LegalHTML: A Representation Language for Legal Acts

Armando Stellato(✉) and Manuel Fiorelli

Department of Enterprise Engineering, University of Rome Tor Vergata, Via del Politecnico 1,
00133 Rome, Italy
{stellato,manuel.fiorelli}@uniroma2.it

Abstract. The Publications Office (OP) of European Union (EU) expressed the need to simplify the Official Journal production workflow, which required different formats and, consequently, document instances at different stages of the process. We met this need by developing LegalHTML, which unifies the formal, structural and semantic representation of legal acts, as well allowing for diverse typographic requirements for publication. This streamlines the production workflow and publication/fruition of content as well, since a single document instance is first drafted and then incrementally enriched. LegalHTML consists of an extension of HTML for the structural representation of legal acts (e.g., articles, paragraphs, items, and references), while a supplementary ontology enables the annotation (using RDFa) of domain references (e.g., signatories, people and their role in organizations, the scope of the document). LegalHTML also supports the consolidation of an act and its subsequent changes into a single document using a tree-based representation. Finally, we implemented a CSS stylesheet for the default rendering of the model and a JavaScript file imbuing documents with an API that supports TOC generation, footnote cross-references and point-in-time visualization of legal acts.

Keywords: legal document · consolidation · metadata · HTML · Semantic Web

Resource type: HTML extension for the legal domain and support ontology
License: European Union Public License 1.2 (EUPL-1.2)
Linked Open Vocabularies (LOV): https://w3id.org/legalhtml/
DOI: https://doi.org/10.5281/zenodo.7454918
URL: https://w3id.org/legalhtml/

1 Introduction

With the large improvements in all aspects of information representation, management and content storage, a new generation of repositories for legal content, with documents rich in metadata, annotations, and cross-references, emerged.

Concrete actions such as the European Directive on open data and the re-use of public sector information [1], also known as the Open Data Directive, and initiatives such as Open Gov [2], have encouraged and possibly pushed institutions, governments, and any

C. Pesquita et al. (Eds.): ESWC 2023, LNCS 13870, pp. 520–537, 2023.
https://doi.org/10.1007/978-3-031-33455-9_31

sort of actor in the public administration on publishing and sharing their data according to open standards and best practices.

With such scenario occurring and evolving at a fast pace, it is important to share and adopt solutions supporting the representation of legal content and of its jurisdictional existence and lifecycle (e.g., when a regulation is published, enters into force, etc..), facilitating all aspects of content production, publication and fruition.

In 2021, we conducted a study, funded by the Publications Office (OP) of the European Union (EU), exploring the possibility of an efficient solution for the publication of legal acts, streamlining an overly complex publication workflow that included several steps, including drafting, proof-reading, finalization and production of several manifestations scoped to different objectives, such as official journal publication, semantic indexing, dissemination, etc.

With the intent of covering the aforementioned objectives through a single solution, we have realized an HTML-based language for representing legal content, dubbed LegalHTML, featuring all structural aspects of an act, such as articles, paragraphs, items, references, supporting a semantically-rich representation of all such elements and references to entities of the legal domain. Furthermore, LegalHTML addresses consolidation of an act and its subsequent modifications into a single document using an efficient tree-based model. Finally, we imbued the document model with API supporting rendering and point-in-time visualization of legal acts. Metadata, consolidation information and other relevant information are represented by a dedicated ontology. In fact, other ontologies and controlled vocabularies can be combined to ground LegalHTML in different legal traditions (i.e., different ways to represent laws in different countries), without violating the integrity and generality of the language. The outcome of the study has been recently accepted and scheduled for adoption by the Publications Office.

2 Related Work

We can identify two classes of works that are relevant to our contribution, either bound by the domain (the legal domain) or the similarity of the approach (extending HTML).

For what concerns the first class, several initiatives for legal document publishing eventually developed XML formats for the representation of legal acts; these initiatives originated from individual countries or from international efforts seeking at accommodating different jurisdictions and legal traditions [3].

CEN MetaLex [4] has been designed in the context of *CEN Workshop on an Open XML Interchange Format for Legal and Legislative Resources* (*MetaLex*), while the name comes from a substantially different model developed in a previous effort focused on Dutch legislation [5]. CEN MetaLex facilitates legal information exchange among software applications, by establishing a least common denominator between different jurisdiction-specific standards and vendor-specific formats. Interestingly, this standard foresees the translation of documents to RDF using a dedicated ontology, and the possibility of external RDF assertions on parts of the document each identified by a URI.

Substantial efforts toward a legal representation standard were pursued on both sides of the Atlantic Ocean, through the foundation of Legal XML [6] in 1998 and LEXML [7] in 2000, in the USA and Europe, respectively. While Legal XML pursued the agreement on a single schema per document type, the diversity of jurisdictions in Europe drove LEXML toward a bottom-up approach with different communities developing their own schemas, in the hope that eventually there would be a limited set of schemas to make the establishment of mappings feasible. With this regard, a Legal RDF Dictionary, being developed in a concerted effort between the two initiatives, would be key to integration of different schemas [8]. This effort was inspired by John McClure's ideas on Legal-RDF [9], utilizing an ontology to represent the structure and meaning of legal documents through inline annotations of XHTML documents.

Akoma Ntoso (Architecture for Knowledge Oriented Management of African Normative Texts using Open Standards and Ontologies) [10, 11], often shortened as AKN, is an international standard for the technological representation of judicial, legislative, and parliamentary documents. AKN was created by the initiative of the "Africa i-Parliament Action Plan", which is a program of the UNDESA (United Nations Department of Economic and Social Affairs). The project aims to achieve transparency, open access, exchange, and ultimately the maximum democratization of legal information produced by parliaments, courts, and government institutions. Past its original conception AKN has further development by the aforementioned Legal XML as an OASIS [12] standard, called OASIS LegalDocumentML, meant to offer specifications for a standard of legal documents of parliamentary, legislative and judicial origin. Akoma Ntoso addresses interoperability concerns including document identification, structure, and semantics. Akoma Ntoso was designed as an XML application, discarding alternatives such as HTML which – specifically – was considered weak on structural constraints, and geared much more towards presentation than structure and semantics. Having partnered with the Library of Congress in a data challenge aiming at the conversion of national legislation schemas to Akoma Ntoso, the United Kingdom National Archives not only adopted that standard but also developed an HTML serialization of Akoma Ntoso [13]. Nonetheless, these are just derivate formats, which are generated from their own XML format, the Crown Legislation Markup Language (CLML).

FORMEX (Formalized Exchange of Electronic Publications) [14] is the standard used by the Publications Office of the EU to exchange data with service providers. Introduced in 1985 as an SGML application defined by a DTD, the fourth revision of the standard became an XML application defined by an XML Schema. FORMEX captures the structure of documents published in the Official Journal of the EU.

A complement to legal document formats is represented by metadata vocabularies for legal acts: the European Legislation Identifier (ELI) is a European effort at harmonizing the way legislation is published. ELI's ontology is based on three pillars [15]:

1. every piece of legislation is identified by an HTTP URI;
2. the same metadata elements are used across the different jurisdictions;
3. metadata is shared in a machine-readable form, reusing the ELI Ontology.

The implementation level of ELI varies (only a core part of ELI is shared among various jurisdictions and indeed each EU country coined its own specialization of the ELI ontology for metadata representation of documents).

A more recent effort combining document formats and ontologies is represented by Lynx [16, 17]. In Lynx, a knowledge graph for the legal domain (Legal Knowledge

Graph, LKG) has been generated and made available for semantic processing, analysis, and enrichment of documents from the legal domain. As the objectives of Lynx did not require the complexity provided by document models such as Akoma Ntoso, they realized a simpler document model based on annotations taken after an ad-hoc ontology, the Legal Knowledge Graph Ontology.

A further, orthogonal, exploration on the semantics dimension of legal acts is given by LegalRuleML [18], an OASIS standard for writing legal rules in a machine-readable format. It extends RuleML with features specific to legal norms, such as defeasibility, deontic operators, negation and temporality. Developed by (part of) the same authors of Akoma Ntoso and thought to complement its representational aspects with reasoning and deontics, LegalRuleML can in fact be adopted in other legal document models.

Moving to other solutions sharing with LegalHTML the approach (extending HTML) rather than the domain, dokieli [19] is a browser-based platform that uses several web standards to enable decentralized authoring, publication, and annotation of documents. The "default UI" of a document in dokieli is a single-page application within the document itself: the necessary JavaScript files are to be linked from the document itself or injected (in any web document) by a dedicated browser extension. Not tied to a specific kind of documents, dokieli suggests combining HTML and RDFa to fully capture the (more specific) document semantics. The dokieli UI allows users to dynamically switch between different stylesheets (e.g., to support different presentation options). Dokieli supports the whole life cycle of a document, as it also covers discussing an already published document. This aspect is of relevance to scientific publication, enabling a more transparent and continuous scrutiny of publications by the scientific community than it is possible with current reviewing approaches. Dokieli is built on a separation between the application logic and the underlying storage, embracing the SOLID [20] platform to "true data ownership" with *personal data storages*.

RASH [21] is a framework for scholarly publishing using a subset of HTML consisting only of 32 elements. RASH leverages semantic elements introduced in HTML5 such as *figure*, *caption*, *section* to represent the components of scholarly publications. A publication can also embed RDF annotations. The framework also includes a schema for the defined subset of HTML using the RELAX NG [22] language.

Originally thought for authoring W3 specifications, ReSpec [23] simplifies writing technical documentation in HTML. Similarly to what has been already discussed about dokieli, a ReSpec document must include the JavaScript file implementing the browser-based support logic and follow certain convention about its content. Unlike dokieli, ReSpec does not implement actual editing of documents, observing that any HTML editor is sufficient. Indeed, the support logic is mostly concerned with generating a live-preview of the document, identifying errors and – most importantly – implementing cross references within the document and across specifications. Actually, ReSpec is not meant to be included in the final published documents, as it offers a few options to export the specification being authored to (X)HTML, EPUB 3 or PDF. ReSpec borrows from software development the idea to store documents on source code hosting sites, such as GitHub [24], Bitbucket [25] and GitLab [26], which support version control, issue management and distributed authoring through pull requests. Indeed, a ReSpec document should include references to these three services, thus fostering feedback and contributions from the wider audience.

3 Motivation

Most attempts at achieving machine-readable and shareable legal texts have resulted in the definition of XML schemas. These attempts also took advantage of the possibility – offered by XML – to combine different schemas for different needs, such as reuse and extensibility. Reused XML applications included (X)HTML for tables and complex formatting, MathML for math, ChemML for chemistry and ATOM for metadata.

However, as a (meta)markup language, XML is in fact representing data for the purpose of information exchange rather than viewable documents that can be consumed by end users. Although XML documents could be associated with CSS stylesheets and XSL transformations for visualization, these technologies are not generally adopted, and initiatives related to legal XML schemas usually assume to generate parallel, distinct versions of the documents for visualization (resorting to common standards, such as HTML). Akoma Ntoso, just to name a notable example, differentiates on purpose semantics and structure from presentation, providing support for structural and semantic markup with the aim to "move digital documents from the presentation to the semantic era" [27]. Along this line of thought, a blog post [28] on the popular akomantoso.io site provides rationale for the choice of XML in Akoma Ntoso as the driving channel for representing legal documents. In particular, HTML was regarded as a simple format designed to primarily support presentation, lacking in support for print publication and semantic service access. In the perspective of the author, HTML has limited support for representing structure (which is also defective in not being imposed strictly), and total lack of support for semantics related to legislation.

We argue that these claims related to HTML are no longer valid because of a general shift of this standard to semantic markup, focusing on the purpose and role of the marked-up content rather than its appearance, this latter depending on a combination of the associated stylesheets and of the user agent visualization preferences [29]. Indeed, simply selecting a different stylesheet can completely alter the presentation (i.e., rendering) of the same document (as discussed about dokieli in Sect. 2). Concerning the lack of support for print publication, it is clear that HTML has all the necessary tools (i.e. CSS) for describing how a document should be represented when printed. About the lack of semantics, the current HTML Living standard features several mechanisms for representing semantics and for extending the language [30], so that a specific language for the legal domain could indeed be developed. HTML as-is is not thought for legal documents, as much as XML as-is is not, it is indeed not even a language, it's a syntax. Finally, concerning the few structural rules and the non-strict application of the existing ones, we observe similar phenomena in both Akoma Ntoso and, on a lesser extent, in FORMEX as well. Akoma Ntoso has different admitted structures because of the respect for "different traditions" which does not make its validation mechanism a strict one. Conversely, HTML has its own validation mechanism that is based on general requirements for documents, not for legal documents. For certain aspects, it is even too strict. The main complication in choosing HTML for the representation of a(ny) specific kind of document is indeed that it might be difficult to embed certain structures given the already existing restrictions of HTML. The opposite phenomenon (too loose validation) described in the blog is indeed not a problem: in the economy of defining a new standard by extending HTML, the creation of an ad-hoc validator is not an issue, it is indeed part of the outcome.

Furthermore, there are two serializations of HTML: traditional HTML and XHTML. The latter is based on XML, enabling the use of validating schemas and the combination of several schemas that address different concerns. While the web usually relies on transport-level security concerning the transmission of HTML documents, an XHTML document could be digitally signed using XAdES [31].

Given the above considerations, we argue that a document format like (X)HTML is not only adequate, but it should indeed be used – being it the standard for representation of documents on the web – unless strong incompatibility is found. Accordingly, we have extended HTML, producing an explicit domain language, featuring all structural aspects of an act, as in Akoma Ntoso or FORMEX, with a neat structure and rich semantics. Our goal is indeed to improve the quality of representation in terms of efficiency and standard compliancy, streamlining the publication workflow by unifying different aspects and phases, such as drafting, semantics and dissemination, into a single format binding them all. Finally, we provided API for the specification oriented at rendering the documents and offering point-in-time visualization of legal acts.

4 Approach

LegalHTML is an extension of HTML that enables a simple and semantically explicit representation of legal acts. Given the current stance of HTML towards the semantics of content rather than its presentation, our aim first required us to focus on the semantics of a legal act. As such, we defined three semantic layers that address different concerns:

- document semantics, which is all about the document (i.e., the legal act) itself, further divided into:
  - global information: document metadata
  - structure: document organization
- external domain knowledge (i.e., non-document classes in Akoma Ntoso).

Table 1 lists the extension mechanisms of HTML against the semantic layers just mentioned. All the Ys represent the mechanisms that could definitely fit the description of a given semantic layer, of which bold ones represent those that we concretely selected for our specification. “N”-valued cells represent solutions not matching the considered layer whereas “P”-valued ones represent mechanisms that may possibly be used but are not convenient – for a matter of clarity or conciseness – for the scope (e.g., refer to the later discussion on RDFa to represent the structure of a legal act).

LegalHTML primarily uses script-embedded RDF (specifically, in Turtle syntax) to represent stand-off metadata describing the document and its editorial/jurisdictional lifecycle. RDFa is used, instead, when metadata is naturally reflected in the act content, as in case of the signatures, which must appear at the end of the act. LegalHTML also uses RDFa to represent external knowledge, again exemplified by signatures, where there is a need to annotate the *people* that sign on behalf of some *organization* playing a given *role*. For these purposes, we introduced a supplementary LegalHTML ontology, which – in most cases – has to be used in conjunction with other ontologies and controlled vocabularies. These are needed to express semantic concepts that are specific to a particular legal tradition and therefore cannot be included in the general LegalHTML model. Describing the use of the model for the European legislation, the examples in this paper often make use of the ELI ontology and authority tables managed by the Publications Office of the European Union.

Table 1. Matching extension mechanisms found in HTML to our semantic layers. Bold faced Y indicates that the corresponding row has been adopted to support the corresponding column

	Structure	Metadata	External knowledge
custom elements	**Y**	Y	N
data- attribute	**Y**	Y	Y
class attribute	Y	Y	Y
reuse of semantic elements	**Y**	Y	Y
embedded web annotations	N	P	N
microformats	Y	Y	Y
HTML **rel** attribute	**Y**	Y	Y
RDFa	P	**Y**	**Y**
microdata	P	Y	Y
script-embedded RDF	N	**Y**	Y
<meta> element	N	**Y**	N

For representing the structure of a legal act, LegalHTML commits on mixing existing HTML elements, when applicable, to new custom elements. As there are two types of custom elements, we defined the following policy for their adoption in legal HTML:

- *customized built-in elements* (`<div is="lh-citation">…</div>`) are used to represent the structural elements of a legal act, by inheriting the semantics of an existing HTML element;
- *autonomous custom elements* (`<lh-version id="art_2">…</lh-version>`), not inheriting from existing HTML elements, are used for control code (e.g., for consolidation), which is in any case beyond the semantics defined by HTML.

The problem with customized built-in elements is that defining a new element, say *lh-citation*, we must choose the one and only one HTML element it inherits from. Compound with the strict constraints associated with some HTML elements, this led to the adoption of somewhat generic elements such as `div` and `span`, as more specific elements might not fit all usages of the customized elements. Customized built-in elements must be used with *data-* attributes, as they cannot introduce new attributes.

LegalHTML borrows from previous standards, including Akoma Ntoso, the idea to avoid generated text, i.e., any visible content should be traceable back to source text. LegalHTML cannot thus rely on automatic numbering of articles, items, and so on.

We discarded RDFa and Microdata to describe the structure of a legal document, as both are not really about the document but about mentions of external entities, and their adoption might require the introduction of additional `div` and `span` elements, which is not tolerable to describe the structure of a document. The *class* attribute should be used to hold *semantic classes*, which in turn are used as anchors by styling rules. However, in practice it is used for all sorts of *classes*, including ones that are solely geared towards presentation concerns. Given the pollution of the value space of this attribute, we discarded it in LegalHTML in favor of custom elements. In fact, CSS classes can be used within a LegalHTML, when it is necessary to fine tune the semantics and the presentation of a legal document. The semantics of LegalHTML includes the "behavior" of the representation elements. Benefiting from the ability of HTML user agents (e.g., the browsers) to execute JavaScript code, we complemented LegalHTML with a programming API, making compliant documents "active" (as detailed in Sect. 8).

Overall, the combination of HTML, its extensions, RDFa, and script+embedded-RDF offers separated "channels" for structure, semantics and presentation, and yet provides a single locus for them all, streamlining production and publication workflows.

5 Evolution and Maturity

LegalHTML evolved across two efforts. A first study proved feasibility of the realization of a language based on HTML for representing the structure and semantics of legal acts. The study also produced a first draft specification and a few sample documents transformed from other existing formats adopted within the Publications Office (OP). Besides the draft specifications, the study did not produce a formal public deliverable, rather a series of presentations reporting on challenges, issues, solutions, etc.. The study was followed by a second stage, aimed at finalizing the specifications and ensuring coverage and soundness of the solution. The latter objective was achieved through the development of a systematic mapping for CoV (Common Vocabulary) [32], an internal standard at the Publications Office reflecting "interinstitutional agreement on business level regarding the semantic concepts that represent the text of exchanged documents". As a further contribution, we validated LegalHTML by converting all the 47 documents that are used as examples in the CoV specifications.

Both study and finalization were supported by legal experts on our side, interviews with staff (several groups with different competencies and duties) from the OP concerning all aspects and steps of the production and publication workflow, and finally various steps of feedback provided by legal document experts from the OP. At the time of writing, these documents are on imminent publication. Additionally, these docs offered a plethora of different use cases and document types (https://w3id.org/legalhtml/specs/#eu-legal-document-types) considered to be sufficient by the OP. These efforts culminated in a final decision to adopt the model for the representation of legal acts on the EUR-Lex portal, equaling a TRL (Technology Readiness Level) of 8.

6 The LegalHTML Document Model

We report the main aspects of the LegalHTML document model, referring to the online specifications for a complete description, which would not fit into this article.

6.1 Overall Document Structure

The overall structure of a document is determined by customized `section` elements, such as `lh-preamble`, `lh-title`, `lh-enacting-terms`, etc. Conversely, we reused the `article` element for the basic unit of the normative content.

6.2 Paragraphs and Items

An *article* is articulated into one or more *paragraphs*, possibly associated with an explicit numeration. The obvious equation of such paragraphs to `p` elements in not possible, as HTML constraints these elements to only contain phrasing content (i.e. text and inline markup), whereas legal paragraphs can be in fact the root of a complex structure, including lists, nested paragraphs, etc.

Consequently, we had introduced a customized `div` element called `lh-paragraph` to surround an entire paragraph. Consecutive unnumbered paragraphs are represented as more elements of this kind in a row. Conversely, for numbered paragraphs we defined a customized `ol` element as in the following example.

```
<ol is="lh-paragraphs">
  <li>
    <span is="lh-number" data-value="1">1.</span>
    <div is="lh-paragraph">
      <p>The specific control and inspection [...]</p>
      <ol is="lh-points">
        <li>
          <span is="lh-number" data-value="a">(a)</span>
          <div is="lh-point">fishing [...]</div>
        </li>
        [...]
      </ol>
    </div>
  </li>
  [...]
  </ol>
```

Adhering to the approach (also found in Akoma Ntoso) to avoid generated content, the paragraph number is made explicit as an `lh-number` paired to an `lh-paragraph` by putting both inside a `li` element. The paragraph in the example contains in a turn a list of items indexed by lowercase Latin letters, which have been modeled in an analogous manner. The first `p` element inside the paragraph corresponds to the introductory phrase of the subsequent list.

6.3 Semantic Annotation

While custom elements are great for representing the structure of a legal act, domain references and, to a certain extent, metadata are best captured as semantic annotations using RDFa. The key insight is the introduction of an ontology providing a core vocabulary to represent such annotations (see (Fig. 2), the values of which are taken from external resources, decoupling LegalHTML from the terminology and procedure of any jurisdiction, including that of the European Union, which guided its development.

In the following example, we annotate the *acting entity* that adopted an act (in this cases, two entities: the European Parliament and the Council of the European Union), thanks to the *corporate body* [33] authority table from the Publications Office.

```
<span rel="lh:actingEntity" resource="corpbody:EP">THE EUROPEAN
PARLIAMENT</span> AND <span rel="lh:actingEntity" re-
source="corpbody:EURCOU">THE COUNCIL OF THE EUROPEAN
UNION</span>
</div>
```

In the following example, we annotate the direct applicability of an EU regulation to all member states, denoted by `corpbody:EUMS` by the *corporate body* authority.

```
<section is="lh-direct-applicability">
  <p rel="lh:applicability" resource="corpbody:EUMS">This Regu-
lation shall be binding in its entirety and directly applicable
in all Member States.</p>
</section>
```

A legal act contains an explicit section for the *concluding formulas*, which hosts the place and date it was approved and who signed it. However, details like the actual date and place, and the identity of who signed for a certain organization with a given role are instead annotated using RDFa accommodating different approaches to write this information. In the example, we used the place [34] authority table (e.g., for Brussels), the role [35] authority table (e.g., for president) and again the corporate body.

```
<section is="lh-concluding-formulas">
  <div is="lh-placedate">
    Done at <span rel="lh:signaturePlace" re-
source="place:BEL_BRU">Brussels</span>, <time property="lh:sig-
natureDate" datatype="xsd:date" content="2014-12-18"
datetime="2014-12-18">18 December 2014</time>.
  </div>
  <div is="lh-signature" rel="lh:signature" re-
source="#borgsign">
    <div>For the <span rel="lh:signatoryOrganization" re-
source="corpbody:EP">European Parliament</span></div>
    <div rel="lh:signatoryRole" resource="role:PRESID">The Pres-
ident</div>
    <div rel="lh:signatory" resource="dbr:Martin_Schulz">M.
SCHULZ</div>
  </div>
  [... ]
</section>
```

7 Consolidation

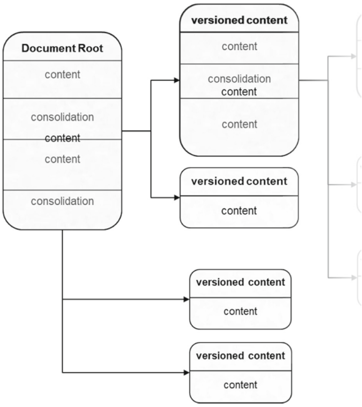
Legal acts evolve over time through subsequent amendments and corrigenda. There is thus a need to consolidate the base act and all documents making changes to it into a single *consolidated resource*, which in turn should allow to build a view of the act in any given point of time according to some criterion. The view of an act as applicable in each date, for example, contains the terms that are binding at that date. However, this view might include corrigenda that have been published on a later date, as they are usually retroactive. In this case, someone defending his good faith in a trial might argue the ignorance of errors corrected by retroactive corrigenda, by referring to the view only containing documents that have published at given and thus excluding corrigenda that were not published at that date.

LegalHTML adopts a tree-based multiversion model for consolidation (see Fig. 1 on the left) allowing to represent a *superposition* of the different versions of a legal act within a single document. Starting from the document root, when a portion of the act has been changed, LegalHTML introduces control `lh-cons` element containing a few `lh-version` elements associated with different "versions" of that part of the document each identified by a distinguished (technical) identifier (see Fig. 1 on the right).

Within an `lh-version` there can be another `lh-cons`, when portions thereof have been changed in turn (before the outer version has been deleted or replaced by a different version). The recursive application of consolidation-related elements produces the tree data structure depicted in Fig. 1 on the left. Indeed, the HTML elements described so far must be complemented by metadata (embedded inside a `script` element) that describes the different sets of changes in time.

We thus equipped the LegalHTML Ontology with a vocabulary related to consolidated resources (Fig. 2), allowing for a description roughly matching the analysis, lifecycle, temporal data and workflow sections of Akoma Ntoso (see Fig. 3).

- A new class: `lh:ConsolidatedResource`, extends `eli:LegalResource`.
- A property `lh:changeSet` links the consolidated resource to different sets of changes in time. A `lh:ChangeSet` describes a set of changes brought by a single amending doc.
- For each change set:



```
<lh-cons>
    <lh-version id="pfc_1.tit_1">
      <h1 is="lh-effective-title">
establishing a specific control and inspection
programme related to the cod stocks in the
Baltic Sea</h1>
    </lh-version>
    <lh-version
id="dec_impl/2012/262/pfc_1.tit_1">
      <h1 is="lh-effective-title">
establishing a specific control and inspection
programme related to the salmon and cod stocks
in the Baltic Sea
      </h1>
    </lh-version>
</lh-cons>
```

Fig. 1. Tree-based multiversion consolidation model (left) and control code to consolidate the effective title (right)

  - the properties `lh:changingAct`, `lh:published`, `lh:entryIntoForce` and `lh:startOfApplicability` link the act providing the amendments described by the change set, and determine when these changes have been published, enter into force and start to be applicable;
  - other properties, such as `lh:forceChange` and `lh:textualChange`, link to the different kinds of change that might occur in the change set described by specific claseses, such as: `lh:EntryIntoForce` or `lh:Substitution`;

- Each type of change is detailed by its dedicated properties, such as – in the case of `lh:Substitution` – `dct:type`, `lh:amendingText`, `lh:amendedText`, `lh:replaced`, `lh:replacement`. The latter two properties hold relative URLs with fragment identifiers referencing the `lh-version` elements by (technical) id.

  Given a date of interest d, it is possible to construct a view of the act v_d as it is applicable on d using the following algorithm.

  Let *act* be the base act of a consolidated resource and CS_{act} be the change sets affecting *act*. Let us write the members of CS_{act} as the sequence $cs_1, cs_2, ..cs_n$, such that $\forall i \in [1, n-1].startOfApplicability(cs_i) < startOfApplicability(cs_{i+1})$ where the function *startOfApplicability* returns when a change set starts to be applicable.

  For $i \in [1, n]$:

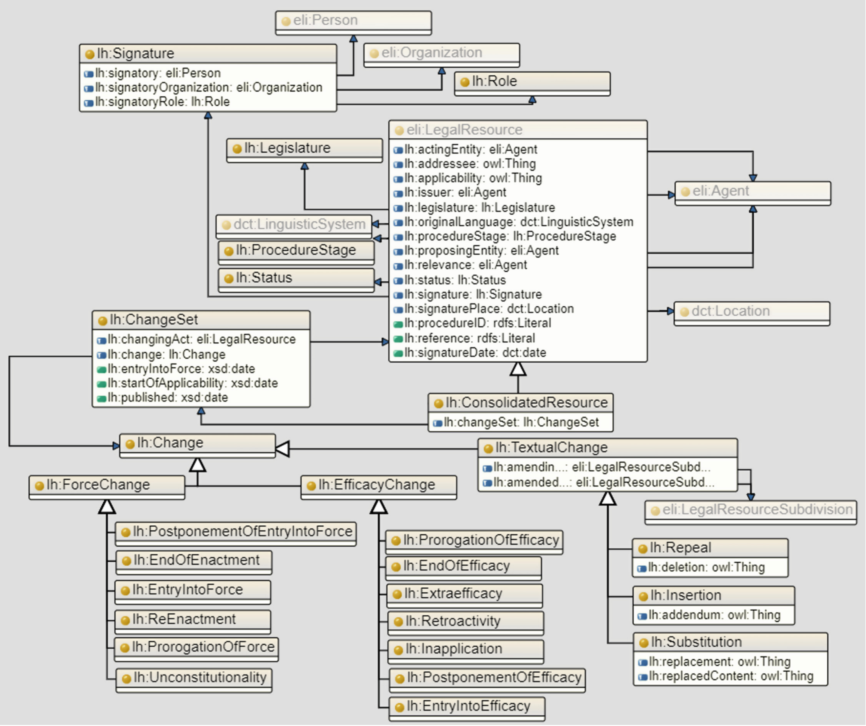

eli:Person
lh:Signature
lh:signatory: eli:Person
lh:signatoryOrganization: eli:Organization
lh:signatoryRole: lh:Role
eli:Organization
lh:Role
eli:LegalResource
lh:actingEntity: eli:Agent
lh:addressee: owl:Thing
lh:applicability: owl:Thing
lh:issuer: eli:Agent
lh:legislature: lh:Legislature
lh:originalLanguage: dct:LinguisticSystem
lh:procedureStage: lh:ProcedureStage
lh:proposingEntity: eli:Agent
lh:relevance: eli:Agent
lh:status: lh:Status
lh:signature: lh:Signature
lh:signaturePlace: dct:Location
lh:procedureID: rdfs:Literal
lh:reference: rdfs:Literal
lh:signatureDate: dct:date
lh:Legislature
dct:LinguisticSystem
lh:ProcedureStage
lh:Status
eli:Agent
dct:Location
lh:ChangeSet
lh:changingAct: eli:LegalResource
lh:change: lh:Change
lh:entryIntoForce: xsd:date
lh:startOfApplicability: xsd:date
lh:published: xsd:date
lh:ConsolidatedResource
lh:changeSet: lh:ChangeSet
lh:Change
lh:TextualChange
lh:amendin...: eli:LegalResourceSubd...
lh:amended...: eli:LegalResourceSubd...
eli:LegalResourceSubdivision
lh:ForceChange
lh:EfficacyChange
lh:PostponementOfEntryIntoForce
lh:EndOfEnactment
lh:EntryIntoForce
lh:ReEnactment
lh:ProrogationOfForce
lh:Unconstitutionality
lh:ProrogationOfEfficacy
lh:EndOfEfficacy
lh:Extraefficacy
lh:Retroactivity
lh:Inapplication
lh:PostponementOfEfficacy
lh:EntryIntoEfficacy
lh:Repeal
lh:deletion: owl:Thing
lh:Insertion
lh:addendum: owl:Thing
lh:Substitution
lh:replacement: owl:Thing
lh:replacedContent: owl:Thing

Fig. 2. Class diagram of the LegalHTML Ontology

- if *startOfApplicability*$(cs_i) < d$, then apply the changeset.
  For each change $c \in cs_i$, depending on the type of c:
  - `lh:Insertion`: ensure that the version pointed by `lh:addendum` is displayed;
  - `lh:Repeal`: ensure that the version pointed by `lh:deletion` is hidden;
  - `lh:Substitution`: ensure that the version pointed by `lh:replacedContent` is hidden while the version pointed by `lh:replacement` is displayed;
- otherwise, the changeset is not applicable and we must ensure that content introduced by any change in this set is hidden. For each change $c \in cs_i$, depending on the type of c:
  - `lh:Insertion`: ensure that the version pointed by `lh:addendum` is hidden;
  - `lh:Substitution`: ensure that the version pointed by `lh:replacement` is hidden.

```
<http://data.europa.eu/eli/dec/2008/589/2012-08-10>
    a           lh:ConsolidatedResource ;
    eli:type_document    <http://publications.europa.eu/resource/authority/resource-
type/CONS_TEXT> ;
    eli:consolidates
            <http://data.europa.eu/eli/dec/2008/589>,    # original doc
            <http://data.europa.eu/eli/dec/2011/114(1)>, # 1st amending doc
            <http://data.europa.eu/eli/dec_impl/2012/262>, # 2nd amending doc
            <http://data.europa.eu/eli/dec_impl/2012/468> ; #3rd amending doc, indirectly
amending this by amending
            <http://data.europa.eu/eli/dec_impl/2012/262>
    lh:changeSet
            <http://data.europa.eu/eli/dec/2008/589/2008-06-12/changeset_0>    ,
            <http://data.europa.eu/eli/dec/2008/589/2011-02-19/changeset_1>    ,
            <http://data.europa.eu/eli/dec/2008/589/2012-05-18/changeset_2>    ,
            <http://data.europa.eu/eli/dec/2008/589/2012-08-10/changeset_3>    .

[...]

<http://data.europa.eu/eli/dec/2008/589/2012-05-18/changeset_2>
    a                                  lh:ChangeSet ;
    lh:changingAct             <http://data.europa.eu/eli/dec_impl/2012/262> ;
    lh:published               "2012-05-16"^^xsd:date ;
    lh:entryIntoForce          "2012-05-18"^^xsd:date ;
    lh:startOfApplicability    "2012-05-18"^^xsd:date ;
    lh:textualChange
            <http://data.europa.eu/eli/dec/2008/589/2012-05-18/change_1>    ,
           <http://data.europa.eu/eli/dec/2008/589/2012-05-18/change_2>,
            <http://data.europa.eu/eli/dec/2008/589/2012-05-18/change_3>    .

<http://data.europa.eu/eli/dec/2008/589/2012-05-18/change_1>
    a                     lh:Substitution ;
    lh:amendingText   <http://data.europa.eu/eli/dec_impl/2012/262/art_1/unp_1/pnt_1/oj>;
    lh:amendedText        <http://data.europa.eu/eli/dec/2008/589/pfc_1/tit_1>  ;
    dct:type              <http://publications.europa.eu/resource/authority/modification-
type/REPLACEMENT> ;
    lh:replacedContent    <#pfc_1.tit_1> ;
    lh:replacement     <#dec_impl/2012/262/pfc_1.tit_1>    .
[…]
```

Fig. 3. Excerpt of consolidation-related metadata

8 Implementation

We developed an implementation of LegalHTML consisting of a JavaScript file and CSS stylesheet, which implement, respectively, the semantics (behavior) and default presentation of a LegalHTML document. Figure 4 shows how the LegalHTML encoding of the Commission Decision 2008/589/EC looks like in a web browser. This example is also available online [36]. Although we did not aim at a pixel-perfect match, the default presentation of LegalHTML is heavily inspired by EUR-Lex, to reinforce the point that a semantically explicit representation can be used to target any presentation template, as long as presentation rules are consistently applied to semantic elements.

The implementation also includes the behavior for LegalHTML documents and an API for them, which support:

- the automatic generation of an interactive table of contents (see the box on the top-left corner of Fig. 4);

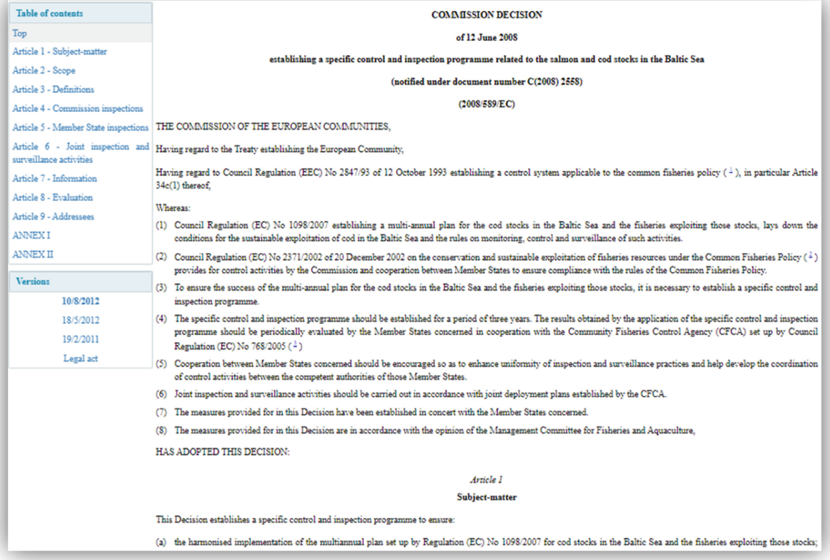


Table of contents

- Top
- Article 1 - Subject-matter
- Article 2 - Scope
- Article 3 - Definitions
- Article 4 - Commission inspections
- Article 5 - Member State inspections
- Article 6 - Joint inspection and surveillance activities
- Article 7 - Information
- Article 8 - Evaluation
- Article 9 - Addressees
- ANNEX I
- ANNEX II

Versions

- **10/8/2012**
- 18/5/2012
- 19/2/2011
- Legal act

COMMISSION DECISION

of 12 June 2008

establishing a specific control and inspection programme related to the salmon and cod stocks in the Baltic Sea

(notified under document number C(2008) 2558)

(2008/589/EC)

THE COMMISSION OF THE EUROPEAN COMMUNITIES,

Having regard to the Treaty establishing the European Community,

Having regard to Council Regulation (EEC) No 2847/93 of 12 October 1993 establishing a control system applicable to the common fisheries policy ([1]), in particular Article 34c(1) thereof,

Whereas:

(1) Council Regulation (EC) No 1098/2007 establishing a multi-annual plan for the cod stocks in the Baltic Sea and the fisheries exploiting those stocks, lays down the conditions for the sustainable exploitation of cod in the Baltic Sea and the rules on monitoring, control and surveillance of such activities.

(2) Council Regulation (EC) No 2371/2002 of 20 December 2002 on the conservation and sustainable exploitation of fisheries resources under the Common Fisheries Policy ([2]) provides for control activities by the Commission and cooperation between Member States to ensure compliance with the rules of the Common Fisheries Policy.

(3) To ensure the success of the multi-annual plan for the cod stocks in the Baltic Sea and the fisheries exploiting those stocks, it is necessary to establish a specific control and inspection programme.

(4) The specific control and inspection programme should be established for a period of three years. The results obtained by the application of the specific control and inspection programme should be periodically evaluated by the Member States concerned in cooperation with the Community Fisheries Control Agency (CFCA) set up by Council Regulation (EC) No 768/2005 ([3])

(5) Cooperation between Member States concerned should be encouraged so as to enhance uniformity of inspection and surveillance practices and help develop the coordination of control activities between the competent authorities of those Member States.

(6) Joint inspection and surveillance activities should be carried out in accordance with joint deployment plans established by the CFCA.

(7) The measures provided for in this Decision have been established in concert with the Member States concerned.

(8) The measures provided for in this Decision are in accordance with the opinion of the Management Committee for Fisheries and Aquaculture,

HAS ADOPTED THIS DECISION:

Article 1

Subject-matter

This Decision establishes a specific control and inspection programme to ensure:

(a) the harmonised implementation of the multiannual plan set up by Regulation (EC) No 1098/2007 for cod stocks in the Baltic Sea and the fisheries exploiting those stocks;

Fig. 4. Proof-of-concept of a Commission Decision in LegalHTML

- switching between different views of the document as it is applicable in different points-of-time (see the box on the bottom-left corner of Fig. 4), starting from the current date;
- management of backlinks from footnotes to in-text references.

The resolution includes different scenarios not limited to the time-variable. While the links provided by the standard view refer to points in time in which a certain modification of the act starts its applicability, the API support other more elaborated cases, e.g. restricting the view of the document to all changes that, at a certain data, have been not only entered into force and made applicable, but have been also published, so to analyze the case of the "good intentions" of a person who, in that date, was not aware of a retroactive modification of a law because it was not yet published.

9 Conclusion

The recent diffusion of linked open data standards and best practices for the publication of legal information produced by institutions, governments and, more in general, the public sector, require effective and efficient solutions for its representation, production, publication and fruition.

In the attempt to realize a single specification covering all the above aspects and thus replacing the different models being currently adopted by several institutions for their implementation, we have developed LegalHTML, an extension of the HTML language thought for representing legal acts. LegalHTML has proven to be a convenient solution for realizing electronic versions of legal acts, since their first drafting and through all other steps of the document preparation and publication, with a dedicated domain markup for structuring the act, a reference ontology for the legal domain, embedded document metadata for describing the editorial and jurisdictional history of the act and a dedicated model for representing the full history of modifications to the act brought by other acts into a single document. The LegalHTML API complete the picture by providing different ways to reconcile this change-tracked document according to different parameters such as a specific point in time and a combination of the jurisdictional characteristics of the modifications (publication, entry into force, efficacy). LegalHTML is scheduled for entering into application at the Publications Office by the end of 2023.

Acknowledgements. The authors want to thank by first Willem van Gemert and Maria Westermann for believing in the concept behind LegalHTML and for starting the discussion within the OP. A special acknowledgement to the work of Edyta Posel-Czescik, Camilo Soares and Dominika Uhrikova, for making it real through the end. Last, but surely not the least, we mention the devoted effort of Véronique Parisse who carefully reviewed our work and of all other staff who supported our preliminary investigation (Zilvinas Bubnys, Maria Kardami, Christian Marien and Tamas Schlemmer, among the many).

References

1. Directive (EU) 2019/1024 of the European Parliament and of the Council of 20 June 2019 on open data and the re-use of public sector information (recast). http://data.europa.eu/eli/dir/2019/1024/oj
2. Open Government Initiative. https://obamawhitehouse.archives.gov/open
3. Lupo, C., et al.: Deliverable 3.1: General XML format(s) for legal Sources. ESTRELLA Project (2007)
4. CEN MetaLex: Open XML Interchange Format for Legal and Legislative Resources. http://www.metalex.eu/
5. Boer, A., Hoekstra, R., Winkels, R.: METALex: legislation in XML. In: Bench-Capon, T.J.M., Daskalopulu, A., Winkels, R. (eds.) Legal Knowledge and Information Systems: Jurix 2002. IOS Press (2002)
6. Legal XML. http://www.legalxml.org/
7. LEXML. http://www.lexml.de/mission_english.htm
8. Muller, M.: Legal RDF dictionary. In: XML Europe 2002, Barcelona, May 2002 (2002)
9. McClure, J.: Legal-RDF vocabularies, requirements & design rationale. In: Proceedings of the V Legislative XML Workshop (2006)

10. Palmirani, M., Vitali, F.: Akoma-Ntoso for legal documents. In: Sartor, G., Palmirani, M., Francesconi, E., Biasiotti, M. (eds.) Legislative XML for the Semantic Web. LGTS, vol. 4, pp. 75–100. Springer, Dordrecht (2011)
11. Akoma Ntoso. http://www.akomantoso.org/
12. OASIS Open. http://www.oasis-open.org/
13. Mangiafico, J.: Legislative Data Challenges, One Year Later. https://blogs.loc.gov/law/2015/01/legislative-data-challenges-one-year-later/. Accessed 20 Jan 2015
14. Formex: Formalized Exchange of Electronic Publications. In: EU Vocabularies. https://op.europa.eu/en/web/eu-vocabularies/formex
15. About ELI. https://eur-lex.europa.eu/eli-register/about.html
16. Schneider, J.M., et al.: Lynx: a knowledge-based AI service platform for content processing, enrichment and analysis for the legal domain. Inf. Syst. **106**, 101966 (2022)
17. Rodriguez-Doncel, V., Montiel-Ponsoda, E.: Lynx: towards a legal knowledge graph for multilingual Europe. Law in Context **37**(1), 175–178 (2020)
18. Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: OASIS LegalRuleML. In: Fourteenth International Conference on Artificial Intelligence and Law, New York, pp.3–12 (2013)
19. Capadisli, S., Guy, A., Verborgh, R., Lange, C., Auer, S., Berners-Lee, T.: Decentralised authoring, annotations and notifications for a read-write web with dokieli. In: Cabot, J., De Virgilio, R., Torlone, R. (eds.) ICWE 2017. LNCS, vol. 10360, pp. 469–481. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60131-1_33
20. Solid. https://solid.mit.edu/
21. Peroni, S., et al.: Research articles in simplified HTML: a web-first format for HTML-based scholarly articles. PeerJ Comput. Sci. **3**, e132 (2017)
22. RELAX NG. https://relaxng.org/
23. ReSpec. https://respec.org/
24. GitHub: Where the world builds software. In: GitHub: Where the world builds software. https://github.com/
25. Bitbucket. https://bitbucket.org/
26. The One DevOps Platform | GitLab. https://gitlab.com/
27. Cervone, L., Palmirani, M., Vitali, F.: What it is | Akoma Ntoso. http://www.akomantoso.org/?page_id=25
28. Hariharan, A.: Why is Akoma Ntoso in XML and not HTML or JSON or PDF? In: AkomaNtoso.io - A Resource on Learning and using the Akoma Ntoso Schema. https://akomantoso.io/faq/why-is-akoma-ntoso-in-xml-and-not-html-or-json-or-pdf/. Accessed 25 Oct 2019
29. MDN contributors: Semantics - MDN Web Docs Glossary: Definitions of Web-related terms. In: MDN Web Docs. https://developer.mozilla.org/en-US/docs/Glossary/Semantics. Accessed 21 Sept 2022
30. WHATWG: Extensibility. In: HTML - Living Standard. https://html.spec.whatwg.org/multipage/introduction.html#extensibility. Accessed 8 Dec 2022
31. Cruellas, J.C., Karlinger, G., Pinkas, D., Ross, J.: XML advanced electronic signatures (XAdES). In: World Wide Web Consortium - Web Standards. http://www.w3.org/TR/XAdES/. Accessed 20 Feb 2003
32. Publications Office of the European Union: IMFC Common Vocabulary. In: EU Vocabularies. https://op.europa.eu/en/web/eu-vocabularies/cov. Accessed 2021
33. Publications Office of the European Union: Corporate body Named Authority List. http://data.europa.eu/88u/dataset/corporate-body. Accessed 2011
34. Publications Office of the European Union: Place Named Authority List. http://data.europa.eu/88u/dataset/place. Accessed 2009

35. Publications Office of the European Union: Role Named Authority List. http://data.europa.eu/88u/dataset/role. Accessed 2009
36. Stellato, A., Fiorelli, M.: Example representation in LegalHTML of the Commission Decision 2008/589/EC. In: LegalHTML. https://w3id.org/legalhtml/examples/OJ/L_2008190EN.01001101/L_2008190EN.01001101.xhtml. Accessed 18 Dec 2022