NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

UNDERGRADUATE THESIS

Parsing Raptarchis' legislative documents for Nomothesi@ platform

Nikolaos K. Matthioudakis

Supervisors: **Manolis Koubarakis,** Professor

Christos Papaloukas, MSc Student

ATHENS

MARCH 2020

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Συντακτική ανάλυση των νομοθετικών αρχείων του Ραπτάρχη για την πλατφόρμα Nomothesi@

Νικόλαος Κ. Ματθιουδάκης

Επιβλέποντες: **Μανόλης Κουμπαράκης,** Καθηγητής
Χρήστος Παπαλουκάς, Μεταπτυχιακός Φοιτητής

ΑΘΗΝΑ

ΜΑΡΤΙΟΣ 2020

UNDERGRADUATE THESIS

Parsing Raptarchis' legislative documents for Nomothesi@ platform

Nikolaos K. Matthioudakis
A.M: 1115201200104

Supervisors: **Manolis Koubarakis,** Professor
Christos Papaloukas, MSc Student

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Συντακτική ανάλυση των νομοθετικών αρχείων του Ραπτάρχη για την πλατφόρμα Nomothesi@

Νικόλαος Κ. Ματθιουδάκης

Α.Μ: 1115201200104

Επιβλέποντες: **Μανόλης Κουμπαράκης,** Καθηγητής

Χρήστος Παπαλουκάς, Μεταπτυχιακός Φοιτητής

ABSTRACT

With the growth of technology and Internet’s scope there is a huge increase in the volume of data which is accessible to its users. In the last few years, more and more countries are participating in attempts to enrich the category of interconnected data that is associated with legislative knowledge. Access to this knowledge is provided by Nomothesi@, a web search platform for Greek Legislation, implemented using REST technologies and powered by a large amount of information based on the RDF data model describing the legislative relationships that are modeled based on an OWL ontology. The purpose of this work is to extend the platform's existing data with Raptarchis’ legal volumes and to provide its users with a wider range of legal archives, thereby contributing to the representation of the legal knowledge contained in open data related to Greek Legislation. The Raptarchis’ Legislative Collection consists of volumes divided into thematic sections, each containing a collection of legislative sources. In the context of the work, in order to obtain the data that would feed the Nomothesi@ platform, the volumes were broken down into individual legislative sources per text file and a parser written in Java programming language was implemented in order to identify the hierarchical component structures that constitute a legislative source. As a result, this project contributes to expanding the volume of legislative texts accessible to open data for information and encourages their further exploitation.

SUBJECT AREA: Syntax Analysis, Open Data, Semantic Web, Artificial Intelligence, Greek Legislation

KEYWORDS: Rest Technologies, RDF Date, OWL Ontology, Raptarchis, Legal Resource, Thematic Sections, Parser

ΠΕΡΙΛΗΨΗ

Με την ανάπτυξη της τεχνολογίας και του Internet υπάρχει μια τεράστια αύξηση στον όγκο των δεδομένων που είναι προσβάσιμα στους χρήστες του. Τα τελευταία χρόνια, όλο και περισσότερες χώρες συμμετέχουν σε μια προσπάθεια εμπλουτισμού της κατηγορίας των διασυνδεδεμένων δεδομένων που σχετίζονται με τη νομοθετική γνώση. Την πρόσβαση στις γνώσεις αυτές την παρέχει η Nomothesi@, μια ηλεκτρονική πλατφόρμα αναζήτησης πάνω στην ελληνική νομοθεσία και η οποία είναι υλοποιημένη με χρήση REST τεχνολογιών και τροφοδοτείται από ένα μεγάλο όγκο πληροφοριών βασισμένο πάνω στον μοντέλο δεδομένων RDF περιγράφοντας τις νομοθετικές σχέσεις σύμφωνα με την μοντελοποίησή τους πάνω σε μια OWL οντολογία. Σκοπός της εργασίας αυτής είναι να επεκτείνει τα ήδη υπάρχοντα δεδομένα της πλατφόρμας με τους νομοθετικούς τόμους του Ραπτάρχη και να προσφέρει στους χρήστες της ένα μεγαλύτερο εύρος νομικών αρχείων συμβάλλοντας με τον τρόπο αυτό στην αναπαράσταση της νομικής γνώσης που συμπεριλαμβάνεται στα ανοιχτά δεδομένα σχετικά με την ελληνική νομοθεσία. Η νομοθετική συλλογή του Ραπτάρχη αποτελείται από τόμους χωρισμένους σύμφωνα με θεματικές ενότητες, καθένας από τους οποίους περιέχει μια συλλογή από νομοθετικές πηγές. Στα πλαίσια της εργασίας, για να επιτευχθεί η εξαγωγή των δεδομένων που θα τροφοδοτήσουν την πλατφόρμα Nomothesi@, οι τόμοι διασπάστηκαν σε μεμονωμένες νομοθετικές πηγές ανά αρχείο κειμένου και στη συνέχεια υλοποιήθηκε ένας συντακτικός αναλυτής σε γλώσσα προγραμματισμού Java για την αναγνώριση των ιεραρχικών συστατικών δομών που αποτελούν μια νομοθετική πηγή. Ως αποτέλεσμα το έργο αυτό συμβάλλει στην διεύρυνση του όγκου των νομοθετικών κειμένων που είναι προσβάσιμα στα ανοιχτά δεδομένα για πληροφόρηση και ενθαρρύνει την περαιτέρω αξιοποίηση τους.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συντακτική Ανάλυση, Ανοιχτά Δεδομένα, Σημασιολογικός Ιστός, Τεχνητή Νοημοσύνη, Ελληνική Νομοθεσία

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Rest Τεχνολογίες, RDF Δεδομένα, OWL Οντολογία, Ραπτάρχης, Νομοθετική πηγή, Θεματικές Ενότητες, Συντακτικός Αναλυτής

ACKNOWLEDGEMENTS

First, I would like to thank my family for their support and their unselfish love that have given to me all those years. Their sentimental and financial contribution have established the foundation of my academic course and made me the person I am today.

Secondly, I would also like to express my gratitude to my supervisor Prof. Manolis Koubarakis for trusting me to complete this thesis and assiduously mentoring me throughout the whole process.

Finally, I would like to thank MSc student Christos Papaloukas for allowing me to contribute on this challenging project and guiding me from start to finish.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Every citizen, as an active and productive member of the society they belong to and interact with, must be able to participate in critical and vital decisions on social paths as well as to elect their representatives. In order to do this, they must have some basic knowledge of political and social organization and be aware of public services' functioning. The access to that information is easier than ever thanks to Nomothesi@[1], a web platform build with RESTful technologies that gives the opportunity to its users to browse through a variety of different Legal Resources and make complicate searching operations with the help of a SPARQL endpoint. Nomothesi@'s main objective is to get citizens acquainted with Greek Legislation simplifying their life, eliminating injustice, fortifying their rights and simultaneously contributing to legislative knowledge representation with linked data.

1.1 Objectives of the thesis

The main objective of this project is the generation of a dataset based on the principles of RDF data model and expressing the relations the Nomothesi@'s OWL ontology defines. We will use that dataset in order to feed the web platform Nomothesi@ with a variety of legal resources that were only accessible from law enforcement. Those legal resources consist the project with code name '*Continuous Legislation Code – Raptarchis*' and for a long period were private property of Pantelis Raptarchis until they were donated to government's public sector. Raptarchis' collection is composed of forty-two volumes with each of them being equivalent to a thematic section. In order to parse those documents, we had to convert them from formatted text files (.doc) to plain text files (.txt). For better extracting results there was a need to separate the legal resources contained in the volumes to a resource per text file.

In order to generate this dataset, we need to implement a parser in Java programming language that will extract the metadata and split legal resources to their structural elements. The difficulties of this goal were modeling Raptarchis' legislative documents and choosing the right parsing techniques in order to achieve the best parsing results. We wanted to separate the parsing procedure into stages in order to simplify the extraction and make our parser more consistent.

This project is focusing on the contribution to the representation of legislative knowledge on the web, making a significant large amount of legal resources accessible to Open Data and making citizens accustomed to Greek Legislation eliminating injustice, abolishing political ignorance and fortifying civil rights.

[1] See http://legislation.di.uoa.gr/

1.2 Thesis structure

This thesis is organized in six chapters. In Chapter 1 we present the thesis's introduction which contains the basic objectives and its main structure. Furthermore, in Chapter 2 we focus on background of Greek Legislation explaining the encoding of legal resources and how Nomothesi@ has been modeled based on an OWL technology and powered by a large dataset following the principles of RDF data model. In Chapter 3 we introduce project 'Continuous Legislation Code – Raptarchis', a collection of volumes containing a massive amount of legal resources that are divided into thematic sections. In the end of this chapter we present a list of all known topics. Next is Chapter 4, the core of this thesis and contains the implementation of the parser we designed for data extraction. In the chapter's beginning (Chapter 4), we explain the conversion flow of Raptarchis' documents in the preprocessing stage and the parser's specification. Subsequently, we elaborate on the modeling of Raptarchis' legal documents with Java programming language and how we achieve the extraction of Legal resource's subdivisions and metadata in each parsing stage. Finally, in Chapter 5 we present the results and the final datasets as they were extracted from our parser and in Chapter 6, we discuss the conclusions of this thesis and possible future work.

2. BACKGROUND AND RELATED WORK

In this chapter we are introducing the structure of Greek Legislation and how is being encoded by the legislative authorities. We focus on the way that web platform Nomothesi@ has been designed and the technologies they have been used in order to achieve the best results in Greek legislation's representation. We are explaining the principles of the RDF data model and the OWL ontology that defines the relationship between Legal Resources and their subdivisions. Finally, we make a reference to a previous project called Nomothesi@ G3 Parser which produced the initial platform's dataset.

2.1 Deconstruction of Greek Legislation

All the legislative knowledge created from the Greek parliament is accessible to the public in gazette form through the website of National Printing House and only in PDF format. National Printing House's archive contains a great number of legal gazettes with the first been dated around 1833 and is being daily updated with new legislation. Because of the wide chronological range, an older gazette may show variance on encoding or document's structure. In order to encounter this inconsistency, the Greek Central Committee of Encoding Standards issued a document with title "Manual Directives for the encoding of legislation" [1] introducing a set of rules for legislation encoding that was further legislated by Law 2003/3133. In 2019, the Law 2019/4606 reformed the institutional framework of the Greek Central Committee of Encoding Standards and made modifications to the legislation encoding manual [2].

2.1.1 Hierarchy of legal subdivisions and basic encoding

As we introduced previously, this set of rules defines a consistent template for legislation encoding so that the legal resources can be easily interpreted and efficiently exploited by the public. The gazettes that have been published by the National Printing House may contain a single or a set of legal resources and each one should apply to that specifications.

Each legal resource starts with a metadata text fragment including information about its type and its serial number that is unique based on the publication year of the resource that can been found on gazette's sheets. After the main metadata information, the legal resource may accommodate a title presenting the thematic topic of the legislation. It is very common for a legal resource to contain a list of citations to existing legislative documents that have been already issued and to refer to them by their unique identifier that is structured from the combination of the document's type, serial number and publication year.

The main body of a legislative resource is structured in a tree hierarchical form of fragments (subdivisions). The body is divided based on the density and the topics that are being referenced to in the resource's content and each fragment should be characterized by simplicity and homogeneity. An article, the fundamental fragment of a legal resource, may be part of a superset containing fragments higher in the hierarchy. More precisely, a legal resource can be divided into books, parts, sections, chapters and finally articles, placed in descending sequence. This means that a book may contain a set of Parts, Sections, Chapters or Articles but a Part cannot include a set of Books because a Book is higher in

hierarchical priority. All the valid combinations of hierarchical division can be seen in Figure 1.

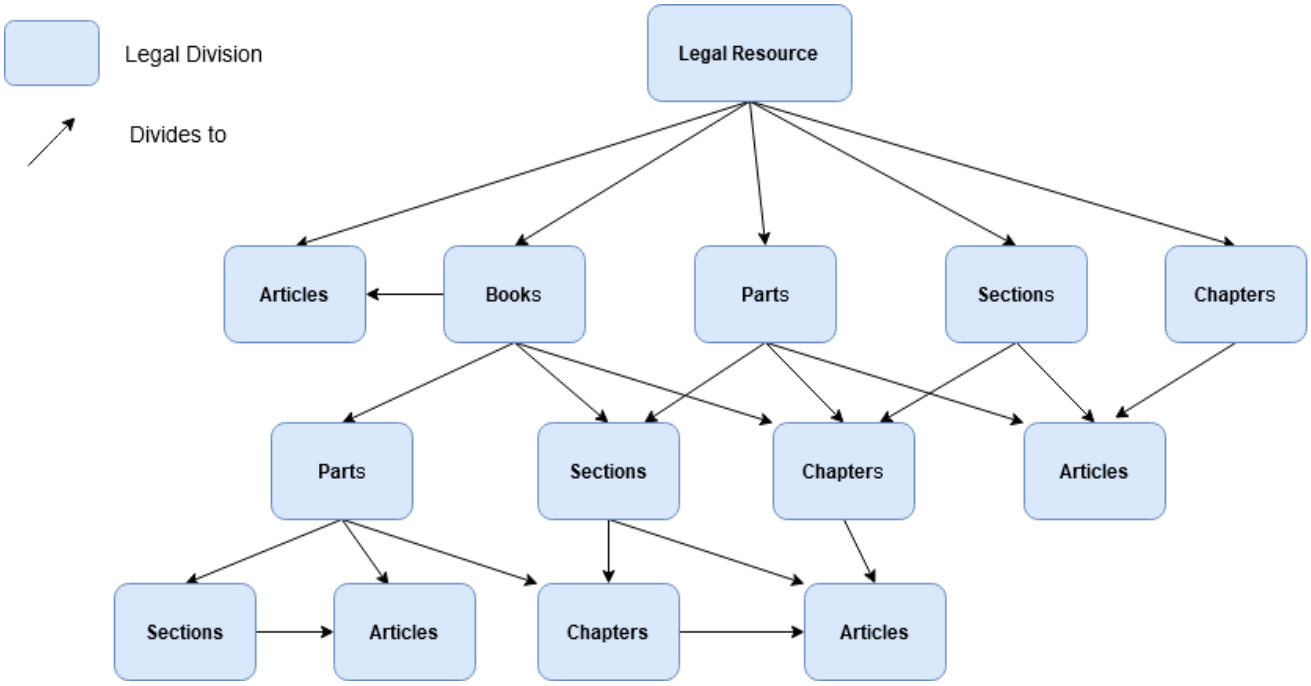
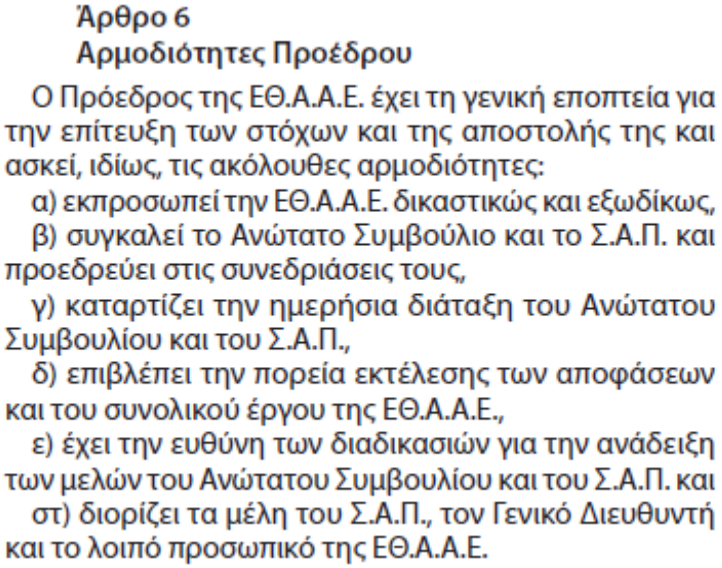


Figure 1: Hierarchical division of Greek Legislation

Books and parts are identified inside the body using numbered ordinals (First, Second, Third, …). Sections are numbers using capital Latin numerals (I, II, III, …). For chapters, we are using capital letters of the Greek Alphabet (Α, Β, Γ, …) for their enumeration. Articles are numbered using Arabic numerals (1, 2, 3, …) and are the most critical subdivision type. They are comprised of at least one paragraph. Paragraphs are identified with Arabic numerals (1, 2, 3, …) exactly as articles and divide legislative context into individual fragments with relative concept. Furthermore, paragraphs consist of passages, indents or a combination of both. Indents are numbered using lower-case letters of Greek Alphabet (α, β, γ, …) and may contain sub-indents that listed using double lower-case Greek letters (αα, αβ, βα, ββ, γγ, …). Passage is defined as the continuous text fragment between two punctuation marks for word's end and is the basic component that structures both paragraphs and indents. The Table 1 contains the enumeration encoding for each legal subdivision and in Figure 2 we can see the main form of an article with one paragraph and indents (Article 6 of Law 4653/2020).

Table 1: Subdivision table of Greek legislation

Legal Subdivision	Enumeration Type	Symbol
Book	Ordinal numbers	First, Second, …
Part	Ordinal numbers	First, Second, …
Section	Latin numbers	I, II, III, IV, …
Chapter	Capital Greek alphabet	Α, Β, Γ, …
Article	Arabic numbers	1, 2, 3, …
Paragraph	Arabic numbers	1, 2, 3, …
Case	Lower Greek alphabet	α, β, γ, …
Subcase	Double lower Greek alphabet	αα, αβ, βα, ββ, γα, γβ, γγ, …
Linea	-	-



Άρθρο 6
Αρμοδιότητες Προέδρου

Ο Πρόεδρος της ΕΘ.Α.Α.Ε. έχει τη γενική εποπτεία για την επίτευξη των στόχων και της αποστολής της και ασκεί, ιδίως, τις ακόλουθες αρμοδιότητες:

α) εκπροσωπεί την ΕΘ.Α.Α.Ε. δικαστικώς και εξωδίκως,

β) συγκαλεί το Ανώτατο Συμβούλιο και το Σ.Α.Π. και προεδρεύει στις συνεδριάσεις τους,

γ) καταρτίζει την ημερήσια διάταξη του Ανώτατου Συμβουλίου και του Σ.Α.Π.,

δ) επιβλέπει την πορεία εκτέλεσης των αποφάσεων και του συνολικού έργου της ΕΘ.Α.Α.Ε.,

ε) έχει την ευθύνη των διαδικασιών για την ανάδειξη των μελών του Ανώτατου Συμβουλίου και του Σ.Α.Π. και

στ) διορίζει τα μέλη του Σ.Α.Π., τον Γενικό Διευθυντή και το λοιπό προσωπικό της ΕΘ.Α.Α.Ε.

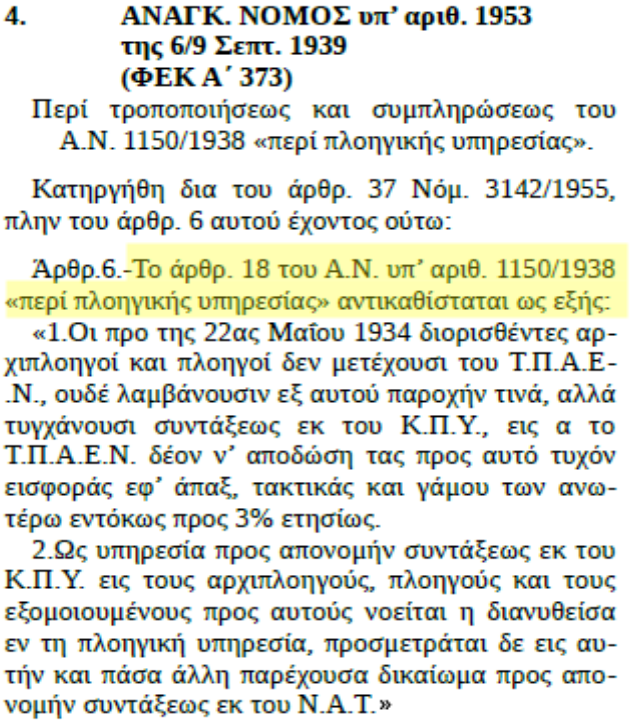
Figure 2: Main form of an Article

2.1.2 Modifications

One of the main characteristics of a legal resource is its life span, which starts from its enforced date to its expiration date. During this period, other enforced legislative documents with conflicting resolutions may affect its provisions. It is very common in Greek legislation for a legal resource to modify a legal fragment of another (legal resource). The fragments that can be amended is the entire legal resource or the subdivision parts (e.g. Book, Article, Paragraph, …) that consist it. There is a variation of modifications that we can come across in Greek legislation but in Nomothesi@ platform we identify the three following basic types [3]:

- Insertion: An entire or a part of legislative fragment is inserted verbatim on a specific place inside patient-resource.
- Repeal: An entire or a part of legislative fragment is removed from the patient-resource.
- Substitution: An entire or a part of legislative fragment is replaced by a new fragment exactly as it appears on the modification section.

Modifications interfere with the linear flow of a legal resource, making the valid resource's representation on the web quite challenging. Additionally, extra complexity is added on the Nomothesia platform's effort on identifying and parsing legislative modifications because the Greek legislation encoding manual does not specify rules about modifications' format and therefore, we observe a large amount of variations in modifications' structure and idioms. In Figure 3 we present an example of legislative modification that belongs to the category of substitutions.



4. ΑΝΑΓΚ. ΝΟΜΟΣ υπ' αριθ. 1953 της 6/9 Σεπτ. 1939 (ΦΕΚ Α΄ 373)

Περί τροποποιήσεως και συμπληρώσεως του Α.Ν. 1150/1938 «περί πλοηγικής υπηρεσίας».

Κατηργήθη δια του άρθρ. 37 Νόμ. 3142/1955, πλην του άρθρ. 6 αυτού έχοντος ούτω:

Άρθρ.6.-Το άρθρ. 18 του Α.Ν. υπ' αριθ. 1150/1938 «περί πλοηγικής υπηρεσίας» αντικαθίσταται ως εξής:

«1.Οι προ της 22ας Μαΐου 1934 διορισθέντες αρχιπλοηγοί και πλοηγοί δεν μετέχουσι του Τ.Π.Α.Ε-.Ν., ουδέ λαμβάνουσιν εξ αυτού παροχήν τινά, αλλά τυγχάνουσι συντάξεως εκ του Κ.Π.Υ., εις α το Τ.Π.Α.Ε.Ν. δέον ν' αποδώση τας προς αυτό τυχόν εισφοράς εφ' άπαξ, τακτικάς και γάμου των ανωτέρω εντόκως προς 3% ετησίως.

2.Ως υπηρεσία προς απονομήν συντάξεως εκ του Κ.Π.Υ. εις τους αρχιπλοηγούς, πλοηγούς και τους εξομοιουμένους προς αυτούς νοείται η διανυθείσα εν τη πλοηγική υπηρεσία, προσμετράται δε εις αυτήν και πάσα άλλη παρέχουσα δικαίωμα προς απονομήν συντάξεως εκ του Ν.Α.Τ.»

Figure 3: Legislative modification example

2.2 The application of Semantic Web technologies on Greek Legislation

The knowledge that Internet contains today as reachable data is unmeasurable, but it is strictly bound to the needs of internet pages and web platforms that make use of it. Contrariwise, Semantic Web is an extension of the Web and its main goal is to make it to be driven by linked machine-readable data which applications can use. This means the existence of public data stores that have been built using Semantic Web's technologies such as Resource Description Framework (RDF) and OWL ontologies and can be exploited by internet pages and applications. In order to support the initiative of making Greek legislative knowledge accessible through linked data on the web, our data population is based on these protocols.

2.2.1 Identifying Legal Resources using URIs

Semantic Web's concept is a web of data where resources are public and linked using relationships. In order to refer to these resources, an identifier protocol called Uniform Resource Identifier (URI) has been developed that defines a string of characters that uniquely identifies a web's resource. In order each country's legislative data to be consistent and unified with the European legislation so it can be used efficiently, European Council provided European Legislation Identifier (ELI) [4], a framework based on Semantic Web's technologies, to be adopted from all European countries who provide legislative data on internet. ELI is an extendable framework that can be enriched according to the needs of each country's legislative peculiarities. Its main idea proposes a URI template, for legal resources identification, and an OWL ontology for expressing relationships and events between them.

In order to identify each legal resource, its subdivisions and the relations between them, Nomothesia uses HTTP URIs[1]. These URIs must be precise and their structure should accurately represent the resource they identify, making it easily guessable. Legal resource's form uses the following template,

http://www.legislation.di.uoa.gr/eli/{type}/{year}/{id}

This URI template [5] identifies uniquely a legal resource based on the selected values of the three following parameters: type, year and id. For parameter type we can assign a value from a collection of special codes, each representing a different legal resource type and can be seen in Table 2. In Chapter 4 we will discuss an extended version of this collection based on Raptarchis' legislative documents. For the year parameter the template expects the publication year of the legal resource and as id its distinctive serial number. For example, if we want to refer to the Presidential Decree with serial number 1 published in 2020, the corresponding URI would be:

http://www.legislation.di.uoa.gr/eli/pd/2020/1

[1] See https://www.w3.org/DesignIssues/HTTP-URI.html

Table 2: Encoding types of Legislation

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

For legal resource's subdivisions, URI form can be extended by concatenating a string suffix in the end of the main resource's URI. The suffix must contain the type of subdivision (book, part, section, chapter, …) and its numbered id. For instance, to present the chapter 2 of Presidential Decree 1 of 2020, we use the following URI:

http://www.legislation.di.uoa.gr/eli/pd/2020/1/chapter/2

In case of nested hierarchical subdivisions, the URI of target division is formed by concatenating the string from the highest division in the nested hierarchy to the target subdivision. For example, to refer to paragraph 1 of article 3 that is under the scope of chapter 2 of Presidential Decree 1 of 2020, we use:

http://www.legislation.di.uoa.gr/eli/pd/2020/1/chapter/2/article/3paragraph/1

2.2.2 The Resource Description Framework (RDF) data model

Previously, we introduced the identification of legal resources using URIs based on the template that European Council proposed. Each of these resources consists of fragments, metadata and refers to other legislative resources creating a relational chain between then, making their storing procedure at web's data stores very challenging. For that purpose, Nomothesia platform uses the Resource Description Framework (RDF), a metadata data model defining the structure of which data is being stored and offers the technology for knowledge representation, data modeling and relationships definition between web resources. RDF decomposes knowledge to expressions in the subject-predicate-object form, known as triples, linking web's data and describing facts. Triples represent relationships between resources that can be identified on the web and can be easily processed by applications. The subject and the predicate are resources and are identified generally by URIs, whereas the object can be either a resource or a literal value. For example, if we want to declare that Presidential Decree 1 of 2020 contains one article, a valid triple would be the following,

e.g.

<http://legislation.di.uoa.gr/pd/2020/1>
<http://data.europa.eu/ontology#has_part>
<http://legislation.di.uoa.gr/pd/2020/1/article/1>.

The example above uses a URI as the triple's object to unambiguously identify the first article of Presidential Decree 1 of 2020. In the next example we are showing how the object can be a string literal, representing the title of the same Presidential Decree,

e.g.

<http://legislation.di.uoa.gr/pd/2020/1>
<http://data.europa.eu/ontology#title>
"Establishment of scientific bodies and committees"@en.

A collection of RDF triples can be visualized by a directed knowledge graph[1]. In this graph, each triple is represented by a node for subject, a node for object and an arc for predicate, directed from subject to object. The RDF graph for previous triples is shown on Figure 4.

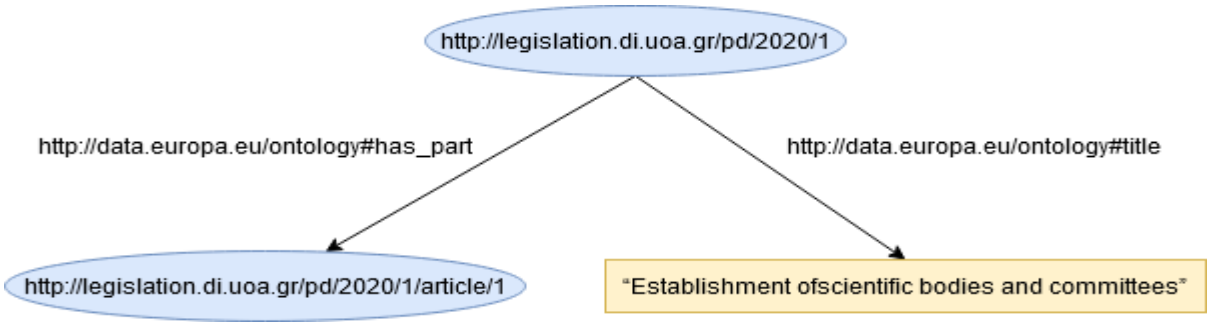


Figure 4: RDF graph example

2.2.3 Nomothesi@ Ontology

In the previous section, we introduced the Recourse Data Framework (RDF) as a data model that defines the structure of web's data in order to represent relationships. Because RDF's capabilities are limited, Semantic Web offers Ontology Web Language (OWL), a technology for rich knowledge representation. OWL languages are built upon the RDF describing powerful and formal semantics for resources' relationships and are flexible on information coming even from heterogeneous data sources. More precisely, the data that an OWL language refers to is interpreted as a set of "individuals" (classes) and a set of "property assertions" which relate these individuals to each other. Furthermore, OWL ontologies are characterized by the existence of a set of axioms that defines constrains to these classes and the relationships between them. In order to model web's resources about Greek legislation, an ontology has been created called Nomothesi@ ontology. Nomothesi@ ontology is an enriched version of ELI's ontology that is adapted on Greek legal resources' peculiarities. For the better understanding we point the basic schema of Nomothesia ontology in figure 5.

[1] See https://www.w3.org/2012/08/RDFNG.html#

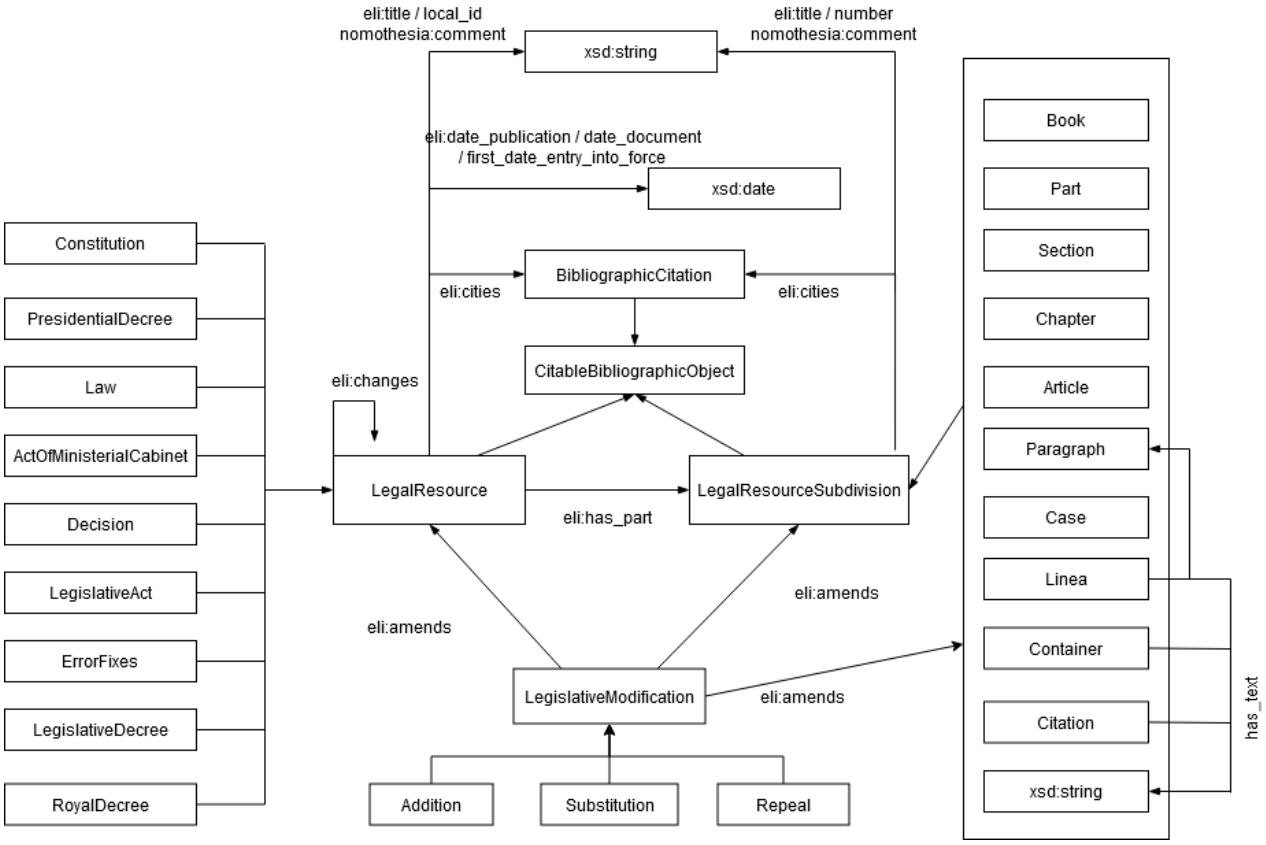


Figure 5: Nomothesi@ ontology

2.3 Related work

The vital part of Nomothesia platform is its data. In the first version of the platform, the dataset population was a demanding task because of the bureaucratic methods in which the legal resources were distributed and the error-prone documents. In order to achieve this goal, a parsing tool was built called Nomothesia G3 Parser.

2.3.1 Nomothesia G3 Parser

Nomothesia G3 Parser [6] is a rule-based tool built to create the initial dataset that would be used to power Nomothesia platform. The chosen programming language for parser's implementation was Java and was designed to be flexible both on typing and structural errors. More than 12.000 legal resources during 1990-2020 were parsed with G3 Parser and produced millions of triples. The basic stages of G3 Parser are the following:

1. Conversion of Greek legal gazettes from PDF format to plain text.

2. Division of gazettes' text into individual legal resources.

3. Extracting legal resource's primary metadata (i.e., title, issue number etc.).

4. Legal resource's structural fragments (i.e., articles, paragraphs, indents, passages) extraction and RDF triple population.

5. Extraction of additional information from bottom-level fragments (i.e., passages).

2.4 Summary

In this Chapter we introduced the basic principles of Greek Legislation's encoding and more precisely, we explained the hierarchical structure of a legal resource and how it is related with others through legislative modifications. Subsequently, we described three basic frameworks of Semantic Web technologies and their appliance in Legislation's representation on the Web. Finally, we presented the parsing stages of Nomothesia G3 Parser, a parsing tool that created the initial dataset for Nomothesi@ platform.

3. CONTINUOUS LEGISLATION CODE – RAPTARCHIS

The main motivation for implementing this thesis was the public sharing of a collection of legal documents that contain a large amount of legal resources from 1834 until today. This legislative collection was created as part of the project Continuous Legislation Code – Raptarchis and offers a significant opportunity for expanding Nomothesia platform's dataset and contributes to the enrichment of web's linked data that refers to national legislation. More precisely, in this chapter we will present the initiative of Raptarchis' project and we will discuss about the structure of legal resources that have been included in the documents, the peculiarities and the differentiation they have compared to those that have been published in government's gazette. Finally, we will introduce the thematic sections that the legal documents are sorted by.

3.1 Project Raptarchis

The Continuous Legislation Code – Raptarchis [7] is an initiative of grouping the Greek legislation from the establishment of Greek state until today, into a collection of legal resources, sorted by thematic topics. The founder and creator of the Continuous Legislation Code was Pantelis Raptarchis who maintained the project as a private enterprise until 1978. In 1978, by law 805/1978 the project was donated to the Greek State and on 01/01/1979, it was passed into the possession of the Ministry of the Presidency of the Government. The legislative collection consists of a variety of different legal resources (i.e., laws, decrees, regulations, etc.), all being extracted by the official Greek government's gazette. All this legislative knowledge is divided in 111 numbered volumes that have been distributed to 40 main thematic sections.

At the beginning of its "public life", the management service of Continuous Legislation Code was an independent department and appertained directly to the general secretary of the Ministry of the Presidency. Today, the service has been integrated to the Finance Directorate of the Directorate-General for Administrative Support of the General Secretariat of Public Administration & E-Government, of Public Administration and Decentralization.

Initially, Raptarchis' legal collection was intended to be used by people related to law, such as lawyers or law enforcers and organizations whose professional activity requires access to all laws in force at that time. Instead, the current version of Raptarchis' project, besides the accessibility that offers to every citizen that want to be informed about Greek legislation, aims to provide the latest legislation knowledge for a multitude of pre-defined subject areas.

The main advantages of the Raptarchis' collection is the thematic categorization of Greek legislation, which facilitates the reader to find the requested legislative source in a short period of time as well as preserving the legality of documents and historically recording the evolution of legislation.

3.2 Structure of legal documents

In Chapter 2 we introduced the suggested legislation encoding as it was proposed by Law 2003/3133. Nevertheless, the legislative collection of Raptarchis contains legal resources that are approximately dated from 1834 until today, making almost impossible to collaboratively follow this encoding. Even for the most recent legal documents we observe deviations, especially on metadata and enumeration types. The Continuous Legislation Code's initial goal was not to make a formal and strictly accurate representation of legislative knowledge, instead it aimed to create a serviceable and quick index of legislative resources based on thematic sections. Furthermore, all the legal fragments the collection contains were extracted by official Greek government's gazettes and written by hand, making them error-prone.

3.2.1 Distribution of legal resources based on thematic sections

Each legislative volume of Continuous Legislation Code project is related to a main thematic topic and has an internal structure based on it. Inside the volume, the main topic is divided into thematic subcategories which are represented by chapters and subsequently each chapter breaks down to subjects which contain the legal resources, creating an interconnected thematic chain. Chapters are being enumerated using capital letters of the Greek alphabet and are followed by their thematic title while subjects are being enumerated with lower-case letters of the Greek alphabet. The design of inner structure of volumes of Raptarchis' legal collection provides an index of legal resources based on the thematic section that are related to, offering their readers a very fast and efficient method for searching the Greek legislative knowledge. In Figure 6 we can see the thematic hierarchy diagram of a volume.

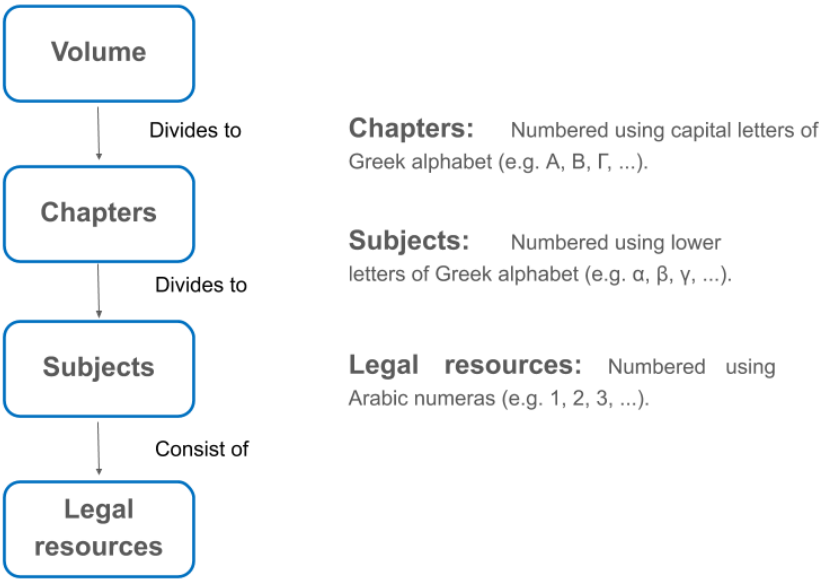
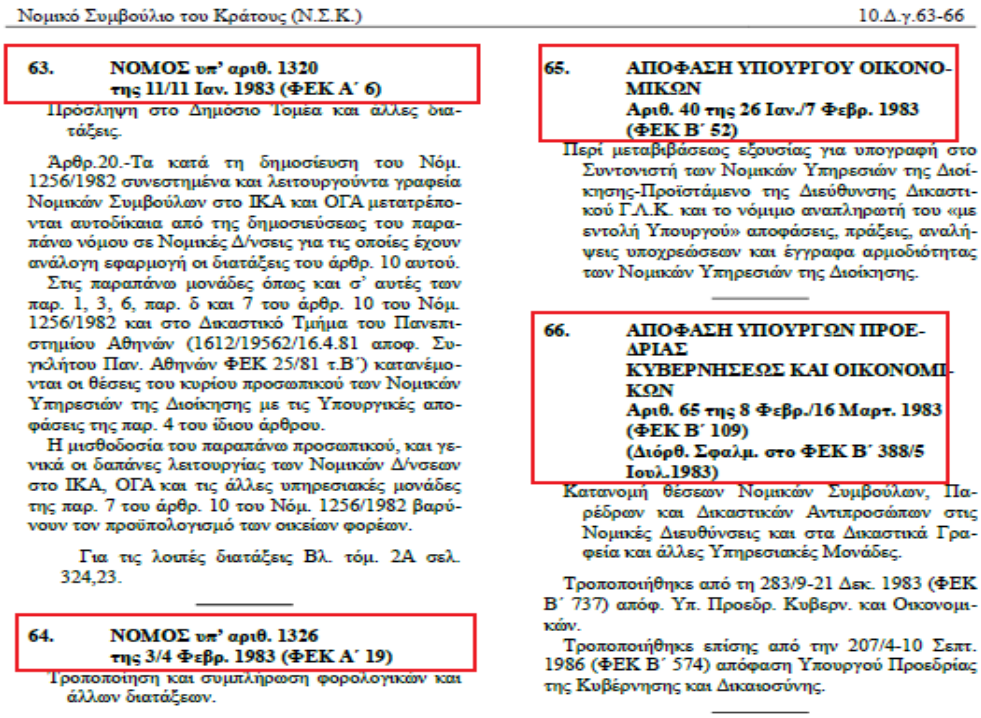


Figure 6: Hierarchical structure of Raptarchis' legislative volume

3.2.2 Differentiation on Raptarchis' structure

Although, the basic guideline for the encoding of Raptarchis' legal resources is similar to the one we introduced in Section 2.1.1 of Chapter 2, we observe several deviations compared to those in Greek government's gazette.

The first differentiation involves the provided information about metadata of each single legislative resource. Unlike government's gazette, Raptarchis' legal collection does not refer to the signers of each legal resource, except for some rare cases. The publication date has been embedded in a main metadata fragment at the start of each resource. Each metadata fragment is numbered in ascending order and contains information about the type of legal resource, its serial number, the publication date and often the legislative gazette's number in which it was published. The form of the metadata fragment can be seen in figure 7.



Νομικό Συμβούλιο του Κράτους (Ν.Σ.Κ.) 10.Δ.γ.63-66

63. ΝΟΜΟΣ υπ' αριθ. 1320 της 11/11 Ιαν. 1983 (ΦΕΚ Α΄ 6)

Πρόσληψη στο Δημόσιο Τομέα και άλλες διατάξεις.

Άρθρ.20.-Τα κατά τη δημοσίευση του Νόμ. 1256/1982 συνεστημένα και λειτουργούντα γραφεία Νομικών Συμβούλων στο ΙΚΑ και ΟΓΑ μετατρέπονται αυτοδίκαια από της δημοσιεύσεως του παραπάνω νόμου σε Νομικές Δ/νσεις για τις οποίες έχουν ανάλογη εφαρμογή οι διατάξεις του άρθρ. 10 αυτού.

Στις παραπάνω μονάδες όπως και σ' αυτές των παρ. 1, 3, 6, παρ. δ και 7 του άρθρ. 10 του Νόμ. 1256/1982 και στο Δικαστικό Τμήμα του Πανεπιστημίου Αθηνών (1612/19562/16.4.81 αποφ. Συγκλήτου Παν. Αθηνών ΦΕΚ 25/81 τ.Β΄) κατανέμονται οι θέσεις του κυρίου προσωπικού των Νομικών Υπηρεσιών της Διοίκησης με τις Υπουργικές αποφάσεις της παρ. 4 του ίδιου άρθρου.

Η μισθοδοσία του παραπάνω προσωπικού, και γενικά οι δαπάνες λειτουργίας των Νομικών Δ/νσεων στο ΙΚΑ, ΟΓΑ και τις άλλες υπηρεσιακές μονάδες της παρ. 7 του άρθρ. 10 του Νόμ. 1256/1982 βαρύνουν τον προϋπολογισμό των οικείων φορέων.

Για τις λοιπές διατάξεις Βλ. τόμ. 2Α σελ. 324,23.

---

64. ΝΟΜΟΣ υπ' αριθ. 1326 της 3/4 Φεβρ. 1983 (ΦΕΚ Α΄ 19)

Τροποποίηση και συμπλήρωση φορολογικών και άλλων διατάξεων.

65. ΑΠΟΦΑΣΗ ΥΠΟΥΡΓΟΥ ΟΙΚΟΝΟΜΙΚΩΝ Αριθ. 40 της 26 Ιαν./7 Φεβρ. 1983 (ΦΕΚ Β΄ 52)

Περί μεταβιβάσεως εξουσίας για υπογραφή στο Συντονιστή των Νομικών Υπηρεσιών της Διοίκησης-Προϊστάμενο της Διεύθυνσης Δικαστικού Γ.Λ.Κ. και το νόμιμο αναπληρωτή του «με εντολή Υπουργού» αποφάσεις, πράξεις, αναλήψεις υποχρεώσεων και έγγραφα αρμοδιότητας των Νομικών Υπηρεσιών της Διοίκησης.

---

66. ΑΠΟΦΑΣΗ ΥΠΟΥΡΓΩΝ ΠΡΟΕΔΡΙΑΣ ΚΥΒΕΡΝΗΣΕΩΣ ΚΑΙ ΟΙΚΟΝΟΜΙΚΩΝ Αριθ. 65 της 8 Φεβρ./16 Μαρτ. 1983 (ΦΕΚ Β΄ 109) (Διόρθ. Σφαλμ. στο ΦΕΚ Β΄ 388/5 Ιουλ.1983)

Κατανομή θέσεων Νομικών Συμβούλων, Παρέδρων και Δικαστικών Αντιπροσώπων στις Νομικές Διευθύνσεις και στα Δικαστικά Γραφεία και άλλες Υπηρεσιακές Μονάδες.

Τροποποιήθηκε από τη 283/9-21 Δεκ. 1983 (ΦΕΚ Β΄ 737) απόφ. Υπ. Προεδρ. Κυβερν. και Οικονομικών.

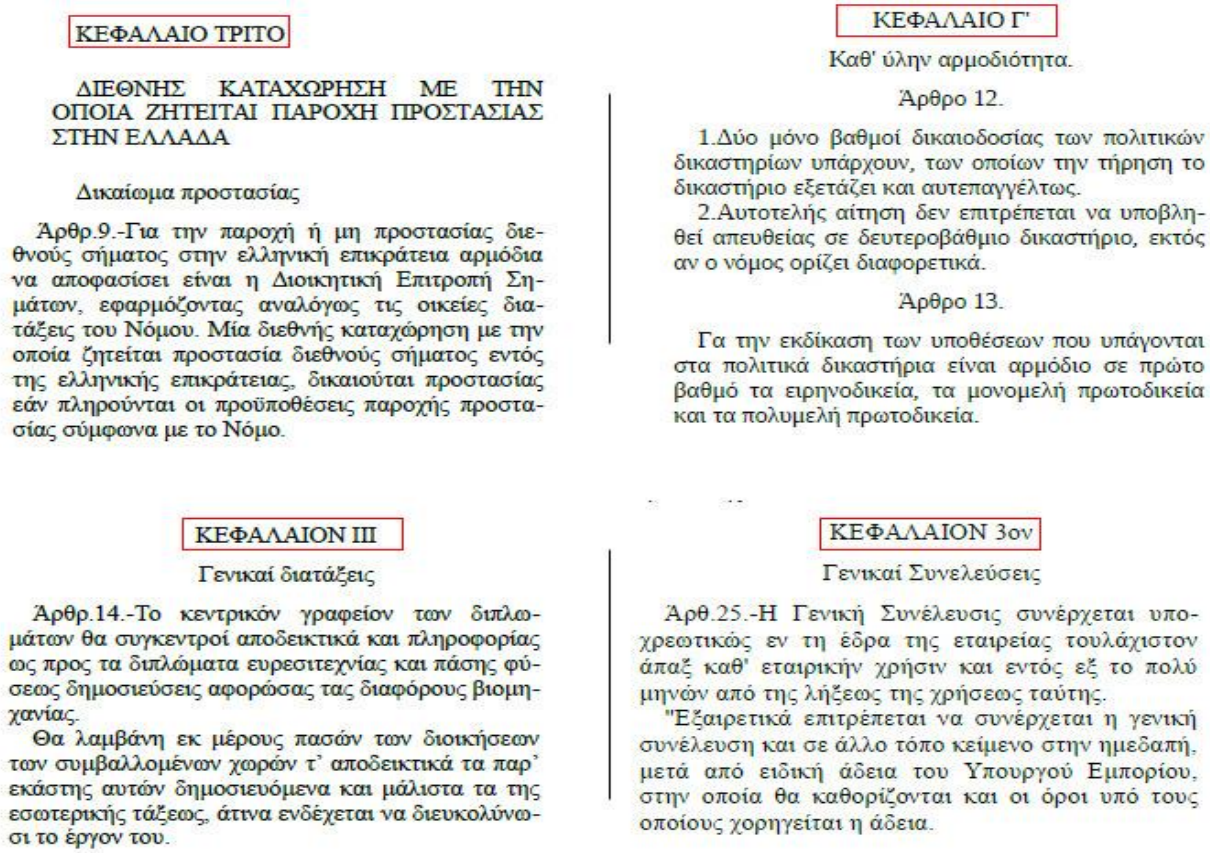
Τροποποιήθηκε επίσης από την 207/4-10 Σεπτ. 1986 (ΦΕΚ Β΄ 574) απόφαση Υπουργού Προεδρίας της Κυβέρνησης και Δικαιοσύνης.

Figure 7: Form of metadata fragments

Another difference we often encounter is the inconsistency in the enumeration of legal subdivisions (fragments) which a legal resource is divided into. As we have shown in Chapter 2, each legal fragment has its own enumeration rules, for example books and parts are using ordinal numerals and sections are ordered with Latin enumeration. Because Raptarchis' collection has a large amount of legal resources published in different chronological periods, we find variations on enumerations which add furthermore complexity to our course of parsing these legal resources and populating our machine-readable database.

Therefore, the subdivisions of books, parts, sections, chapters and articles may occur in a legal resource with all possible combinations of their original enumeration type. This means that each of the previous subdivisions is possible to be enumerated using Greek ordinal words, Greek capital letters of the alphabet, Latin numerals or rarely a form of Arabic number followed by abbreviations (1st, 2nd, 3rd, …). In Figure 8 we can see the enumeration

types that have been used in different legal documents, in order to present a chapter with the same numerical order in a legal resource.



ΚΕΦΑΛΑΙΟ ΤΡΙΤΟ

ΔΙΕΘΝΗΣ ΚΑΤΑΧΩΡΗΣΗ ΜΕ ΤΗΝ ΟΠΟΙΑ ΖΗΤΕΙΤΑΙ ΠΑΡΟΧΗ ΠΡΟΣΤΑΣΙΑΣ ΣΤΗΝ ΕΛΛΑΔΑ

Δικαίωμα προστασίας

Άρθρ.9.-Για την παροχή ή μη προστασίας διεθνούς σήματος στην ελληνική επικράτεια αρμόδια να αποφασίσει είναι η Διοικητική Επιτροπή Σημάτων, εφαρμόζοντας αναλόγως τις οικείες διατάξεις του Νόμου. Μία διεθνής καταχώρηση με την οποία ζητείται προστασία διεθνούς σήματος εντός της ελληνικής επικράτειας, δικαιούται προστασίας εάν πληρούνται οι προϋποθέσεις παροχής προστασίας σύμφωνα με το Νόμο.

ΚΕΦΑΛΑΙΟ Γ'

Καθ' ύλην αρμοδιότητα.

Άρθρο 12.

1.Δύο μόνο βαθμοί δικαιοδοσίας των πολιτικών δικαστηρίων υπάρχουν, των οποίων την τήρηση το δικαστήριο εξετάζει και αυτεπαγγέλτως.

2.Αυτοτελής αίτηση δεν επιτρέπεται να υποβληθεί απευθείας σε δευτεροβάθμιο δικαστήριο, εκτός αν ο νόμος ορίζει διαφορετικά.

Άρθρο 13.

Για την εκδίκαση των υποθέσεων που υπάγονται στα πολιτικά δικαστήρια είναι αρμόδιο σε πρώτο βαθμό τα ειρηνοδικεία, τα μονομελή πρωτοδικεία και τα πολυμελή πρωτοδικεία.

ΚΕΦΑΛΑΙΟΝ ΙΙΙ

Γενικαί διατάξεις

Άρθρ.14.-Το κεντρικόν γραφείον των διπλωμάτων θα συγκεντροί αποδεικτικά και πληροφορίας ως προς τα διπλώματα ευρεσιτεχνίας και πάσης φύσεως δημοσιεύσεις αφορώσας τας διαφόρους βιομηχανίας.

Θα λαμβάνη εκ μέρους πασών των διοικήσεων των συμβαλλομένων χωρών τ' αποδεικτικά τα παρ' εκάστης αυτών δημοσιευόμενα και μάλιστα τα της εσωτερικής τάξεως, άτινα ενδέχεται να διευκολύνωσι το έργον του.

ΚΕΦΑΛΑΙΟΝ 3ον

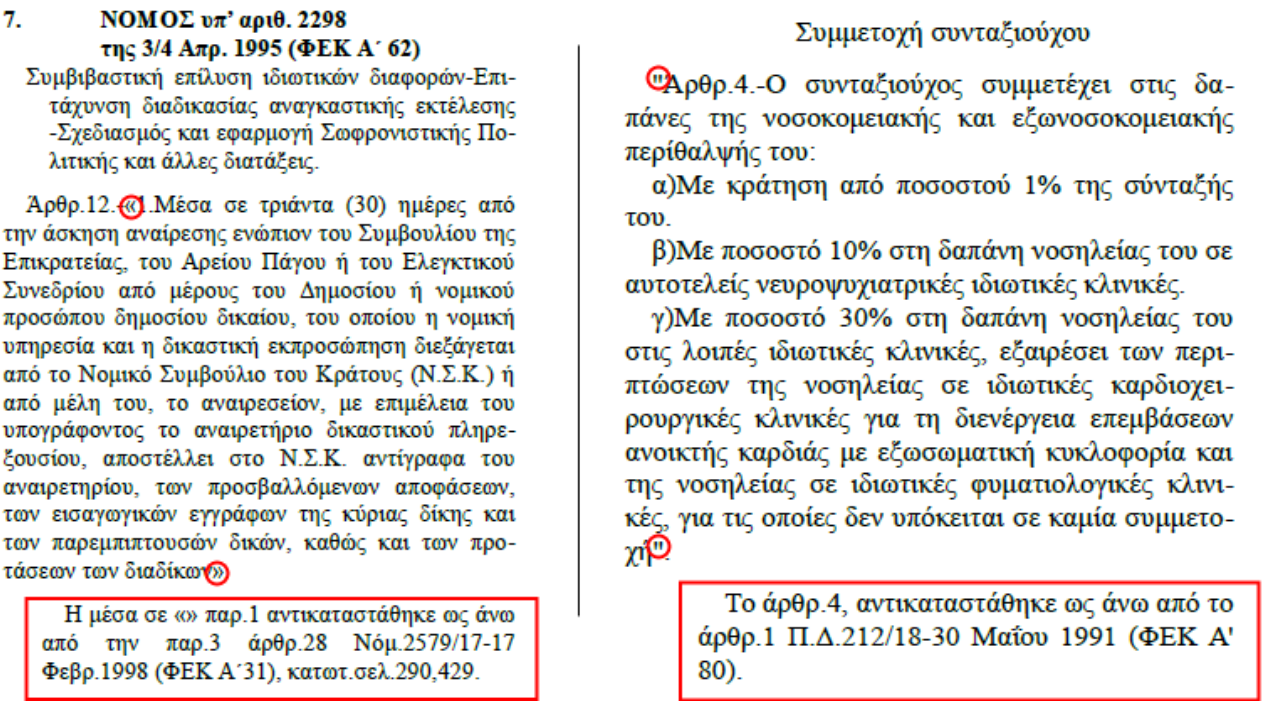
Γενικαί Συνελεύσεις

Αρθ.25.-Η Γενική Συνέλευσις συνέρχεται υποχρεωτικώς εν τη έδρα της εταιρείας τουλάχιστον άπαξ καθ' εταιρικήν χρήσιν και εντός εξ το πολύ μηνών από της λήξεως της χρήσεως ταύτης.

"Εξαιρετικά επιτρέπεται να συνέρχεται η γενική συνέλευση και σε άλλο τόπο κείμενο στην ημεδαπή, μετά από ειδική άδεια του Υπουργού Εμπορίου, στην οποία θα καθορίζονται και οι όροι υπό τους οποίους χορηγείται η άδεια.

Figure 8: Enumeration differences on legal subdivisions

Furthermore, in the legal resources contained in the Raptarchis' legislative volumes, a significant divergence is observed in the modifications in which one legal resource modifies another. In section 2.1.2 of Chapter 2 we explained the main form of a legal modification in Greek legislation and the relationship between the amending legal resources and those who are being amended. In order to provide a useful, quick and serviceable representation of Greek legislative knowledge, Raptarchis' collection has already integrated the modifications to the patient legal resources that are being amended. After each modified legal fragment that have been embedded, a metadata section is following that states the original legal resource that caused the modification. For better understanding, Figure 9 contains an example of legal modification that may often occur.



7. **ΝΟΜΟΣ υπ' αριθ. 2298**
της 3/4 Απρ. 1995 (ΦΕΚ Α΄ 62)

Συμβιβαστική επίλυση ιδιωτικών διαφορών-Επιτάχυνση διαδικασίας αναγκαστικής εκτέλεσης -Σχεδιασμός και εφαρμογή Σωφρονιστικής Πολιτικής και άλλες διατάξεις.

Άρθρ.12.«1.Μέσα σε τριάντα (30) ημέρες από την άσκηση αναίρεσης ενώπιον του Συμβουλίου της Επικρατείας, του Αρείου Πάγου ή του Ελεγκτικού Συνεδρίου από μέρους του Δημοσίου ή νομικού προσώπου δημοσίου δικαίου, του οποίου η νομική υπηρεσία και η δικαστική εκπροσώπηση διεξάγεται από το Νομικό Συμβούλιο του Κράτους (Ν.Σ.Κ.) ή από μέλη του, το αναιρεσείον, με επιμέλεια του υπογράφοντος το αναιρετήριο δικαστικού πληρεξουσίου, αποστέλλει στο Ν.Σ.Κ. αντίγραφα του αναιρετηρίου, των προσβαλλόμενων αποφάσεων, των εισαγωγικών εγγράφων της κύριας δίκης και των παρεμπιπτουσών δικών, καθώς και των προτάσεων των διαδίκων»

Η μέσα σε «» παρ.1 αντικαταστάθηκε ως άνω από την παρ.3 άρθρ.28 Νόμ.2579/17-17 Φεβρ.1998 (ΦΕΚ Α΄31), κατωτ.σελ.290,429.

Συμμετοχή συνταξιούχου

"Αρθρ.4.-Ο συνταξιούχος συμμετέχει στις δαπάνες της νοσοκομειακής και εξωνοσοκομειακής περίθαλψής του:

α)Με κράτηση από ποσοστού 1% της σύνταξής του.

β)Με ποσοστό 10% στη δαπάνη νοσηλείας του σε αυτοτελείς νευροψυχιατρικές ιδιωτικές κλινικές.

γ)Με ποσοστό 30% στη δαπάνη νοσηλείας του στις λοιπές ιδιωτικές κλινικές, εξαιρέσει των περιπτώσεων της νοσηλείας σε ιδιωτικές καρδιοχειρουργικές κλινικές για τη διενέργεια επεμβάσεων ανοικτής καρδιάς με εξωσωματική κυκλοφορία και της νοσηλείας σε ιδιωτικές φυματιολογικές κλινικές, για τις οποίες δεν υπόκειται σε καμία συμμετοχή"

Το άρθρ.4, αντικαταστάθηκε ως άνω από το άρθρ.1 Π.Δ.212/18-30 Μαΐου 1991 (ΦΕΚ Α' 80).

Figure 9: Modifications on Raptarchis' documents

3.2.3 List of thematic sections

As we mentioned on Section 3.1, Raptrachis's collections consist of 111 numbered legislative volumes with each one of them belonging to one of the 40 existing thematic sections. In this section we will list all the main thematic areas that can be found on Raptarchis' legal documents and the volumes that are related to them. The list is the following:

1. *CONSTITUTIONAL LEGISLATION*
   - *Associated volumes: 2, 2A, 2B*
2. *ADMINISTRATIVE LEGISLATION - PUBLIC OFFICIALS*
   - *Associated volumes: 2, 2A, 2B*
3. *MUNICIPALITIES AND COMMUNITIES - PREFECTURE AUTHORITY*
   - *Associated volumes: 3, 3A*
4. *POLICE LEGISLATION – FIRE BRIGADE*
   - *Associated volumes: 4, 4A*
5. *PURCHASING LEGISLATION – CODING OF MARKETING PROVISIONS*
   - *Associated volumes: 5, 5A*

6. *JUSTICE ADMINISTATION*
   - *Associated volumes: 6, 6A, 6B*
7. *URBAN LEGISLATION*
   - *Associated volumes: 7, 7A*
8. *PENAL LEGISLATION*
   - *Associated volume: 8*
9. *PENAL PROCEDURE*
   - *Associated volume: 9*
10. *CIVIL LEGISLATION*
   - *Associated volumes: 10, 10A*
11. *COMMERCIAL LEGISLATION*
   - *Associated volume: 11*
12. *ANONYMOUS COMPANIES - LTD - STOCK EXCHANGE – BANKS*
   - *Associated volumes: 12, 12A*
13. *INDUSTRIAL LEGISLATION - DEVELOPMENT LEGISLATION – FISHERY*
   - *Associated volumes: 13^1, 13^2, 13A, 31B*
14. *LEGISLATION OF CHAMBER OF ASSOCIATIONS AND UNIONS*
   - *Associated volume: 14*
15. *LABOUR LEGISLATION – SOCIAL INSURANCE*
   - *Associated volumes: 15^1, 15^2, 15^3, 15B1, 15B2*
16. *AGRICULTURAL LEGISLATION - WATER RESOURCES*
   - *Associated volumes: 16, 16A, 16B, 16Γ*
17. *RURAL LEGISLATION – FORESTS – HUSBANDRY*
   - *Associated volumes: 17^1, 17^2, 17A, 17B*
18. *PRESS – MEDIA – TOURSIM*
   - *Associated volumes: 18, 18A*
19. *MERCHANT SHIPPING*
   - *Associated volumes: 19, 19A, 19B, 19Γ*
20. *PORT LEGISLATION*
   - *Associated volumes: 20, 20A*

21. *PUBLIC TRANSPORT*
    - *Associated volumes: 21, 21A, 21B, 21Γ*
22. *POST OFFICES – TELECOMMUNICATIONS*
    - *Associated volumes: 22, 22A*
23. *PUBLIC CONSTRUCTIONS – ENGINEERS – URBAN PLANNING*
    - *Associated volumes: 23, 23A, 23B, 23Γ*
24. *FINANCIAL ADMINISTRATION – ENVIRONMENT*
    - *Associated volumes: 24, 24A*
25. *PUBLIC ACCOUNTING*
    - *Associated volumes: 25, 25A*
26. *CURRENCY - PUBLIC PROPERTY*
    - *Associated volumes: 26, 26A*
27. *DIRECT TAXATION*
    - *Associated volumes: 27, 27A*
28. *INDIRECT TAXATION*
    - *Associated volumes: 28, 28A*
29. *COURT OF AUDITORS - PUBLIC PENSIONS*
    - *Associated volume: 29*
30. *CUSTOMS LEGISLATION*
    - *Associated volumes: 30, 30A, 30B*
31. *SCIENCES AND ARTS - ACADEMY – ANCIENTS*
    - *Associated volumes: 31, 31^{A}, 31B, 31Γ*
32. *EDUCATIONAL LEGISLATION (1^{ST}-2^{ND} GRADE, TECHNOLOGICAL INSTITUTIONS) – SPORTS*
    - *Associated volumes: 32, 32A, 32B, 32Γ*
33. *ECCLESIASTICA LEGISLATION*
    - *Associated volumes: 33, 33^{A}*
34. *HEALTH LEGISLATION*
    - *Associated volumes: 34, 34^{A}, 34B*
35. *SOCIAL WELFARE*
    - *Associated volumes: 35, 35^{A}*

36. *NATIONAL DEFENSE – ARMY*
    - *Associated volumes: 36, 36A, 36B, 36Γ, 36Δ*
37. *NAVY*
    - *Associated volumes: 37, 37A*
38. *AIR FORCE - CIVIL AVIATION*
    - *Associated volumes: 38, 38A, 38B, 38*
39. *PUBLIC ENTITIES - SOCIAL SECURITY FUNDS*
    - *Associated volumes: 39, 39A, 39B, 39Γ, 39Δ, 39E*
40. *DIPLOMATIC LEGISLATION - INTERNATIONAL ORGANIZATIONS*
    - *Associated volumes: 40, 40A*

3.3 Summary

In this Chapter, we presented the initiative of Pantelis Raptarchis to create a legislative collection consisting of 111 volumes that are divided based on 40 thematic sections. Next, we discussed about Raptarchis documents' structure and we pointed out the main differences we observe compairing to the legal resources that are published in Greek government's gazettes. Finally, we listed all the thematic topics that can we found in Raptarchis' collection.

4. IMPLEMENTATION OF RAPTARCHIS DOCUMENTS' PARSER

In this chapter we will present the vital component of our initiative to parse the legislative collection of Continuous Legislation Code. Having a large set of error-prone volumes, which contain legal resources from the beginning of the Greek State that are incosistent with the encoding, makes our parsing process quite complex and challenging. Because of this, we had to plan and choose the optimal approach based on the form and the type of our data. Accordingly, we will explain the stages of preprossesing of the initial legislative documents until the population of the final text files and how we modeled legal resources and the relationships between them and their subdivisions. Finally, we will look into the extraction methods we used to identify the structural components and metadata for the legal documents.

4.1 Preprocess of documents

4.1.1 Conversion flow of initial documents

The initial collection we had in our possession consited of 111 legislative volumes distributed in 40 thematic sections. In the collection that was donated from Pantelis Raptarchis to the Greek State and consists the Continuous Legislation Code project, each volume had been encoded in .word file format. Because of the additional metadata contained in a .word file (e.g. font size, style, page margins), we wanted a simplier version for our parser and to accomplish that we had to convert them to plain text files. In order to achieve the best valid results in this conversion process, we exploited two reliable tools called Apache Poi[1] and iText[2], respectively.

Both Apache Poi and iText are tools for file manipulation but each one refers to different range of file formats. Apache Poi is a Java project providing a set of libraries for creating and modifying files in Microsoft Office formats. On the other hand, iText is a Java library for manipulating and editing PDF files, providing support for the most advanced PDF features. Therefore, all the volumes are converted from .word files to PDF files with the help of Apache Poi and subsequently to plain text files, using iText. The conversion flow can be seen in Figure 10.

Figure 10: Conversion flow of volumes

[1] See https://poi.apache.org/

[2] See https://itextpdf.com/en

4.1.2 Population of final text files

As we introduced in Chapter 3, each Raptrarchis's legislative volume is related to a thematic section and contains a large amount of legal resources, being characterized by their thematic consistency. In order to achieve better parsing results, after the conversion of initial documents to plain text files, we attempted to divide each volume to its individual legal resources, creating a collection of text files per legal resource.

Taking advantage of the Raptarchis volume's inner structure, with the legal resources been indexed to thematic sections, we created an autonomous Java program to split the legislative volumes into legal resources. Our program iterated over the collection of the volumes and with the use of Java Regexes[1], it divided the context into chapters and subsequently it separated each chapter into thematic subjects. After gathering the sets of legal resources for each thematic subdivision, we proceeded on extracting each individual resource by identifying the start and the end of it. We defined the context of an individual resource as the starting point of the numbered fragment that contains the resource's metadata and the ending point as the next sequential metadata section.

Following this procedure, we end up having sorted all the legal resources based on the thematic hierarchy they are under. At this point of program's runtime, we have reached our main goal on identifying and extracting each legal resource, knowing in which volume, chapter and subject it belongs to and its medatada fragment. Next, having gathered all the needed information, for each legal resource we transfer its contex to a text file applying a xml based format that matches the following template:

<vol> *VOLUME'S IDENTIFIER* **</vol>**

<chap> *CHAPTER'S IDENTIFIER* **</chap>**

_{** *SUBJECT'S IDENTIFIER* **}

<start_law>

<law_info>

METADATA FRAGMENT

</law_info>

MAIN CONTEXT

<end_law>

Finally, the text files are saved in a hierarchical directory consisted of folders that are organized in a tree structure. This directory is composed of four layers. The first layer is the root folder that represent the legislative volume. The second layer contains all the thematic chapters of the volume and the third layer includes the thematic subjects of the individual chapters. The final layer consists of the generated text files with legal resources' context. In Figure 11 we present the structrure of the hierarchical directory that we implemented. Applying the preprocess method we achieved to populate 55.515 individual legal documents that they have been distributed based on their thematic topic and indexted by our tree structural directory.

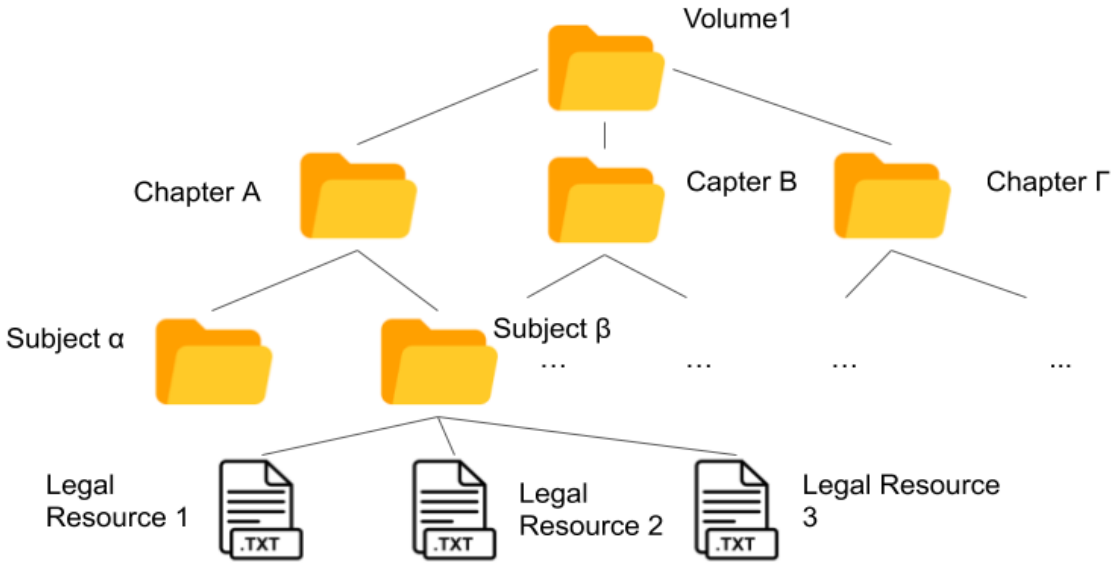


Figure 11: Hierarchical directory for Raptarchis' documents

4.2 Specifications and design of parser

As we explained in Chapter 3, Raptarchis' collection includes many legal resources that are chronologically old without having a stable encoding and even modern Greek legislation contained in the volumes does not always comply with the proposed encoding. Consequently, these encoding variations are making our parsing approach quite difficult and challenge us to come up with new ideas to overcome these problems by designing a parser that will fulfill our needs and model the Greek legislation precisely.

4.2.1 Parser's specifications

As for the implementation of the parser, the programming language that was chosen was version 8 of Java. In order to exploit the hierarchical structure of the components that consist a legislative resource, we wanted to use an object-oriented programming language. The parser is implemented in a Java project named 'Raptarchis_Parser' and contains a main Java class and five Java packages as following:

- **Api**: Package that contains the parser's interfaces and their implementations.
- **Divisions**: Package that contains all the Java classes that represent legislative components
- **Enums:** Package that contains all the Java enumerations that the parser uses.
- **Helpers:** Package that contains all converters and helpers of the parsing process.
- **Loggers:** Package that contains all the classes that offer logging functionality.

For input, Raptarchis parser expects two arguments. The first must be the string that represents the path of the location in the file system where the directory which contains the text files we want to parse is stored. The second must be the path of the location where we want the produced RDF file to be stored. For the input directory there are no constrains, which means that is acceptable to contain from a single to a large amount of text files, even containing a complex sub-directory system as we presented in Figure 11. The only requirement is the format of the text files to follow the principles of xml-based format we proposed in section 4.1.2. The running command for the parser is the following:

java Raptarchis_Parser **-i** path_of_input_directory **-o** path_of_output_file

4.2.2 Modeling Raptarchis' legislative documents with Java programming language

In previous chapters we introduced both the general directives for legislation encoding as they were proposed by the Greek Central Committee of Encoding Standards and the peculiarities that Raptarchis' documents have. Unquestionably, the inalterable component of Greek legislation is its structural hierarchy that consists of legal components such as books, sections, articles, etc. Having chosen an object-oriented programming language, it is very beneficial as it makes our parsing process more direct and comprehensible.

Our main goal, in order to populate the final dataset that is going to feed Nomothesia platform, was to analyze and decompose each legal resource as they were produced by the initial preprocess. The modeling approach we applied was to map all the legal resource's subdivisions, including itself, into Java Entity Objects. Therefore, each legal subdivision corresponds to a Java class that contains attributes that represent the division's real-life characteristics and methods that add functionalities to these components. *LegalResource*, our most important class, represents real-life legal documents and contains attributes such as publication date, the date it came in force, number of Government's gazette it was published, etc.

In our entity model, we have defined two main classes, the *Division* and *Passage* classes, to represent legislative subdivisions. The class *Division* is an abstract class that is being inherited by all subclasses that correspond to thematic subdivisions of a legal resource. With the term 'thematic subdivisions' we refer to subdivision fragments such as books, parts, sections, chapters, articles and paragraphs, with each of these fragments having their own subclasses in the parser's model. *Division* class provides attributes for fragment's Uri, numerical identifier inside the hierarchical structure it belongs to, fragment's text, title and a *Division* type attribute that contains its parent component. Likewise, the class *Passage* represents the legislative subdivisions of a paragraph and is been used as the inheritable class for components such as lineas (Latin word for lines) and indents (cases). *Passage*, like class *Division*, provides attributes for fragment's Uri, initial text and an identifier representing the component's order inside the hierarchy.

In Section 2.1.1 of Chapter 2, we explained that each legal component is divided into legislative subdivisions based on the hierarchical constrains we presented and can be seen on Figure 1. Accordingly, each subclass in our model that extends the *Division* class, except *Paragraph* class, contains a list of *Division* classes that represents its subdivisions. Having a list of 'children' fragments and an attribute that points to the class of parent-division on each model's class, we achieve the creation of a hierarchical tree structure of legislative

components, giving as the ability to iterate over the components during parser's runtime. Because a real-life paragraph may consist of lineas, indents or modifications, our *Paragraph* class contains a list of *Passage* classes and a *Modification* class. A *Passage* can be either a *Linea* or a *Case* class. The *Linea* class belongs to the last layer of our model's hierarchy and it cannot be divided into further components. It represents the legislative lineas which are the building blocks of paragraphs and indents, containing the final text of the parsing resource. On the other hand, the *Case* class refers to indents and contains a list of *Passage* classes that may be divided into sub-indents (sub-cases) or lineas. Finally, the *Modification* class represents legislative modifications and contains information about the amended resource and the inserted or modified text.

Our goal is to reach to a parsing stage where we have identified all the legal resource's articles and from that point to proceed to the extraction of the final passages. Both in Greek Government's gazette and in Raptarchis' collection, it is possible to come up to cases where legal documents have incomplete hierarchical structure. That means that articles may be missing from division's hierarchy or even the order of some components is violating the constrains of hierarchical layers that we introduced in Section 2.1.1 of Chapter 2. For cases that match this concept, Nomothesia ontology provides the class *Container* which is represented by our parser's model with the java class *Container* and it is used as a text wrapper for unidentified legal resource's fragments. In Figure 12 we present the class diagram of Raptarchis parser's model in UML.

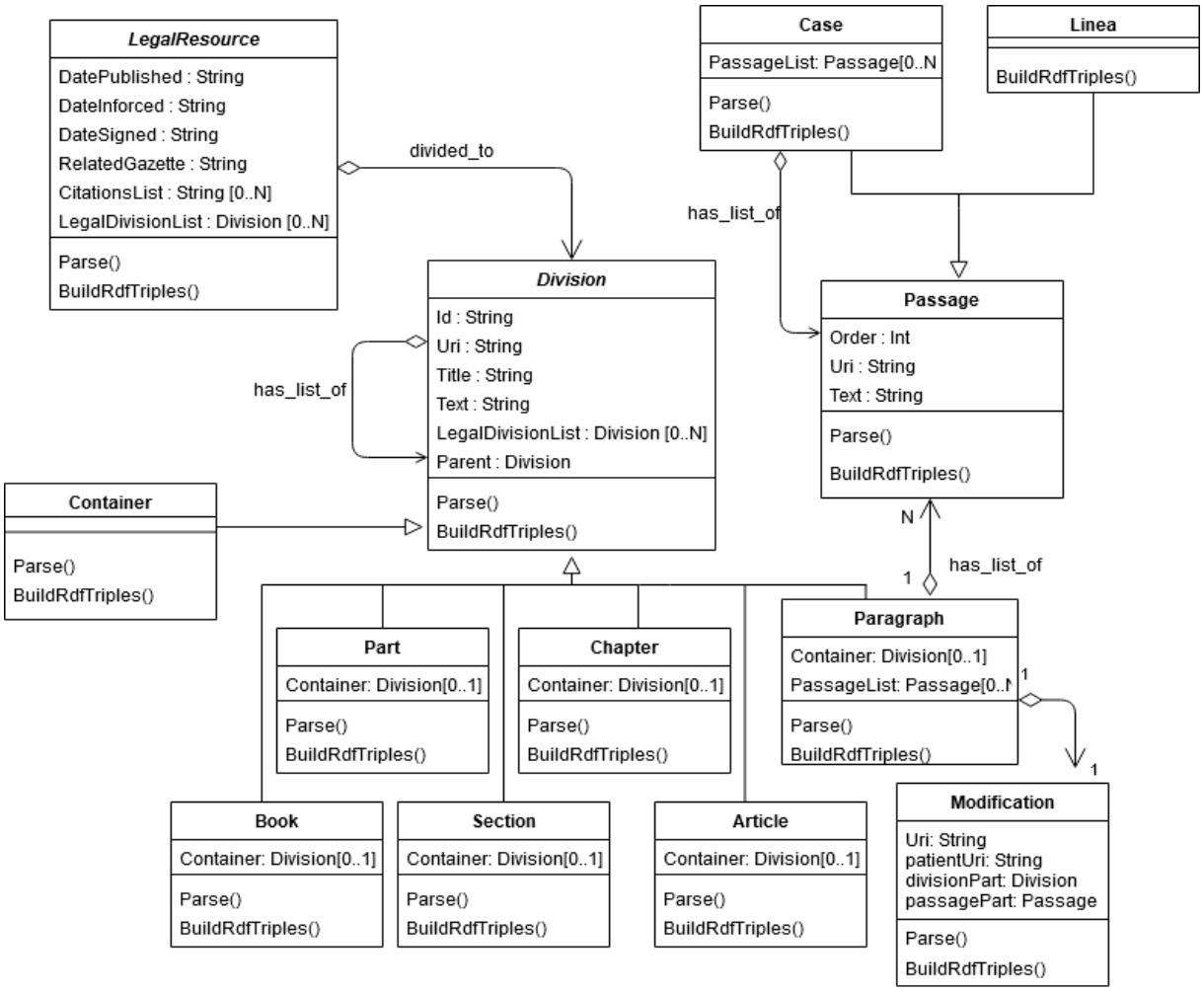


Figure 12: Class diagram of Raptarchis' parser

4.3 Parsing stages of a Legal Resource

The preprocess of the initial text documents (.doc) produced over 55.515 plain text files containing a legal resource per file. In order to analyze the files semantically and produce the final RDF dataset that is going to enrich Nomothesia platform's knowledge, we had to iterate over legal resources individually and apply certain parsing methods. Firstly, each legal resource is mapped to class *LegalResource* and the file's text is assigned to attribute *text* of the class. Subsequently, the text is going through a parsing procedure that can be grouped in the following stages:

a) Extracting legal resource's information and metadata (e.g., type, identifier, publication date).

b) Identification and extraction of citations.

c) Split of text into legislative subdivisions, identification of their metadata (i.e., title) and creation of the hierarchical division tree.

d) Decomposition of articles into paragraphs and paragraphs into final lineas by identifying existing cases or modifications.

e) Validations on produced hierarchical structure.

f) Population of RDF triples.

In summary, Raptarchis parser reads the initial xml based text file and with the appropriate use of the defined tags extracts the number of the volume that had been published, the chapter's and subject's number, the metadata fragment and the core text fragment of the legal resource. From the metadata fragments, the parser identifies the legal type of the legislation along with the unique identifier, the publication date, the signature date, the date it came into force and commonly the number of the government's legislative gazette that was included in.

After the extraction of the metadata, the parser tries to identify the existence of citations or title and splits the core fragment of legal resource into two parts. The first contains the title and the citations and the second the legal subdivisions (e.g., chapters, articles). If citations or title have been detected it extracts the corresponding information, otherwise, it proceeds directly to legal division separation. Raptarchis parser creates a subdivision tree structure where its nodes are classes of *Book*, *Part*, *Section*, *Chapter*, *Article*, *Paragraph* or *Case* types and the tree's leaves can only be either *Linea* or *Container* classes.

Immediately after identifying every resource's components, it validates that the produced structure does not violate any hierarchical constrains. Finally, the parser iterates overs each component of the tree and generates the final RDF triples that is being saved to the pointed text file. In Figure 13 we can see the method *parse* of *LegalResource* class that initiates the previous parsing stages.

```
public void parse() {

    extractLegalResourceInfo(lawInfo);
    buildUri();
    if (uri == null || legalType == null || (legalResourceId == null && datePublished == null)) {
        Custom_Logger.appendlnProblematicId( msg: lawInfo + " ----> " + " UNKNOWN LEGAL RESOURCE  "
                + ParsingLegalResourceInfo.getRaptarchisVolumeInfo());
        return;
    }
    divideLaw(text, citationsAndTitleContainer, coreLawContainer);

    StringBuilder remainText = new StringBuilder();
    parseTitleAndCitations(remainText);

    parseCoreLaw();

    if ((legalDivisionList==null || legalDivisionList.size() == 0) && remainText.length() > 0) {
        if (legalDivisionList == null)
            legalDivisionList = new ArrayList<Division>();
        legalDivisionList.add(new Container(remainText.toString(), id: "1", parent: this));
        parseLegalDivisionList();
    }

    if (containsInvalidDivision)
        correctDivisionHierarchy();

    if (InvalidArticlesHelper.containsInvalidArticles())
        InvalidArticlesHelper.correctInvalidArticles();

    buildRdfTriples();

    legalDivisionList.forEach(Division::buildRdfTriples);

}
```

Figure 13: parse method of LegalResource class

4.4 Extracting Legal Resource's fragments

Previously, we presented the stages of the parsing flow that each text file is been through. In this Section we are going to delve into the basic parsing stages from a technical scope and explain the manipulation methods we chose in order to achieve the best possible results. In our parser, the implemented methods make use of the predefined tools for text manipulation that Java programming language offers. Java provides an API called Regex that defines patterns for advanced searching and extraction techniques upon strings. In

Raptarchis' parser we exploit two of the basic objects provided by Regex API, the Pattern[1] and the Matcher[2] objects. The Pattern objects is a compiled representation of a regular expression [8], while Matcher object is the engine that interprets the pattern and performs match operations against an input string.

4.4.1 Metadata and citations extraction

During the initial preprocess where the Raptarchis' legislative volumes were converted into plain text files, some of the distinguishable components of the legal resources were wrapped up to custom xml-based tags. One of these components is the metadata fragment that is been enclosed in the following tags,

<law_info>

METADATA FRAGMENT

</law_info>

From the metadata fragment we can extract the legislation's type, its unique identifier, the publication date and often the enforcement date and the number of legislative gazzete it belongs to. In Figure 14 we can see how that information is represented in the legal resource's metadata section using as example the section of the Law 3068 of 2002.

Figure 14: Metadata fragment of legal resouce

For the extraction of the identifier and the gazette's number we use as a regular expression, specified as a string, the predifined character class *d* that the Regex API provides and matches a range of digits (0-9). On the other hand, the extraction of the publication date and enforcment date, was more challenging due to the variations on months' abreviations that legal resources use. For this purpose we created the DateFormatHelper class which converts every string to the *MM/DD/YY* date format, containing only Arabic numerals, substituting any previous date format used.

We have already explained that the Raptarchis' legislative collection consists of different legislation types and a part of them has not even been included in Nomothesi@'s OWL ontology. We tried to group the legal types with common or related characteristics (i.e., decisions per legal entity) by creating regular expressions with the combination of string literals and character classes. As a result, Raptarchis parser can support a large variety of types that can be seen in Table 3 along with their mapping codes for use on URIs.

[1] See https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html
[2] See https://docs.oracle.com/javase/7/docs/api/java/util/regex/Matcher.html

Table 3: Compatible legislation types and code

Types of Legislation	Codes
Legislative Presidential Decree	lpd
Presidential Decree	pd
Legislative Decree	ld
Royal Decree	rd
Regulatory Decree	regd
Degree	dg
Decision	dec
Regulatory Provision	regprov
Provision	prov
Act Of Ministerial Cabinet	amc
Legislative Act	la
Act Of Bank Of Greece	abg
Act	act
Announcement	ann
Resolution	res
Circular	cir
Protocol	pro
Constitution	con
Agreement	agr
Concordat	conc
Civil Code	civilcode
Penal Code	penalcode
EU Directive	eudir
Parliament Regulation	pareg
GOC Regulation	gocreg
Mandatory Law	ml
Law	law

After metadata extraction, the parser proceeds to identify any existing citation. Mainly, citations are placed at a certain point inside legislation body and are presented as a numbered list. In parser's implementation, we define the fragment that contains the citations as the text region between the resource's main title and its first subdivision. In addition, citations are always being introduced in text using certain keywords that denote relationship with other legislative resources and are always end with keywords in imperative form. The parser groups these keywords with the use of regular expressions and applies pattern matching in the possible citation area. After it has successfully matched the keywords, we split the extracted text using the string regex *"(?<=\s)(?=\d+[\.\)])".* This regex is trying to match a whitespace followed by number and a right parenthesis inside a text region and splits the text into a string list using this pattern as a delimiter. After the creation of the list, we filter the elements of the list to validate that the citations are in an ascending order based on the numbered they were listed.

4.4.2 Hierarchical divisions extraction

The next step of the parsing process is to divide the core component of the legal resource into subdivisions. Because of the polymorphic design of the parser's model, we wanted to make our implementation as simple as possible. Starting from *LegalResource* class, inside the *parse* method we presented in Figure 13, the parser's flow triggers the *splitToSubDivisions* method that initiates the division procedure. After the extraction of its children-fragments, it iterates over its subdivisions and invokes their corresponding methods for subdivision extraction until the creation of the final hierarchical tree structure. All the classes in our model, except *Linea* and *Container* which are the building blocks of the classes' tree, contain a list of elements that can be either *Division* or *Passage* type. This means that during the extraction process we do not know the final form of the list, except that the possible types can only be classes that extend one of the two class types. Therefore, there was a need to create a flexible algorithm and separate it from the object structure on which it operates. For that purpose, we designed an implementation based on the Visitor pattern [9] that give us the ability to work across several independent class hierarchies. The overall design can be seen in Figure 15.

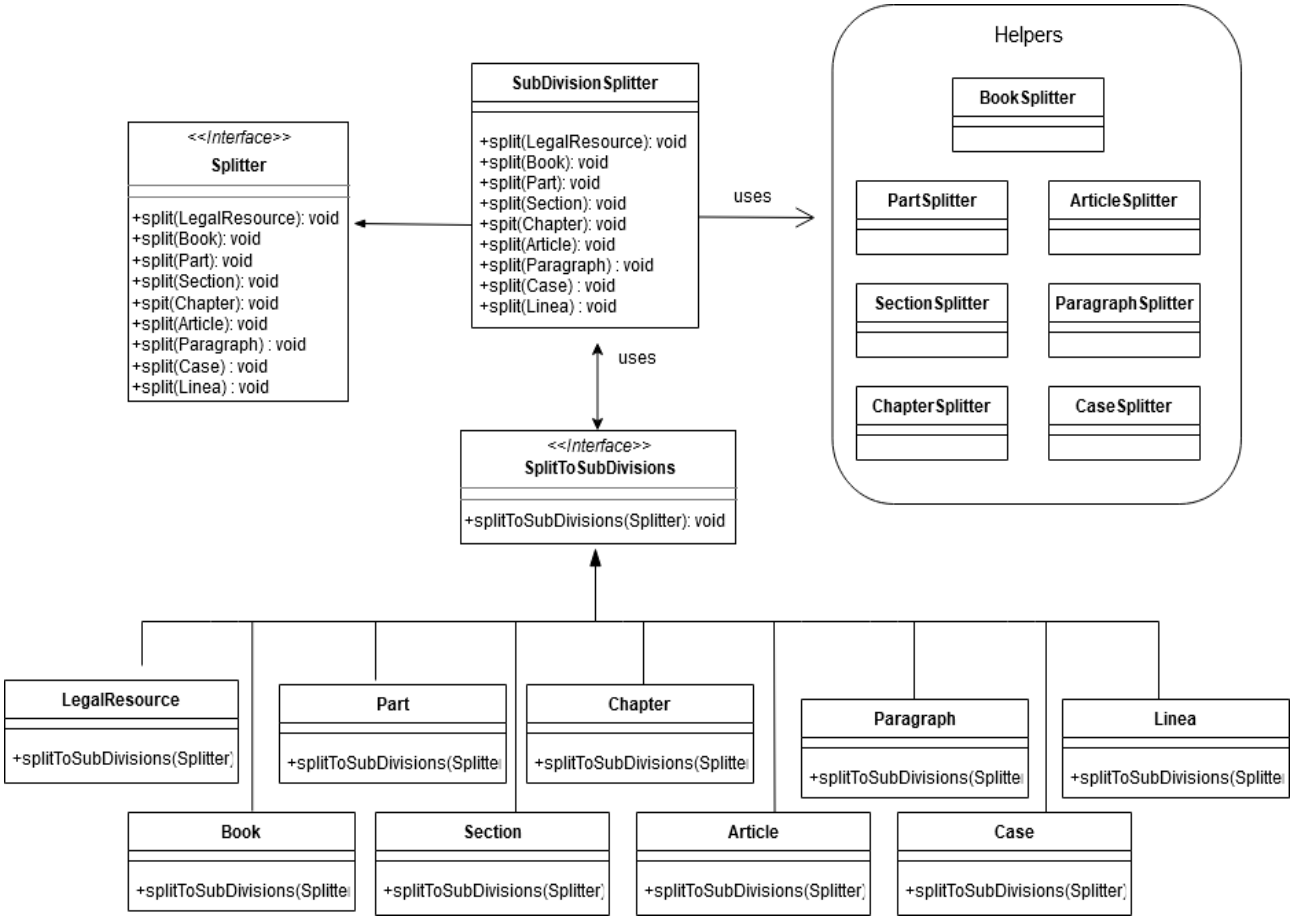


Figure 15: Class diagram of subdivision extraction procedure

Each of our concrete classes implement the *splitToSubDivisions* method which expects an object that implements the *Splitter* interface as parameter. All the implemented *splitToSubDivisions* methods in concrete classes have the following form:

```
public void splitToSubDivisions(Splitter splitter) {
        splitter.split(this);
}
```

Finally, we have created the *SubDivisionSplitter* class that extends the *Spliiter* interface and contains all the main methods' implementations of the designed splitting algorithm. This class is using a set of helper classes that provide extra functionality and converting methods about Greek legislation's enumerations. For better understanding the parsing process of the legal subdivisions we succinctly present the basic stages of the flow which is almost identical for all classes except *Paragraph*, *Case* and *Linea* on which we will present their flow in the next section of this chapter. The main parsing steps are the following:

1. With the help of regular expressions, we are trying to match in the text section all the patterns that legislation uses to present legal subdivisions.

2. We split the text into fragments using the regular expressions as a delimiter and we run validations in order to be certain that the extracted divisions are legitimate.

3. We iterate over the extracted divisions in order to check for existing title and identify the division's ordinal which is necessary for the creation of its URI.

4. The extracted list is assigned to its parent's class and the extraction procedure continues to each individual component until we reach classes that cannot been divided further (i.e., *Linea*, *Container*).

In Section 3.2.2 we introduced one of the difficulties we face on Raptarchis' legislation collection due to the variations on divisions' numerals that makes their identification very difficult. In order to overcome this challenge, we created a universal regular expression that is capable to match the following types of numerals:

a) Arabic numerals (e.g., *1, 2, 3*).

b) Latin numerals (e.g., *i, ii, iii*).

c) Ordinal numbers in Greek language (e.g., *Πρώτος, Δεύτερος, Τρίτος*).

d) Greek alphabetical numbers (e.g., *Α, Β, Γ*).

Along with the regular expression, we have created a converter that transforms each of the last previous types to the corresponding value in Arabic numeral. For example, the Latin number *i* is converted into 1 and the Greek ordinal "*Τρίτος"* to 3. Both the regular expression and the converter belong to the implemented *LegislationNumericalHelper* class.

4.4.3 Extraction of paragraphs and their building components

After we have identified the articles of the legal resource, we proceed on dividing it into paragraphs. As we have already mentioned, articles consist of at least one paragraph based on the thematic consistency. Article's paragraphs are numbered using Arabic numerals and are easily distinguishable since they are introduced in the text with their first line been indented compared to the rest text and starting in a new line. Because of the conversion of the initial documents from formatted files to plain text files, the paragraphs' indents are eliminated and in our parsing process we can only identify them by the new line and their number.

For our parsing approach, we divided the initial article's text in fragments based on the regular expression *\n(?=\d+[\.)])*. Using this pattern, the java's regex mechanism tries to match a new line character followed by at least one number that ends with either a dot or a right parenthesis. With this technique we split the article's text into numbered fragments. However, this procedure is not flawless due to the form of our text files. Inside our files, the legal resource's sentences may have been broken into multiple lines and with break line characters between them. Therefore, we cannot rely on the fact that each new line that starts with a number followed by a dot or a right parenthesis is the start of a new paragraph. For this reason, we have implemented a method named *validateSequenceOfParagraphs* that takes as input a list of possible paragraphs and tries to detect the valid paragraphs. The method's algorithm relies on the fact that paragraphs are numbered in sequential order. For example, after paragraph one we except only to see paragraph with number 2. For better understanding, we present the algorithm in pseudo code.

```
expectedNumber = 1
List finalParagraphs
List initialParagraphs = splitInitialsParagraphs()
for each index in initialParagraphs
        number = initialParagraphs[index].getNumber()
        if number == expectedNumber
                finalParagraphs.add(initialParagraphs[index])
                expectedNumber = expectedNumber + 1
        else
                finalParagraphs.appendStringToLastElement(initialParagraphs[index])
```

After we have finished the paragraphs' identification, we continue the extraction of their building components. We have defined two cases about paragraph's context and for each one we apply different parsing approach. In the first case we include the paragraphs that contain legal modification and in the second those who do not. The parsing process about first case we are going to present it in the next Section of this Chapter. Paragraphs that belong to the second category are consisting of lineas and legislative cases. Same as paragraphs, because of the conversion of files to plain text we have lost the ability to

recognize them from their indentation inside the text. We can only be certain that a legislation case is been introduced in a paragraph by starting in a new line and being numbered with Greek lower-case alphabetic characters (e.g., α, β, γ). The algorithm is trying to detect that pattern and splits the paragraph into sub fragments with each of one containing a case. If it does not find cases proceeds directly to lineas' extraction. Subsequently, each case is being searched in order to determinate the existence of further legislative sub-case division. We can easily identify sub-cases from their double lower-case Greek alphabetic enumeration (e.g., αα, αβ, βα). Both legislative cases and sub-cases are represented with *Case* class in our parser's model and as we mentioned in Section 4.2.2 is not our final class in the hierarchical tree we are building. For this reason, we must divide the cases and sub-cases, if they exist, into lineas that include the actual sentences of the text. In order to extract the text's lineas we must identify where a sentence is ending and where the next is starting. In the parser's implementation we split the text into individual lineas having in mind the following basic criteria:

a) A linea always ends with a punctuation mark (i.e., ., :).

b) Linea's last word cannot end with a Greek consonant letter (e.g., γ, δ, ζ).

c) A linea must starts with a capital letter.

Finally, in order to minimize the fault possibility in the lineas' extraction procedure we filter them based on a blacklist that contains known legislative abbreviations that the legal resources often use and we must not consider the as the end word of a sentence.

4.4.4 Identifying modifications

As we explained in Section 3.2.2, most of the modifications in Raptarchis' legislative collection are already being integrated inside patient-resource's text. Nevertheless, we have implemented methods in order to identify the few modifications that may be contained in resources. We have come across two modification types, the insertion and the modification. During paragraph's parsing, we search in its text for patterns that indicate modification's existence. The modifications are being introduced in text fragment using a metadata sentence that indicates the legal component that is being amended. After this sentence follows the verbatim legal component that is going to be inserted on the patient-component or replace it. For a better understanding we are presenting the parsing approach of paragraph tree of article 38 of Law 1806/1988 with the following text:

3. In article 2 of Presidential Decree 360/1985 (Government Gazette 129) is added paragraph 3 as follows:
«3. By decision of the Minister of National Economy, following the opinion of the Securities and Exchange Commission, it is mandatory for public limited companies whose shares have been listed the obligation to publish reports on Greek language for their activity and results in the first quarter of each year and in the first nine months. From the Bank of Greece and the Funds are exempt from this obligation. The time of publication of these reports shall be set by the above decision. The provisions of the following Articles. 4, 5, 6, 7 and 8 apply to the above reports».

The combinations of words "*added"* and *"as follows:"* in the first sentence of paragraph make the parser to assume that we have a case of modification and more precisely an insertion. From this metadata sentence, with the use regular expressions we built the patient-resource's Uri and the Uri of the components is going to be added. For example, the corresponding Uris will be:

Patient-resource component's Uri : *http://legislation.di.uoa.gr/pd/1985/1806/article/2*

Added component's Uri : *http://legislation.di.uoa.gr/pd/1985/1806/article/2/paragraph/3*

The actual text of the modification that follows the metadata sentence, we wrap it up with one of the two superclasses (i.e., *Division*, *Passage*) of our model based on which of these classes the added component's class is subclass. In our example the added component is a paragraph, so the corresponding class in our model is *Paragraph* class. As a result, we are going to wrap it up with *Division* class because *Paragraph is subclass of the Division.* Consequently, we can proceed to further division of the component until we extract its building components.

4.5 Summary

In this Chapter, we explained the preprocessing stages of the initial documents and how we extracted the included legal resources, creating a hierarchical folder directory that contains legal resources per text files. Subsequently, we introduced the specifications and the design of the parser we created in order to extract the legal resources' data that we need for Nomothesi@'s dataset. Next, we proposed the modeling approach that our parser was based on and we presented the basic parsing stages of an individual legal resource. Finally, we delved into each parsing stage of the parser and more precisely, we analyzed the procedures of the metadata extraction, the hierarchical division extraction, the modifications' identification and the extraction of paragraphs and their subcomponents.

5. RESULTS AND DATASET

5.1 Parsing results

At the beginning of this thesis, we had 111 legislations volumes divided based on their thematic section. After the initial preprocess, we achieved to extract each individual legal resource that was included in Raptarchis' collection and store it to different files, producing 55.515 legislative components. Each of these resources was processed from the implemented parser and went throught severals parsing stages creating their hierarchical tree structure. Finally, based on that tree we were able to produced the corresponding RDF triples. In the end of this parsing effort, we managed to populate 3.395.419 RDF triples and to recognize 27 different types of legislation.

5.2 Demostration of dataset's population

In Section 4.3 we introduced the parsing stages of a legal resource. The final stage is about populating the final RDF triples based on the hierarchical tree structure we have built. Because showing the RDF triples would be very inconvenient, we will make an example using the RDF graph of the produced RDF triples of Royal Decree 135 of 1956 as it follows:

ΒΑΣΙΛΙΚΟΝ ΔΙΑΤΑΓΜΑ 135 της 11/19 Δεκ.1956

Περί παρατάσεως της διαρκείας προστασίας των τέως εχθρικών σημάτων και της προθεσμίας μη ισχύος ενίων λόγων διαγραφής τούτων.

Έχοντες υπ' όψιν τας διατάξεις της παρ. 4 του άρθρ. 26 του Ν.Δ. 1138/1949 «περί των εχθρικών περιουσιών» ως και την υπ' αριθ. 648/1956 γνωμοδότησιν του Συμβουλίου της Επικρατείας, προτάσει των Ημετέρων επί των Οικονομικών και του Εμπορίου Υπουργών, απεφασίσαμεν και διατάσσομεν:

Άρθρο μόνο

1.Η διάρκεια της προστασίας των σημάτων, περί ων το άρθρ. 1 του Β.Δ της 12/13 Σεπτ. 1951 «περί προστασίας τέως εχθρικών σημάτων κλπ.», παρατείνεται δυνάμει του παρόντος και άνευ ετέρας διατυπώσεως μέχρι της 31 Δεκ. 1958.

2.Παρατείνεται μέχρι της 31 Δεκ. 1958 η εν άρθρ. 2 του, περί ου η προηγουμένη παράγραφος του παρόντος άρθρου, Β.Δ/τος προθεσμία μη ισχύος των εν τω άρθρω τούτω αναφερομένων λόγων διαγραφής, περί ων το άρθρ. 15 του Α.Ν. 1998/1939, ως το άρθρον τούτο αντικατεστάθη δια του άρθρ. 8 του Νόμ. 3205/1955.

3.Αι κατα τας δύο προηγουμένας παραγράφους του παρόντος άρθρου παρατάσεις περιλαμβάνουσι μόνον σήματα περιελθόντα τω Ελληνικώ Δημοσίω βάσει των διατάξεων του Α.Ν. 1530/50, ως και τοιαύτα περιελθόντα αυτώ κατόπιν επεκτάσεως των διατάξεων τούτων δια Β.Δ/των εκδοθέντων συμφώνως προς το άρθρ. 16 αυτού, δεν περιλαμβάνουσιν όμως εκ των σημάτων τούτων εκείνα, εφ' ων κατά την δημοσίευσιν του παρόντος δεν διατηρεί εξ οιουδήποτε λόγου δικαιώματα το Ελληνικόν Δημόσιον. Εις τον Ημέτερον επί των Οικονομικών Υπουργόν ανατίθεμεν την δημοσίευσιν και εκτέλεσιν του παρόντος Δ/τος.

In Figure 16 we present the RDF graph of the produced RDF triples from Raptarchis parser.

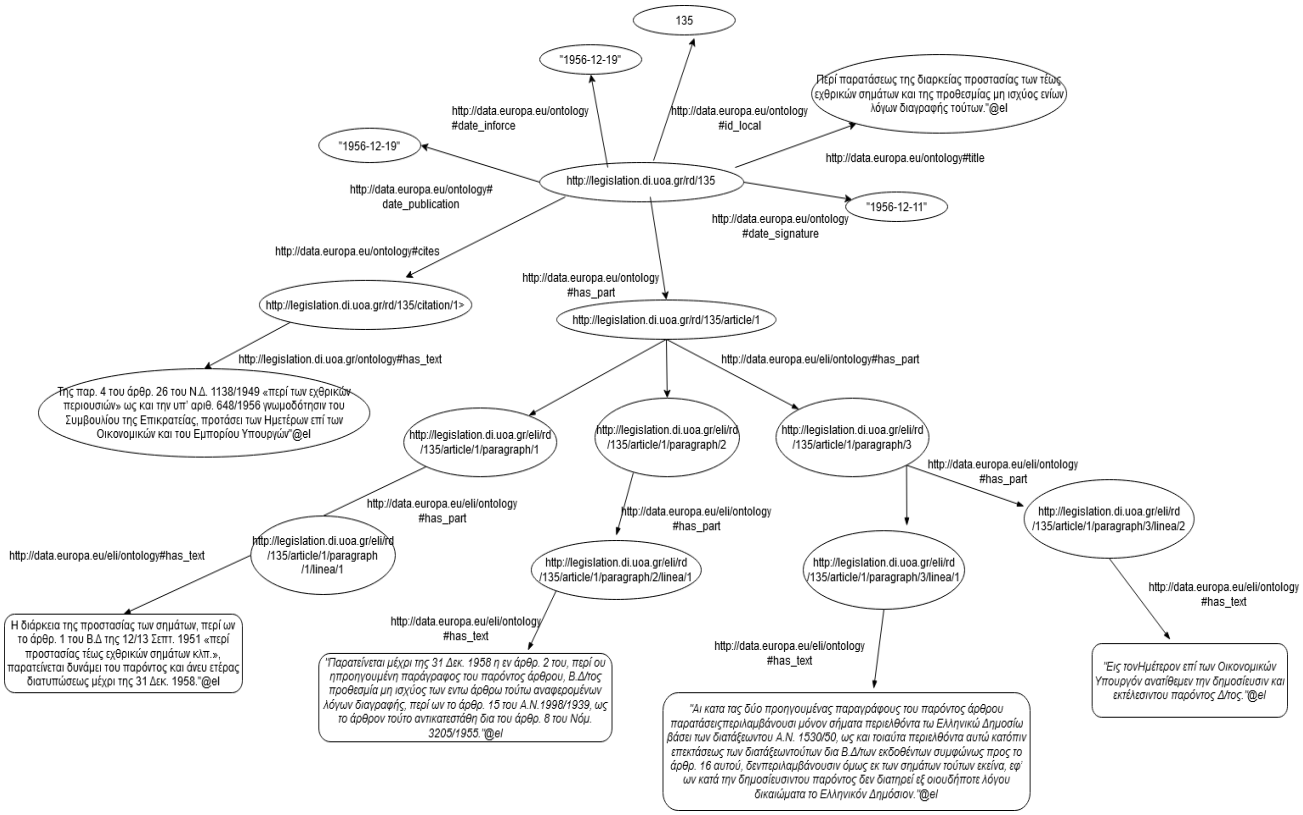


Figure 16: RDF graph of Royal Decree 135/1956

5.3 Summary

In this Chapter, we cited the results of the parsing process that are related to the produced RDF triples and the types of legislation that we recognized during the parsing. Finally, we demonstrated the RDF graph of the Royal Decree 135 of 1956 as it would be created from Raptarchis' parser.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusions

During this period of technological innovation and internet's heyday, Nomothesi@ platform must be kept updated in order to offers its users reliable legislative knowledge and advanced searching capabilities. The main idea of this thesis was based on the need of data enrichment with the legislative resources that Raptarchis' collection includes. Depending on previous works on Nomothesi@ platform, we used its implemented infrastructure and we exploited the provided technological tools, such as the Nomothesi@ OWL ontology.

We processed the Raptarchis' collection and divided it into individual resources based on their thematic topics and created a serviceable legislation index. Subsequently, we introduced a modeling approach for Greek legislation using an object-oriented programming language by deconstructing its building components and defined the relationships between them. In order to achieve the best results, we tried to design efficient algorithms for legislative divisions' recognition and effective extraction techniques. After the application of a sequence of parsing stages we produced approximately 3,39M RDF triples.

Concluding, although the dataset population of Raptarchis' in going to be exploited directly by Nomothesi@ platform, this parsing initiative aims to a bigger and more abstract idea of contributing in the open liked data of the web that are related to the legislative knowledge. Every person should be able to access and refer to legislative data in order to know their rights and be responsible and active members on the society they belong to.

6.2 Future work

In Chapter 2 we explained that the Raptarchis' legal documents are divided based on thematic topics and each one is being subdivided into thematic chapters and subjects. After the preprocess of the legislative volumes we presented in Section 4.1.2, the individual legal resources have been sorted based on their thematic consistency. Therefore, the creation of the resources' index can be affectively exploited by tools of Data Analysis sector such as classifiers. Theoretically, training classifiers with Raptarchis' legislative collection can lead to easily categorization of future legislative documents, powering and adding flexibility to searching capabilities of legislative knowledge on the web.

ABBREVIATIONS - ACRONYMS

Table 4: Abbreviations - Acronyms

API	Application Programming Interface
REST	Representational State Transfer
RDF	Resource Description Framework
OWL	Web Ontology Language
SPARQL	SPARQL Protocol and RDF Query Language
URI	Universal Resource Identifier
ELI	European Legislative Identifier
HTTP	Hypertext Transfer Protocol
PDF	Portable Document Format
UML	Unified Modeling Language
XML	Extensible Markup Language

REFERENCES

[1] Greek Central Committee of Encoding Standards, Manual Directives for the encoding of legislation, 2003, http://www.ggk.gov.gr/wp-content/uploads/2010/02/teliko_egxeiridio_odigion_gia_tin_kodikopoiisi_tis_nomothesias.pdf

[2] Greek Central Committee of Encoding Standards, Manual Directives for the encoding of legislation – Final form, 2019

[3] Chalkidis I., Nikolaou C., Soursos P., Koubarakis M, Modeling and Querying Greek Legislation using Semantic Web Technologies, National and Kapodistrian University of Athens, 2017

[4] ELI Task Force, ELI A Technical Implementation Guide, Publications Office of the European Union, 2015

[5] S. G. Phil Archer and N. Loutas.Study on persistent URIs, with identification of best practices and recommendations on the topic for the MSs and the EC, 2012.

[6] Chalkidis I., Nomothesi@: Greek Legislation Platform. B.Sc Thesis. National and Kapodistrian University of Athens, 2014

[7] Association of Greek lawyers, https://www.ethemis.gr/nomothesia.html

[8] Yunyao L., Rajasekar K., Sriram R., Shivakumar V., Regular Expression Learning for Information Extraction. IBM Almaden Research Center, 2008, https://dl.acm.org/doi/10.5555/1613715.1613719

[9] Bruno C. d. S. Oliveira, Meng Wang, and Jeremy Gibbons, The Visitor Pattern as a Reusable, Generic, Type−Safe Component, ACM SIGPLAN Notices, October 2008