NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

GRADUATE THESIS

Nomothesi@: Greek Legislation Platform

Ilias I. Chalkidis

Supervisors: Professor, Manolis Koubarakis
PhD Candidate, Charalampos Nikolaou

ATHENS
June 18, 2014

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Νομοθεσί@: Πλατφόρμα για την Ελληνική νομοθεσία

Ηλίας Ι. Χαλκίδης

Επιβλέποντες: Καθηγητής, Μανώλης Κουμπαράκης
Διδακτορικός Φοιτητής, Χαράλαμπος Νικολάου

ΑΘΗΝΑ
June 18, 2014

GRADUATE THESIS

Nomothesi@: Greek Legislation Platform

Ilias I. Chalkidis

Supervisors: Professor, Manolis Koubarakis
PhD Candidate, Charalampos Nikolaou

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Νομοθεσί@: Πλατφόρμα για την Ελληνική νομοθεσία

Ηλίας Ι. Χαλκίδης

Επιβλέποντες: Καθηγητής, Μανώλης Κουμπαράκης
Διδακτορικός Φοιτητής, Χαράλαμπος Νικολάου

Abstract

Nowadays, unlike many EU countries, there is not a machine-readable standard for Greek legislation, therefore there can be no advanced legal web services in Greece. Our objective is to contribute in legal knowledge's representation and its integration in the area of Open Data in Greece, both from a technological perspective and in terms of transparency. This paper introduces Nomothesi@, a platform to provide access to Greek legislation, by means of a legal XML syntax and linked data. Nomethesi@ proposes a XML standard for Greek legal documents and adopts CEN Metalex OWL ontology's core, alongside with other RDF metadata, to describe the legislative relationships and events. Linking data is about inter-linking and publishing openly Greek public data and legislative data across EU in order to enhance E-Government. On these fundamentals, we build a Content Management System (CMS) in order to manage legislation's archive and web services for searching and browsing legislation.

SUBJECT AREA: Semantic Web AI, Software Engineering
KEYWORDS: Legislative Knowledge Representation, Open Data, E-Government, RDF/OWL Metadata, XML, Content Management System, Universal Resource Identifiers

Περίληψη

Στις μέρες μας, σε αντίθεση με πολλές χώρες της Ευρωπαϊκής Ένωσης, δεν υπάρχει κάποιο μηχανικά αναγνώσιμο πρότυπο για την ελληνική νομοθεσία και επομένως δεν μπορούν να προκύψουν προηγμένες διαδικτυακές νομικές υπηρεσίες. Στόχος μας είναι να συνισφέρουμε στον τομέα της αναπαράστασης νομικής γνώσης και στην ενσωμάτωση αυτής στην περιοχή των ανοιχτών δεδομένων στην Ελλάδα, τόσο από τεχνολογική σκοπιά όσο και από άποψη διαφάνειας. Η συγκεκριμένη εργασία παρουσιάζει την Νομοθεσί@, μια πλατφόρμα, που σκοπό έχει να δώσει πρόβαση στην ελληνική νομοθεσία, με την χρήση ενός νομικού XML προτύπου και τα διασυνδεδεμένα δεδομένα (linked data). Η Νομοθεσί@ προτείνει ένα νεό XML πρότυπο για την αναπαράσταση τον νομοθετικών εγγράφων και ταυτόχρονα υιοθετεί την OWL οντολογία CEN Metalex, παράλληλα με άλλες, με σκοπό να περιγράψει τις νομοθετικές σχέσεις και γεγονότα. Η σύνδεση δεδομένων αφορά την διασύνδεση και ανοιχτή δημοσίευση ελληνικών δημόσιων δεδομένων και των νομοθετικών δεδομένων κατά μήκος της Ευρωπαϊκή Ένωσης, με σκοπό την ενίσχυση της ηλεκτρονικής διακυβέρνησης. Πάνω σε αυτές τις αρχές, χτίζουμε ένα Σύστημα Διαχείρισης Περιεχομένου (ΣΔΠ) για την διαχείρηση του νομοθετικού αρχείου και διαδικτυακές υπηρεσίες με σκοπό την αναζήτηση και παρουσίαση της ελληνικής νομοθεσίας.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Σημασιολογικός Ιστός, Τεχνητή Νουμόσυνη, Τεχνολογία Λογισμικού
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αναπαράσταση Γνώσης νομοθεσίας, Ανοιχτά Δεδομένα, Ηλεκτρονική Διακυβέρνηση, RDF/OWL Μεταδεδομένα, XML, Σύστημα Διαχείρισης Περιεχομένου, Καθολικά Αναγνωριστικά Πηγής

Acknowledgements

First of all I would like to acknowledge my supervisor Prof. Manolis Koubarakis. His courses Artificial Intelligence I and Artificial Intelligence II had directly contributed in discovering my personal interest in AI and more specifically assisted me in selecting this topic. I would also like to acknowledge PhD candidate Charalampos Nikolaou, for his valuable assistance and guidance as a second supervisor in the whole process of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 State of the art on legislative knowledge representation

In the last decade, many countries have directly provided to the public web access to legislation. Various legislative Content Management Systems (CMS) provide on-line access to legislation via thematic search, verbal or chronological using different types of data models. These systems also aim to provide services and data to third party developers.

Content and information of legal documents are drafted and archived on Information Systems, where these documents are accessible and retrievable. Web standards such as XML and RDF facilitate the separation between content based information and ontological information to help best presentation and utilization of existing information presented and / or drawn from documents therefor [1].

The usability and quality of these systems is determined on the basis of opportunities related to the search of legal documents and their structure (e.g. articles, paragraphs etc.) based on criteria (thematic, verbal, chronological), on the representation of chronological versions (e.g. enacted, current, 3rd revised), on the publication of metadata and generally capabilities of reprocessing by third party CS engineers. The quality of these capabilities is immediate consequence of the adoption of the new web technologies and more specifically the semantic web ones, along with the revised XML ones [1].

For the development of such legislative Open Data systems, international practice is concentrated in the following standards:

- Create an XML schema to represent the content of legal documents.
- Create an ontology mainly for the representation of legislative relations (modification, citation etc.) between the legal documents and legislative events (publication, amendment, withdrawal etc.). Crucial entity metadata, also.
- Create a permanent system of Universal Resource Identifiers (persistence URIs).

Creating an XML schema for the representation of legal documents for a country or an international organization (e.g., EU) requires both subtractive approach, to ensure the usability and scalability of the figure, and also the full cover of specificities on each legislative system (e.g. Greek Legislation). In this context, international practice has not implemented a widespread standard, despite very promising developing standards like CEN Metalex [2, 3], but mainly develops standards “from scratch” according to the specific legislation.

In contrast to the last decade, they have been presented several ontologies for legislative representation, in which the proposal of CEN Metalex, seems to be adopted more consequently, mainly regarding the core ontology. The starkest example is the United Kingdom (legislation.gov.uk API) [4].

Finally, the creation of stable Universal Resource Identifiers (URIs) concerns a standard, which is fully adopted by all projects involving legal knowledge internationally, and has generally been researched as well as given as a directive from international programs dealing with e-Government as ISA [5, 6].

All these fundamental standards are going to be unified in an EU level, with the first step towards this approach being the European Legislative Identifier (ELI) standard [7].

1.2 Objectives of the thesis

The objective of this thesis is, based on international practice, to design such a CMS for the Greek legislation and to implement it in demo version, hoping that it would set a foundation for the discussion of the representation of legal knowledge and its integration in the area of Linked Data in Greece. The project aims to enhance transparency and the right for information through the idea of Open Data. The need of transparency becomes clearer when we look at who the users of such systems are. People using public legislation web services are not just lawyers, but a much wider group of people who need to be informed, to cite, or use legislation as part of their job, e.g., Civil servants, freelancers, citizens defending their rights. They are the people who need to understand the legislation, and they traditionally used to search web services like Google. All these people can directly benefit from such systems. Last but not least, our research wants to challenge the whole legislative drafting process, in order to lead in new better services. As has rightly been said:

> “We shape our tools and then our tools shape us”, Martin McLuhan

1.3 Organization of the thesis

The organization of the thesis is separated in five chapters. Chapter 2 discuss the background of Greek legislation and helps us to understand the structure of Greek legislation. Based on this research, Chapter 3 analyzes the design of our platform, by means of the XML standard, RDF/OWL metadata and URI encoding. Thereafter, in Chapter 4 we present the architecture of our platform and its different parts. Chapter 5 is about the demonstration of this platforms follows to navigate the reader of this thesis through the upcoming web services. In summary (Chapter 6), there are possible matters that can be dealt with in future work.

Chapter 2

Background

In this chapter we discuss issues of legislation's structure, general behavior and some technical aspects. Firstly we approach in a glance the ongoing legislation publishing process. Subsequently we present the encoding of Greek legislation and we mention some crucial points. Summarizing the first section,we present some conclusions about the legislative modifications. The last section refers to XML, RDF, OWL, URIs spes and share a taste of what Linked Data is all about.

2.1 Structure of Greek Legislation

2.1.1 From Government Gazette to legislative Linked Data

According to Article 7 of Law 3861/2010 for the Diavgei@ program about transparency through mandatory disclosure of government's decisinstitutionions and acts, the Government Gazette is now available to citizens free of charge from the website of the National Printing House for reading, storing and printing. The Article 3 of the same Law refers to the publication on web, for all public documents as separate entities from individual institutions, something that currently happens through / via Diavgei@ website. Diavgei@ website publish any kind of public document, including legislation, totally inconsistently either in text or pdf format. Due to the lack of machine-readable format, there can be no advanced search functionality and any kind of advanced services.

Main goal of this thesis is to create a platform that will publish legislation, both in terms of legal documents in machine-readable format and also legislative knowledge (legislative interactions - events) as linked data, which can offer crucial semantic information that currently is "ruined", in order to deliver advanced legislative services open publicly.

There are six main types of Legislation, which are written sources of law, that we intend to initially publish:

Constitution
Constitution is the fundamental law upon which is based the entire configuration of a country's legislation concerning the rights and obligations of citizenship, organization and basic rules of the state and its institutions. The content of the constitution should be consistent by all other written sources of legislation. Constitutional revision can occur every 5 years, if and only if it has been approved by the previous government, with an increased majority by the House of Parliaments.

(Typical) Law
Law is issued by the prime legislative body of the State, which are House of Parliaments and the President of Hellenic Republic combined(Article 26 of the Constitution). According to the constitutionally prescribed legislative process, formal law is the source of legislation which, once introduced in the House of Parliaments as a future law, discussed in House of

Parliaments, put to vote and having received the required majority, is issued by the President and published in the Government Gazette.

Presidential Decree
The Presidential Decree is issued by the President as Head of State. The decrees are divided into the following three categories : normative decrees, which contain rules of law and issued after legislative authorization, the executive decrees, issued for the implementation of laws, and the regulatory decrees, issued under the powers of the President as a regulator of the regime.

Act of Ministerial Cabinet
Act of Ministerial Cabinet is a decree for executive issues, which is made and signed by all members of the ministerial cabinet, including the prime minister. It is issued by the President and published in the Government Gazette.

Ministerial Decision
It is called the act adopted by the (responsible) Minister, necessary delegated law and always within the framework of this authorization. By Ministerial Decision, are released secondary laws and so called “regulation” or are regulated administrative issues (e.g. recruitment, dismissal, promotion, transfer of public employees etc.) and so called “executive”. Both these types of Ministerial Decisions are published in the Government Gazette.

2.1.2 Encoding of legislation

An important part of this work is to understand the encoding of Greek legislation, which means knowing the components of a legal document, the connection between those components, their syntax, their value, any other crucial information. The encoding of Greek legislation follows the rules set out in “Manual directives for encoding of legislation” [8], which was adopted from the Central Committee of Encoding standards as required by Law 3133/2003. As we observed, consistent encoding is implemented after 2003 and still the whole persistency of the draft process is been challenged. The appropriate structure throughout the legal text has the following parts:

Articles
The basic divisions in the text of legal document are the articles, which are numbered with Arabic numeral (1, 2, 3 ...) or, in the case of insertion of a new article in an existing legal document, by combining Greek and Arabic numeral chapter letter (Α, Β, Γ...). Adding titles in the articles is useful because it helps in the systematic classification of the substance of the legal document. The numbering of subdivisions of the content of the article should be uniform in all the articles of the legal document.

Paragraphs
Paragraphs should be numbered with Arabic numerals (1, 2, 3 ...). If an article has a single paragraph, this single paragraph is not numbered, which means that number 1 is not placed.

Cases
The cases should be numbered with small Greek letters (α, β, γ ..), the subindents double small Greek letters ($\alpha\alpha$, $\beta\beta$ ...).

Passages

A passage is defined as the verbal period between two dots. The passages are numbered and set out in writing contiguous, ie without break new line in text.

Books - Chapters - Sections - Parts

The legal document may also be subdivided according to their size at larger units, such as books, chapters or sections, which are numbered with Greek uppercase letter characters. The “books”, “sections” and “parts” are numbered verbally and written in capital letters (e.g. BOOK ONE). The chapters are numbered with capital letters of Greek alphabet (e.g. A, B, Γ ...). In each case precedes the numbered subset of the code and follow the number (e.g. BOOK ONE, PART ONE, SECTION ONE, CHAPTER A).

From the above it can be easily concluded that the main division of legal documents is the article. Articles comprise of sections. Paragraphs may enumerate cases. Paragraphs consist of passages, namely recommendations. In some cases it is appropriate to construct larger divisions of legal text therefore used subdivisions book, section, chapter, part. This is likely to happen with legislative works such as the “Civil Law”, the “Criminal Law” etc.

Important elements of the structure of the legislation is also the following :

- Any legal document is characterized by the following: Type, Year, Number (ID), Title. The first three of them form a unique persistent key to each legal document.
- Then follows the title of the ministry/ies (e.g., MINISTRY OF HEALTH), which is/are in charge for promoting this legal document.
- Optionally below that, there are citations, numbered with Arabic numerals (1, 2, 3 ...), to provisions of prior legal documents, European directives, even thoughts - assessments.
- At the end of the legislative document are the signatures of the competent Ministers of the Greek Government and the President of Greek Republic. These are considered as the legislators.

2.1.3 Legislative Modifications

It is common international practice, the amendment of divisions of legal document by later legal documents. This change is not fundamental, but merely modifying and / or removing divisions (such as article, paragraph, sentence). In Greek legislation, there are often modifications through replacements, additions or extensions (add-on phrases) in passages and paragraphs’ cases, but generally in any kind of text’s subdivision. Therefore a legislative modification has independent significance but also expressed in a certain way (replacement or extension) on the legislative text. It is necessary to mention that some times modifications are called in the level of words and phrases. This type of lawmaking is close to “non-legitimate”[1], at least careless, and it creates some issues to the whole automatic updating process, which we want to generate.

[1]There is not any encoding protocol of how to call legislative modifications, but calling substitutions of words and phrases by avoiding to redefine complete passages, complicates the whole legislative process.

Example of Presidential Decree[1]

(Type) *(Number)*
PRESIDENTIAL DECREE **No. 54**

(Title)
Flood Protection Special Service Public Works of the valley of the river Evros and its tributaries (EYDE EVROY)

THE PRESIDENT OF THE HELLENIC REPUBLIC

Having regard to:

(Citation)

1. Paragraph 1 of Article 5 of Law 679/1977, "On increase increase staff posts for the Public Works Ministry and regulation of related matters" (A' 245) as the last two passages of this paragraph amended and supplemented by par. 1 of the Article 23 of Law 1418/1984 (A 23).

[...]

Article 1
Establishment, Title , Responsibilities

1. A Special Service entitled "Flood Protection Special Service Public Works of the valley of the river Evros and its tributaries" (EYDE EVROY) , with responsibility for all matters relating to the special and important work of flood protection of the valley of the river Evros, Erythropotamos, Arda, tributaries and Orestiadas' Regional Moat.
2. This Service shall exercise all responsibilities of the Management Service in accordance with the provisions applicable to the design and construction of Public Works.
3. Registered office of EYDE EVROY is set Soufli and its running period is specified at eight years after the entry into force of this Decree.

[...]

(Publication Date)
Athens, 24 May 2011

(Signatures of the competent Ministers)
PRESIDENT OF REPUBLIC
KAROLOS G. PAPOULIAS

DEPUTY MINISTERS

INFRASTRUCTURES, TRANSPORT AND NETWORKS
IOANNIS MAGKRIOTIS

[1]As the Greek legislation does not formally be translated, the translation of this presidential decree is unofficial for the purposes of this thesis.

2.1.4 The passage and case as the core of legislation

Studying legal documents that have been published in the Government Gazette, we noticed that most modifications are related with passages or cases (e.g. “At the end of paragraph X add passage as follows”, “The X passage of paragraph Y of Article Z is replaced as follows”). This directly affects the design of our data models. Looking the legislative document as a tree structure from top to bottom, we understand that our “children” nodes should be at the level of passages and cases as structural entities in our data models.

Example of Legislative Modifications

Article 1 of PD 54/2011 has the following structure:

Article 1
Establishment, Title , Responsibilities

1. A Special Service entitled “Flood Protection Special Service Public Works of the valley of the river Evros and its tributaries” (EYDE EVROY) , with responsibility for all matters relating to the special and important work of flood protection of the valley of the river Evros, Erythropotamos, Arda, tributaries and Orestiadas’ Regional Moat.
2. This Service shall exercise all responsibilities of the Management Service in accordance with the provisions applicable to the design and construction of Public Works.
3. Registered office of EYDE EVROY is set Soufli and its running period is specified at eight years after the entry into force of this Decree.

There are couple of legislative modifications, an addition and a replacement, in the previous article by Article 1 of P.D. 10/2012.

Article 1

1. After the end of paragraph 1 of Article 1 of Presidential Decree 54/2011 is added paragraph as follows: “The maintenance of these projects remain within the remit of the region of Eastern Macedonia and Thrace in accordance with the provisions of Law 3852/2010 (Government Gazette A 87).”
2. Paragraph 3 of Article 1 of Presidential Decree 54/2011 is replaced as follows: “Registered office of EYDE EVROY is set Alexandroupoli and its running period is specified at eight years after the entry into force of this Decree.”

Bearing in mind the legislative modification, the updated version of Article 1 of PD 54/2011 follows.

Article 1
Establishment, Title , Responsibilities

1. A Special Service entitled “Flood Protection Special Service Public Works of the valley of the river Evros and its tributaries” (EYDE EVROY) , with responsibility for all matters relating to the special and important work of flood protection of the valley of the river Evros, Erythropotamos, Arda, tributaries and Orestiadas’ Regional Moat. The maintenance of these projects remain within the remit of the region of Eastern Macedonia and Thrace in accordance with the provisions of Law 3852/2010 (Government Gazette A 87).
2. This Service shall exercise all responsibilities of the Management Service in accordance with the provisions applicable to the design and construction of Public Works.
3. Registered office of EYDE EVROY is set Alexandroupoli and its running period is specified at eight years after the entry into force of this Decree.

2.2 Technologies for data exchange and knowledge representation

2.2.1 Extensible Markup Language (XML)

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. The characters making up an XML document are divided into markup and content, which may be distinguished by the application of simple syntactic rules [9].

XML is the first appropriate machine-readable representation format for our legal documents. XML markup environment can cover Greek legislation encoding. By archiving XML -legal- documents, we have the potential to publish and also query our information.

2.2.2 Universal Resource Identifiers (URIs)

The Web is an information space. Human beings have a lot of mental machinery for manipulating, imagining, and finding their way in spaces. URIs are the points in that space. Unlike web data formats, where HTML is an important one, but not the only one, and web protocols, where HTTP has a similar status, there is only one Web naming/addressing technology: URIs.

Uniform Resource Identifiers (URIs) are short strings that identify resources in the web: documents, images, downloadable files, services, electronic mailboxes, and other resources. Such identification enables interaction with representations of the web resource over a network, typically the World Wide Web, using specific protocols. Schemes specifying a concrete syntax and associated protocols define each URI. The parts in URIs are frequently used for representing words of natural languages. This usage has many advantages: Such URIs are easier to memorize, easier to interpret, easier to transcribe, easier to create, and easier to guess [10].

In our platform every single legal document, its subparts, its versions or services are resources, which need to be adressed by a specific URIs system. URIs system must be persistent and build so as URI for any kind of resource to be highly guessable.

2.2.3 The data model RDF and the query language SPARQL

The Resource Description Framework (RDF) is a directed, labeled graph data format for representing information in the Web (web resources). As a metadata data model, the core structure is a set of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph. An RDF graph can be visualized as a node and directed-arc diagram, in which each triple is represented as a node-arc-node link (See Figure 2.1). RDF datasets are used to organize collections of RDF graphs [11].

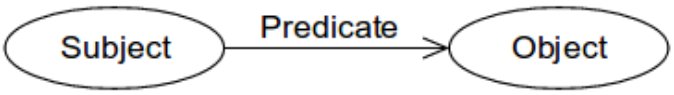

Figure 2.1: An RDF graph with two nodes (Subject and Object) and a triple connecting them (Predicate)

SPARQL is an RDF query language able to retrieve and manipulate data stored natively as RDF or viewed as RDF via middleware [12].

So we will use RDF to store metadata for Greek legislation for 2 main reasons. First of all to query them and infer the appropriate knowledge needed for advanced services. Secondly to publish our data sets and link them with third-party data sets across web.

2.2.4 The Web Ontology Language (OWL)

The Web Ontology Language (OWL) is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. The data described by an ontology in the OWL is interpreted as a set of “individuals” and a set of “property assertions” which relate these individuals to each other. An ontology consists of a set of axioms which place constraints on sets of individuals (called “classes”) and the types of relationships permitted between them. These axioms provide semantics by allowing systems to infer additional information based on the data explicitly provided. OWL has more facilities for expressing meaning and semantics than XML and RDF and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. [13].

2.2.5 Linked Data

Traditionally, data published on the Web has been made available as raw dumps in formats such as CSV or XML, or marked up as HTML tables, sacrificing much of its structure and semantics. The Semantic Web isn’t just about putting data on the web. The term “Linked Data” refers to a set of best practices for publishing and connecting structured data on the Web. These best practices have been adopted by an increasing number of data providers over the last three years, leading to the creation of a global data space containing billions of assertions — the Web of Data (See Figure 2.2). It builds upon standard Web technologies such as HTTP, RDF and URIs, but rather than using them to serve web pages for human readers, it extends them to share information in a way that can be read automatically by computers. This enables data from different sources to be connected and queried [14].

Tim Berners-Lee (2006) outlined a set of ’rules’ for publishing data on the Web in a way that all published data becomes part of a single global data space, in his Linked Data - Design Issues note [15]:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
- Include links to other URIs. so that they can discover more things.

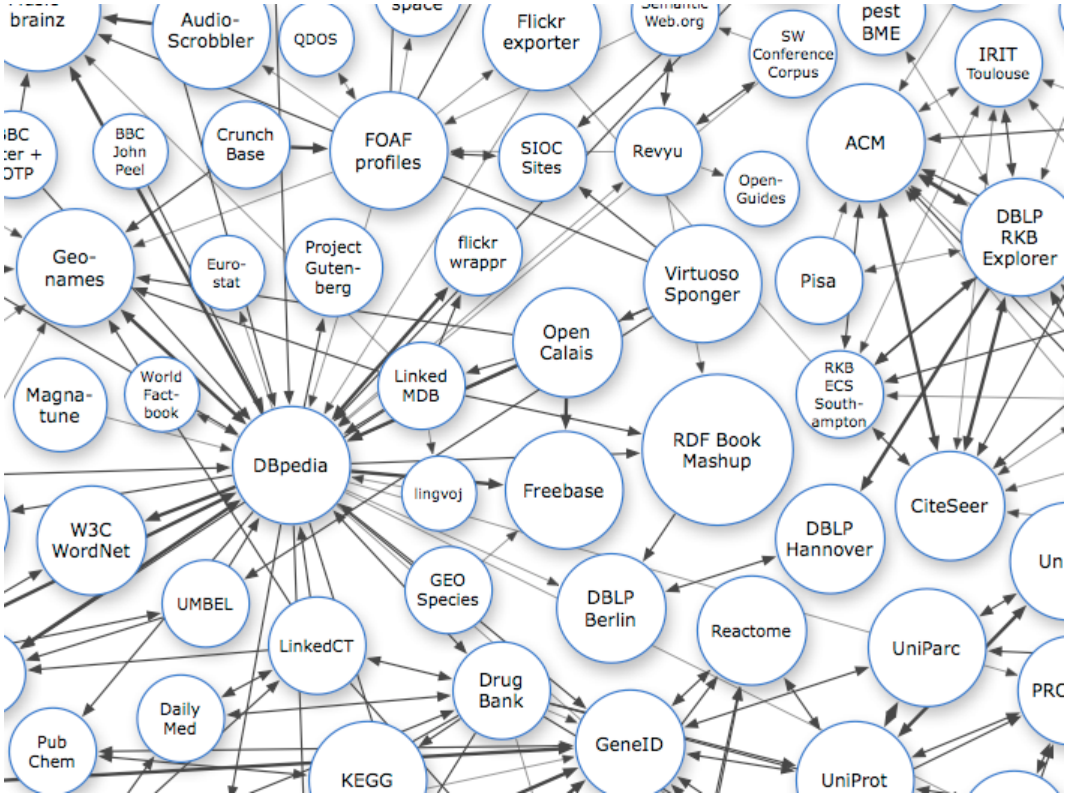

Figure 2.2: Part of the Linking Open (LOD) Data Project Cloud Diagram

The goal of the W3C Semantic Web Education and Outreach group's Linking Open Data community project is to extend the Web with a data commons by publishing various open datasets as RDF on the Web and by setting RDF links between data items from different data sources. In October 2007, datasets consisted of over two billion RDF triples, which were interlinked by over two million RDF links. By September 2011 this had grown to 31 billion RDF triples, interlinked by around 504 million RDF links. There is also an interactive visualization of the linked data sets to browse through the cloud.

2.3 Summary

After our review of the structure and the nature of the Greek legislation, with particular reference to legislative modifications, we can use this knowledge for the selection and design of modern standards and mechanisms for knowledge representation. The choices we make should cover every aspect of the legal knowledge to be functional and of course legitimate.

Chapter 3

The Design of the Platform Nomothesi@

In this chapter we present the design of the platform Nomothesi@. The design is divided in the creation of an XML standard, which follows Greek legislation encoding, adoption of the appropriate ontologies and vocabularies, in which we will built our data sets, discussion on chronological aspects of legislative knowledge and the creation of a persistent URI system.

3.1 XML Schema

The XML schema, which is essentially the structure of a new markup language, as mentioned earlier, must meets two criteria. To be as abstract and as extensible as possible and simultaneously to apply in the conditions (rules) of the Greek legislation encoding. Also we have to foresight the case of amendment and to include the appropriate elements and attributes for this requirement. In our CMS each legal document is represented by a separate XML file and without storing any updates. Updates happens mechanically instead of manually. Based on international practices and the specific criteria, mentioned above, we propose the following schema (standard), on Figure 3.1. The meaning of the XML elements used in this schema, are reported on Table 3.1.

Table 3.1: XML elements definitions

Element	Definition
Legislation	defines a legal document
Metadata	defines the metadata of the legal document *(e.g. title, signer, creation date etc.)*
Introduction	defines the introductory part of the legal document, which includes the citations
Citation	defines a citation
Body	defines the main legal text
Article	defines an article
P	defines a paragraph in an article
P2	defines a passage in a paragraph *(Same for Book, Section, Chapter, Part)*
List	defines a group of cases in a paragraph
Case	defines a case in a group of cases
Modification	defines a modification in a paragraph
Table	defines a table in an article
Title	defines the title of an XML element (e.g. Article)
Text	defines the text in a passage or a citation
Number	defines the serial number of an XML element in a group of XML elements

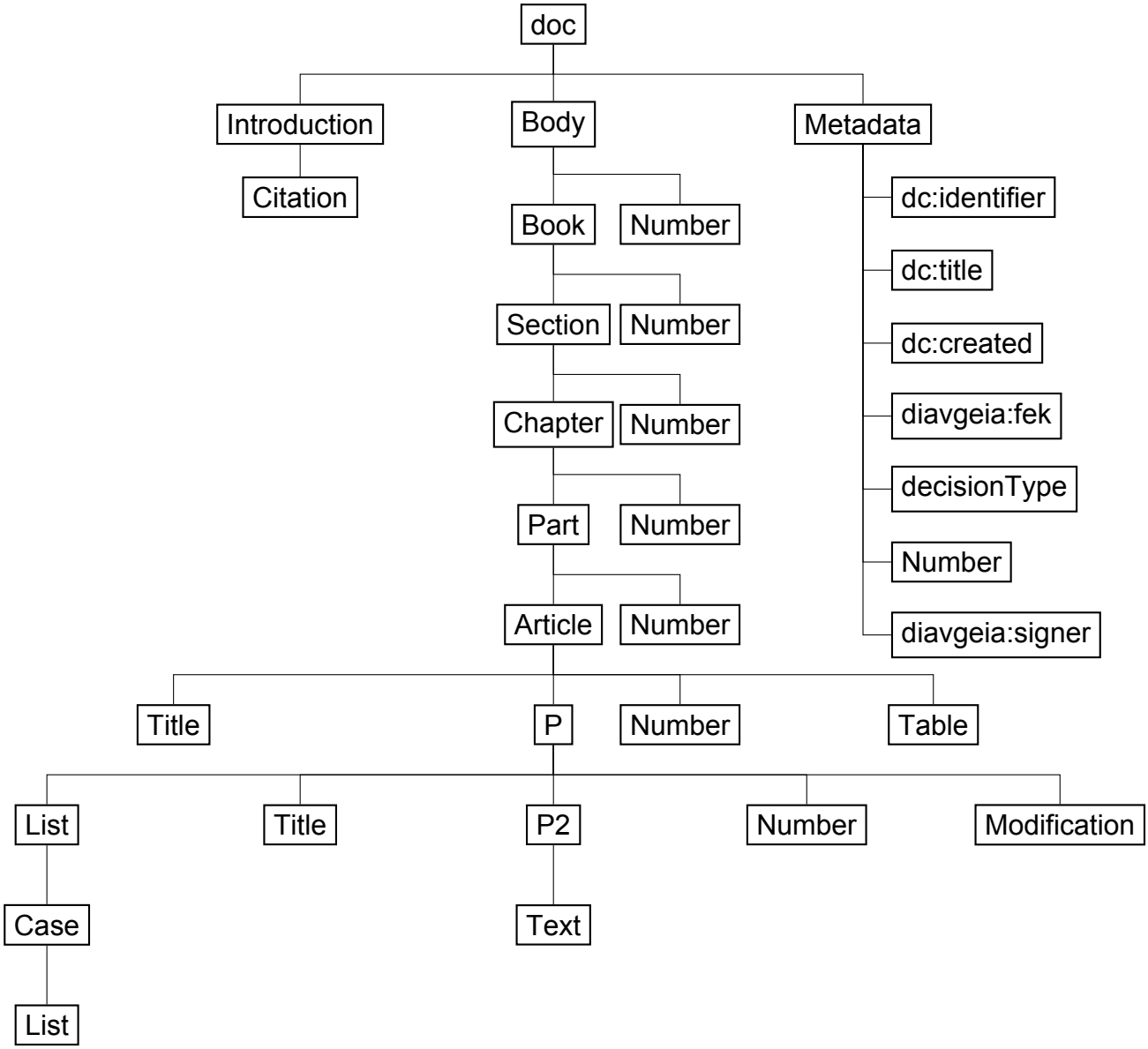

Figure 3.1: Greek Legislation XML schema

We have now a machine-readable standard for Greek legislation, which means that we are able to computerize legal documents. Based on this great first step, we still need to store legislative knowledge about the interactions between legal documents and any legislative events occurred on these legal documents.

3.2 OWL Ontology

3.2.1 Legal Documents - Versions - Files

The use of Semantic Web technologies in the context of this project focuses mainly to representing information on legislative events (publication, modification, repeal), which affect the validity, the content and general the behavior of the legal document (work) and also the interactions between legal documents arising from specific events. For this matter, we adopt the standard CEN Metalex OWL ontology [16, 17], whose core is the most suitable to represent such information. The CEN Metalex OWL ontology uses a model based on legal representation of legislative events involving the legal documents and the features that they entail. The main entities used for describing interactions in this model are:

Bibliographic Work is any legal (bibliographic) document (e.g. law, presidential decree, etc.), created and published by legislative procedures at a given time.

Bibliographic Expression is every expression (version), which is expressed by modifying the original content of a specific legal (bibliographic) work.

Bibliographic Manifestation is every representation (type) of file (e.g. PDF, XML, RDF/XML), which represents a different standard to form an legal document.

Legislative Modification defines any modification in content (e.g. article, passage, case etc.) on a legal document.

Legislative Competence Ground defines legal documents, which contains modifications referring to other legal documents.

In Figure 3.2, we describe the interactions (relationships) between these elements. For each legal document, we have the same cognitive knowledge. Which versions (BibliographicExpression) have occurred, which modifications (LegislativeModification) led on these versions, which legal document (LegislativeCompetenceGround) placed these modifications. This dataset, through queries, will indicate those individuals that are appropriate each time in order to produce suitable results.

We also need to store crucial RDF metadata for each entity. These means chronological data (dates), thematic (title, tags, organizations involved etc.) data and many other government information. For this purpose we started to use some DC Terms[1]. In the future we will enrich our metadata using e-Gorvernment Core Vocabularies[2] (Core Person, Core Business, Core Location), assisted by ISA in the effort to integrate RDF metadata across EU and combat roadblocks to metadata governance and management.

[1] See `http://dublincore.org/documents/dcmi-terms/`

[2] See `https://joinup.ec.europa.eu/community/core_vocabularies/description`

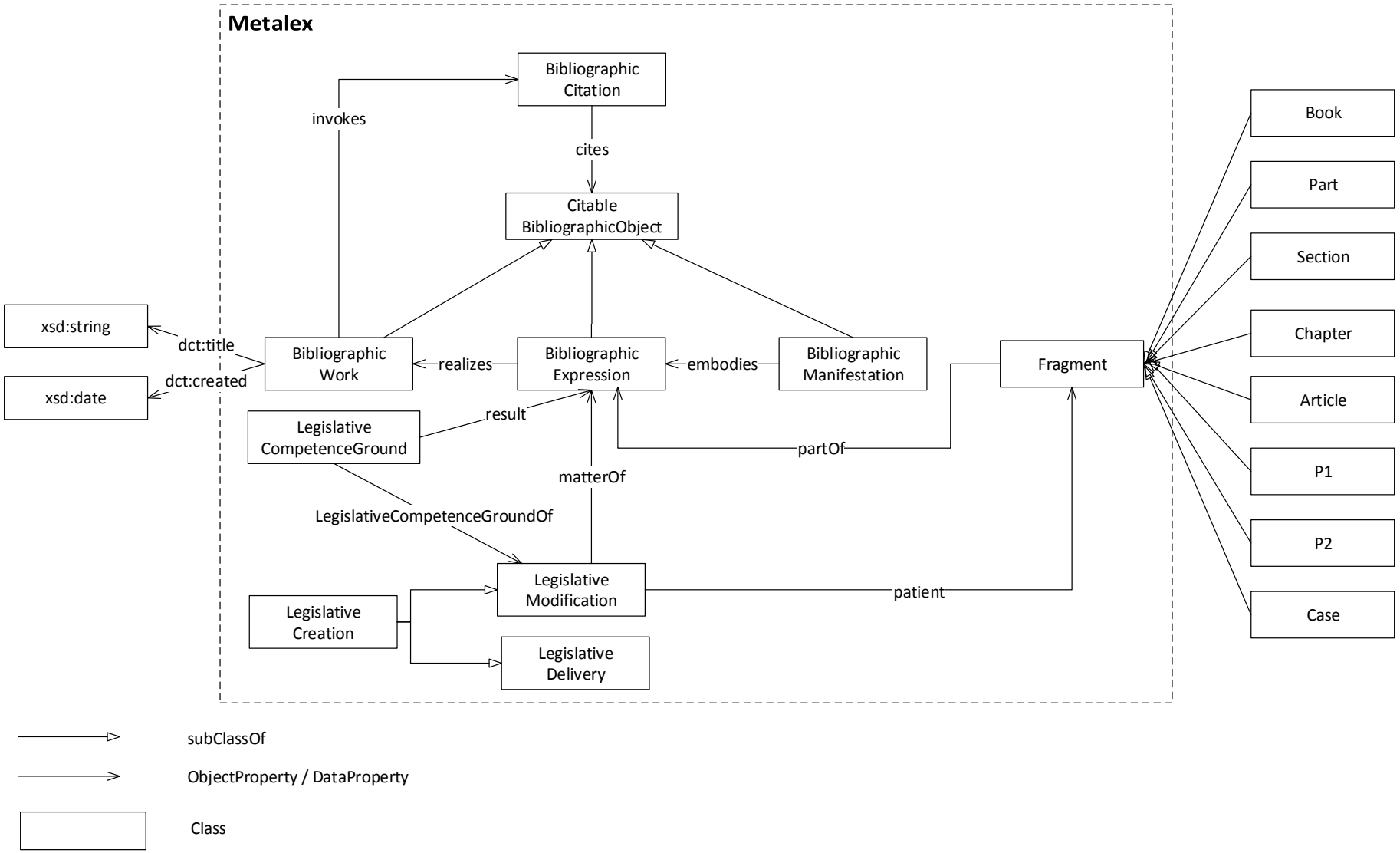

Figure 3.2: CEN Metalex OWL Ontology Core

[0]In Figure 3.2, we only the core elements of Metalex OWL ontology, which are inside the frame, and also some other very basic elements(DC Terms, add-ons).

3.2.2 Legislative timestamps

An important piece of legislation knowledge are the timestamps framing legislative events [18]. Thus we have the following timestamps for each legislative procedure:

date-publication is the time the element is officially published or announced, in the sense that the element becomes an entity at that point in time to which references can be made from outside the containing document.

date-enacted is the time the content becomes applicable in decisionmaking and is always later than or the same time as date-publication

date-modified is the time the content was modified by the content of other legal documents, which were published at a later time.

date-repealed is the time the content becomes inapplicable in decisionmaking.

The recording and appropriate representation of timestamps provide us with the ability to observe the life cycle of legal documents and infer crucial information according to the legislative events (publication, enactment,amendment, repeal) that happened in these specific times. A timeline of the life cycle of legal documents can be seen in Figure 3.3. The inferred information helps us to search and present legal documents based on chronological criteria and compute interesting statistics around the whole legislative process.

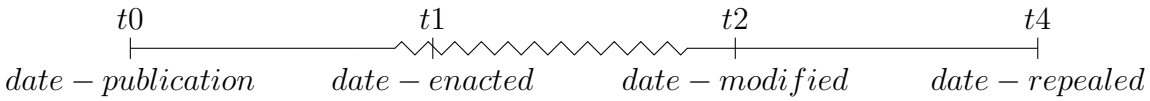

Figure 3.3: Life Cycle of legal documents

3.2.3 Time versions

We have multiple versions (BibliographicExpression) of the same legal documents (BibliographicWork). Any legal document, which contains modifications for a precedent legal document, leads to a new version of this legal document. In technical words, an XML element called Modification encapsulate a newly published regulation refers to another XML element of the same conceptual unit (e.g. article, paragraph) in a precedent legal document to repeal, enact, or change it. in Three different types of versions could be standardized:

enacted version is the version of the legal document, as it was published in the Government Gazette on date-enacted.

current version is the version of the legal document, as amended and applied, taking into account all the modifications in the legal text, which probably have occurred during the time.

date version is the version of legal document as applied on a specific date after its publication, taking into consideration all the modification, which have probably occurred until the specific date.

3.3 Persistent URIs

The Universal Resource Identifiers (URIs) provide a high level of usability as they recognize the objects in each field, applied to a structural way, and facilitate the stability and recovery of information described [6]. Fixed URIs to divisions of legislation are very important, as they are on legal documents in general. Various initiatives are trying to upgrade reliable classification for the legislation to existing bibliographic scheme. Their aim is to facilitate the process of creating URIs for legal sources, regardless of the availability of a document on the web, location of a document, and the way to access it. Initiatives such as PRESTO in the UK describe a system for legislation and public information in which *"All documents, views and metadata at all significant levels of granularity and composition should be available in the best formats practical from their own permanent hierarchical URIs."* [19]. Based on international practice and the particularities of Greek legislation, we proposed a schema of URIs, which is very similar with the UK.

The schema of URIs defines the following pattern:

`http://www.nomothesia.gov.gr/legislation/{typeoflegislation}/{year}/{id}`

Any field between curly brackets needs to take specific value. As a type of legislation, we mean all different types of Greek legislation, we are using the encoding of Table 3.2. Year is actually the year of publication (e.g., 2012) and ID is the Number (ID) of the specific legal document. So for example if we want to address in Law 12 of 2014, the corresponding URI is:

`http://www.nomothesia.gov.gr/legislation/law/20014/12`

Correspondingly, we can follow a structural search:

`http://www.nomothesia.gov.gr/legislation/search?keywords={titleoflegaldocument}&type={typeoflegislation}&year={year}&id={id}&date={YYYY-MM-DD}`[1]

Table 3.2: Encoding Types of Legislation

Type of Legislation	Code
Constitution	con
Law	law
Presidential Decree	pd
Act of Ministerial Cabinet	amc
Ministerial Decision	md

Keywords are possible words of the title, separated with comma (,). Type of legislation, year and ID follow the same rules, as we already mentioned. Date of publication must be in a specific format, which contains 4-digit year, 2-digit month and 2-digit day separated with dashes (-). Any of these criteria is optional.

[1] See `http://en.wikipedia.org/wiki/ISO_8601`

We also refer to specific conceptual unit of the legal document, by collocate the branch of the calling division after the ID. So for example if we want to address in the third paragraph of the fourth Article in Law 3/2003, the corresponding URI is:

`http://www.nomothesia.gov.gr/legislation/law/2003/3/article/4/paragraph/3`

As we mentioned already, there are three type of versions for legal documents. In order to refer on them, we follow the URI's extensions, in Table 3.3.

Table 3.3: Encoding Versions of Legislation

Version	URI extention
current	-
enacted	enacted
date-version	YYYY-MM-DD

e.g. `http://www.nomothesia.gov.gr/legislation/law/2003/3/2007-12-23`

Any legal source has multiple file formats available , both for internal use of the system and also to share open with third party CS engineers. The available file formats are: XML file (.xml), RDF file (.rdf) and PDF file (.pdf). For the disposal of these files we use the following extensions, mentioned on Table 3.4.

Table 3.4: Encoding Types of File

File - Manifestation type	URI extension
XML	data.xml
RDF	data.rdf
PDF	data.pdf

There are also specific URIs, referring to useful services for legal documents and their versions. We use the following extensions, mentioned on Table 3.5.

Table 3.5: Services' URI extensions

Services	URI extension
Table of Citations	citations
Table of Contents	contents
Timeline	timeline
Fullscreen	plain

3.4 Summary

Having regard to these key design elements: the XML schema, that we developed, the modern RDF/OWL metadata, useful chronological knowledge representantion and the persistent URIs system, we proceed to implement a first version of the platform Nomothesi@ on a technical level.

Chapter 4

The Implementation of the Platform Nomothesi@

In this chapter we analyze the architecture of Nomothesi@ platform and its components. We try to explain the functionality of Apache Cocoon framework, which is deployed as our Application server, and also the functionality of the components, in which we store and query our data.

4.1 Apache Cocoon Framework

Searching the best solution to merge our theoretical work (XML schema, OWL ontology, URIs), based on the international experience [20], we conclude we should be build our CMS on Apache Cocoon Framework[1]. *"Apache Cocoon is a Spring-based (since version 2.2 of Cocoon) framework built around the concepts of separation of concerns and component-based development. Cocoon implements these concepts around the notion of component pipelines, each component on the pipeline specializing on a particular operation. This makes it possible to use a Lego(tm)-like approach in building web solutions, hooking together components into pipelines, often without any required programming."* Cocoon relies on the pipeline model: an XML document is pushed through a pipeline, that exists in several transformation steps of your document, as we see in Figure 4.1. Every pipeline begins with a generator, continues with zero or more transformers, and ends with a serializer. The main three components could be described like this:

Generators Generator is the starting point of an XML pipeline. Usually, it generates XML content as SAX[2] events and initializes pipeline processing. XML content can be either retrieved from an XML file or produced from other data models based on request parameters.

Transformers A transformer is the central point in a sitemap pipeline. Within a pipeline match, transformers consume SAX events and emit SAX events. It can be compared to an XSL: it gets an XML document (or SAX events), and generates another XML document (or SAX events).

Serializers A Serializer is responsible for transforming SAX events to a presentation format. We have Serializers for generating HTML, XML, PDF and of course you can create your own.

[1] See `http://cocoon.apache.org`

[2] SAX (Simple API for XML) is an event-based sequential access parser API, which provides a mechanism for reading data from an XML document. SAX parsers operate on each piece of the XML document sequentially.

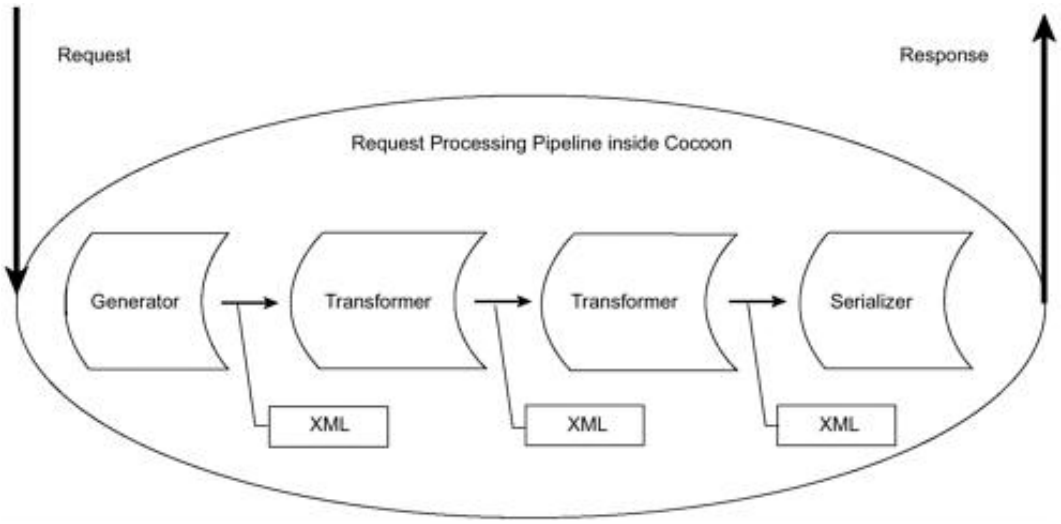

Figure 4.1: Apache Cocoon Pipeline

4.2 XML Database Server

In order to organize our XML documents for querying processes, we use a native XML DataBase, BaseX[1]. BaseX is a scalable and high-performance, yet very light-weight XML Database engine and XPath/XQuery Processor, which can run remotely as a Server in a Client/Server architecture, whose client is Cocoon. The XQuery language give us the capabilities to search based on multiple criteria and also our project is supported with full-text search engine of Apache Lucene[2], which is used by BaseX.

4.3 RDF Store Server

Our project needs a RDF Store to organize RDF triples. OpenRDF Sesame[3] is a de-facto standard framework for processing RDF data. This includes parsers, storage solutions (RDF databases a.ka. RDFStores), reasoning and querying, using the SPARQL query language, that we introduced in Section 2.2. It offers a flexible and easy to use Java API that can be connected to all leading RDF storage solutions. So like BaseX, using Sesame remotely as a Server in a Client/Server architecture, whose client through Java API is Cocoon, we can handle CEN Metalex OWL ontology data.

4.4 Architecture of Nomothesi@ platform

To recap, we have Apache Cocoon as a web application Server, BaseX as a native XML DB Server and Sesame as an RDFStore Server. So every HTTP request pass from Cocoon, then if needed we settle a connection with BaseX and/or Sesame to retrieve appropriate data and turn back to Cocoon to form the HTTP response for the user. The abstract architecture is depicted in Figure 4.2.

[1] See `http://basex.org`
[2] See `http://lucene.apache.org`
[3] See `http://www.openrdf.org`

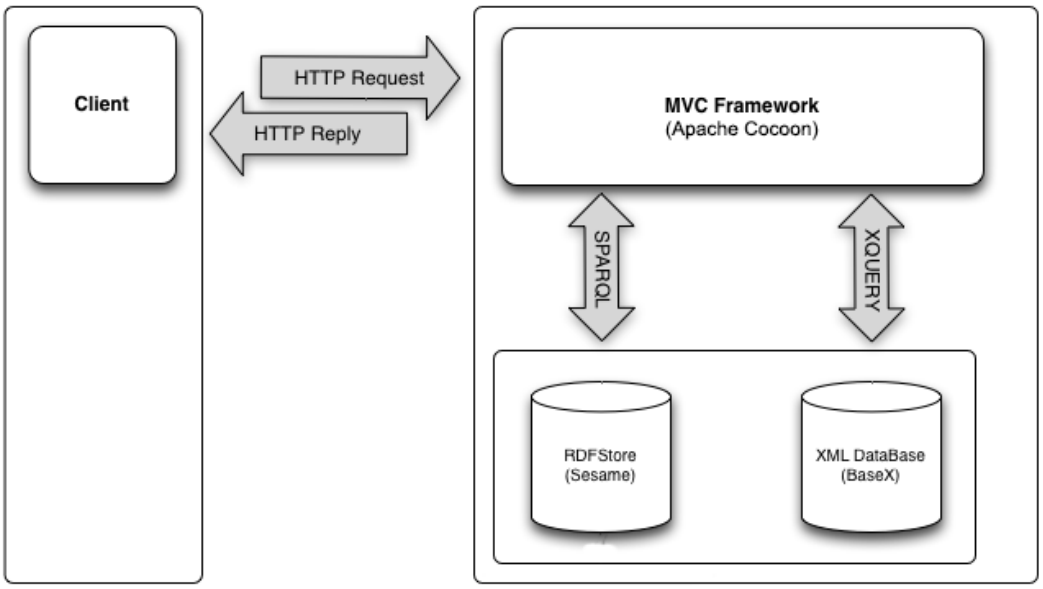

Figure 4.2: Architecture of the platform Nomothesi@

4.5 Use Cases

4.5.1 Use Case 1 - Present updated legislative version

Suppose we take the following request: `http://www.nomothesia.gov.gr/legislation/pd/2011/54`, which means to return the last updated version of Presidential Decree (PD) 2011/54. We have the following steps (Figure 4.3):

1. Cocoon match request URI with the appropriate pipeline for PDs.
2. Pipeline calls Version Generator.
3. Version Generator connects with Sesame Server and apply SPARQL query to find any modifications, which have occurred on PD 2011/54. It accepts a list of modifications with 2 main elements: modification's URI and modification's Source File.
4. Version Generator connects with BaseX Server. For each modification in the list, query the modification's Source File and get as a result modification's content (XML branch). Form an XML block of all these XML parts.
5. Pipeline calls XMLGenerator to take the content of XML file, called PD201154.xml .
6. Pipeline calls XSLT Transformer.
7. XSLT Transformer use an XSLT file to apply modifications on XML file's divisions, which have to be modified (replaced, enriched, deleted, added).
8. Pipeline calls XSLT Transformer.
9. XSLT Transformer use another XSLT file to produce HTML page from XML output of the previous XSLT file.
10. Pipeline calls XHTML Serializer to give HTML page as HTTP response.

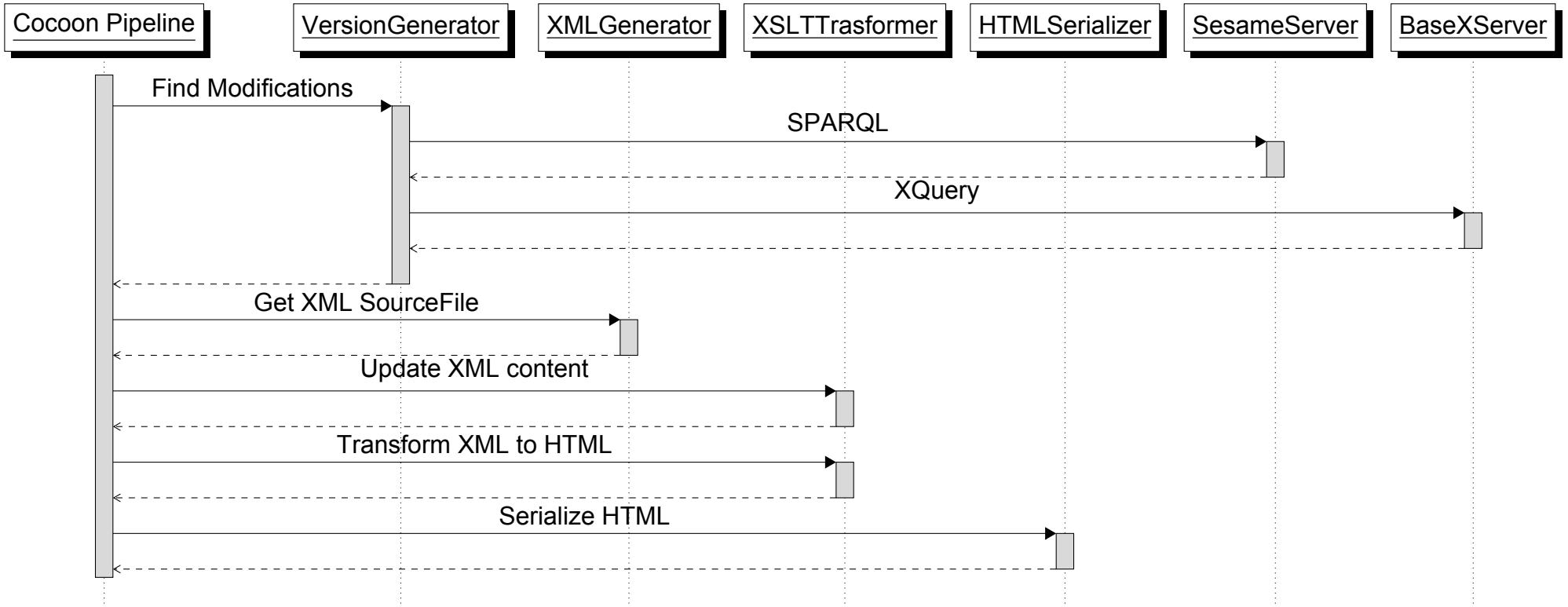

Figure 4.3: UML sequence diagram Use Case 1

4.5.2 Use Case 2 - Search legal documents, based on criteria

We fill search form to find all PDs of 2012. The request URI for this functionality is `http://www.nomothesia.gov.gr/legislation/search?type=pd&year=2013`. We have the following steps (Figure 4.4):

1. Cocoon match request URI with the appropriate pipeline for search function.
2. Pipeline calls Search Generator.
3. Search Generator connects with BaseX to query XML DB with type and year criteria. We find the suitable XML documents and set for each one a record with title, date, typ and URI characteristics. Then we form an XML block with all these records.
4. Pipeline calls XSLT Transformer.
5. XSLT Transformer produces HTML page with a table for all these records.
6. Pipeline calls XHTML serializer to give HTML page as HTTP response.

```
<map:pipeline id="search">
  <map:match pattern="search">
        <map:generate type="searchgen"/>
        <map:transform src="search.xslt"/>
        <map:serialize type="xhtml"/>
  </map:match>
</map:pipeline>
```

⇓ Call SearchGenerator ⇓

```
public class SearchGenerator extends AbstractGenerator{

    static Context context = new Context();
    Request request;

  @Override
  public void setup(SourceResolver resolver, Map objectModel, String src, Parameters params)
  throws ProcessingException, SAXException, IOException {...6 lines }

  public void generate() throws IOException, SAXException, ProcessingException {...191 lines }

  static void xquery(final String query) throws BaseXException, IOException {...3 lines }
```

⇓ Call search.xslt ⇓

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
    <head></head>
    <body>
        <div id="wrapper">
            ....................................
                <table id="results" >
                    <thead><tr><th>Τίτλος</th><th>Κωδικός</th><th>Ημερομήνια</th></tr></thead>
                    <tbody>
                    <xsl:for-each select="results/result">
                        <tr>
                            <td>
                                 <a><xsl:attribute name="href">
                                    <xsl:value-of select="id"/>
                                 </xsl:attribute>
                                 <xsl:value-of select="title"/>
                                 </a>
                            </td>
                            <td>
                                <xsl:value-of select="type"/> 
                                <xsl:value-of select="year"/>/<xsl:value-of select="number"/>
                            </td>
                            <td>
                                <xsl:value-of select="date"/>
                            </td>
                        </tr>
                    </xsl:for-each>
                    </tbody>
                    </table>
        </div>
    </body>
    </html>
  </xsl:template>
```

Figure 4.4: Search Pipeline Transactions

Chapter 5

Demonstration of the Nomothesia@ Platform

First we will see Home Page of Nomothesi@ (See Figure 5.1). Home Page provides a search form for legal documents, two tables, one for most requested acts and one for new legislation and an introduction text about our web application.

Figure 5.1: Home Page of Nomothesi@

Let's see a page of enacted version about Presidential Decree (PD) 2011/54 (See Figure 5.2). We notice two abstract divisions. The main division provides the legal text, a table of basic information and a toolbar for table of contents, timeline, citations and full-screen mode. The other division - left - is a menu, where we provide last and enacted version of this legal document and three different formats (XML,RDF,PDF).

Figure 5.2: HTML Page of Enacted version of PD 2011/54

We saw above the enacted version of PD 2011/54. We are also able to read the latest available version of PD 2011/54 (See Figure 5.3). The GUI environment is exactly the same, but we can notice that divisions of the legal document have been modified through legal modifications, as we presented in Section 2.1.

Figure 5.3: HTML Page of Latest version of PD 2011/54

As we mentioned home page includes a search form. We search for any legislation published in 2013. The search results are printed in a table with basic information, from which we can linked to resulted legal documents (See Figure 5.4).

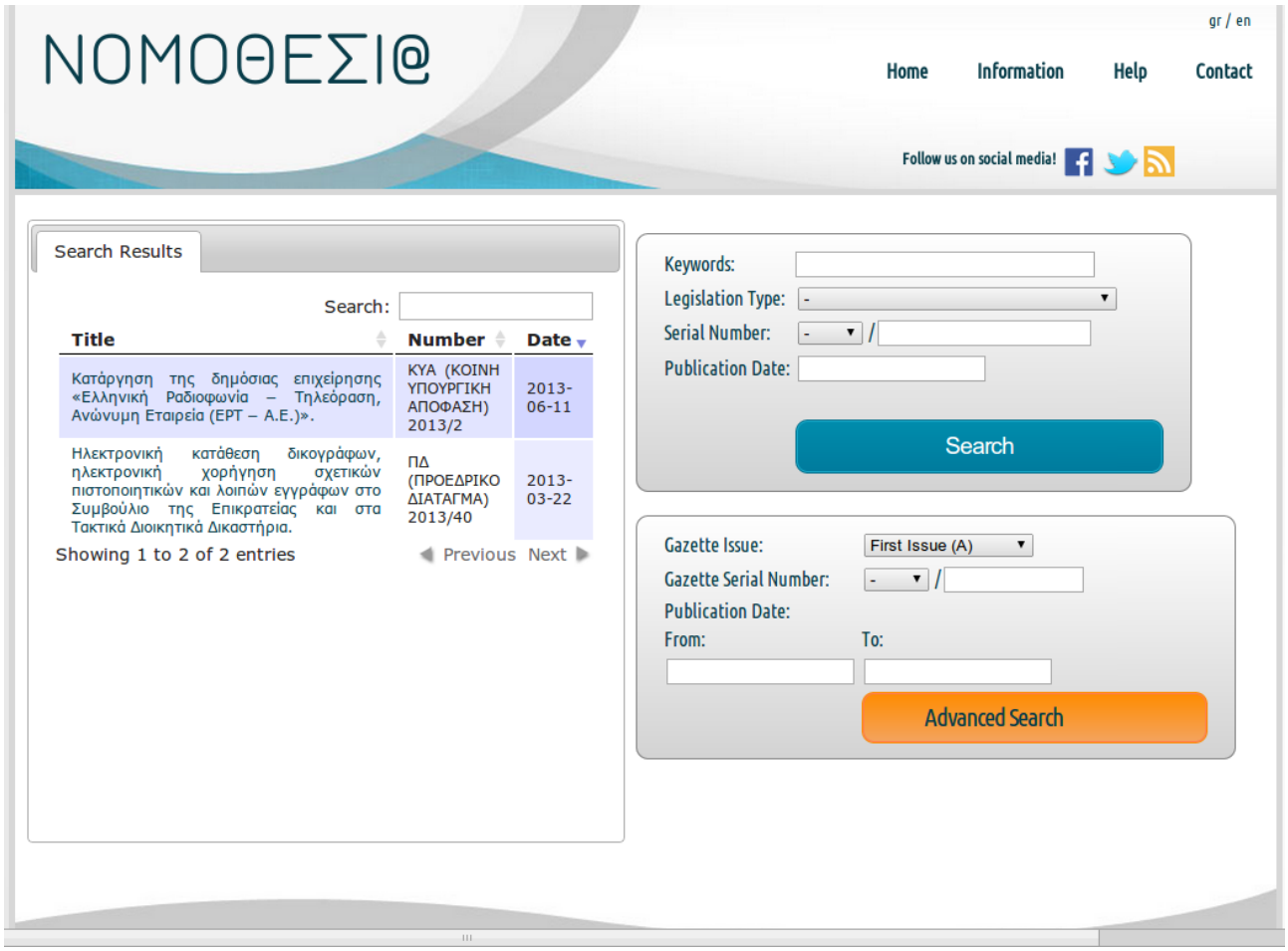

Figure 5.4: Page of Search Results

Chapter 6

Conclusions

This thesis is just the first brief step in the development of a robust platform for Greek legislation. We achieved to design the first version of very basic data models and a persistent URI system for legislative entities. We also demonstrated Nomothesi@ web services to present some obvious results and benefits arising from such a platform. Although the research presented here have demonstrated the effectiveness of such a platform for Greek Legislation, it could be further developed in a number of ways, that are described in the following.

6.1 Future Work

Legal Documents Converter

As we mentioned, Greek legal documents are stored as text files (.doc) and published as PDF files (.pdf). In order to apply our platform in the real world, we need to enrich our data with as many legal documents as possible. Of course, it cannot happen manually, so we need a tool (converter) to accomplish the hard and lengthly job. n order to transform legacy XML to Metalex and RDF, Metalex research group implemented the MetaLex converter, an open source Python script [20].

Core Vocabularies

The last three years, the Core Vocabularies Working Group of the Interoperability Solutions for European Public Administrations (ISA) Programme has the objective to promote the specification of the Core Person and develop the specification of the Core Business and Core Location vocabularies. These vocabularies will promote interoperability through different data sets, which are published by many organizations across EU. With cautious circumspection in the great issue of interoperability, we will use these vocabularies to redefine and enrich our data sets.

SPARQL Endpoint

It is in our interest to make our platform as useful as possible for third party Computer Software (CS) engineers. In this approach the implementation of a SPARQL endpoint is the next big thing. This service will give the opportunity for everyone to query our data sets to draw legislative knowledge in his own special interests. The whole architecture of our system is redefined as shown in Figure 6.1.

Greek e-Government Linked Data

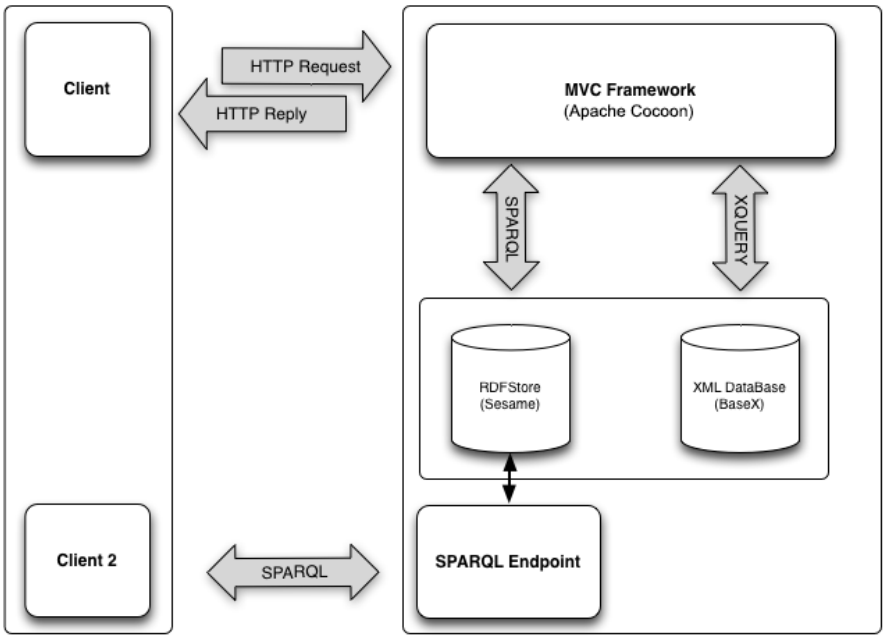

Figure 6.1: Future Architecture of Nomothesi@ platform

A key objective of our project is to link our data sets with any other e-Government data sets, such as Public Services and Organizations and Government staff (See Figure 6.2). The adoption of Core Vocabularies will accelerate this effort [21]. Linking Data has to be both in RDF/OWL metadata level and also in content's markup. Marking and linking all available knowledge in legal documents' content widen the whole concept of our services' quality and usability. This specific aspect is one of the most important that undergoes in UK legislative services and it has great interest for the working group of The National Archives (UK).

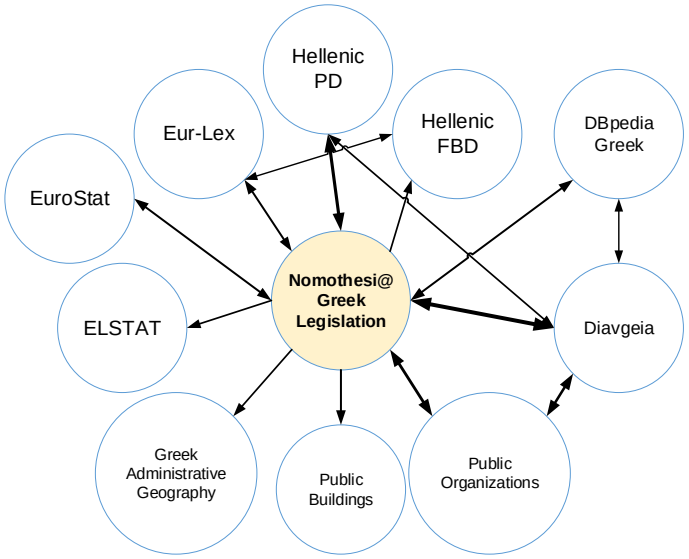

Figure 6.2: Part of Future Greek Linked Data Cloud Diagram

Local Government Decisions

In our project, we worked with the core of Greek Legislation (Laws, Presidential Decrees, Acts of Ministerial Cabinet, Legislative Acts and Ministerial Decrees). Other secondary legal documents have to be handled in the same way. Something really innovative is to include Local Government Decisions. Greek Local Government consists of 7 decentralized administrations, 13 regions and 325 municipalities. Each of them publish Local decisions in Government Gazette and there is a great interest in all these paralegal documents. Research groups in our department have published an ontology of Kallikratis plan, the administrative system of Greece [22]. We can enrich our metadata with other from Greek Administrative Geography in order to infer knowledge about Local Government Decisions.

Case Law

Case Law plays an important role in judicial activities. The term “res judicata” describes the application of legal documents in a specific trial, where similar cases in the future should be handled using the same legal documents in the same manner. It would be a great improvement for our project to cite case laws, which are linked on specific legal documents.

Legislative Statistics

Having all this legislative knowledge in our CMS and bearing in mind any further knowledge that will be gathered by web users’ requests, there is a great opportunity to start organizing statistic data for Greek legislative data. Interesting statistics could be published dynamically via our web service. How often legislators publish each type of legislation, how many modifications occur in a legal document’s lifecycle, which are the most requested legal documents by the users (public). Statistics have the answer to all these, which are very useful in assessment of the legislative process. Both the public and special legal analysts will find available these statistic data.

Appendix A

Abbreviations & Translations

Table A.1: LIST OF ABBREVIATIONS

CMS	Content Management System
XML	Extensible Markup Language
RDF	Resource Description Framework
CS	Computer Science
URI	Universal Resource Identifier
EU	European Union
ELI	European Legislative Identifier
CEN	European Committee for Standardization
OWL	Web Ontology Language
SPARQL	SPARQL Protocol and RDF Query Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
SAX	Simple API for XML

Table A.2: TABLE OF TRANSLATIONS

CMS	Συστήματα Διαχείρισης Περιεχομένου (ΣΔΠ)
URIs	Καθολικά Αναγνωριστικά Πηγής
Semantic Web	Σημασιολογικός Ιστός

Appendix B

Nomothesi@ Web Application

Nomothesi@ is implemented as an Apache Cocoon Project. Apache Cocoon is a Spring-based (since version 2.2 of Cocoon) framework built around the concepts of separation of concerns and component-based development. Cocoon implements these concepts around the notion of component pipelines, each component on the pipeline specializing on a particular operation. There are 3 fundamental components: generators, transformers and serializers.

Software / Library dependencies

Java Version: Java™ SE 7
Build Tool: Apache Maven 3.2.1
Web Server: Apache Tomcat 7 / Jetty Server 6.1.7
RDFStore: Sesame Server 2.7.9 (on Apache Tomcat 7)
XML DB Server: BaseX 7.8 (on Apache Tomcat 7)

Installation instructions

The project of Nomothesi@ platform is compressed in nomothesia-1.0.0.war file. The .war file needs to be deployed in a Web Application Server (Apache Tomcat, Jetty) and also collaborates / communicates with 2 external web applications Sesame and BaseX, which also needs to be deployed. We will give instructions of how-to complete installation. After completing installation, you can browse Nomothesi@ platform at `http://localhost:8080/nomothesia-1.0.0/GRLegislation/`.

1. Apache Tomcat (`http://tomcat.apache.org/`)

Apache Tomcat can be downloaded here (`http://tomcat.apache.org/download-70.cgi`), depends to the operating system and its distribution. You can find installation instructions here (`http://tomcat.apache.org/tomcat-7.0-doc/setup.html`). After installation, you can point your browser at this location (http://localhost:8080/) to verify that the deployment succeeded. The easiest way to deploy Nomothesi@ Platform is placing GRLegislation.war in Tomcat's folder /webbapps. You can find information of how-to deploy an application here (`http://tomcat.apache.org/tomcat-7.0-doc/deployer-howto.html`)

2. Sesame Server (`http://www.openrdf.org/`)

Sesame Server can be downloaded here (`http://sourceforge.net/projects/sesame/files/Sesame%202/2.7.11/openrdf-sesame-2.7.11-sdk.zip/download`). You can also find the .war files in /webapps folder. The .war files needs to be deployed in Apache Tomcat. The easiest way to deploy this application is to place the .war files in webapps folder of Apache Tomcat and restart it. After you have deployed the Sesame Server webapp, you should be able to access it, by default, at path /openrdf-sesame. You can point your browser at this

location to verify that the deployment succeeded. You can also manage repositories from the OpenRDF Workbench, which should be available at path /openrdf-workbench.

For the purposes of our project, you need to visit /openrdf-workbench create a new repository called “Legislation” with type Memory store / persist or Native store and then load our data set from file TEST.owl. You can complete any appropriate action from OpenRDF Workbench’s menu.

Any additional information od downloading, installing and interacting with Sesame Server can be found in the website’s user’s documentation.

3. BaseX XML DB Server (`http://basex.org/`)

BaseX XML DB Server can be downloaded here (`http://files.basex.org/releases/7.8.2/BaseX782.war`). You can also find the .war file in /webapps folder. Like Sesame, the .war file needs to be deployed in Apache Tomcat. The easiest way to deploy this application is to place the .war file in webapps folder of Apache Tomcat and restart it. After you have deployed the BaseX XML DB Server, you should be able to access it, by default, at path /BaseX782. You can point your browser at this location to verify that the deployment succeeded.

Currently our XML repository is stored in the filesystem, so there is no need for extra steps with BaseX configuration.

Any additional information for downloading, installing and interacting with BaseX XML DB Server can be found in the website’s user’s documentation. There are also some useful Java examples at `http://docs.basex.org/wiki/Java_Examples`, which really helps to understand how it works.

Configuration instructions

There is a possibility that you need to configure Nomothesi@. For this purpose there is a configuration file called properties.properties, which specify the following information:

XMLDBServer = (BaseX Server path)
XMLDBName = (BaseX DataBase name)
SesameServer = (Sesame Server path)
SesameRepositoryID = (Sesame Server repository name)
BaseURI = (BaseURI of RDF metadata)

Bibliography

[1] María Hallo Carrasco, M. Mercedes Martínez-González, and Pablo De La Fuente Redondo. Data models for version management of legislative documents. *J. Inf. Sci.*, 39(4):557–572, 2013. http://jis.sagepub.com/content/39/4/557.abstract.

[2] A. Boer, R. Hoekstra, R. Winkels, T. van Engers, and F. Willaert. ^{META}lex: Legislation in XML. In T. Bench-Capon, Aspassia Daskalopulu, and R.G.F. Winkels, editors, *Legal Knowledge and Information Systems. Jurix 2002: The Fifteenth Annual Conference*, Frontiers in Artificial Intelligence and Applications, pages 1–10, Amsterdam, 2002. IOS Press. http://www.jurix.nl/pdf/j02-01.pdf.

[3] Rinke Hoekstra. The metalex document server - legal documents as versioned linked data. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist, editors, *International Semantic Web Conference (2)*, volume 7032 of *Lecture Notes in Computer Science*, pages 128–143. Springer, 2011. http://iswc2011.semanticweb.org/fileadmin/iswc/Papers/In-Use/70320129.pdf.

[4] John Sheridan. LEGISLATION.GOV.UK, 2010. http://blog.law.cornell.edu/voxpop/2010/08/15/legislationgovuk/.

[5] PwC EU Services EESV. Case study: How Linked Data is transforming egoverment. Technical report, European Commission, 2013. http://ec.europa.eu/isa/actions/documents/isa_1.1-how-linked-data-is-transforming-egovernement_en.pdf.

[6] Stijn Goedertier Phil Archer and Nikolaos Loutas. Study on persistent URIs, with identification of best practices and recommendations on the topic for the MSs and the EC. Technical report, European Commission, 2012. https://joinup.ec.europa.eu/sites/default/files/D7.1.3%20-%20Study%20on%20persistent%20URIs.pdf.

[7] John Dann. European Legislation Identifier "ELI". Technical report, European Commission, 2014. https://joinup.ec.europa.eu/sites/default/files/ckeditor_files/files/11_SEMIC2014_John%20Dann_European%20Legislation%20Identifier%20ELI%20towards%20a%20legal%20semantic%20interoperability.pdf.

[8] Central Codification Committee of Goverment's SG. *Manual directives for encoding of legislation*, 2003. http://www.ggk.gov.gr/wp-content/uploads/2010/02/teliko_egxeiridio_odigion_gia_tin_kodikopoiisi_tis_nomothesias.pdf.

[9] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and Francois Yergeau. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. W3C Recommendation, 2008. http://www.w3.org/TR/xml/.

[10] Tim Berners-Lee. *Universal Resource Identifiers (URIs)*. W3C, 2008. http://www.w3.org/Addressing/URL/uri-spec.html.

[11] Richard Cyganiak, David Wood, and Markus Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation, 2014. http://www.w3.org/TR/rdf11-concepts/.

[12] Steve Harris and Andy Seaborne. *SPARQL 1.1 Query Language*. W3C Recommendation, 2013. http://www.w3.org/TR/sparql11-query/.

[13] W3C OWL Working Group. *OWL 2 Web Ontology Language Document Overview*. W3C Recommendation, 2012. http://www.w3.org/TR/owl2-overview/.

[14] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009. http://www.igi-global.com/article/linked-data-story-far/37496.

[15] Tim Berners-Lee. Linked Data - Design Issues. *W3C*, (09/20), 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[16] European Committee for Standardization (CEN). CEN Workshop Agreement: Metalex (Open XML Interchange Format for Legal and Legislative Resources). Technical report, European Committee for Standardization (CEN), 2006. ftp://ftp.cen.eu/CEN/Sectors/List/ICT/CWAs/CWA15710-2010-Metalex2.pdf.

[17] Alexander Boer. *Legal Theory, Sources of Law and the Semantic Web*, volume 195 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009. http://www.leibnizcenter.org/wp-content/uploads/2012/04/lt-sl-sw-aboer2009.pdf.

[18] Alexander Boer, Radboud Winkels, Tom van Engers, and Emile de Maat. Time and versions in ^{META}lex XML. In *Proceeding of the Workshop on Legislative XML*, Kobaek Strand, 2004. http://www.metadat.com/LegXML-20040922!24-Kobaek/TimemanegementMetaLex.pdf.

[19] Rich Jelliffe. PRESTO - A WWW information architecture for legislation and public information systems, 2008. http://www.oreillynet.com/xml/blog/images/PRESTO.pdf.

[20] Alexander Boer, Radboud Winkels, Tom van Engers, and Emile de Maat. A content management system based on an event-based model of version management information in legislation, 2004. http://jurix.nl/pdf/j04-04.pdf.

[21] ISA programme. eGovernment Core Vocabularies, 2012. https://joinup.ec.europa.eu/sites/default/files/a7/3d/18/ISA_eGovernment-Core-Vocabularies_February2012.pdf.

[22] Greek Linked Open Data: Greek Administrative Geography. http://www.linkedopendata.gr/dataset/greek-administrative-geography.