DELIVERABLE

Project Acronym: EUCases
Grant Agreement number: 611760
Project Title: European and National Legislation and Case Law Linked in Open Data Stack

D4.1 Publication of EUCases Legal Linked Open Dataset and Web Interface Querying EUCases Linking Platform

Authors:

Kiril Simov IICT-BAS
Petya Osenova IICT-BAS
Iliana Simova IICT-BAS
Ivajlo Radev IICT-BAS

Project co-funded by the European Commission within FP7-ICT-2013-SME-DCA

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
</tr>
<tr>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission Services)</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
</tbody>
</table>
# Revision History, Status, Abstract, Keywords, Statement of Originality

## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author</th>
<th>Organisation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>01-05-2015</td>
<td>Petya Osenova</td>
<td>IICT-BAS</td>
<td>First draft</td>
</tr>
<tr>
<td>0.2</td>
<td>15-05-2015</td>
<td>Kiril Simov</td>
<td>IICT-BAS</td>
<td>Ontology overview</td>
</tr>
<tr>
<td>0.3</td>
<td>21-05-2015</td>
<td>Alexander Simov</td>
<td>IICT-BAS</td>
<td>GraphDB Workbench</td>
</tr>
<tr>
<td>0.4</td>
<td>31-05-2015</td>
<td>Kiril Simov, Tenyo Tyankov</td>
<td>IICT-BAS, APIS</td>
<td>Modeling EUCases Documents - first draft</td>
</tr>
<tr>
<td>0.5</td>
<td>20.06.2015</td>
<td>Alexander Simov</td>
<td>IICT-BAS</td>
<td>Validation</td>
</tr>
<tr>
<td>0.6</td>
<td>25-06-2015</td>
<td>Kiril Simov, Tenyo Tyankov</td>
<td>IICT-BAS, APIS</td>
<td>Document modeling - second draft</td>
</tr>
<tr>
<td>0.8</td>
<td>25.07.2015</td>
<td>Kiril Simov, Alexander Simov</td>
<td>IICT-BAS</td>
<td>Validation</td>
</tr>
<tr>
<td>1.0</td>
<td>10.08.2015</td>
<td>Kiril Simov, Petya Osenova</td>
<td>IICT-BAS</td>
<td>Final version</td>
</tr>
<tr>
<td>1.01</td>
<td>25.08.2015</td>
<td>Kiril Simov</td>
<td>IICT-BAS</td>
<td>Final version, revision, namespace changed</td>
</tr>
<tr>
<td>2.0</td>
<td>25.01.2016</td>
<td>Kiril Simov</td>
<td>IICT-BAS</td>
<td>Revision as per reviewer requests</td>
</tr>
</tbody>
</table>

Date of delivery

<table>
<thead>
<tr>
<th>Contractual: 30-05-2015</th>
<th>Actual: 10-08-2015</th>
</tr>
</thead>
</table>

Status

<table>
<thead>
<tr>
<th>final</th>
<th>draft</th>
<th>Revised version: 27-01-2016</th>
</tr>
</thead>
</table>

**Abstract**

This document presents Publication of EUCases legal linked open dataset and Web interface querying EUCases Linking Platform.

**Keywords**

Linked Open Data, Ontology, SPARQL

**Statement of originality**

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.
# Table of Content

1. The EUCases Legal Linked Open Dataset ................................................................. 6
2. Technological Basis of EUCases-LLOD ................................................................. 7
   2.1 Linked Data ........................................................................................................ 7
   2.2 Resource Description Format ......................................................................... 8
   2.3 SPARQL Query and Update Languages ......................................................... 9
   2.4 Reason-able Views ......................................................................................... 9
3. Ontologies .............................................................................................................. 11
   3.1 PROTON Ontology ........................................................................................ 11
   3.2 SKOS Ontology ............................................................................................. 13
   3.3 Dublin Core ..................................................................................................... 15
   3.4 EuroVoc Thesaurus ......................................................................................... 16
   3.5 Legal Taxonomy Syllabus .............................................................................. 17
4. EUCases Document Modeling ............................................................................. 18
   4.1 Namespaces for EUCases project .................................................................. 18
   4.2 Ontology modelling ....................................................................................... 18
   4.3 Creation of RDF representation of the EUCases documents ....................... 22
5. GraphDB RDF Repository .................................................................................... 24
6. EUCases-LLOD Support and Availability ............................................................ 26
7. Conclusion ............................................................................................................ 30

References ................................................................................................................ 31
Executive Summary

The deliverable introduces the methodology and technical details behind the EUCases Legal Linked Open Dataset (EUCases-LLOD) and Web Interface Querying EUCases Linking Platform. Methodologically, the EUCases-LLOD consists of a set of ontologies and RDF triples. EuroVoc and Syllabus are in the center of the model, since they are used as domain specific ontologies. Additionally, other supporting ontologies have been added, such as GeoNames for the named entities; PROTON as an upper ontology; SKOS as a mapper between ontologies and terminological lexicons; Dublin Core as a metadata ontology. For an efficient reasoning, the so-called FactForge reason-able view was explored as an approach to linked data management.

Technically, EUCases-LLOD is represented in RDF graphs and uses the SPARQL query language. The input documents have been encoded into the legal XML schema Akoma Ntoso. Thus, the schema has been converted into an appropriate RDF representation for the purposes of linking.

For Web Interface the EUCases Linking Platform relies on customized version of the GraphDB Workbench, developed by Ontotext AD. In the underlying repository engine reasoning and query evaluation are performed over a persistent storage layer. Loading, reasoning and query evaluation proceed extremely quickly even against huge ontologies and knowledge bases. More specifically, for the project purposes GraphDB Standard Edition was selected as a scalable semantic platform. The address of the Web Interface to EUCases Linking Platform is http://graphdb.eucases.eu/.
Abbreviations

DCMI - Dublin Core Metadata Initiative
EUCases-LLOD - EUCases Legal Linked Open Dataset
HTTP - HyperText Transfer Protocol
LOD – Linked Open Data
LLOD – Legal Linked Open Data
OWL – Web Ontology Language
QL – Query Language
RL – Rule Language
RDF – Rich Data Format
SKOS - Simple Knowledge Organization System
SPARQL - Protocol and RDF Query Language
URI – Uniform Resource Identifier
XML – eXtensible Mark-up Language
1 The EUCases Legal Linked Open Dataset

In this deliverable we report on the design and development of EUCases Legal Linked Open Dataset (EUCases-LLOD) for representation of metadata and annotation of EUCases documents. The EUCases-LLOD is a set of ontologies and instance data in RDF model. The ontologies are used to define the conceptual structure of EUCases-LLOD and all RDF data is extracted from EUCases documents. They build on the best practices, such as Dublin Core, EuroVoc, Syllabus, SKOS, etc. The software infrastructure for the EUCases-LLOD is based on the GraphDB RDF store.

The RDF datasets loaded within EUCases-LLOD include the following ontologies and RDF datasets:

1. **EuroVoc** – a thesaurus of European Commission comprising terms for classification of European documents. Each term is represented within all official languages of EU and in some additional languages. The RDF encoding of EuroVoc is done via SKOS ontology and it is available from the following site: http://eurovoc.europa.eu/. It is used in the project for annotation of the content of the EUCases documents.

2. **Legal Taxonomy Syllabus** – an ontology containing European and National legal concepts, developed by UNITO, Turin. It is used in the project for annotation of the content of the EUCases documents.

3. **GeoNames** – a dataset of geographical data represented within RDF. It is used in the project for annotation of the content of the EUCases documents.

4. **Dublin Core** – an ontology for representation of document metadata like authors, date of creation, and other features of the documents. We have implemented a module for conversion from the XML metadata of the document into the RDF representation.

5. **SKOS** – a meta-ontology for mapping between ontologies and lexicons. SKOS was selected for representation of the data, since the main lexical resource used in the project (EuroVoc thesaurus) has been already encoded within SKOS.

6. **PROTON** – a top ontology, used in the creation of several integrated LOD datasets. In the project it will be used as an alternative modeling of geopolitical data in GeoNames. The robustness of its usage is guaranteed by the mapping from PROTON to GeoNames ontology. In future, if necessary, PROTON can be extended in order to play a common ontology for all the datasets included in EUCases-LLOD.

This document represents the main technology behind the creation and maintenance of EUCases-LLOD.
2 Technological Basis of EUCases-LLOD

In this section we briefly describe the main components and technologies used in the implementation of the infrastructure for EUCases-LLOD.

2.1 Linked Data

The notion of *linked data* was defined by Tim Berners-Lee in (Berners-Lee 2006) and (Bizer et al. 2009a) as RDF graphs, published on the WWW so that one can explore them across servers by following the links in the graph in a manner similar to the way the HTML web is navigated. It is considered a method for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web, using URIs and RDF.

Linked data have been created by publishing and interlinking open data sources, following the principles of:

1. using URIs as names of things;
2. using HTTP URIs, so that people can look up these names;
3. providing useful information when someone looks up a URI, using standards like RDF, SPARQL, etc.;
4. including links to other URIs, so people can discover more things.

In fact, most of the RDF datasets fulfil principles 1 and 3 by design. The new element in these principles concerns the requirement for enabling Semantic Web browsers to load HTTP descriptions of RDF resources, based on their URIs. The fourth requirement guarantees that the datasets are not isolated from each other. To this end, data publishers should make sure that:

- the "physical" addresses of the published pieces of data are the same as the "logical" addresses, used as RDF identifiers (URIs);
- upon receiving an HTTP request, the server should return a set of triples that describe the resource;
- re-use identifiers and vocabularies from other datasets, which are also linked data.

Linking Open Data (LOD)⁷ is a W3C SWEO community project aiming to extend the Web by publishing open datasets as RDF and by creating RDF links between data items from different data sources. Linked Open Data provides sets of referenceable, semantically interlinked resources with defined meaning. The central dataset of the LOD is DBPedia, (Bizer et al. 2009b). Due to the many mappings between other LOD datasets and DBPedia, the latter serves as a sort of a hub in the LOD graph, assuring a certain level of connectivity. LOD is rapidly growing – as of September 2014 it contains more than 1014 datasets.

The technology behind LOD and the currently available LOD datasets will be used in EUCases in the following way. First, the semantic data produced via different services within the project will be interconnected with each other and also with datasets from LOD using the technologies of LOD. Also, some of the LOD datasets will be exploited as a source of additional background factual knowledge to be used by the services of EUCases. We will design a URI schema for the entities discovered by the services of the project. Where possible, these URIs will be connected to the conceptual information represented within the other datasets in the infrastructure for EUCases-LLOD. These are EuroVoc, GeoNames, among others. They will include classification of the entities with respect to ontologies loaded

---

⁷ [http://www.w3.org/wiki/SweolG/TaskForces/CommunityProjects/LinkingOpenData](http://www.w3.org/wiki/SweolG/TaskForces/CommunityProjects/LinkingOpenData)
in the infrastructure. For example, GeoNames location entities will be used when recognized in the text.

![Figure 1: Map of Datasets in Linking Open Data](http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/)

### 2.2 Resource Description Format

RDF is a data model based on statements made about resources that can be anything with an associated URI (Universal Resource Identifier) or, recently IRI (Internationalized Resource Identifier). Each statement is a triple: (Subject) – (Predicate) – (Object). Subject is always a URI. Predicate is always a URI. Object is a URI or Literal (RDF/XML) data type.

Examples:

Subject:    <http://dbpedia.org/resource/Robinson_Crusoe>
Predicate: <http://dbpedia.org/ontology/author>
Object:     <http://dbpedia.org/resource/Daniel_Defoe>

In Turtle\(^9\) format:

```
```

The URI can be abbreviated by defining name spaces. In Turtle format of RDF this is done by the @prefix declaration:

```
@prefix dbpont: <http://dbpedia.org/ontology> .
@prefix dbpres: <http://dbpedia.org/resource> .
```

---

\(^8\) [http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/](http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/)

\(^9\) [http://www.w3.org/TR/turtle/](http://www.w3.org/TR/turtle/)
RDF is a standard model for data interchange on the Web. It is the main model for data representation in LOD, also a basis for representation on the Web of ontological information in RDF(S) and OWL. Each LOD dataset is a set of RDF triples which form an RDF Graph. More information can be found at: http://www.w3.org/RDF/.

2.3 SPARQL Query and Update Languages

The first version of SPARQL (Prud'hommeaux and Seaborne, 2008) is an SQL-like query language for RDF data, specified by the RDF Data Access Working Group of W3C. It differs from SQL in that it does not contain specific Data Definition Language (DDL) provisions, because the schemata are represented in both RDFS and OWL as standard RDF graphs, so no specific language is required to deal with them.

Version 1.1 of the SPARQL specification extends the query language with new features, (Harris and Seaborne, 2010), but also adds data modification capabilities with the SPARQL Update specification (Schenk and Gearon, 2010).

The SPARQL Query Language supports four types of queries:

- **SELECT queries** – return tuples of results just like the SELECT queries in SQL;
- **DESCRIBE queries** – return an RDF graph. The resulting graph describes the resources, which match the query constraints. Usually, a description of a resource is considered an RDF-molecule (Li et al, 2009), forming the immediate neighborhood of a URI;
- **ASK queries** – provide a positive or a negative answer indicating whether or not the query pattern can be satisfied;
- **CONSTRUCT queries** – return an RDF graph constructed by substituting the variables in the graph template and combining the triples into a single RDF graph by set union.

The SPARQL Update language allows a range of actions, related to remote management and modification of an RDF repository:

- Updates of data contained in the single RDF graph, including insertion of new data, deletion of data, and loading of new files of data into the repository;
- Creation, deletion and management of named graphs.

SPARQL language will be the primary interface between different services within the EUCases framework and the semantic knowledge represented within the semantic infrastructure of the project. Services that discover new world facts or new conceptual information will use the updated constructs of the language in order to store the new information in the semantic repository. The services that will use the represented semantic knowledge, will search the semantic repository via specially designed SPARQL templates to be filled with service specific information during the actual usage. In this way, the SPARQL language will not be visible to the end user of the EUCases services.

2.4 Reason-able Views

The exploitation of linked data for data management is considered to have a great potential. On the other hand, several challenges need to be handled in order to make this possible. Reason-able views (Kiryakov and Momtchev, 2009) represent an approach for reasoning with and management of linked data defined at Ontotext and implemented in two systems, namely, FactForge (http://factforge.net) and LinkedLifeData (http://www.linkedlifedata.com). FactForge is based on general world knowledge LOD datasets like the DBPedia dataset.
LinkedLifeData is domain oriented and it is used to support biomedical research. In the project we exploit FactForge as a source of background world knowledge.

Reason-able view is an assembly of independent datasets, which can be used as a single body of knowledge with respect to reasoning and query evaluation. The key principles can be summarized as the following instruction:

- Group selected datasets and ontologies in a compound dataset;
- Clean up, post-process and enrich the datasets if necessary. Do this conservatively, in a clearly documented and automated manner, so that (i) the operation can easily be performed each time a new version of one of the datasets is published and (ii) the users can easily understand the intervention made to the original dataset;
- Load the compound dataset in a single semantic repository and perform inference with respect to tractable OWL dialects;
- Define a set of sample queries against the compound dataset. These determine the level of service or the scope of consistency contract offered by the reason-able view. Each reason-able view is aiming at lowering the cost and the risks of using specific linked data datasets for specific purposes. The design objectives behind each reason-able view are as follows:
  - Make reasoning and query evaluation feasible;
  - Lower the cost of entry through interactive user interfaces and retrieval methods such as URI auto-completion and RDF search (a search modality where RDF molecules are retrieved and ranked by relevance to a full-text style query, represented as a set of keywords);
  - Guarantee a basic level of consistency – the sample queries guarantee the consistency of the data in the same way regression tests guarantee the quality of the software;
  - Guarantee availability – in the same way web search engines are usually more reliable than most of the web sites; they also do caching;
  - Easier exploration and querying of unseen data – sample queries provide reusable extraction patterns, which reduce the time for getting to know the datasets and their interconnections.

The reason-able view is important for EUCases project as an approach to linked data, because the linked data extracted from the EUCases documents will interact with other LOD datasets. Our goal is to support the consistency of the domain specific data as much as possible. Cases of contradictory information will be also useful to the business scenarios of EUCases because they would result from different sources and opinions. The implementation of reason-able view is done within GraphDB Workbench using the RDF repository and inference supported by GraphDB.
3 Ontologies

In order to model the data extracted from EUCases documents and to construct a reasonable view, we have considered several ontologies to be put together:

- *EuroVoc* and *Syllabus* are domain modeling ontologies which are used for the annotation of the content in the EUCases documents;
- *GeoNames* ontology describes the structure of GeoNames LOD dataset;
- *Dublin Core* provides a vocabulary for description of document metadata;
- *PROTON* is a general ontology. It plays the important role of a joined ontology for the reason-able view;
- *SKOS* is a metaontology for mapping lexicons and ontologies.

SKOS is used in the distribution of EuroVoc for supporting lexicons for several languages aligned to the EuroVoc term identifiers. It has to be noted that the Lemon Ontology was not exploited for aligning the lexicons due to the following reasons: (1) SKOS is good enough for the goals of EUCases project; (2) The conversion and maintenance of EuroVoc is not a task of the EUCases consortium and it is better to be done by the developers of EuroVoc; (3) The developers of Lemon ontology promised\(^{10}\) that in near future they will provide a convertor from SKOS to Lemon data.

3.1 PROTON Ontology

The PROTON (PROTo ONtology) ontology has been developed in the SEKT project\(^{11}\) as a light-weight upper-level ontology, serving as a modelling basis across different tasks and domains. PROTON serves better than other popular ontologies for tasks related to automatic entity recognition and, more generally, Information Extraction (IE) from text, for the sake of semantic annotation (metadata generation). It also provides solid basis of data integration and RDF-based extraction, transformation and loading (ETL) of data. PROTON is constructed according to the requirements of OntoClean methodology and in this respect it is compatible to DOLCE ontology. This fact would facilitate the usage of sub-hierarchies from DOLCE as extensions of PROTON ontology.

This section describes the third version of PROTON ontology. This new version extends the version from 2005 with new classes and properties in order to cover the conceptual knowledge encoded within the most popular Linking Open Data datasets. The modularization of PROTON is also changed – the number of modules is reduced from four to two for convenience reasons. PROTON v3.0 defines in OWL 2 about 500 classes and 200 properties. Mappings are available between PROTON and the schemata of DBPedia, FreeBase and Geonames. PROTON ontology has been developed by Ontotext and available for usage under Creative Commons license\(^{12}\). The top part of PROTON is given in Fig. 2.

PROTON was designed as a light-weight upper-level ontology for use in Knowledge Management and Semantic Web applications, which implies the following specifics:

- PROTON is relatively un-restrictive. It specifies only a hierarchy of classes and domain and range of properties defined within it, but it does not impose any other restrictions on the meaning of the classes and properties.

---

\(^{10}\) Personal communication.

\(^{11}\) [http://www.ontotext.com/research/sekt](http://www.ontotext.com/research/sekt)

\(^{12}\) [http://creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)
PROTON intentionally does not go deep in modeling some general notions, for instance, the conceptualization of space and time. This is partly because proper models for these notions would require using a logical apparatus, which is beyond the limits acceptable for many of the tasks that PROTON is used for, e.g. querying and management of huge datasets and knowledge bases. Another reason to “keep it simple” is because it is very hard to craft strict and precise conceptualizations for these concepts, which are adequate for a wide range of domains and applications.

Figure 2: A view of the top part of the PROTON class hierarchy
Two additional requirements were set to PROTON, namely to allow for low cost of adoption and maintenance and for scalable reasoning. The first point requires an easy understanding of a set of classes and properties. The second requires a tractable algorithm for reasoning with the represented conceptual knowledge. Thus, the goal is to make the usage of ontologies and the related reasoning infrastructure as a replacement for the use of DBMSs feasible.

Being lightweight, PROTON matches the intuition behind the arguments coming from the Information Science community (Sparck Jones (2004) and Shirky (2005)) that the Semantic Web is more likely to yield solutions to real world information management problems if it is based on partial and relatively simple models of the world, used for semantic tagging.

The design principles behind the ontology can be summarized as follows:

- domain-independence;
- light-weight logical definitions;
- alignment with popular metadata standards;
- good coverage of named entity types and concrete domains (i.e. modelling of concepts such as people, organizations, locations, numbers, dates, addresses); and
- good coverage of instance data in Linked Open Data Reason-able view Fact Forge\(^{13}\).

The design principles behind PROTON are presented in greater detail in (Terziev et al. 2005).

PROTON ontology has two modules: PROTON Top module and PROTON Extension module. Their namespace definitions are:

```xml
@prefix ptop:    <http://www.ontotext.com/proton/protontop#> .
@prefix pext:    <http://www.ontotext.com/proton/protonext#> .
```

### 3.2 SKOS Ontology

SKOS (Simple Knowledge Organization System) is a W3C standard, based on other Semantic Web standards (RDF and OWL). It provides a structured way to represent the content of various controlled vocabularies, taxonomies and thesauri. SKOS itself is an OWL ontology and as such it can be presented in any RDF-based syntax. In this way, the resources, encoded in SKOS, are ready to be mapped as LOD ones. The namespace definition is:

```xml
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
```

In basic SKOS, conceptual resources (concepts) are identified with URIs, labeled with strings in one or more natural languages, documented with various types of note, semantically related to each other in informal hierarchies and association networks, and aggregated into concept schemes.

In advanced SKOS, conceptual resources can be mapped across concept schemes and grouped into labeled or ordered collections. Relationships can be specified between concept labels. The SKOS vocabulary itself can be extended to suit the needs of particular communities of practice or combined with other modeling vocabularies.

SKOS provides three properties to attach labels to conceptual resources: `skos:prefLabel`, `skos:altLabel` and `skos:hiddenLabel`.

\(^{13}\) [http://www.ontotext.com/factforge](http://www.ontotext.com/factforge)
The `skos:prefLabel` property makes it possible to assign a preferred lexical label to a resource. For example, the concept “lawyer” has a preferred label ‘lawyer’.

```
ex:lawyer rdf:type skos:Concept;
   skos:prefLabel "lawyer".
```

The `skos:altLabel` property makes it possible to assign an alternative lexical label to a concept. This is especially helpful when assigning labels beyond the one that is preferred for the concept, for instance when synonyms need to be represented. For example, the concept “lawyer” can have a synonym, presented as an alternative – ‘attorney’.

```
ex:lawyer rdf:type skos:Concept;
   skos:prefLabel "lawyer";
   skos:altLabel "attorney".
```

A hidden lexical label, represented by means of the `skos:hiddenLabel` property, is a lexical label for a resource, where a KOS designer would like that character string to be accessible to applications performing text-based indexing and search operations, but would not like that label to be visible otherwise. Hidden labels may for instance be used to include misspelled variants of other lexical labels.

SKOS offers the following types of relations: hierarchical, associative and mapping. The hierarchical relationships are `skos:broader` and `skos:narrower`. The associative relation is `skos:related`. It represents various types of relations among the concepts that are of non-hierarchical nature.

Here is an example, where the concept “Computer” is broader that the concept “Laptop” and also is related to the concept “Software”.

```
ex:Computer rdf:type skos:Concept;
   skos:prefLabel "Computer"@en;
   skos:broader ex:Laptop ;
   skos:related ex:Software .
```

SKOS has two mapping relationships, `skos:closeMatch` and `skos:exactMatch`, which allows to represent a mapping between different concepts. The `skos:closeMatch` indicates that two concepts are sufficiently similar and both concepts may be used interchangeably, but it is not transitive.

The `skos:exactMatch` denotes a higher degree of similarity, namely, both concepts have the same meaning. This relationship here is transitive. In the next example the concept “Netbook” is matched closely to the concept “Laptop”, while the concept “Portatil” is exactly matched to it, but it is in Spanish.

```
ex:Laptop rdf:type skos:Concept;
   skos:prefLabel "Laptop"@en;
   skos:prefLabel "Portatil"@es;
   skos:narrower ex:Computer ;
   skos:closeMatch ex:Netbook ;
   skos:exactMatch ex2:Portatil.
```

In the project we exploit SKOS to represent the hierarchy of EuroVoc terms and the aligned lexicons.
3.3 Dublin Core

The Dublin Core Metadata Element Set\textsuperscript{14} is a vocabulary of fifteen properties for use in resource description. The elements are broad and generic, thus usable for describing a wide range of resources.

At the moment this set is maintained by Dublin Core Metadata Initiative (DCMI), including properties, vocabulary encoding schemes, syntax encoding schemes, and classes.

The 15 core elements are as follows:

1. Contributor (An entity responsible for making contributions to the resource.)
2. Coverage (The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant)
3. Creator (An entity primarily responsible for making the resource)
4. Date (A point or period of time associated with an event in the lifecycle of the resource)
5. Description (An account of the resource)
6. Format (The file format, physical medium, or dimensions of the resource)
7. Identifier (An unambiguous reference to the resource within a given context)
8. Language (A language of the resource)
9. Publisher (An entity responsible for making the resource available)
10. Relation (A related resource)
11. Rights (Information about rights held in and over the resource)
12. Source (A related resource from which the described resource is derived)
13. Subject (The topic of the resource)
14. Title (A name given to the resource)
15. Type (The nature or genre of the resource)

There is also an extended list of elements/terms\textsuperscript{15}. Each term is specified with the following minimal set of attributes:

1. Name: A token appended to the URI of a DCMI namespace to create the URI of the term.
2. Label: The human-readable label assigned to the term.
3. URI: The Uniform Resource Identifier used to uniquely identify a term.
4. Definition: A statement that represents the concept and essential nature of the term.
5. Type of Term: The type of term as described in the DCMI Abstract Model [DCAM].

Additional information about the term or its application can be specified:

1. See: Authoritative documentation related to the term.
2. References: A resource referenced in the Definition or Comment.
3. Refines: A Property of which the described term is a Sub-Property.
4. Broader Than: A Class of which the described term is a Super-Class.

\textsuperscript{14} http://dublincore.org/documents/dces/
\textsuperscript{15} http://dublincore.org/documents/2012/06/14/dcmi-terms/
5. Narrower Than: A Class of which the described term is a Sub-Class.
6. Has Domain: A Class of which a resource described by the term is an Instance.
7. Has Range: A Class of which a value described by the term is an Instance.
8. Member Of: An enumerated set of resources of which the term is a Member.
9. Instance Of: A Class of which the described term is an instance.
11. Equivalent Property: A Property to which the described term is equivalent.

We have selected classes and properties from the two specifications. We use the following namespace definition:

```xml
@prefix dcterms: <http://purl.org/dc/terms/> .
```

We are using the extended version in order to represent the metadata of the EUCases documents and the annotation of the documents as described in Deliverable D2.3. Here is a list of used elements:

- `dcterms:accessRights` - the rights for accessing the document
- `dcterms:abstract` - an abstract of the content of the document (usually generated automatically)
- `dcterms:contributor` - a person or an organization contributed to the content of the document
- `dcterms:coverage` - geographical location in which the document is valid
- `dcterms:creator` - the creator of the document: a person or an organization
- `dcterms:date` - the date of the document
- `dc:description` - the keywords of the document
- `dc:identifier` - EUCases identifier (URI) of the document
- `dcterms:isReplacedBy` - a new ECLI identifier
- `dcterms:issued` - the date of publication of the document
- `dcterms:isVersionOf` - an ECLI identifier
- `dcterms:language` - a language code
- `dcterms:publisher` - the publisher of the document (an organization)
- `dcterms:references` - references to external entities
- `dcterms:subject` - classification of the document
- `dcterms:title` - the full title of the document
- `dcterms:type` - type of decision

### 3.4 EuroVoc Thesaurus

EuroVoc is one of the most popular multilingual, multidisciplinary thesauri covering mainly the activities of the EU, and the European Parliament in particular. It contains terms in 23 EU languages (Bulgarian, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovenian, Spanish and Swedish), plus Serbian. It also covers many specific domains, such as Law, Finance, Trade, Energy, Science, etc.
EuroVoc is managed by the Publications Office, which uses ontology-based thesaurus management and semantic web technologies conformant to W3C recommendations. EuroVoc users include the European Parliament, the Publications Office, national and regional parliaments in Europe, plus national governments and private users around the world. EuroVoc is a controlled set of vocabulary which can be used outside the EU institutions, particularly by parliaments. The aim of the thesaurus is to provide the information management and dissemination services with a coherent indexing tool for the effective management of their documentary resources and to enable users to carry out documentary searches using controlled vocabulary.

The EuroVoc ontology is an extension of SKOS (Simple Knowledge Organization System) – W3C recommendation. The EuroVoc thesaurus "eu:EuroVoc" is defined as an instance of "eu:Thesaurus", itself a subclass of the SKOS "Concept Scheme" class. To this "eu:EuroVoc" instance is attached the list of languages for which the EuroVoc thesaurus concepts have a preferred label (via skos:prefLabel or xml:literalForm). The thesaurus supported languages are defined by values of the property "eu:supportedLanguage". Values of this property are instances of the class "eu:Language". The attribute rdfs:label is used to represent the language independent name of the Thesaurus (EuroVoc). The "eu:Language" class is a convenience class where each instance represents a language. The language names of the instances are represented using "rdfs:label" in each of the supported languages. The property value "eu:language" contains the ISO 639-2 Language Code (conforming the xsd:language value space). By convention, the URI of an "eu:Language" instance is the registered public subject indicator (see http://psi.oasis-open.org/iso/639/).

EuroVoc thesaurus is used in the project for classification of legal documents and for in text annotation of EUCases documents. In the LOD dataset we use them as annotation of the whole documents.

### 3.5 Legal Taxonomy Syllabus

The Legal Taxonomy Syllabus is described in detail in Deliverable D1.1. The Legal Taxonomy Syllabus was designed to help legal professionals acquire deep understanding of laws in European and other national jurisdictions. The goal of the ontology necessitated a completely new design and level of detail in comparison to ontologies designed for information retrieval or other semantic web applications. The most important insight from lawyers, which informed the design, was that the meaning of a legal term depends on context (jurisdiction, domain, legislation, timeframe). The ontology framework was designed in a way which makes these considerations explicit, so that users can determine which definitions are most relevant. To the ontology lexicons in the project languages were created and aligned.

Syllabus is used in the project, similarly to EuroVoc, for classification of legal documents and for in text annotation of EUCases documents.

In the next section we will present how we use these ontologies in order to model EUCases documents.
4  EUCases Document Modeling

Each EUCases document is an XML document valid with respect to the legal XML schema Akoma Ntoso presented in deliverable D2.2 Legal XML-schema (XSD). Here we present the main elements of the header of each EUCases document and its mapping to the ontologies selected for EUCases-LLOD.

There are two types of EUCases documents: Act (covering legislative documents) and Judgment (covering case law documents). The structure of each EUCases document is divided into two: metadata and content. The metadata determines the type of the document, its life cycle, including the date of creation, the author(s), the place of creation, keywords, abstract, etc. The content of the document is the actual text of the document. The RDF representation of each EUCases document encodes information coming from both sources—the metadata and the content. The metadata is already represented explicitly during the creation of the XML representation of the document. The content of the document is described on the basis of the text annotation of the document via NLP and the linking tools developed in workpackages 2 and 3 of the project. Each annotation is represented as having a body and a target. The target of each annotation in EUCases project is an EUCases document whose content is annotated with the corresponding information. The body is the additional information provided by the annotation. This additional information is represented as a URI pointing to a geopolitical entity described in the GeoNames dataset; an external document represented via the URL of the document; and/or a concept from an annotation ontology (in the project we exploit EuroVoc Thesaurus and Legal Taxonomy Syllabus).

4.1 Namespaces for EUCases project

All created instances of EUCases documents, metadata elements and annotations are represented as instances in the EUCases dataset. In order to represent them uniformly, we use for all EUCases instances the following namespace:

@prefix eucinst: <http://www.eucases.eu/lod/instances#> .

In our work we model all the necessary information using existing ontologies, but in case of necessity to extend some of the ontologies, new classes and properties will be defined within the following namespace:

@prefix eucont: <http://www.eucases.eu/lod/ontology#> .

The usages of these name spaces are given below.

4.2 Ontology modelling

In this section we describe the main classes of EUCases ontology exploited within the project to represent conceptually the EUCases documents. We start with PROTON ontology and extend it with a definition of new classes and properties where necessary or via mapping to the other ontologies mentioned above. In EUCases-LLOD we represent the two types of documents: acts and judgments. For each of them we encode information coming from metadata section of the document and the content of the document. The EUCases document is modelled as a sub-class of the PROTON class ptop:Document:

```turtle
ptop:Document
  rdf:type owl:Class ;
  rdfs:comment "The information content of any sort of document. The tangible aspects are ignored. It is usually a document in free text with no formal structure or semantics."@en ;
  rdfs:label "Document"@en ;
  rdfs:subClassOf ptop:InformationResource .
```
ptop:Document has the following important for EUCases properties (directly defined for it or inherited): ptop:documentAbstract; ptop:documentSubTitle; ptop:derivedFromSource; ptop:hasContributor; ptop:hasDate; ptop:hasSubject; ptop:inLanguage; ptop:informationResourceCoverage; ptop:informationResourceIdentifier; ptop:informationResourceRights; ptop:resourceType; ptop:title. These properties comply with Dublin Core ones.

The EUCases document has two sub-classes for acts and for judgments. The following image represents the hierarchy of the documents:

On the level of eucont:EUCDocument class we define all the properties that are necessary for the representation of EUCases documents and that are shared by the two subclasses of documents - acts and judgments, for example eucont:reference which has domain and range eucont:EUCDocument. The properties specific for eucont:Act and eucont:Judgment are defined locally.

Document identifiers can be of several types: Akoma Ntoso identifier, National identifier, EUCases identifier, ELI identifier. Each identifier is an URI pointing to the different representation of the document. The first identifier is used as instance identifier of the document.

Language of the document is represented as language code with respect to ISO 639-2 standard. The language is represented by DC property dcterms:language which is mapped to PROTON property ptop:inLanguage (see below). Behind the actual language of the document we also represent the original language of the document via the property eucont:originalLanguage.

A EUCases document could have several titles. The full title is represented by the DC property dcterms:title. The other types of title (short title, abbreviation, colloquial title) are
represented by EUCases specific properties: eucont:shortTitle, eucont:abbreviation, eucont:colloquialTitle. All of them are subproperties of the PROTON property ptop:title.

The EUCases document type is represented by the DC property dcterms:type which is defined to be a subproperty of PROTON property ptop:resourceType.

The history of a EUCases document is presented as a sequence of events (or states) like document creation, document publication, document signature, etc. Events and states are modelled by the PROTON class ptop:Happening. Each happening is determined by its beginning and end time moment as well as the participants. For example, the document creation is done in some period of time, by some legislative authority; a change of document also can be done by legislative authority in a given time interval. Although such document related events (or states) are well perceived by the users of EUCases-LLOD, their representation in many cases is partial or unnecessarily complex. In order to avoid the complex nature of the document events in the current version of the EUCases ontology we encode only time stamps for some of the events. These time points or intervals are encoded by several EUCases properties whose names reflect the document events that happened at a given moment or interval. Such properties are eucont:documentDate - the date of the creation of the document; eucont:publicationDate - the date of the publication of the document; eucont:effectDate and eucont:validityDate - determine the period in which the document is in force. The time properties are subproperties of the PROTON property ptop:hasDate. In many cases the events and their participants are unique. In these cases the participants in the events can be also expressed by appropriate properties like eucont:hasPublisher. If in future it is necessary to extend the representation of provenance of EUCases documents to more detailed descriptions ontologies like PROV\(^\text{16}\) can be exploited.

Classification of a EUCases document is done via a set of keyword references. The keyword references point to terms in a thesauri like EuroVoc or Legal Taxonomy Syllabus in our case. Such a classification has a source which can be some real agent - Person or Organization; or software agent like EUCases NLP ToolKit. Each classification is represented by the EUCases class eucont:Classification. The class eucont:Classification is a subclass of the PROTON class ptop:InformationResource. The property eucont:hasSource represents the source of the classification. A classification is attached to a document by the property eucont:hasClassification. The property eucont:hasKeyword connects a classification with its keywords. Each keyword is an instance of the class eucont:Keyword which is a subclass of the PROTON class ptop:Topic. The SKOS class skos:Concept is a subclass of eucont:Keyword. Thus, using SKOS representation of EuroVoc thesaurus we can use EuroVoc terms as keywords in the EUCases classifications. This hierarchy is represented in the following diagram:

\(^{16}\) http://www.w3.org/TR/prov-o/
Geographical characteristics of the EUCases documents are modelled in two ways. First, the metadata geographical features determine the area in which the document is in force. These features are modelled by DC property dcterms:coverage with rdfs:range restricted to PROTON class pext:PoliticalRegion. The second kind of geographical information is extracted automatically from the content of the document. This kind of information is represented by property eucont:hasLocationReference which is also restricted to take instances of the concept pext:PoliticalRegion as values.

The summary of an EUCases document is extracted automatically by the EUCases NLP ToolKit. It is modelled by the DC property dcterms:abstract.

In order the different ontologies to be exploited together we also provide mapping rules between them. The mapping rules are expressed by RDFS properties rdfs:subClassOf and rdfs:subPropertyOf. Here we provide some examples of such rules.

**Mapping from PROTON ontology to GeoNames ontology**

The namespace for GeoNames is:

```prefix geo-ont: <http://www.geonames.org/ontology#> .
[rdf:type owl:Restriction ;
  owl:onProperty <http://www.geonames.org/ontology#featureCode> ;
  owl:hasValue <http://www.geonames.org/ontology#A.ADM4> ]
rdfs:subClassOf pext:PoliticalRegion .```

The first part of the rule (with the brackets []) expresses the concept of political region in the terms of GeoNames. The same concept is modelled within PROTON as the class pext:PoliticalRegion. The PROTON Ontology is provided with a full set of such mapping rules to cover the whole GeoNames dataset.

**Mapping from PROTON ontology to EUCases ontology**

EUCases specific classes and properties are represented as an extension of the PROTON ontology. The rules for the mapping between them are of the following types:

eucontLEUCDocument rdfs:subClassOf ptop:Document .
eucont:shortTitle rdfs:subPropertyOf ptop:title .

The same kind of rules are used for mapping between the other ontologies. We are using subclasses and subproperties statements for the mapping in order not to enforce the equivalence between PROTON classes, properties and statements and the other ontologies.
In this way we believe that the resulting EUCases-LLOD dataset will not depend too much on PROTON ontology and, if necessary, other ontologies can be used instead of PROTON.

In the next section we describe the procedure for converting EUCases documents from Akoma Ntoso XML representation into a set of RDF triples with respect to the EUCases ontology.

### 4.3 Creation of RDF representation of the EUCases documents

The process for RDFization of a given EUCases document comprises the following steps:

2. For each element of the document which provides information for the RDF representation one or more appropriate new XML elements are added.
3. For each added triple element the module computes the subject, predicate and object URIs. In some cases object can be a literal – concrete value: text, number, date, etc.
4. The triples with defined subject, predicate and objects are extracted from the document and converted into actual RDF Triples.
5. The created sets of RDF triples for a given document are loaded into the EUCases RDF repository. During this process the new RDF triples are checked for consistency with the rest of the EUCases Dataset; new information following from the inference rules is also added to the repository.

During the addition of the RDF representation to the RDF repository the annotations from the document resolve their bodies to the corresponding instances that are already loaded into the repository, such as GeoNames instances, EuroVoc terms or Syllabus classes.

Here are some examples of rules:

**Creation of document instance:**

**XML representation:**

```xml
<FRBRthis value="/akn/eu/act/regl/cons/2000-12-22/32001r0044/main" />
```

**RDF representation:**

```xml
eucinst:akn/eu/act/regl/cons/2000-12-22/32001r0044/main
rdf:type eucont:EUCDocument .
```

**Language of the document:**

**XML representation:**

```xml
<FRBRlanguage language="bul" />
<FRBRlanguage eId="#OriginalLanguage" language="bul" />
```

**RDF representation:**

```xml
eucinst:akn/eu/act/regl/cons/2000-12-22/32001r0044/main
dcterms:language "bul"@en ;
eucont:originalLanguage "bul"@en .
```

**Document classification:**

**XML representation:**

```xml
<classification source="#eucases.eu/nlp-toolkit/summarization"/>
```
<keyword dictionary="eurovoc" eId="4622" showAs="transport user" value="eurovoc/4622" />
<keyword dictionary="eurovoc" eId="195" showAs="airport" value="eurovoc/195" />
<keyword dictionary="eurovoc" eId="497" showAs="damage" value="eurovoc/497" />
<keyword dictionary="eurovoc" eId="1339" showAs="indemnification" value="eurovoc/1339" />
</classification>

RDF representation:

@prefix eurovoc: <http://eurovoc.europa.eu/> .
eucinst:akn/eu/act/regl/cons/2000-12-22/32001r0044/main
eucont:hasClassification
eucinst:akn/eu/act/regl/cons/2000-12-22/32001r0044/main_classification001 .
eucinst:akn/eu/act/regl/cons/2000-12-22/32001r0044/main_classification001
eucont:hasSource
eucinst:eucases.eu/nlp-toolkit/summarization .
eucinst:akn/eu/act/regl/cons/2000-12-22/32001r0044/main_classification001
eucont:hasKeyword eurovoc:4622 ;
eucont:hasKeyword eurovoc:195 ;
eucont:hasKeyword eurovoc:497 ;
eucont:hasKeyword eurovoc:1339 .

Here the actual terms in different languages will be available by the SKOS representation of EuroVoc.
5 GraphDB RDF Repository

GraphDB is a high-performance semantic repository created by Ontotext. It is implemented in Java and packaged as a Storage and Inference Layer (SAIL) for the Sesame RDF framework. GraphDB is a native RDF rule-entailment and storage engine. The supported semantics can be configured through rule-set definition and selection. Included are rule-sets for OWL-Horst, unconstrained RDFS with OWL Lite and the OWL2 profiles RL and QL. Custom rule-sets allow tuning for optimal performance and expressivity.

Reasoning and query evaluation are performed over a persistent storage layer. Loading, reasoning and query evaluation proceed extremely quickly even against huge ontologies and knowledge bases.

GraphDB can manage billions of explicit statements on desktop hardware and can handle tens of billions of statements on commodity server hardware.

Features of GraphDB include:

- Pure Java implementation, ensuring ease of deployment and portability;
- Compatible with Sesame 2, which brings interoperability benefits and support for all major RDF syntaxes and query languages;
- Compatible with Jena with a built in adapter layer;
- Customisable reasoning, in addition to RDFS, OWL-Horst, and OWL 2 RL support;
- Optimized owl:sameAs handling, which delivers dramatic improvements in performance and usability when huge volumes of data from multiple sources are integrated.
- Clustering support brings resilience, fail-over and scalable parallel query processing;
- Geo-spatial extensions for special handling of 2-dimensional spherical data allowing data using the WGS84 RDF vocabulary to be indexed and processed quickly using a variety of special geometrical query constructions and SPARQL extensions functions;
- Full-text search support, based on either Lucene or proprietary search techniques;
- High performance retraction of statements and their inferences – so inference materialisation speeds up retrieval, but without delete performance degradation;
- Powerful and expressive consistency/integrity constraint checking mechanisms;
- RDF rank, similar to Google’s PageRank, can be calculated for the nodes in an RDF graph and used for ordering query results by relevance, visualisation and any other purposes;
- RDF Priming, based upon activation spreading, allows efficient data selection and context-aware query answering for handling huge datasets;
- Notification mechanism, to allow clients to react to statements in the update stream.

GraphDB family versions are:

- **GraphDB Lite** is fast semantic repository, supporting non-trivial inference with tens of millions of statements on contemporary desktop hardware.
- **GraphDB Standard** is an extremely scalable semantic repository: it can load tens of billions of RDF statements, using non-trivial inference and delivering outstanding performance.

---

17 Some features are not available in GraphDB-Lite edition (refer http://ontotext.com/products/ontotext-graphdb/ for details)
multi-user query performance. GraphDB Standard is a robust engine packed with advanced features that bring unmatched efficiency to a huge variety of application scenarios:

- optimized owl:sameAs handling that delivers dramatic improvements in performance and usability when huge volumes of data from multiple sources are integrated
- hybrid querying capabilities that combine SPARQL with efficient full-text search, geo-spatial constraints and ranking of query results

- **GraphDB Enterprise** is a replication cluster infrastructure based on GraphDB-Standard. It offers industrial strength resilience and linearly scalable parallel query performance, with support for load-balancing and automatic fail-over.

- **GraphDB Cloud** is an extension of GraphDB Standard with the added advantage that it has been architected to run on the Amazon Cloud as a pay-by-the-hour service.

For the EUCases project we use GraphDB Workbench, part of the GraphDB Standard Edition. The front page of the user interface to the repository is shown in Fig. 3.

![EUCases Workbench](image)

**Figure 3:** The front page of the EUCases installation of GraphDB Workbench.

The EUCases-LLOD dataset is uploaded within GraphDB Workbench and gradually updated. SPARQL endpoint provides access to the dataset. For end user applications appropriate SPARQL query templates will be implemented for facilitating the retrieval of the necessary information. The address of the Web Interface to EUCases Linking Platform is http://graphdb.eucases.eu/.
6  EUCases-LLOD Support and Availability

The EUCases-LLOD is available for use by third parties in two ways:

- SPARQL endpoint to the GraphDB repository;
- As an archive of RDF files in n3 format.

The access via SPARQL endpoint allows the user to specify SPARQL queries and evaluate them over the repository. The screen for SPARQL endpoint is given below:

The users could evaluate some of the existing queries or write their own. Here we present a query to provide an example of interaction with the system. The query is:

```
PREFIX eucont: <http://www.eucases.eu/lod/ontology#>
PREFIX gn: <http://www.geonames.org/ontology#>
PREFIX ptop: <http://www.ontotext.com/proton/protontop#>
PREFIX geonames: <http://sws.geonames.org/>
WHERE {
  ?c gn:name ?cname
} LIMIT 1000
```

The query is searching for a document (?doc) which is EUCases document; its title (?title); its source URI (?uri), which is where the document was downloaded from; the country in which the document was created (?c) and its name (?cname). This country has to be located in Europe (<http://sws.geonames.org/6255148/>). The query is entered in the edit box for SPARQL query. If necessary, the system adds the necessary prefixes for space names. The number of answers is limited to 1000. If the users need more answers, then they have to delete this restriction. Here are the screenshots of the interactions with the system.
When pressing the Submit button, the query is sent for evaluation. The result is given in a table format. Each column corresponds to a variable which the users have been searching for.

By clicking on the corresponding links in the table, the users could explore the additional information for the objects pointed by the links. First, the system shows the information available in the repository for the corresponding object. Then, the user could exploit some external links to reach information on the web that lacks in the RDF repository of EUCases-LLOD. In the following description we provide two types of such information. The first is related to the source URI of the document. Inside the repository there is no additional information for the external URI (the document on the web) because the extracted information is assigned to the document internal URI (presented in the first column of the table). Thus, the only possibility here is to follow the external link to the original document. The two screenshots below show this possibility for the first row about document from United Kingdom of Great Britain and Northern Ireland.
Following the source link we go to the original document:

**PRESS SUMMARY**

28 October 2012

Rahie and another (Respondents) v Euromine SA and others (Appellants) and New Cap Reinsurance Corporation (In Liquidation) and another (Respondents/Cross Appellants) v A R Grant and others as Members of Lloyd's Brokers 991 for the 1997 Year of Account and another (Appellants/Cross Respondents) [2002] UKSC 46


**JUSTICER** Lord Collins Lord Walker, Lord Mance, Lord Clark, Lord Sumption.

**BACKGROUND TO THE APPEALS**

The two appeals concern whether, and if so, in what circumstances, an order or judgment of a foreign court in proceedings to set aside prior transactions, such as preferences or transactions at an

Following the link for the country provides much more information inside repository and also a link to the original GeoNames page:

The table shows the properties defined for the country and their values.
This example illustrates how EUCases-LLOD is linked to other information on the web. The second way for giving access to the data is as XML sources and RDF triples. The webpage providing access to the two archives is:

http://download.webclark.org/EUCasesLOD/

From there the users could download the necessary information. The dataset EUCases-LLOD has been also published using the Open Data Platform (ODN) developed by the FP7 SME-DCA call COMSODE project:

http://eucasesodn.di.unito.it:8890/sparql

Both mechanisms for access to the EUCases-LLOD data will be supported by the Institute of Information and Communication Technologies at Bulgarian Academy of Sciences (IICT-BAS) for at least two years after the end of the project. Taking into account the serious professional interest of IICT-BAS in Language and Semantic Technologies, we expect the data and the service to be available for a longer period. Additional requests, comments and bug reports are very welcome. They have to be sent to Kiril Simov: kivs@bultreebank.org.

The dataset EUCases-LLOD will be further extended by adding new documents when they are available. Also the conceptual model will be further developed.
7 Conclusion

In the deliverable we described the ontological modeling of EUCases documents as conceptual basis for the creation and maintenance of EUCases-LLOD dataset. Description of the relevant technologies is provided with appropriate references. The dataset is implemented as a reason-able view including existing LOD dataset uploaded in a common RDF store. In this way all the necessary information is represented on the same server and efficient performance will be ensured.
References


Harris, S.; Seaborne, A. (eds.). 2010. SPARQL 1.1 Query Language.
   http://www.w3.org/TR/2010/WD-sparql11-query-20101014/


   http://www.shirky.com/writings/ontology_overrated.html


   (http://proton.semanticweb.org/D1_8_1.pdf).