-free grammar (CFG) [4] to express document structure and content. A method for document structure analysis with syntactic model and parsers for Japanese legal judgments presented in [13] is closer to ours. In contrast with [13] our method does not rely on PEG [10] rules, which cannot be ambiguous, but on CFG parser which allows for flexibility in regards to ambiguity and context-dependent grammar by means of predicates [21].

2.3 Greek legislation data

A few studies have been also published handling Greek legislation data. A system for the organization, management and retrieval of law sources by utilizing technologies of the semantic web has been firstly studied in [14] and afterwards a Content Management System (CMS) in order to manage legislation's archive and web services for searching and browsing Greek legislation has been proposed in [8]. Limitations, caveats and challenges for the automated analysis of Greek legislative texts are presented in [12]. In this work we continue our preliminary study in [14], providing a missing link for [8], taking in consideration findings of [12].

3. APPROACH OVERVIEW

In this section, we examine the requirements on the structure of Greek legal documents and present extensions in the legal meta-schema. Then, we present an overview of our DSL approach and its syntax rules, and how these rules are used for generating the parsing mechanism[5].

3.1 Data Model

Documents in the legal domain possess some noteworthy characteristics, such as being intrinsically multi-topical, relying on well defined structure, and possessing a broad and unevenly distributed coverage of legal issues [16]. Several guidelines/ principles of good legislative drafting, both at National and E.U. level (e.g., Joint Practical Guide for persons involved in the drafting of European Union legislation[6]) have established common formats[7], which most legal documents abide by. In a simplified view, a legislative legal document has the following structure:

- *Introductory part*, containing information enabling us to identify the type of document as well as most of its metadata, such as its title, number, the issuing authority, etc.

- *Text body*, which is the main part of the document. The text body may be structured differently depending on the legal document type; however it usually follows a well-defined hierarchical layout of text blocks (*legal blocks*), in

[2] http://www.publicrevenue.gr/elib/
[3] http://www.opsi.gov.uk/legislation/schema/
[4] https://en.wikipedia.org/wiki/Context-free_grammar
[5] Supplementary material is provided at www.dbnet.ntua.gr/ mkoniari/LawParser
[6] http://eur-lex.europa.eu/content/pdf/techleg/joint-practical-guide-2013-en.pdf
[7] In this work the term "format" is used as to include topics such as structure, numbering, preferred word use and grammar and template sentences.

which high level parts of a document, such as the chapter of a law, contains other blocks, such as articles and paragraphs.

- *End part.* It contains closing formulas, the date, and the signatures.

Figure 1 provides a visual aid of the aforementioned structure for a law, where we manually annotated structural parts and metadata values.

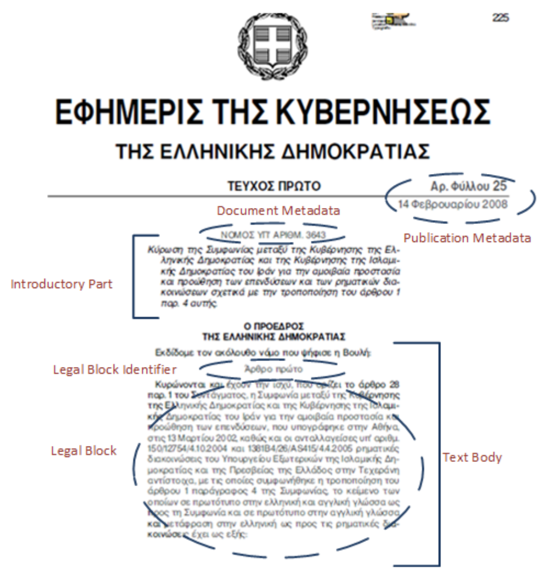


Figure 1: Overview of the structure of a legal document, with manually annotated structural parts

As stated in Section 2 we model legal resources following Akoma Ntoso meta schema. which allows structural and semantic markup of legislative documents in means of a meta XML schema. To comply with Greek legal documents i.e. laws, presidential decrees and regulatory acts of the Council of minister, we analyzed Greek legal documents and created mapping rules between all the legal blocks (e.g., title, chapters, paragraphs) and available metadata types and their corresponding elements in Akoma Ntoso schema. We also, appropriately extended the schema to accommodate custom Greek metadata based on the Dublin Core vocabulary.

Overall, in accordance with the five levels of compliance to the Akoma Ntoso schema, our output documents conform to level 3. That is we follow the document structure defined in the Akoma Ntoso specification (e.g., preface, preamble, body, conclusion, annexes) for the entire document; the naming convention of URI/IRI (FRBR metadata) and IDs defined in the Akoma Ntoso Naming Convention and the basic metadata FRBR, publication, normative reference.

Furthermore, in alignment with the ELI[8] our approach offers the minimum set of metadata required by the ELI standard and assigns a URI at each different legal block modeled in Akoma Ntoso. For example the '.../ gr/ act/ 2008/ 3643/ main/ art/ 1/' URI identifies article 1 of the main part of the act with no 3643, published in 2008 by the Greek Parliament, as shown in Fig. 1. In this way the mark-up of each structural unit of the document complies with the ELI standard, as to facilitate the precise linkage of legal citations for each respective structural unit.

3.2 A DSL Languange for legal documents

Domain-specific modeling [9] is a software engineering methodology for designing and developing systems directly from the domain-specific models, offering tailor-made solutions to problems in a particular domain. The structure of a legal document as aforementioned described can be expressed by a domain-specific model and thus extracted by employing syntax rules (DSL) [20] [11]. Thus, document structure analysis methods provide us with the proper framework in order to describe syntactic structure of documents with an abstract document model. When combined with a method to implement a document structure parser by a combination of syntactic parsers, it equips us with a parser that has high generality and extensibility.

A context-free grammar (GFG), described in Extended Backus-Naur Form [9], G is a tuple $G = (V, T, P, S)$ where

- V is the (finite) set of variables (or nonterminals or syntactic categories). Each variable represents a language, i.e., a set of strings

- T is a finite set of terminals, i.e., the symbols that form the strings of the language being defined

- P is a set of production rules that represent the recursive definition of the language.

- S is the start symbol that represents the language being defined. Other variables represent auxiliary classes of strings that are used to help define the language of the start symbol.

Thus, a syntax rule for the top level document structure of the legal document presented in Figure 1 can be described as shown in Figure 2, where nonterminals such as body and conclusions are defined separately. Preface and preamble compose the introductory part and body the text part.

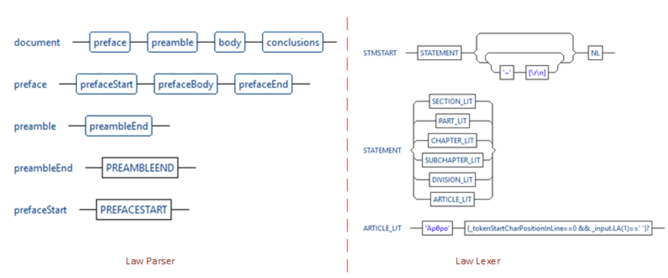


Figure 2: Overview of Document Structure Parser

Figure 2 provides a segmented railway overview of parser rules (left pane) and lexer rules (right pane) for laws. As a rule of thumb legal block definition is kept in the parser and string declaration in the lexer.

Combining different syntax rules together enables to identify the type of the document and the different text block

[8]EU proposed standard for a European Legislation Identifier (ELI) that provides, among others, a solution to uniquely identify and access national and European legislation online

[9]EBNF is a formal way of representing context free grammars

elements it contains. For the identification of the syntax rules, we heavily rely on domain knowledge from the legal experts who provide with feedback on the structural parts and their relationships (nesting, succession, etc) within the legal documents.

Next, based on the defined GFG and the set of syntax rules defined, we employ parser generators for implementing the parsing mechanism. Among the most powerful and popular is ANTLR [10]. ANTLR accepts as input any context-free grammar that does not contain indirect or hidden left-recursion and generates a lexer and a recursive-descent parser that uses an ALL(*) production prediction function [21]. Experiments presented in [21] have shown that ALL(*) outperforms general (Java) parsers by orders of magnitude, exhibiting linear time and space behavior for various languages.

The lexer, parser, and tree walker are generated by ANTLR from the corresponding ANTLR grammars. In this way our method has high generality and extensibility. Among others the advantages of our methodology are:

- we use a powerful abstraction that seperates programming from legal domain knowledge
- we can easily extend our grammars to provide for more legal documents eg. judgements
- it is easier to maintain/ evolve the procedure
- our layered implementation allows to easily adopt to new schemas/ standards.

4. PARSING PROCESS

In this section we present details of our parsing mechanism. Our goal is to provide automatic structuring and semantic indexing of legal resources. The main steps of our approach are:

- identify the structure of the legal documents
- identify legal documents metadata
- validate produced files against the selected schema

As a pre-processing step, in our method, all legal documents are converted into plain text files. Notably, this step does have the downside of discarding valuable style and layout information found in the presentation document format (pdf/ word). Several open source content analysis frameworks can efficiently and effectively perform this task e.g. Apache Tika. Also, scanned text and stored as an image in the supplied document can be adequately handled by an OCR process.

Figure 3 provides a visual overview of our parsing process. In technical terms, firstly, the input document is transformed by the lexer into a token stream. In the stream, tokens are associated with a regular expression that define the set of possible character sequences that are used to form the token. Afterwards, the parser, parses syntactic structure of the token stream creating an Abstract Syntax Tree (AST). AST is a tree data structure representing the abstract syntactic structure of the input stream, which is processed by a

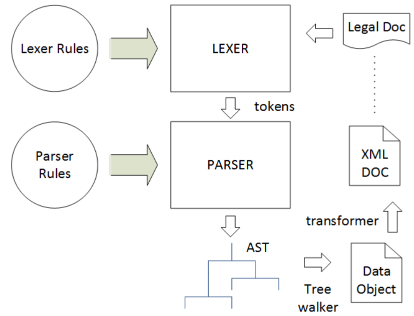


Figure 3: Overview of Document Structure Parser

tree walker that generates the in memory document model. Afterwards a transformer formats the in memory document model to the chosen legal schema (Akoma Ntoso). Finally, upon successful completion of the semantic checker/ validator, the output document is serialized.

Our parsing methodology follows a pipeline strategy, utilizing a top-down approach, that can be summarized into the following 5 steps:

1. Document Type Identification. Identification of the type of legal document is a fundamental task to effectively model legal documents. The type of the document defines both the document structure, available metadata and the internal semantic organization of the resulting document. Thus a first step in our methodology is to properly identify the type of document. This is achieved by scanning the beginning of the text for several predefined keywords that effectively distinguish document types. We do identify and process all legal documents published by the Government Gazzette i.e., laws, presidential decrees and regulatory acts of the Council of minister.

2. Structural Analysis. Following document type identification, legal blocks (e.g. front matter, body, conclusions, annexes) in the document are distinguished. Structural Analysis differs between various document types as predefined in the corresponding grammar and implemented in the appropriate parser.

3. Legal Blocks Isolation. Each legal block identified in the previous step, structural analysis step, is iteratively broken down, according to the corresponding grammar, into distinct elements. In this step we try to identify and describe the structure of the legal documents – this includes identification and mark-up of each structural unit of the document (title, chapters, sections, articles, paragraphs, annexes, etc.), so that they can be later precisely referred by linking tools where there is a legal citation quoting the respective structural unit. Document metadata values are also identified in this step. By design we assume that no metadata is available to the parser and thus our parser follows an eager approach as to identify as many metadata values as possible.

4. Legal Modeling. In this step, an in memory model of the document is iteratively constructed as new elements are identified in the text source. Also, this step assigns permanent URI to legal resources based on the technical specifications of the chosen schema.

[10] ANother Tool for Language Recognition, a language framework for constructing recognizers, interpreters, compilers and translators from grammatical descriptions containing actions in a variety of target languages.

Table 1: Accuracy analysis of structuring legal documents

Method	Precision	Recall	F-measure
Structural Analysis	1.0	0.923	0.960
Legal Blocks Isolation	0.987	0.96	0.973

5. Semantic check and validation. As a final step semantic check and validation detects any inconsistencies the text may contain from the legal point of view or any discrepancies with the chosen legal meta model.

5. EXPERIMENTAL SETUP

This section presents the evaluation of our approach. Our experimental studies were performed in a two-fold strategy: a) a qualitative/ quantitative analysis for evaluating the accuracy of our method with respect to the optimal set and b) scalability analysis for evaluating the performance behaviour when increasing the input parameters. We firstly describe the experimental dataset and then proceed with the experimental results.

5.1 Legal Corpus

General Secretariat of Public Revenue provided a corpus of more than 600 legal documents, such as laws, presidential decrees and regulatory acts, in pdf format, on which iterative tests have been carried out, aiming at modeling and refining our method. For the purpose of this study we have randomly selected 20 laws. These documents were manually structured and modeled according to the Akoma Ntoso schema by two domain experts, as to produce the optimal set. We used Apache Tika[11] as to extract textual information from the pdf files. Finally, a text pre-processing step involved removal of purely presentation information from the text files, such as page numbers. The refined text files were provided as input to our method, which produced Akoma Ntoso compliant XML files.

5.2 Accuracy analysis

In a relaxed form, parser evaluation [7] is based on computing recall and precision measures over grammatical relations. Thus, we evaluated the performance of our parser using the following established metrics: *Precision* measures the percentage of dependencies with a specific type in the parser output that were correct; *Recall* measures the percentage of dependencies with a specific type in the test set that were correctly parsed; and *F-measure* is the harmonic mean of precision and recall.

Our goal was to evaluate the effectiveness (i.e., accuracy) in extracting and modeling legal resources out of document files. We do note that based on the complexity of the input, a precise statistical analysis about the accuracy on the whole data-set (structural analysis, metadata detection, uri assignment, semantic modeling) is a demanding task, requiring to manually annotate all the metadata found in the legal documents. Thus, we compared the human annotated files with the parser output for the second and third step of parsing methodology (Section 4) considering only structural analysis identification. Results are reported in Table 1.

Our parser is very effective in identifying and structuring legal documents. In terms of Structural Analysis (identification of front matter, body, conclusions, annexes) errors that occurred during the evaluation involved the identification of annexes. In the Legal Blocks Isolation part most common parse were errors related to the identification of tables and mismatches in amending documents (misquoted parts of documents). Another, rather rare, type of errors that we identified was the proper paragraph title identification, as it is not always possible to determine if the text beginning in the first line of a section is a title or part of a paragraph.

[11] https://tika.apache.org/

5.3 Scalability analysis

We evaluated the performance of our mechanism with respect to the size (KB) and the complexity (number and variety of legal blocks) of the input legal documents. The size of our dataset varies from a few KB for the smallest files up to $\sim 900KB$ for the largest one; also the complexity varies from ~ 40 up to 600 legal blocks.

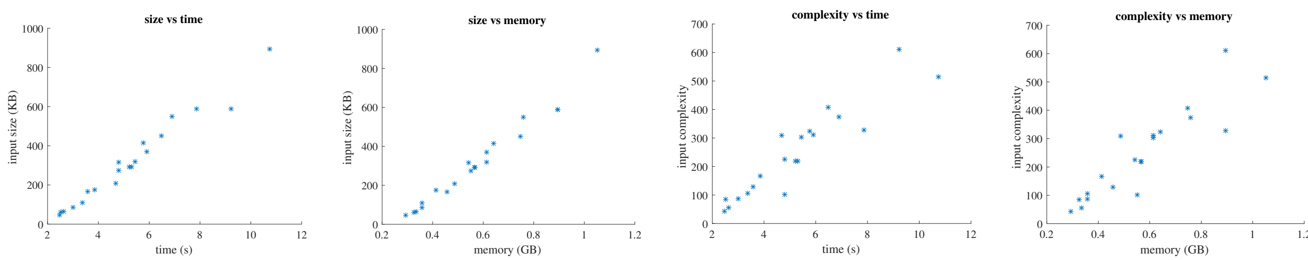
We measured the performance in terms of time needed for producing the output, and memory allocation. Reported time sums up the time of each individual step i.e., reading the input, parsing the text and serializing the output. Reported memory is the total memory allocated for the executable and not directly allocated for the core parsing module.

Fig. 4 (a) - (d) summarizes the performance results. Fig. 4 (a) and (b) shows the performance of our method as the file size increases, whereas (c),(d) presents the performance for increasing file complexity. Most files, pertaining to small sizes and complexity, are processed within a few seconds allocating less than 0.4 GB of memory. Large files are also efficiently processed within affordable time and memory requirements. Note that the rightmost outlier in Fig. 4 (c),(d) is due to the complex nesting structure of the input legal document, that forces the parser to assess a large number of syntax rules. From these results, we can infer that our parsing strategy follows a linear pattern in terms of time/ memory consumption and input size/ legal complexity.

6. CONCLUSIONS

In this paper, we presented a method for extracting semantic representations of legal documents from unstructured formats. To this end, we extended Akoma Ntoso, a popular legal meta schema, for modeling the structure and metadata of legal documents. We utilized ELI specifications, as to assign unique identifications, in the form of URIs, to legal blocks within a document thus, realising the availability of legal resources on the web. Our method expresses legal documents structure in the form of a set of syntactic rules, i.e., a domain-specific language for legal documents, used for generating a syntactic document parser. We evaluated our approach on a set of Greek legal documents. Results show that our mechanism extracts the document structure and metadata with high accuracy with linear-time performance (speed/memory).

A challenge we faced in this work was the detection of annexes and tables, which can not be recognized correctly by our current method, since they do not follow a common identifiable pattern. We plan to further improve the accuracy of document structure extraction, by utilizing machine learning methods. However, further standardization of legal documents structure and drafting procedures can greatly improve the parsing results.



(a) size vs time (b) size vs memory (c) complexity vs time (d) complexity vs memory

Figure 4: Plots of size and structural complexity of the input legal documents vs time used for parsing and memory requirements. We notice a linear pattern in terms of time/ memory consumption and input size/ legal complexity.

Several extensions of this research are currently under implementation or investigation. These include: the application of natural language processing for the identification of named entities i.e., people, places, organizations, events; the automatic soft encoding of legislative sources e.g., generation of amending documents based on the editing of an existing law or proposal; the temporal management of legal resources e.g., the time validity of a legal block; the semantic representation of amendments and types of amendments within a document and finally the automatic consolidation of (proposed) legislation based on the original and its amending documents.

7. REFERENCES

[1] T. Agnoloni, E. Francesconi, and P. Spinosa. xmlegeseditor: an opensource visual xml editor for supporting legal national standards. In *Proceedings of the V legislative XML workshop*, pages 239–251, 2007.

[2] L. Bacci, P. Spinosa, C. Marchetti, R. Battistoni, I. Florence, I. Senate, and I. Rome. Automatic mark-up of legislative documents and its application to parallel text generation. In *Proceedings of LOAIT Workshop*, pages 45–54, 2009.

[3] G. Barabucci, L. Cervone, M. Palmirani, S. Peroni, and F. Vitali. Multi-layer markup and ontological structures in akoma ntoso. In *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue*, pages 133–149. Springer, 2010.

[4] V. R. Benjamins, P. Casanovas, J. Breuker, and A. Gangemi. Law and the semantic web, an introduction. In *Law and the Semantic Web*, pages 1–17. Springer, 2005.

[5] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[6] A. Boer, R. Winkels, and F. Vitali. Metalex xml and the legal knowledge interchange format. In *Computable models of the law*, pages 21–41. Springer, 2008.

[7] J. Carroll, T. Briscoe, and A. Sanfilippo. Parser evaluation: a survey and a new proposal. In *Proceedings of LREC '98*, pages 447–454, 1998.

[8] I. Chalkidis. Nomothesi@: Greek Legislation Platform. Bachelor thesis, University of Athens, 2014.

[9] E. Evans. *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley, 2004.

[10] B. Ford. Parsing expression grammars: a recognition-based syntactic foundation. In *ACM SIGPLAN Notices*, volume 39, pages 111–122, 2004.

[11] M. Fowler. *Domain-specific languages*. Pearson Education, 2010.

[12] J. Garofalakis, K. Plessas, and A. Plessas. Automated analysis of greek legislative texts for version control: limitations, caveats and challenges. In *Proceedings of the 19th Panhellenic Conference on Informatics*, pages 115–116, 2015.

[13] H. Igari, A. Shimazu, and K. Ochimizu. Document structure analysis with syntactic model and parsers: Application to legal judgments. In *JSAI International Symposium on A.I.*, pages 126–140, 2011.

[14] M. Koniaris. Organization, management and retrieval of law sources by utilizing technologies of the semantic web. Master's thesis, Hellenic Open University, 2012.

[15] C. F. Lima JAO. LexML Brasil, Parte 3 - LexML XML Schema, version 1.0. Technical report, 2008.

[16] Q. Lu, J. G. Conrad, K. Al-Kofahi, and W. Keenan. Legal document clustering with built-in topic segmentation. In *Proceedings of CIKM '11*, pages 383–392, 2011.

[17] C. Lupo, F. Vitali, E. Francesconi, M. Palmirani, R. Winkels, E. de Maat, A. Boer, and P. Mascellani. ESTRELLA Project, Deliverable D3.1 - General XML format(s) for legal Sources, version 1.0. Technical report, 2007.

[18] A. Marchetti, F. Megale, E. Seta, and F. Vitali. Using xml as a means to access legislative documents: Italian and foreign experiences. *ACM SIGAPP Applied Computing Review*, 10(1):54–62, 2002.

[19] M. Palmirani and R. Brighi. An xml editor for legal information management. In *International Conference on Electronic Government*, pages 421–429, 2003.

[20] T. Parr. *Language implementation patterns: create your own domain-specific and general programming languages*. Pragmatic Bookshelf, 2009.

[21] T. Parr, S. Harwell, and K. Fisher. Adaptive ll (*) parsing: the power of dynamic analysis. In *ACM SIGPLAN Notices*, volume 49, pages 579–598, 2014.

[22] I.-P. Union. World e-Parliament Report 2010. 2010. http://www.ictparliament.org/wepr2010.html.

[23] S. Van De Ven, R. Hoekstra, R. Winkels, E. de Maat, and Á. Kollár. Metavex: Regulation drafting meets the semantic web. In *Computable Models of the Law*, pages 42–55. Springer, 2008.