



**NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS**

**SCHOOL OF SCIENCE  
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION**

**BSc THESIS**

**A Government Gazette Parser for the Platform Nomothesi@**

**Realnto V. Gkatzi**

**Supervisors: Manolis Koubarakis, Professor UoA  
Ilias Chalkidis, Ph.D. Candidate**

**ATHENS**

**MARCH 2020**



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Συντακτικός Αναλυτής της Δομής του Φύλλου Εφημερίδας της  
Κυβερνήσεως για την Πλατφόρμα Nomothesi@**

**Ρεάλντο Β. Γκάτζι**

**Επιβλέποντες: Μανόλης Κουμπάρκης, Καθηγητής ΕΚΠΑ  
Ηλίας Χαλκίδης, Διδακτορικός Φοιτητής**

**ΑΘΗΝΑ**

**ΜΑΡΤΙΟΣ 2020**

## **BSc THESIS**

A Government Gazette Parser for the Platform Nomothesi@

**Realnto V. Gkatzi**

**S.N.: 1115201100028**

**Supervisors: Manolis Koubarakis, Professor UoA**  
**Ilias Chalkidis, Ph.D. Candidate**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Συντακτικός Αναλυτής της Δομής του Φύλλου Εφημερίδας της Κυβερνήσεως για τη  
Πλατφόρμα Nomothesi@

**Ρεάλντο Β. Γκάτζι**

**A.M.: 1115201100028**

**Επιβλέποντες: Μανόλης Κουμπάρκης, Καθηγητής ΕΚΠΑ  
Ηλίας Χαλκίδης, Διδακτορικός Φοιτητής**

## **ABSTRACT**

Up until recent years, unlike many EU countries, no attempt had been made in order to create a machine readable standard for Greek legislation, therefore advanced legal web services can not be built in Greece. A major effort to contribute to the representation of legal knowledge and its integration into the open data area in Greece, both from a technological perspective and in terms of transparency, is Nomothesi@ platform. The objective of this dissertation is to create a government gazette parser for the Nomothesi@ platform, therefore giving us the ability to enrich the dataset of machine readable Government Gazette issues by parsing all A issues of Government Gazette from 1990 to present, uploaded in the National Printing Office's website and providing it with a RDF/OWL dataset which is the final output of this thesis.

**SUBJECT AREA:** Semantic web, Software engineering

**KEYWORDS:** Legislative Knowledge Identifiers, Open Data, E-Government RDF/OWL Metadata, Universal Resource Identifiers, Government Gazette, Parser

## ΠΕΡΙΛΗΨΗ

Μέχρι πριν από λίγα χρόνια, σε αντίθεση με πολλές χώρες τη Ευρωπαϊκής Ένωσης, δεν είχε γίνει καμία προσπάθεια να δημιουργηθεί κάποιο μηχανικά αναγνωρίσιμο πρότυπο για την ελληνική νομοθεσία με συνέπεια να μην μπορούν να δημιουργηθούν προηγμένες διαδικτυακές νομικές υπηρεσίες. Μια σημαντική προσπάθεια συνεισφοράς στον τομέα της αναπαράστασης νομικής γνώσης και στην ενσωμάτωση αυτής στην περιοχή των ανοιχτών δεδομένων στην Ελλάδα, τόσο από τεχνολογικής σκοπιάς όσο και από άποψη διαφάνειας, είναι η πλατφόρμα Νομοθεσί@. Στόχος της εργασίας αυτής, είναι η δημιουργία ενός συντακτικού αναλυτή δομής του Φύλλου Εφημερίδας της Κυβερνήσεως για την πλατφόρμα Νομοθεσί@ με σκοπό την ανάλυση της δομής όλων των τευχών Α των ΦΕΚ απο το 1990 έως σήμερα, όπως αυτά αναρτώνται στη σελίδα του Εθνικού Τυπογραφείου με τελικό σκοπό να εμπλουτίσει το σύνολο δεδομένων των μηχανικά αναγνωρίσιμων ΦΕΚ της πλατφόρμας Νομοθεσί@, παρέχοντάς της ένα σύνολο RDF/OWL δεδομένων το οποίο είναι και το τελικό παραδοτέο της εργασίας.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:** Σημασιολογικός ιστός, Τεχνολογία λογισμικού

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Αναπαράσταση Γνώσης νομοθεσίας, Ανοιχτά Δεδομένα, Ηλεκτρονική Διακυβέρνηση, RDF/OWL Μεταδεδομένα, Καθολικά Αναγνωριστικά Πηγής, Φύλλο Εφημερίδας της Κυβερνήσεως, Αναλυτής Δομής

## **ACKNOWLEDGMENTS**

First of all I would like to acknowledge my supervisor Prof. Manolis Koubarakis. His course “Artificial Intelligence (I)” has directly contributed to discovering my personal interest in AI and Big Data and assisted me in selecting this specific topic. I would also like to acknowledge PhD candidate Ilias Chalkidis for his valuable assistance and guidance throughout the whole thesis implementation.

# CONTENTS

<b>1. INTRODUCTION</b>	<b>11</b>
1.1 Thesis objective	11
1.2 Organisation of the thesis	11
<b>2. BACKGROUND</b>	<b>12</b>
<b>2.1 Structure of Greek Legislation</b>	<b>12</b>
2.1.1 Types of legislation	12
2.1.2 Encoding of legislation	13
2.1.3 Legislative modifications	15
2.1.4 The linea and case - The most frequent modifications	15
2.1.5 Legislation lifecycle	16
<b>2.2 Technologies for knowledge representation</b>	<b>17</b>
2.2.1 Universal Resource Identifiers (URIs)	17
2.2.2 The RDF data model and SPARQL query language	17
2.2.3 The Web Ontology Language (OWL)	18
2.2.4 Linked Data	19
<b>2.3 Summary</b>	<b>20</b>
<b>3. DESIGN OF NOMOTHESI@: A GREEK GOVERNMENT GAZETTE PARSER</b>	<b>21</b>
<b>3.1 Explanation of parsing workflow</b>	<b>21</b>
3.1.1 Preprocessing	21
3.1.2 Parsing	24
<b>3.2 JavaCC Grammar - A Context Free Grammar</b>	<b>24</b>
<b>3.3 Using JavaCC and JTB</b>	<b>26</b>
<b>4. PARSER DEMONSTRATION IN NOMOTHESI@ PLATFORM</b>	<b>29</b>
<b>5. CONCLUSIONS AND FUTURE WORK</b>	<b>31</b>
<b>APPENDIX A</b>	
<b>A GOVERNMENT GAZETTE PARSER</b>	<b>34</b>



## LIST OF FIGURES

Figure 2.1: Lifecycle of a legal document	17
Figure 2.2: An RDF graph with two nodes (Subject and Object) and a triple connecting them (Predicate)	18
Figure 2.3: RDF triple stating the publication date of Presidential Decree 24 of 09/02/2006	18
Figure 2.4: RDF triples stating that the Presidential Decree 2 of 09/02/2006 has three articles	19
Figure 2.5: RDF triples stating that the Presidential Decree has a citation and it's content	19
Figure 2.6: Nomothesi@ ontology v.2	20
Figure 2.7: Displays how Nomothesi@ can be connected to other platforms	21
Figure 3.1: Presidential Decree 1 of 02/01/2007 (Original)	23
Figure 3.2: Presidential Decree 1 of 02/01/2007 (after preprocessing)	24
Figure 3.3: Presidential Decree 1 of 02/01/2007 as a set of RDF triples	25
Figure 3.4: Goal rule, each document should have a date, an identifier and is one of the aforementioned types of legislation	26
Figure 3.5: Date published rule, matches the date at the beginning of the document	26
Figure 3.6: Law rule, matches a document of type law	27
Figure 3.7: The big picture	27
Figure 3.8: Document structure example result	29
Figure 4.1: Document's structure with the corresponding titles	30
Figure 4.2: Article 2 expanded	30
Figure 4.3: Document's general info	31

## LIST OF TABLES

Table 1: TABLE OF TERMINOLOGY	32
Table 2: ABBREVIATIONS - ACRONYMS	33

# 1. INTRODUCTION

## 1.1 Thesis objective

This thesis constitutes an attempt to enrich the database of machine readable Government Gazette issues of Nomothesi@ platform by creating a workflow where a Government Gazette gets decoded and then converted into a set of RDF triples [1], thus creating a foundation where legal web services can be built upon to, making Greek legislation more transparent, easy to access and to understand. The need for such systems becomes much clearer when we look into how time consuming and difficult traditional ways of accessing the legislation are. Manual search or use of web services like Google is the only way Greek legislation can be accessed. Making legislation more accessible, helps not only professionals, but also citizens fighting for their rights.

## 1.2 Organisation of the thesis

This thesis is organized in five chapters. Chapter 2 gives us an insight of Greek legislation background and helps us understand how it is composed and structured. Based on the aforementioned research, Chapter 3 thoroughly described the design of the parser, by means of CFG grammar and how it is represented in a Parse Tree using JavaCC tool. In Chapter 4 we present the parsing results and how Nomothesi@ platform utilises those parsed documents.. Thereafter, in Chapter 5 we summarise our work and cite any possible issues or improvements that can be dealt with in future work.

## 2. BACKGROUND

In this chapter, we discuss how the Greek legislation is decoded and structured and have a better insight of what the main issues are and what this thesis, as an improvement of Nomothesi@ platform, proposes as a solution.

### 2.1 Structure of Greek Legislation

#### 2.1.1 Types of legislation

Before we delve into the codification of legislation, we must first look at all types of Greek legislation and better understand each one.

There are eight main types of Legislation which we intend to examine and publish:

##### Constitution

The Constitution of Greece is the fundamental law upon which all Greek legislation is based regarding the rights and obligations of citizens, the organization and basic rules of operation of the Greek state and institutions. The parliament has the right to revise or amend the Constitution, except for the articles dealing with the "Form of the State" and the articles safeguarding human rights and freedoms, which are unalterable. The current Constitution of Greece, was created by the Fifth Revisionary Parliament of the Hellenes in 1974, after the fall of the Greek military junta and the start of the current Third Hellenic Republic. It entered into force in 1975 and has been revised three times since, most significantly in 1986, and also in 2001 and in 2008.

##### Presidential Decree

A presidential decree is called in the Greek legal order the regulation issued by the President of the Republic as Head of State. Presidential decrees fall into the following three categories:

- the regulatory presidential decrees, which contain rules of law and are adopted after legislative authorization,
- the executive presidential decrees issued to enforce laws,
- the Regulatory Presidential Decrees, which are issued within the remit of the President of the Republic as the regulator of the state.

##### Law

The law is issued by the main legislature of the state, which is the House of Parliament and the President of the Hellenic Republic (Article 26 of the Constitution). According to the

constitutionally foreseen legislative process, official law is the source of legislation which, once introduced in the House of Parliaments as a future law, is discussed in the House of Parliament, put to vote and having received the required majority, is issued by the President and published in the Government Gazette.

#### Act of Ministerial Cabinet

The act of ministerial act is adopted by the cabinet and regulates issues not regulated by a formal law.

#### Ministerial Decisions

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.

#### Legislative Act

According to the Greek constitution, an act of ministerial cabinet is called the legislation adopted by the President of the Republic, following a proposal by the Ministerial Cabinet, for the legislative resolution of an emergency in case of an urgent and unforeseen need

#### Legislative Decree

It is issued after the government has authorized Parliament to adopt rules that are applicable to the law on specific issues.

#### Error Fixes

Typically published to make corrections to any of the aforementioned types of legislation already published in the Government Gazette.

### **2.1.2 Encoding of legislation**

By law 3133/2003, the “Central Codification Commission”, prescribes the “Instruction Manual for Codification of Legislation” as it is necessary to enforce it as the collection and systematic classification of legislative documents contributes to the safeguarding of citizens' rights [2]. Although these standards were adopted in 2003, still the whole persistency of the draft process is being challenged.

The appropriate structure throughout the legal text consists of the following parts:

## Books - Parts - Sections - Chapters

Depending on the content and extent of the document, the text may be divided into larger units, such as 'Books', 'Parts', 'Sections' and 'Chapters'. 'Parts' may be used where necessary depending on content, scope and structure of the material. 'Books', 'Parts' and 'Sections' are numbered verbally and capitalized (e.g. BOOK ONE). Chapters are numbered in Greek capital letters in alphabetical order (e.g. A, B, C, and so on). In each case precedes the numbered subset of the code and follows the number (e.g. BOOK ONE, PART ONE, SECTION ONE, CHAPTER A).

## Articles

Articles are the main subdivision of the text, they are numbered in Arabic (e.g. 1, 2, 3 etc.) or, in the case of insertion of a new article in an existing legal document, by combining Greek letters and Arabic numerals (e.g. Article 2B).

## Paragraphs

Articles may, depending on their essential content subdivided into paragraphs. The paragraphs begin with indentation and are numbered with Arabic numerals (1, 2, 3...). If the article has only one paragraph, that single paragraph is not numbered.

## Cases

If the content of a paragraph or the content of an article that is not divided into paragraphs contains a list of cases, these are numbered in small Greek letters (e.g. α, β, γ etc.), each of which begins with a recess. If needed, sub cases are numbered in double Greek letters (e.g. αα, ββ, γγ etc.)

## Lineas

A linea is defined as the verbal period between two dots. The lineas are numbered and set out in writing contiguous, i.e. without a new break line in text.

Some other important elements of the structure of the legislation, are the following:

- A group of homogenous articles, that is, articles with the same subject, should fit into a "Chapter" with the common subject as a title.
- Groups of homogenous "Chapters" should fit into a "Section"
- Groups of homogenous "Sections" should fit into a "Part"
- Groups of homogenous "Parts" should fit into a "Book"
- A "Book" may, where suitable, be divided or not into "Sections". The same goes for the rest of subdivisions of the code.
- A 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 ministries, which are in charge of publishing this legal document.
- Optionally below that, there are citations to provisions of prior legal documents that are taken into consideration.

### 2.1.3 Legislative modifications

During the research on Greek Legislation, we have observed that there are frequent modifications of subdivisions of legal documents by subsequent legal documents. This practice does not fundamentally change a document but simply modifies, extends or removes a linea, a case, a paragraph or an article resulting in an updated version of the document.

### 2.1.4 The linea and case - The most frequent modifications

While studying the Government Gazette, we found that most modifications are related to lineas and cases. With the most frequent ones being, replacements and deletions (e.g. 'At the beginning of paragraph 1 of article 1 add the following linea...', 'Second paragraph of article 3 is being removed').

Example of legislative modification<sup>1</sup>

Article 45 of Law 154/2014 has the following structure:

<b>Article 45</b> <b>ASFA issues arrangement</b>
[...]
5. Paragraph 5 of article 166 of Law 4001/2011 is replaced by the following: «Decision of the Secretary-General of the Decentralized Administration Macedonia-Thrace of paragraph 4 determining the value of the land is published within six months of publication on Government Gazette of the decision of the Minister of Environment, Energy and Climate Change for the route and installation of the pipeline or for temporary occupation and use of the above zones.»
[...]

Article 166 of Law 4001/2011 has the following structure:

<b>Article 166</b> <b>Obligations of property owners, bans, payment of market value percentage</b>
[...]
5. Decision of the Secretary-General of the Decentralized Administration Macedonia-Thrace of paragraph 4 determining the value of the land is published within one year of publication on Government Gazette of the decision of the Minister of Environment, Energy and Climate Change for the route and installation of the pipeline or for temporary occupation and use of the above zones.
[...]

---

<sup>1</sup> The Government Gazette is not officially translated. This is an unofficial translation for the purpose of this thesis.

As we can see, article 45 of Law 154/2014 modifies article 166 of Law 4001/2011 by replacing its paragraph 5 which results in an updated document as shown below:

**Article 166**  
**Obligations of property owners, bans, payment of market value percentage**

[...]

5. Decision of the Secretary-General of the Decentralized Administration Macedonia-Thrace of paragraph 4 determining the value of the land is published within **one year** of publication on Government Gazette of the decision of the Minister of Environment, Energy and Climate Change for the route and installation of the pipeline or for temporary occupation and use of the above zones.

5. Decision of the Secretary-General of the Decentralized Administration Macedonia-Thrace of paragraph 4 determining the value of the land is published within **six months** of publication on Government Gazette of the decision of the Minister of Environment, Energy and Climate Change for the route and installation of the pipeline or for temporary occupation and use of the above zones.

[...]

### 2.1.5 Legislation lifecycle

An important part of legislative knowledge is the time frames that surround legislative events. So we have the following timestamps for each legislative process (shown in Figure 2.1):

date-signed is the time the document has been signed.

date-published is the time the document is being published to the Government Gazette.

date-into-force is the time the content becomes applicable in decision making and is always the same or later than the publication date

version-date(s) are the dates occurred by a legislative modification.

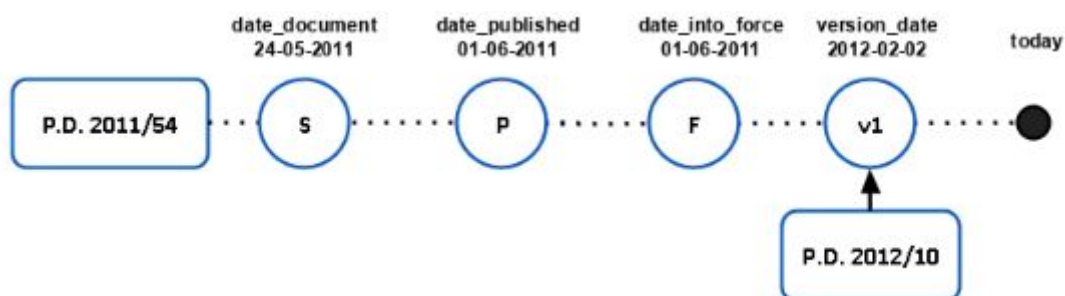


Figure 2.1: Lifecycle of a legal document



## 2.2 Technologies for knowledge representation

### 2.2.1 Universal Resource Identifiers (URIs)

The Web is an information space. Human beings have many mechanisms for manipulating, imagining, and finding their way in spaces. URIs are the points in this 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, aka URLs) are short strings that identify resources on the web: documents, images, downloadable files, services, electronic mailboxes, and other resources. They make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail addressable in the same simple way. They reduce the tedium of "log in to this server, then issue this magic command ..." down to a single click. [3]

In our system every single legal document, its subparts, its versions, those involved and its modifications are resources, being addressed by a specific URIs system.

### 2.2.2 The RDF data model and SPARQL query language

Resource Description Framework (RDF) is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web [4].

The core structure of the model 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 (Show in Figure 2.2) [5].

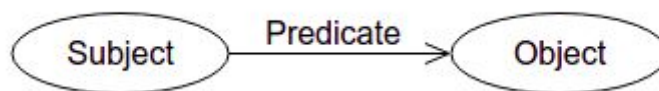


Figure 2.2: An RDF graph with two nodes (Subject and Object) and a triple connecting them (Predicate)

Below are some RDF triples produced from our parser for the Presidential Decree 2006/24:

```

<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24>
  <http://data.europa.eu/eli/ontology#date_publication>
    "2006-02-09"^^<http://www.w3.org/2001/XMLSchema#date>.
  
```

Figure 2.3: RDF triple stating the publication date of Presidential Decree 24 of 09/02/2006

```
<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24>
  <http://data.europa.eu/eli/ontology#has_part>
<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24/article/1>.

<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24>
  <http://data.europa.eu/eli/ontology#has_part>
<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24/article/2>.

<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24>
  <http://data.europa.eu/eli/ontology#has_part>
<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24/article/3>.
```

**Figure 2.4: RDF triples stating that the Presidential Decree 24 of 09/02/2006 has three articles**

```
<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24>
  <http://data.europa.eu/eli/ontology#cites>
<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24/citation/1>

<http://legislation.di.uoa.gr/eli/pd/2006-02-09/24/citation/1>
  <http://legislation.di.uoa.gr/ontology/context>
"Paragraph 1 of Article 20 of Law 2503 1997..."@en.
```

**Figure 2.5: RDF triples stating that the Presidential Decree has a citation and it's content**

SPARQL is a query language for RDF which can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.

Having the above in mind, it became clear that representing the Greek legislation with the RDF model was the best option, giving us the possibility to query them, publish them and link them with third-party datasets across the web.

### 2.2.3 The Web Ontology Language (OWL)

The OWL Web Ontology Language, is an ontology language for the Semantic Web with formally defined meaning. OWL ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL ontologies can be used along with information written in RDF, and OWL ontologies themselves are primarily exchanged as RDF documents [6].

The ontology consists of a set of axioms that place restrictions on groups of individuals (called "classes") and the types of relationships that are allowed between them. The second version of Nomothesi@ ontology is displayed in Figure 2.6 below:

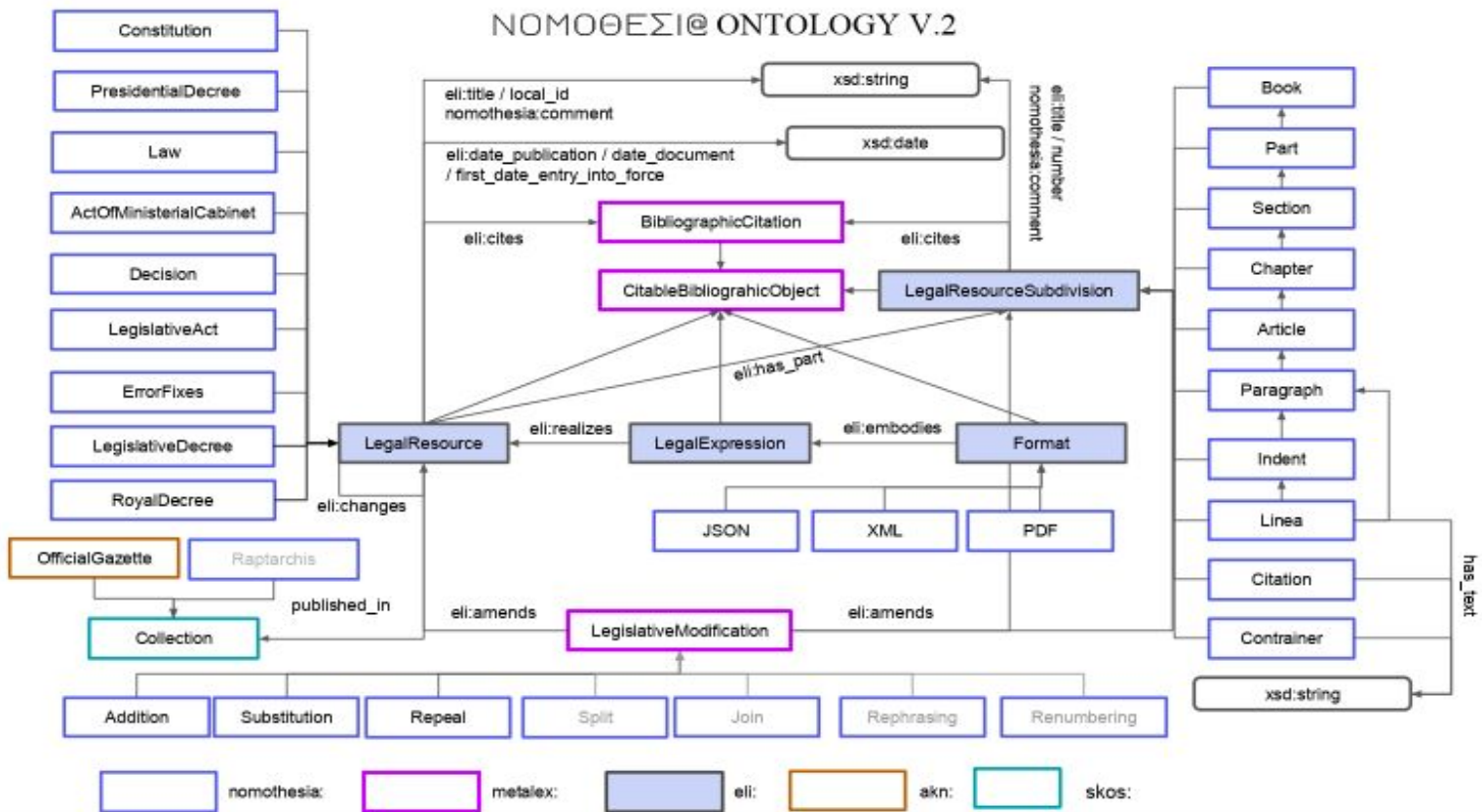


Figure 2.6: Nomothesi@ ontology v.2

### 2.2.4 Linked Data

In computing, Linked Data is structured data which is interlinked with other data so it becomes more useful through semantic queries. It builds upon standard Web technologies such as HTTP, RDF and URIs, but rather than using them to serve web pages exclusively for human readers, it extends them to share information in a way that can be read automatically by computers. This enables data of different sources to be connected.

Part of the vision of linked data is for the Internet to become a global database. In order for that interconnection to happen, there is a set of rules for publishing data on the Web [7]:

- 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.

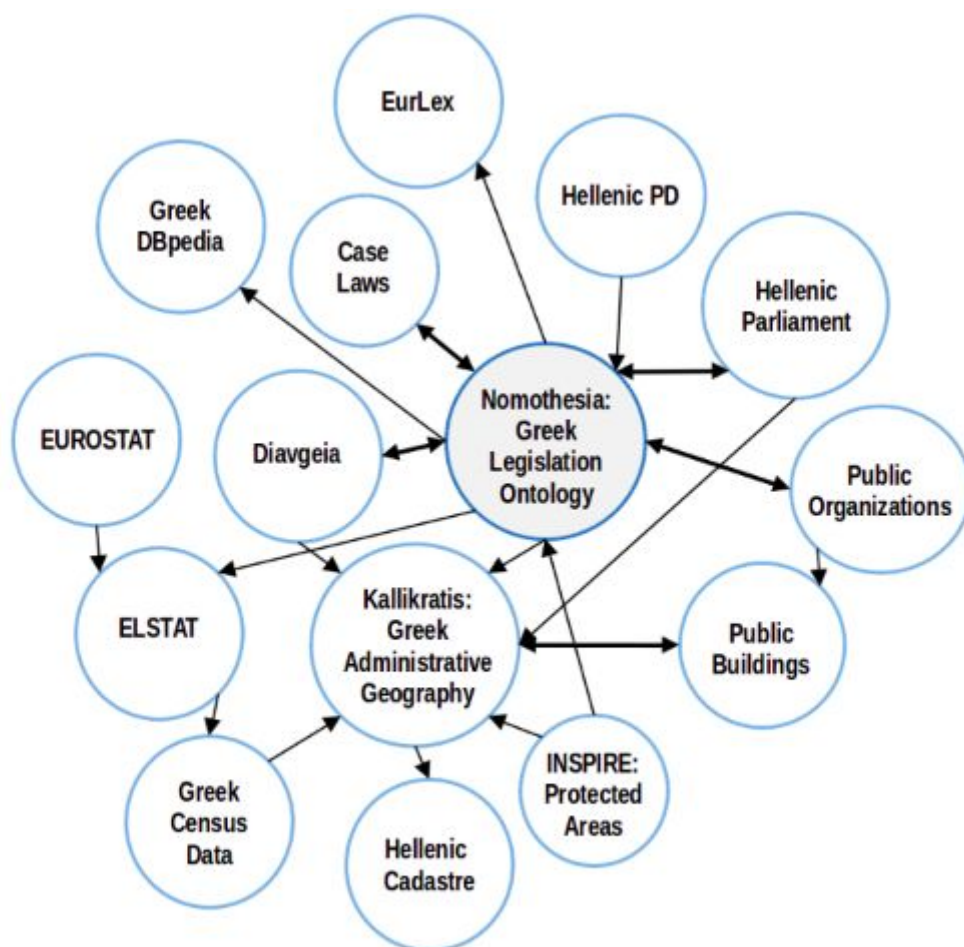


Figure 2.7: Displays how Nomothesi@ can be connected to other platforms

### 2.3 Summary

After having thoroughly examined the structure and codification of Greek legislation, we can now create a better tool for extracting as much information as possible and enriching the data sets of Nomothesi@ platform.

### **3. DESIGN OF NOMOTHESI@: A GREEK GOVERNMENT GAZETTE PARSER**

#### **3.1 Explanation of parsing workflow**

Legal documents are uploaded in the National Printing Office's website in PDF format and many of them have part of their text synthesised into two columns. In order to extract a set of RDF triples from these documents, each document has to go through two stages, the preprocessing and parsing.

##### **3.1.1 Preprocessing**

Before we delve into what this stage does, we have to note that there are cases of documents where more than one legislative act is cited. When we come across these cases, the first thing that needs to be done is to separate these acts into different documents.

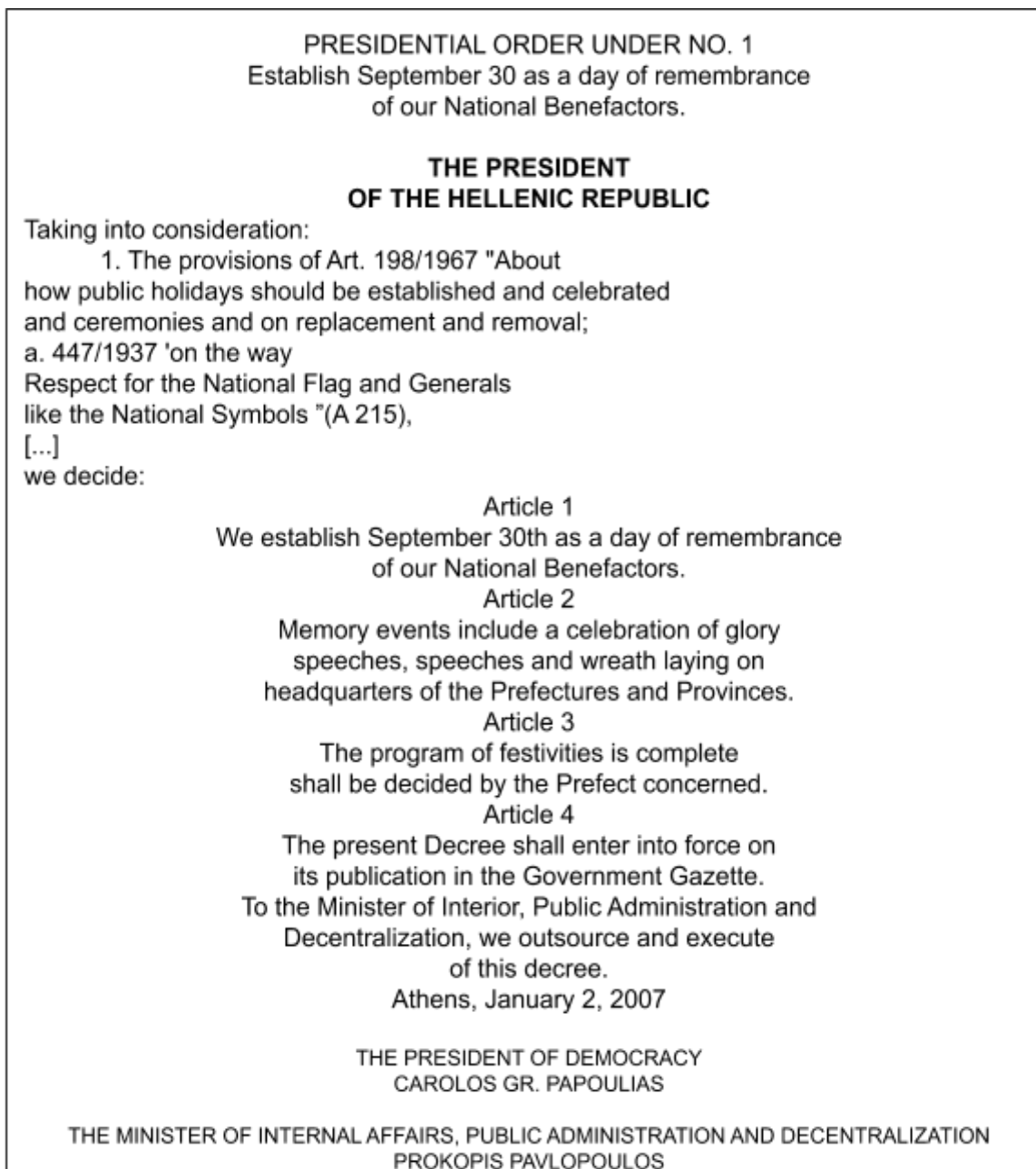
As we mentioned, all legislative documents are uploaded in PDF format, thus in order to extract any information from them, they have to get converted into plain text documents. This is achieved by using iText pdf Reader.

Thereafter, after all the aforementioned steps have been applied, the next step is to wrap any structural point of the document within a corresponding XML tag and add some metadata at the beginning of the document. Such structural points are the document's publication date, title, citations, books/parts/sections/chapters/articles etc. and the date signed.

Below is an example of Presidential Decree in its original format (Figure 3.1) and the result obtained after preprocessing stage (Figure 3.2)<sup>2</sup>:

---

<sup>2</sup> The Government Gazette is not officially translated. This is an unofficial translation for the purpose of this thesis.



**Figure 3.1: Presidential Decree 1 of 02/01/2007 (Original)**

```

<date_published>2007/01/02</date_published>
<gazette>a/2007/1</gazette>
<pd_id>1</pd_id>
Establish September 30 as a day of remembrance of our National Benefactors.
<main_body_start></main_body_start>
<citations_start>
<citation>1</citation>The provisions of Art. 198/1967 "About how public holidays
should be established and celebrated and ceremonies and on replacement and
removal; a. 447/1937 'on the way Respect for the National Flag and Generals like
the National Symbols "(A 215)
[.]
<citations_end>
<article>1</article>
We establish September 30th as a day of remembrance of our National
Benefactors.
<article>2</article>
Memory events include a celebration of glory speeches, speeches and wreath
laying on headquarters of the Prefectures and Province
<article>3</article>
Memory events include a celebration of glory speeches, speeches and wreath
laying on headquarters of the Prefectures and Province
<article>4</article>
The present Decree shall enter into force on its publication in the Government
Gazette.
To the Minister of Interior, Public Administration and Decentralization, we outsource
and execute of this decree.
<end_of_main_body>Athens, January 2, 2007</end_of_main_body>
THE PRESIDENT OF DEMOCRACY
CAROLOS GR. PAPOULIAS
THE MINISTER OF INTERNAL AFFAIRS, PUBLIC ADMINISTRATION AND
DECENTRALIZATION
PROKOPIS PAVLOPOULOS

```

**Figure 3.2: Presidential Decree 1 of 02/01/2007 (after preprocessing)**

These XML tags will be used during the parsing process where all structural points can now be automatically recognised giving us the ability to create an embedded structure which describes the whole document's structure.

Below is an extensive list of these tags:

- <date\_published> is a metadata stating the date the document was published
- <gazette> is a metadata stating issue numbering.
- <pd\_id | law\_id | dec\_id | la\_id | aomc\_id > states the document's id depending on the Legislation's type (e.g. pd\_id for Presidential Decree).
- <main\_body\_start> and <main\_body\_end> surround the citations and the main content of the document.
- <citation> tags enumerate all the citations.
- <book>, <part>, <section> etc. surround all main structural points of the documents described in chapter 2.1.2.



### 3.1.2 Parsing

Parsing is the stage where the documents resulting from the preprocessing stage are converted into a set of RDF triples by means described in sections 3.2 and 3.3. Below is an example of what the final results look like (Figure 3.3):

```
<http://legislation.di.uoa.gr/gazette/a/2007/1> <http://data.europa.eu/eli/ontology#date_publication>
"2007-01-02"^^<http://www.w3.org/2001/XMLSchema#date>.
<http://legislation.di.uoa.gr/gazette/a/2007/1> <http://purl.org/dc/terms/title> "A/2007/1".
<http://legislation.di.uoa.gr/gazette/a/2007/1> a <http://legislation.di.uoa.gr/ontology/GovernmentGazette>.
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://data.europa.eu/eli/ontology#published_in> <http://legislation.di.uoa.gr/gazette/a/2007/1>.
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://data.europa.eu/eli/ontology#id_local> "1".
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://legislation.di.uoa.gr/ontology/tag> "Ελλάδα"@el.
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://legislation.di.uoa.gr/ontology/views> "0"^^<http://www.w3.org/2001/XMLSchema#integer>.
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://data.europa.eu/eli/ontology#date_publication>
"2007-01-02"^^<http://www.w3.org/2001/XMLSchema#date>.
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://data.europa.eu/eli/ontology#date_inforce> "2007-01-02"^^<http://www.w3.org/2001/XMLSchema#date>.
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://data.europa.eu/eli/ontology#date_document> "2007-01-02"^^<http://www.w3.org/2001/XMLSchema#date>.
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://data.europa.eu/eli/ontology#has_part> <http://legislation.di.uoa.gr/eli/pd/2007/1/article/1>.
[...]
<http://legislation.di.uoa.gr/eli/pd/2007/1> <http://data.europa.eu/eli/ontology#cites> <http://legislation.di.uoa.gr/eli/pd/2007/1/citation/1>.
[...]
<http://legislation.di.uoa.gr/eli/pd/2007/1/citation/1> <http://legislation.di.uoa.gr/ontology/context> "The provisions of article 1 of a.d. 198/1967 "On the
manner in which public holidays and ceremonies are instituted and performed and on the replacement and abolition of the provisions of the Hon. 447/1937
"On the manner in which the National Flag was honored and in general the National Symbols" (A 215),"@el.
<http://legislation.di.uoa.gr/eli/pd/2007/1/citation/1> a <http://www.metalex.eu/metalex/2008-05-02#BibliographicCitation>.
[...]
<http://legislation.di.uoa.gr/eli/pd/2007/1/article/1> <http://data.europa.eu/eli/ontology#has_part> <http://legislation.di.uoa.gr/eli/pd/2007/1/article/1/linea/1>.
<http://legislation.di.uoa.gr/eli/pd/2007/1/article/1> a <http://legislation.di.uoa.gr/ontology/Article>.
<http://legislation.di.uoa.gr/eli/pd/2007/1/article/1/linea/1> <http://legislation.di.uoa.gr/ontology#has_text> "We establish September 30th as a day of
remembrance of our National Benefactors"@el.
<http://legislation.di.uoa.gr/eli/pd/2007/1/article/1/linea/1> a <http://legislation.di.uoa.gr/ontology/Linea>.
[...]
```

Figure 3.3: Presidential Decree 1 of 02/01/2007 as a set of RDF triples

Extensive details of this parsing method works are presented in the next Chapters.

### 3.2 JavaCC Grammar - A Context Free Grammar

Greek legislation’s structure can be described in a way, similar to how programming languages or any written language can be described. This is possible because Greek legislation’s structure follows a set of rules regarding how it should be composed (described in chapter 2.1.2). That technique is to create a Context Free grammar, which makes possible the creation of a parser, using the JavaCC parser generator tool.

In formal language theory, a context-free grammar (CFG) is a set of recursive rewriting rules (or productions) used to generate patterns of strings [8].



A CFG consists of the following components:

- a set of terminal symbols, which are the characters of the alphabet that appear in the strings generated by the grammar.
- a set of nonterminal symbols, which are placeholders for patterns of terminal symbols that can be generated by the nonterminal symbols.
- a set of productions, which are rules for replacing (or rewriting) nonterminal symbols (on the left side of the production) in a string with other nonterminal or terminal symbols (on the right side of the production).
- a start symbol, which is a special nonterminal symbol that appears in the initial string generated by the grammar.

To generate a string of terminal symbols from a CFG, we:

- Begin with a string consisting of the start symbol.
- Apply one of the productions with the start symbol on the left hand side, replacing the start symbol with the right hand side of the production.
- Repeat the process of selecting nonterminal symbols in the string, and replacing them with the right hand side of some corresponding production, until all nonterminals have been replaced by terminal symbols.

Using the xml tags created in the preprocessing stage, we can now create a grammar that can recognise those tags. Below are some example rules of the context free grammar created for the purposes of this thesis:

```
void Goal() :
{
{
DATE_PUBLISHED()
GAZETTE()
(LAW()|PD()|DEC()|LEGISLATIVEACT()|ANN()|AOMC()|REG()|AGR())
}
}
```

**Figure 3.4: Goal rule, each document should have a date, an identifier and is one of the aforementioned types of legislation**

```
void DATE_PUBLISHED():
{
{
"<date_published>"
IntegerLiteral()
<SPECIAL_CHAR>
IntegerLiteral()
<SPECIAL_CHAR>
IntegerLiteral()
"</date_published>"
}
}
```

**Figure 3.5: Date published rule, matches the date at the beginning of the document**

```

void LAW():
{
{
  LAW_ID()
  SIMPLE_PARAGRAPH()
  MAIN_BODY_START_TAGS()
  MAIN_BODY()
  MAIN_BODY_END_TAGS()
  [SIMPLE_PARAGRAPH()]
  <EOF>
}
}

```

Figure 3.6: Law rule, matches a document of type law

### 3.3 Using JavaCC and JTB

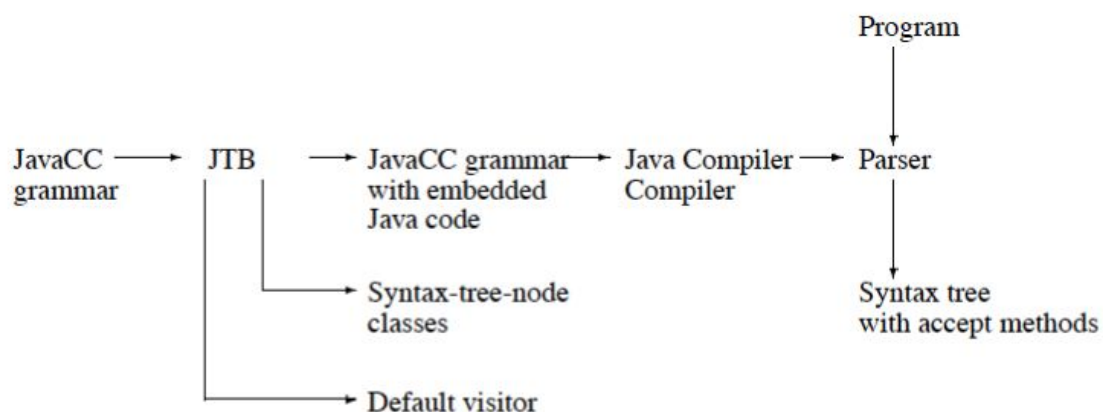


Figure 3.7: The big picture

The above figure (Figure 3.7) shows the process of how the JavaCC grammar described in section 3.2 is converted into a Java program that can recognise matches to the grammar.

The Java Tree Builder (JTB) is a syntax tree builder to be used with the Java Compiler Compiler (JavaCC) parser generator [9]. It takes a plain JavaCC grammar file as input and automatically generates the following:

- A set of syntax tree classes based on the productions in the grammar, utilizing the Visitor design pattern.
- Two interfaces: Visitor and GJVisitor. Two depth-first visitors: DepthFirstVisitor and GJDepthFirst, whose default methods simply visit the children of the current node.
- A JavaCC grammar `jtb.out.jj` with the proper annotations to build the syntax tree during parsing.

JavaCC, is a parser generator which provides a Java language extension for specifying a programming language's grammar. Moreover, JavaCC allows us to define grammars in a fashion similar to EBNF, making it easy to translate EBNF grammars into the JavaCC format [10].

The output of the JavaCC is the following (i.e. `nomothesia.jj` is the JavaCC grammar):

- `TokenMgrError` is a simple error class; it is used for errors detected by the lexical analyser and is a subclass of `Throwable`.
- `ParseException` is another error class; it is used for errors detected by the parser and is a subclass of `Exception` and hence of `Throwable`.
- `Token` is a class representing tokens. Each `Token` object has an integer field `kind` that represents the kind of the token (`PLUS`, `NUMBER`, etc) and a `String` field `image`, which represents the sequence of characters from the input file that the token represents.
- `JavaCharStream` delivers characters to the lexical analyser.
- `NomothesiaParserConstants` is an interface that defines a number of classes used in both the lexical analyser and the parser, associating token classes with symbolic names.
- `NomothesiaParserTokenManager` is the lexical analyser.
- `NomothesiaParser` is the parser.

The result of the parsing process, is a structure with all the identified structural points of a legislative document and the relations between them.

```
{
  "type": "part",
  "id": 1,
  "content": [{
    "type": "article",
    "id": 1,
    "content": "Article 1 text content"
  },
  {
    "type": "article",
    "id": 2,
    "content": [{
      "type": "paragraph",
      "id": 1,
      "content": "Content 1 paragraph"}]
  }
]
```

**Figure 3.8: Document structure example result**

Having regard to aforementioned structure, we can construct the RDF representation of the document as described in section 3.1.2.

## 4. PARSER DEMONSTRATION IN NOMOTHESI@ PLATFORM

In this chapter, we demonstrate the parsing results as they are illustrated in the redesigned Nomothesi@ platform.

**PART ONE**  
ΝΕΟ ΡΥΘΜΙΣΤΙΚΟ ΣΧΕΔΙΟ ΑΘΗΝΑΣ - ΑΤΤΙΚΗΣ

**CHAPTER I**  
ΠΕΡΙΕΧΟΜΕΝΟ, ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΙ ΧΡΟΝΙΚΗ ΔΙΑΡΚΕΙΑ

Article 1	"Όρισμός και περιεχόμενο του νέου Ρυθμιστικού Σχεδίου Αθήνας - Αττικής"
Article 2	"Χρονική διάρκεια"

**CHAPTER II**  
ΣΤΡΑΤΗΓΙΚΗ ΤΟΥ ΝΕΟΥ ΠΣΑ ΒΑΣΙΚΟΙ ΚΑΙ ΕΙΔΙΚΟΙ ΣΤΟΧΟΙ

Article 3	"Στρατηγικοί στόχοι του νέου ΠΣΑ"
Article 4	"Ισορροπη οικονομική ανάπτυξη, ενίσχυση του διεθνούς ρόλου της Αθήνας - Αττικής, βελτίωση της ανταγωνιστικότητας, αύξηση της παραγωγής και της απασχόλησης σε όλους τους τομείς δραστηριοτήτων"
Article 5	"Βιώσιμη χωρική ανάπτυξη, εξοικονόμηση πόρων, αποτελεσματική προστασία του περιβάλλοντος και της πολιτιστικής κληρονομιάς και προσαρμογή στην κλιματική αλλαγή"
Article 6	"Βελτίωση της ποιότητας ζωής των κατοίκων και εξισορρόπηση στην κατανομή των πόρων και των ωφελειών από την ανάπτυξη"

**Figure 4.1: Document's structure with the corresponding titles**

**PART ONE**  
ΝΕΟ ΡΥΘΜΙΣΤΙΚΟ ΣΧΕΔΙΟ ΑΘΗΝΑΣ - ΑΤΤΙΚΗΣ

**CHAPTER I**  
ΠΕΡΙΕΧΟΜΕΝΟ, ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΙ ΧΡΟΝΙΚΗ ΔΙΑΡΚΕΙΑ

Article 1	"Όρισμός και περιεχόμενο του νέου Ρυθμιστικού Σχεδίου Αθήνας - Αττικής"
Article 2	"Χρονική διάρκεια"

1. Το νέο ΠΣΑ στηρίζεται σε προβολές πληθυσμιακών και οικονομικών μεγεθών και αναπτυξιακών προβλέψεων με χρονική διάρκεια έως και το έτος 2021. Για την εφαρμογή του νέου ΠΣΑ δεν προβλέπονται ενδιάμεσες προγραμματικές περιόδους. Η προσέγγιση των στόχων, η προώθηση των πολιτικών, καθώς και ο ρυθμός και η έκταση υλοποίησης των προβλεπόμενων ρυθμίσεων παρακολουθούνται με σταθερά μέσα από το σύστημα κριτηρίων, με βάση τις περιγραφόμενες στα άρθρα 35, 36, 37, 39 του παρόντος διαδικασίες, λαμβάνοντας τα αναγκαία οργανωτικά και λειτουργικά μέτρα για τη βελτίωση των επιδόσεων.

**Figure 4.2: Article 2 expanded**

## GENERAL DATA

TYPE	Law
REF.NUMBER	2014/4277
FEK	A/2014/156 

## DATES

PUBLISHED	2014-08-01
IN-FORCE	2014-08-01
SIGNATURE	2014-07-31

**Figure 4.3: Document's general info**

## 5. CONCLUSIONS AND FUTURE WORK

This thesis sets the foundation for the next version of the Nomothesi@ platform, by enriching its database of machine readable legal documents and being able to identify legal document types that up until now the system couldn't process (e.g. Government Decisions). We achieved this by creating a new parser for the government gazette which improves the structure identification of the documents. We demonstrated how this process is divided into two steps, the preprocessing step and the parsing step and how from a pdf file we can create a set of RDF triples.

Although the parser developed in the thesis is a big improvement over the existing parser, there is an issue that the method followed in this thesis couldn't solve. That is the identification of the relevant ministries and the signers, thus some post process work is required in the future.

Having all this structured information about the legal documents, the next stage of the Nomothesi@ platform, is to be able to recognise references in the legal documents. A legal document could have references to other documents (e.g. reference to a law), people, geographic areas etc.

## TABLE OF TERMINOLOGY

URIs	Καθολικά Αναγνωριστικά Πηγής
RDF	Πλαίσιο Περιγραφής Πόρων



## ABBREVIATIONS - ACRONYMS

FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
EU	European Union
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
RDF	Resource Description Framework
OWL	Web Ontology Language
SPARQL	SPARQL Protocol and RDF Query Language
PDF	Portable Document Format
XML	Extensible Markup Language
CFG	Context Free Grammar
JavaCC	Java Compiler Compiler
JTB	Java Tree Builder

## APPENDIX A A GOVERNMENT GAZETTE PARSER

### Installation instructions

Nomothesi@ parser, is a java program divided into two projects. Inside the zipped file, there are two java projects:

#### nomothesia.pre-processor:

This project takes as an input a set of pdf files and converts them into a set of txt files with the xml tags described in section 3.1.1.

In order to get the expected results, the source and destination folder paths must be specified inside the project and then the java project must get executed.

#### nomothesia.parser:

This project is responsible for producing the final output, a set of RDF triples describing all the parsed documents.

In order to get the expected result, the input is the already processed by the preprocessor documents and the output file paths must be specified inside the project and then the java project must get executed.

### The JavaCC Grammar

```
void Goal() :  
{  
    DATE_PUBLISHED()  
    GAZETTE()  
    (LAW())|PD())|DEC())|LEGISLATIVEACT())|ANN())|AOMC())|REG())|AGR()  
}
```

void **AGR**():

```
{  
  AGR_ID()  
  [SIMPLE_PARAGRAPH()]  
  MAIN_BODY_START_TAGS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **AGR\_ID**():

```
{  
  "<contr_id>"SIMPLE_PARAGRAPH("</contr_id>"  
}
```

void **REG**():

```
{  
  REG_ID()  
  SIMPLE_PARAGRAPH()  
  MAIN_BODY_START_TAGS()  
  CITATIONS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **REG\_ID**():

```
{  
  ("<reg_id>"|"<rp_id>")  
  SIMPLE_PARAGRAPH()  
  ("</reg_id>"|"</rp_id>")  
}
```

void **AOMC**():

```
{  
  AOMC_ID()  
  SIMPLE_PARAGRAPH()  
  MAIN_BODY_START_TAGS()  
  CITATIONS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **AOMC\_ID**():

```
{  
  "<aomc_id>" SIMPLE_PARAGRAPH() "</aomc_id>"  
}
```

void **ANN**():

```
{  
  ANN_ID()  
  MAIN_BODY_START_TAGS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **ANN\_ID**():

```
{  
  "<ann_id>" SIMPLE_PARAGRAPH() "</ann_id>"  
}
```

void **LEGISLATIVEACT**():

```
{  
  LA_ID()  
  SIMPLE_PARAGRAPH()  
  MAIN_BODY_START_TAGS()  
  CITATIONS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **LA\_ID**():

```
{  
  "<la_id>" SIMPLE_PARAGRAPH() "</la_id>"  
}
```

void **DEC**():

```
{  
  DEC_ID()  
  [SIMPLE_PARAGRAPH()]  
  [CITATIONS()]  
  MAIN_BODY_START_TAGS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **DEC\_START\_TAGS\_1**():

```
{  
  ( DEC_START_TAGS_3()| DEC_START_TAGS_4())  
}
```

void **DEC\_START\_TAGS\_3**():

```
{  
  CITATIONS()  
}
```

void **DEC\_START\_TAGS\_4**():

```
{  
  SIMPLE_PARAGRAPH()  
  CITATIONS()  
}
```

void **DEC\_ID**():

```
{  
  "<dec_id>" SIMPLE_PARAGRAPH() "</dec_id>"  
}
```

void **PD**():

```
{  
  PD_ID()  
  SIMPLE_PARAGRAPH()  
  MAIN_BODY_START_TAGS()  
  CITATIONS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **PD\_ID**():

```
{  
  "<pd_id>" IntegerLiteral() "</pd_id>"  
}
```

void **CITATIONS**():

```
{  
  "<citations_start>"  
  (MULTIPLE_CITATIONS()|SIMPLE_PARAGRAPH())  
  "<citations_end>"  
}
```

void **MULTIPLE\_CITATIONS**():

```
{  
  (SINGLE_CITATION())+  
}
```

void **SINGLE\_CITATION**():

```
{  
  "<citation>" (WORD()|IntegerLiteral()) "</citation>"  
  SIMPLE_PARAGRAPH()  
}
```

void **LAW**():

```
{  
  LAW_ID()  
  SIMPLE_PARAGRAPH()  
  MAIN_BODY_START_TAGS()  
  MAIN_BODY()  
  MAIN_BODY_END_TAGS()  
  [SIMPLE_PARAGRAPH()]  
  <EOF>  
}
```

void **GAZETTE**():

```
{  
  "<gazette>"  
  WORD()  
  <SPECIAL_CHAR>  
  IntegerLiteral()  
}
```

```
<SPECIAL_CHAR>  
IntegerLiteral()  
"/>gazette>"  
}
```

void **DATE\_PUBLISHED**():

```
{  
  "<date_published>"  
  IntegerLiteral()  
  <SPECIAL_CHAR>  
  IntegerLiteral()  
  <SPECIAL_CHAR>  
  IntegerLiteral()  
  "</date_published>"  
}
```

void **LAW\_ID**():

```
{  
  "<law_id>" IntegerLiteral() "</law_id>"  
}
```

void **MAIN\_BODY**():

```
{  
  (SIMPLE_PARAGRAPH())|SKIP_CONTENT()|RECURSIVE_ARTICLE()|RECURSIVE_CHAPTER_ARTICLE()  
  |RECURSIVE_SECTION_CHAPTER_ARTICLE()|RECURSIVE_PART_SECTION_CHAPTER_ARTICLE()|RECURSIVE_BOOK_PART_S  
  ECTION_CHAPTER_ARTICLE()  
}
```

void **RECURSIVE\_BOOK\_PART\_SECTION\_CHAPTER\_ARTICLE**():

```
{  
  (BOOK_PART_SECTION_CHAPTER_ARTICLE())+  
}
```



void **BOOK\_PART\_SECTION\_CHAPTER\_ARTICLE**():

```
{  
    BOOK()  
  
    (RECURSIVE_PART_SECTION_CHAPTER_ARTICLE() | RECURSIVE_SECTION_CHAPTER_ARTICLE() |  
    RECURSIVE_CHAPTER_ARTICLE() | RECURSIVE_ARTICLE())  
}
```

void **BOOK**():

```
{  
    "<book>"  
  
    SIMPLE_PARAGRAPH()  
  
    "</book>"  
  
    [BOOK_TITLE()]  
  
    [SIMPLE_PARAGRAPH()]  
}
```

void **BOOK\_TITLE**():

```
{  
    "<book_title>"  
  
    SIMPLE_PARAGRAPH()  
  
    "</book_title>"  
}
```

void **RECURSIVE\_PART\_SECTION\_CHAPTER\_ARTICLE**():

```
{  
    (PART_SECTION_CHAPTER_ARTICLE())+  
}
```

void **PART\_SECTION\_CHAPTER\_ARTICLE**():

```
{  
    PART()  
  
    (RECURSIVE_SECTION_CHAPTER_ARTICLE() | RECURSIVE_CHAPTER_ARTICLE() | RECURSIVE_ARTICLE())  
}
```

void **PART()** :

```
{  
  "<part>" SIMPLE_PARAGRAPH() "</part>"  
  [PART_TITLE()]  
  [SIMPLE_PARAGRAPH()]  
}
```

void **PART\_TITLE()**:

```
{  
  "<part_title>"  
  (WORD()|IntegerLiteral()|SPECIAL_CHAR()|DOT())*  
  "</part_title>"  
}
```

void **RECURSIVE\_SECTION\_CHAPTER\_ARTICLE()**:

```
{  
  (SECTION_CHAPTER_ARTICLE())+  
}
```

void **SECTION\_CHAPTER\_ARTICLE()**:

```
{  
  SECTION()  
  (RECURSIVE_CHAPTER_ARTICLE() | RECURSIVE_ARTICLE())  
}
```

void **SECTION()**:

```
{  
  "<section>" SIMPLE_PARAGRAPH() "</section>"  
  [SECTION_TITLE()]  
  [SIMPLE_PARAGRAPH()]  
}
```

void **SECTION\_TITLE**():

```
{  
  "<section_title>"  
  SIMPLE_PARAGRAPH()  
  "</section_title>"  
}
```

void **RECURSIVE\_CHAPTER\_ARTICLE**():

```
{  
  (CHAPTER_ARTICLE())+  
}
```

void **CHAPTER\_ARTICLE**() :

```
{  
  CHAPTER()  
  [RECURSIVE_ARTICLE()]  
}
```

void **CHAPTER**() :

```
{  
  "<chapter>"SIMPLE_PARAGRAPH()</chapter>"  
  [CHAPTER_TITLE()]  
  [CONTENT()]  
}
```

void **CHAPTER\_TITLE**() :

```
{  
  "<chapter_title>"  
  SIMPLE_PARAGRAPH()  
  "</chapter_title>"  
}
```

void **RECURSIVE\_ARTICLE**():

```
{  
  (ARTICLE())+  
}
```

```
}
```

```
void ARTICLE() :
```

```
{  
  "<article>"SIMPLE_PARAGRAPH()"/>"  
  [ARTICLE_TITLE()]  
  (CONTENT() | SKIP_CONTENT())  
}
```

```
void ARTICLE_TITLE():
```

```
{  
  "<article_title>"SIMPLE_PARAGRAPH()"/>"  
}
```

```
void PARAGRAPH():
```

```
{  
  "<paragraph>" SIMPLE_PARAGRAPH() "</paragraph>"  
  [SIMPLE_PARAGRAPH()]  
  (CASE())*  
}
```

```
void CASE() :
```

```
{  
  "<case>" SIMPLE_PARAGRAPH() "</case>"  
  [SIMPLE_PARAGRAPH()]  
  (SUBCASE())*  
}
```

```
void SUBCASE() :
```

```
{  
  "<subcase>" SIMPLE_PARAGRAPH() "</subcase>"  
  SIMPLE_PARAGRAPH()  
}
```

void **CONTENT**():

```
{  
  CONTENT_CASE_1()  
  |  
  CONTENT_CASE_2()  
  |  
  SIMPLE_PARAGRAPH()  
}
```

void **SKIP\_CONTENT**():

```
{  
  "$$SKIP$"  
  SIMPLE_PARAGRAPH()  
  "$/SKIP$"  
  
}
```

void **CONTENT\_CASE\_1**():

```
{  
  (CASE())+  
}
```

void **CONTENT\_CASE\_2**():

```
{  
  (PARAGRAPH())+  
}
```

void **SIMPLE\_PARAGRAPH**():

```
{  
  (WORD())|IntegerLiteral()|SPECIAL_CHAR()|DOT()+  
}
```

void **DOT**():

```
{  
  "."  
}
```

void **SPECIAL\_CHAR**() :

```
{  
  <SPECIAL_CHAR>  
}
```

void **IntegerLiteral**() :

```
{  
  <INTEGER_LITERAL>  
}
```

void **WORD**() :

```
{  
  <WORD>  
}
```

void **MAIN\_BODY\_START\_TAGS**():

```
{  
  "<main_body_start></main_body_start>"  
}
```

void **MAIN\_BODY\_END\_TAGS**():

```
{  
  "<end_of_main_body>"[SIMPLE_PARAGRAPH()]"</end_of_main_body>"  
}
```

```
void MAIN_BODY_END_TAGS_DATE():
```

```
{  
    IntegerLiteral()  
    <SPECIAL_CHAR>  
    IntegerLiteral()  
    <SPECIAL_CHAR>  
    IntegerLiteral()  
}
```

## REFERENCES

- [1] Ora Lassila, Ralph R. Swick. "Resource Description Framework (RDF) Model and Syntax Specification", 1999. <https://www.w3.org/TR/PR-rdf-syntax/>
- [2] Central Codification Committee of Government's, "Instruction Manual for Codification of Legislation", 2003. [http://www.ggk.gov.gr/wp-content/uploads/2010/02/teliko\\_egxeiridio\\_odigion\\_gia\\_tin\\_kodikopoiisi\\_tis\\_nomothesias.pdf](http://www.ggk.gov.gr/wp-content/uploads/2010/02/teliko_egxeiridio_odigion_gia_tin_kodikopoiisi_tis_nomothesias.pdf)
- [3] Dan Connolly. "Naming and Addressing: URIs, URLs", 1993. <https://www.w3.org/Addressing/>
- [4] Ora Lassila, Ralph R. Swick. "Resource Description Framework (RDF) Model and Syntax Specification", 1999. <https://www.w3.org/TR/PR-rdf-syntax/>
- [5] Graham Klyne, David Wood, Markus Lanthaler. "RDF 1.1 Concepts and Abstract Syntax", 2014. <https://www.w3.org/TR/rdf11-concepts/>
- [6] Michael Schneider. "OWL 2 Web Ontology Language RDF-Based Semantics", 2012. <https://www.w3.org/TR/owl2-rdf-based-semantics/>
- [7] Tim Berners-Lee. "Linked Data", 2006. <https://www.w3.org/DesignIssues/LinkedData.html>
- [8] Nelson, R. "*Context-Free Grammars*", 2016. [https://www.cs.rochester.edu/~nelson/courses/csc\\_173/grammars/cfg.html](https://www.cs.rochester.edu/~nelson/courses/csc_173/grammars/cfg.html)
- [9] Enseling, Oliver. "Build your own languages with JavaCC", 2020. <https://www.javaworld.com/article/2076269/build-your-own-languages-with-javacc.html>
- [10] Federico Tomassetti. "EBNF: How to Describe the Grammar of a Language", 2017. <https://tomassetti.me/ebnf/>