Computer Law & Security Review 51 (2023) 105888

Contents lists available at ScienceDirect

Computer Law & Security Review: The International Journal of Technology Law and Practice

journal homepage: www.elsevier.com/locate/clsr

LegalHTML: Semantic mark-up of legal acts using web technologies

Armando Stellato *, Manuel Fiorelli

University of Rome, Tor Vergata Department of Enterprise Engineering, Rome, Italy

ARTICLE INFO

Keywords:
Legal acts
Legal document
Consolidation
Metadata
HTML
Semantic web
Ontologies
Official journal
Law publishing

ABSTRACT

We introduce here LegalHTML, an extension of the HTML language thought for representing legal acts. LegalHTML has been conceived in the context of an exploratory study conducted for the Publications Office of the European Union, with the objective of overcoming the proliferation of formats for the electronic redaction of legal acts, dedicated to different steps of the editorial process (e.g. first draft, content editing, proof reading, introducing semantics, publishing) and of realizing a model and a language that could bind all processes and exigencies under a common umbrella. LegalHTML satisfies these requirements by providing an explicit domain language addressing all structural aspects of an act, such as articles, paragraphs, items, references and an associated ontology (foreseeing both inline annotations through RDFa and explicit RDF code within script elements) providing rich semantics to describe the editorial and jurisdictional history of the act and to insert references to entities of the domain. Being based on HTML, presentation is also offered by the same language, an aspect missing from all most notable standards for the legal domain. Furthermore, LegalHTML addresses consolidation of an act and its subsequent modifications into a single document using a tree-based representation of the original content and of its modified versions. Finally, alongside the language & ontology, we implemented a CSS stylesheet for the default rendering of LegalHTML documents and a JavaScript file imbuing documents with an API supporting TOC generation, footnote cross-references and the said point-in-time visualization of consolidated legal acts.

1. Introduction

In several countries, official journals have traditionally fulfilled the need for public notice of legal acts and other information concerning the public and private sectors. Originating in paper form, these journals and the dissemination of law more generally have undergone a digital transformation aimed at improving public access to law and offering value-added services to legal professionals. Repositories of legal content made available on the web in different countries have thus evolved over time, to support rich metadata, semantic annotations and cross-referencing, while the precise definition of representation models – first – and the standardization of shared models – then – have increased reusability and interoperability not only at the technical level (between applications), but also at the juridical level (e.g., combining norms from different jurisdictions in cross-border scenarios, search for precedents relevant to a given case, etc.).

In 2021, we conducted a study, funded by the Publications Office of the European Union (simply, Publications Office from now on), looking for an efficient solution to the complexity of its publication workflow made up of several stages, including drafting, proof-reading, finalization, and production of several manifestations scoped to different objectives, such as official journal publication, semantic indexing and search, dissemination, etc... We have addressed this complexity through the introduction of LegalHTML [1], an extension of HTML that unifies the formal, structural, and semantic representation of legal acts with the need for a viewable rendering suitable for publication, making it possible to represent these different aspects all within a same document. LegalHTML also supports the representation of a legal act together with changes made by amendments and corrigenda as a single consolidated document. The document model is supplemented by a dedicated LegalHTML Ontology to represent metadata, consolidation information and semantic annotations within the document. Finally, the document model is associated with an API supporting the rendering and visualization of (consolidated) documents as they apply at different points in time. Welcoming the model with interest, the Publications Office is going to adopt LegalHTML to represent its legal document base.

We deposited the specifications of LegalHTML and other support material to Zenodo, while the companion ontology has been published

* Corresponding author: University of Rome, Tor Vergata Department of Enterprise Engineering, via del Politecnico 1, 00133, Rome, Italy
E-mail address: stellato@uniroma2.it (A. Stellato).

https://doi.org/10.1016/j.clsr.2023.105888

Available online 24 October 2023

as Linked Open Data and archived to the Linked Open Vocabularies catalog[1].

2. Related work

Our own effort to represent legal texts using HTML and Semantic Web technologies in general can be related to two categories of work: earlier efforts to improve sharing and processing of legal texts, and applications of HTML to represent different types of documents.

Concerning the first category, we note that projects to publish legal content within or across jurisdictions have developed different representation models, usually, based on XML [2].

EnAct is a system originally developed to support the drafting, management, and delivery of legislation in Tasmania [3]. “Norme in Rete” [4] (NIR) was a portal providing citizens with free and unified access to Italian legislation. This portal has been superseded by "Normattiva" [5]. Similar initiatives in the Old Continent include Lex Dania [6] in Denmark, eLaw [7] in Austria, CHLexML [8] in Switzerland and the UK’s official legislation portal by The National Archives [9]. The Japan government created the Japanese Law Translation Database System [10,11] (JLTS), which provides unofficial English translations of Japanese laws, to facilitate international transactions and help non-natives comply with Japanese law. Subsequently, Japan adopted the e-LAWS [12] system that supports the drafting and publication of regulations and laws in electronic form.

The Publications Office uses FORMEX (Formalized Exchange of Electronic Publications) [13] for data interchange with external service providers that are involved in the production and publication process of the Official Journal of the EU. This format was created in 1985 as an SGML application defined by a DTD (Document Type Definition, a standard model for validation of XML documents), and in 2004 the fourth revision of FORMEX adopted XML and was defined by an XML Schema. Opened to the public in 2001, EUR-lex provides free access to the European Union Legislation [14].

In the United States, there are a few XML standards in use to support machine-readable legal texts. The Office of the Law Revision Counsel [15] (OLRC), which prepares and publishes the United States Code, introduced an XML distribution of the code in 2013 using the United States Legislative Model (USLM) [16]. Congressional bills, amendments, and resolutions are published (among other distributions) as XML documents conforming to dedicated DTDs [17] on the Congress [18] web site (operated by the Library of Congress [19]), GovInfo [20] (operated by U.S. Government Publishing Office [21]) and other places.

While the efforts and related models discussed above mainly originated from individual countries and thus focus on the specific needs of individual jurisdictions (including supranational entities like the European Union), there have also been attempts to establish standards for the representation of legal texts that accommodate different jurisdictions and legal traditions.

Legal XML [22] was founded in the USA in the 1998 to find agreement on schema per document type. Shortly thereafter, LEXML [23] was founded in Europe in 2000 with a different bottom-up strategy (motivated by the diversity of the legal landscape in Europe) which on the one hand allowed for different community-specific schemas, and on the other hand facilitates the convergence on a few of them, at which point mappings could be easily created.

Legal XML and LEXML collaborated on the development of the Legal RDF Dictionary [24], inspired by John McClure’s Legal-RDF [25], which would support the integration of different document schemas.

The *CEN Workshop on an Open XML Interchange Format for Legal and Legislative Resources* (*MetaLex*) hosted the collaborative development of CEN MetaLex [26] (not to be confused with an earlier model thought for Dutch legislation [27]). Among the various contributors, we mention LexDania, CHLexML, NormeInRete, and Formex. CEN MetaLex prescribes a least common denominator between different jurisdiction-specific standards and vendor-specific formats, primarily intended to support information interchange between national XML legislation formats. This can be important for settling legal debate in cross-border transactions (already mentioned for the Japanese JLTS), combining heterogeneous legal sources (e.g., case law and statutory law), and other unforeseen applications for legal information.

Akoma Ntoso (Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies) [28,29] supports the representation of various types of documents, including judicial, legislative, and parliamentary documents. Originally conceived in the context of the "Africa i-Parliament Action Plan", which is a program of the UNDESA (United Nations Department of Economic and Social Affairs), Akoma Ntoso has become widely adopted worldwide thanks to its flexibility to accommodate different legal traditions [30]. In fact, Akoma Ntoso defines many XML elements with a few constraints on their use, to accommodate the needs of different legal traditions. For example, the `section` element can be used both as a higher subdivision (e.g., in EU legislation) and a basic unit (e.g., in the United Kingdom). The disadvantage of this approach is the ambiguity of how a legal text should be annotated, which can be circumvented by developing a systematic annotation policy for a specific legal tradition, for example as a set of transformation rules for a national representation standard (as has been done for JLTS [31]). In August 2018, Akoma Ntoso became an OASIS standard called LegalDocumentML, thanks to the work of the aforementioned LegalXML organization aforementioned, which has since joined the OASIS consortium.

Legal document formats are complemented by metadata vocabularies for legal texts. The European Legislation Identifier (ELI) is a framework for harmonizing legislation publication that is based on three pillars [32]:

1. Every part of legislation is identified by an HTTP URI (based on URI templates at different jurisdictional levels, using URI components defined by ELI).
2. A common metadata model that follows the conceptual model FRBR [33], which has been encoded in an OWL ontology.
3. Machine-readable metadata are advertised on the legislative website using RDFa [34] or JSON-LD.

Individual jurisdictions adopting ELI build upon this common framework and define their own specializations of the ELI ontology.

Lynx [35,36] is a more recent effort that combines document formats and ontologies. On the one hand, Lynx has created the Legal Knowledge Graph (LKG) to support the semantic processing, analysis, and enrichment of legal documents. As Lynx did not need a sophisticated document model like Akoma Ntoso, they realized a simpler alternative based on annotations taken after a dedicated ontology, the Legal Knowledge Graph Ontology.

A work worth mentioning, even though not in the same category of the others and of LegalHTML itself, is represented by LegalRuleML. LegalRuleML [37] is an OASIS standard for formalizing the norms expressed in the textual provisions of a legal source as machine-readable rules. As an extension to the RuleML standard, LegalRuleML addresses concerns of particular interest to legal norm modeling, such as defeasibility, temporal reasoning, deontic operators, negation, jurisdiction, and isomorphism (i.e., associating rules and their components with relevant parts of the textual source). LegalRuleML was adopted by the same LegalXML organization that introduced LegalDocumentML, as a companion for semantic modeling of norms. Despite its natural connection to LegalDocumentML, LegalRuleML can also be used in conjunction with other legal document formats and thus can, in principle, be adopted within LegalHTML.

[1] specifications and ontology: https://w3id.org/legalhtml/; zenodo link: https://doi.org/10.5281/zenodo.7454918; Linked Open Vocabularies: https://lov.linkeddata.es/dataset/lov/vocabs/lh

Moving on to works related to ours by using HTML to represent different types of documents, we first mention the Legal-RDF approach to represent legal texts as XHTML documents embedding annotations that reflect both the structure and the semantics of the legal content. As extensively discussed in Section 3, the adoption of Akoma Ntoso is instead associated with the neglect of HTML as being more related to presentation rather than structure and semantics, which are of interest to Akoma Ntoso. Despite that, the United Kingdom National Archives, which adopted Akoma Ntoso, successfully created an HTML equivalent of this standard [38]. However, both the XML serialization and the new HTML serialization are really just generated from their own XML format, the Crown Legislation Markup Language (CLML), which is still the main one.

Recently, the idea of using (extensions of) HTML to represent specific types of documents flourished beyond the legal domain. Dokieli [39] and RASH [40] promoted the use of HTML for scientific publications, while ReSpec [41] and the W3C templates have influenced technical specs authoring in HTML.

Dokieli builds on the SOLID [42] platform and its concept of *personal data storages* that enable "true data ownership", to allow decentralized authoring, publishing, and discussion of any type of document directly in the browser using multiple web standards. The in-browser experience is realized by turning a document into a single page application (SPA) that supports its own editing. The inline editor also makes possible to alternate between different presentation options by switching between different stylesheets. The necessary JavaScript implementation file can be included in any HTML document, or it can be injected afterwards by a dedicated browser extension. Dokieli is certainly similar in spirit to WebDAV [43], the IETF protocol for collaborative authoring on the Web. While adopting HTML as the general document model, Dokieli relies on RDFa annotations to capture the semantics of specific document types. Even after a document is published, Dokieli continues to help by supporting conversations about it. The continuous, rigorous and transparent review process that Dokieli makes possible is certainly important to the Open Science [44] movement, which is challenging the traditional way in which scientific literature is produced, reviewed and disseminated.

RASH is a framework for scholarly publishing that combines a small subset of HTML and inline RDF annotations. The use of this document model is encoded as an RELAX NG [45] schema.

ReSpec was conceived for authoring W3 specifications, and although it somewhat bound to the requirements of its original scope, it can be used to write technical specifications writing in general using HTML. ReSpec adopts Specref [46] as a bibliographic format and "community-maintained database of Web standards & related references". ReSpec also has automatic external reference linking (xref), by means of which the mention of a term is linked to its definition in external specifications. ReSpec requires the inclusion of a JavaScript file in the document that provides a live preview of the document including automatically generated content (e.g., the table of contents), cross-references (possibly across external specifications) and conformance validation. A ReSpec document is not directly editable in the bowser (unlike Dokieli), but any HTML editor can be used for this purpose. While Dokieli takes a more web-centric approach to collaborative authoring, ReSpec assumes that documents are stored on source code hosting sites, such as GitHub [47], Bitbucket [48] and GitLab [49], which support a decentralized, collaborative authoring through mechanisms such as forks, issue trackers, and pull requests (PR). In fact, ReSpec encourages the inclusion of references to these services, in order to make them easily discoverable by anyone stumbling upon a published document. This lowers the barrier to providing feedback and contributing to the evolution of the document – the same goal as Dokieli. While Dokieli embraces the entire document lifecycle past its publication, ReSpec is only a development tool, not to be included in the published documents, which can be obtained through an export facility (supporting (X)HTML, EPUB 3 and PDF).

3. Motivation

The predominant approach to representing legal texts in a machine-readable and easily sharable form has been the definition of dedicated XML schemas, possibly mixing them with existing ones for common concerns, such as (X)HTML for tables and complex formatting, MathML for mathematics, ChemML for chemistry and ATOM for metadata.

However, XML schemas only support the representation of legal texts for the purpose of data interchange (between applications) rather than fruition of viewable documents by end users. Generic XML viewers and web browsers, for example, simply display documents that conform to these schemas as source code or, at best, as a hierarchical view defined by the nesting of XML elements. These limitations could be overcome by linking CSS stylesheets or XSL transformations, but these solutions are neither widespread nor supported by nowadays tooling. In fact, the current solution is to generate parallel, distinct manifestations of the legal texts for visualization, including rendering them as HTML pages for the publication on the Web. Akoma Ntoso, just to cite a notable example of an XML-based language for legal content, argues for the need to separate structural and semantic markup on the one hand and presentation concerns on the other, in order to “move digital documents from the presentation to the semantic era” [50]. In [51] the reasons for choosing XML over alternatives are discussed, dismissing HTML as a presentation format lacking support for print publication and semantic service access. In the view of the author, HTML does not suffice for structural markup, also being flawed in that it has too loose structural constraints, as well lacking any grounding in the domain of legislation.

We content that this position on HTML is today outdated, as we should take into account the current transition to semantic markup, which focuses on the meaning of the marked-up content rather than its appearance, delegated to a combination of the associated stylesheets and of the user agent visualization preferences [52]. HTML5 famously introduced some *semantic elements* in the area of sectioning (e.g., `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, and `<footer>`), as well as additional elements that address areas such as text-semantics (e.g., `<time>`). The semantic actually originated earlier, since even in previous revisions of the language we can see changes geared towards semantics that have developed progressively. For instance, the element `<strike>` was deprecated in HTML4 and then obsoleted in HTML5 in favor of the semantic element `<del>` to represent deleted content (while different rendering options existing, including strikethrough text and the use of a red background). Another notable example of this semantic restyling of the language is provided by the element `<i>`, which was originally conceived for the formatting to change the contained text to italics (hence, the name of the element). Concerning this and other "font style" elements, HTML3 retained the original definition, while noticing that "alternative means should be used to render the differences in emphasis" if fonts are constrained or for speech output. HTML4 goes further discouraging the use of these elements in favor of stylesheets. HTML5 shuffles the deck, as it deprecates some elements (e.g., `<tt>`) that are too bound to visual appearance, while others are ascribed to the broader family of *text-level semantics*, which includes the above mentioned `<time>` element, while rewriting their definition in more semantic terms. For instance, the semantic element <i> is now intended to convery *idiomatic text* that differs from the surrounding content in quality or modality, such as such alternative voice or mood, taxonomic designations, etc. In fact, the shift to a more semantic approach to content markup goes beyond the addition of new elements, the depreciation of some, and the redefinition of others, as it manifests itself pervasively in the language, best-practices, and application guidelines. Rather than mark-up content to achieve a desired appearance, the recommendation (see again, [52], and [53]) is today to select the elements that best describe the content and structure it according to its intended meaning. This is also very important to impaired users, as semantic markup enabled better assistive technologies. The current HTML living standard is well equipped with mechanisms for

expressing meaning and for extending the language to address the structure and semantics of domain-specific documents (discussed in Section 4).

While HTML is intended primarily for "continuous media", CSS clearly supports "paged media", and thus it supports (together with JavaScript) proper rendering of documents in print.

Regarding loose validation rules, we note that both Akoma Ntoso and, similarly, other models for legal content suffer from a similar problem. Akoma Ntoso, for example, admits different structures to accommodate "different traditions" (i.e., different ways to represent laws in different countries), obviously resulting in weak validation rules. HTML as it stands certainly cannot impose constraints on how different structures of a legal text should be composed, since it is based on a general document model that is independent of any specific domain. Nonetheless, HTML imposes a number of constraints related to its document model, being sometimes very strict about permissible structures. The strictness of HTML on certain structural requirements is the main obstacle to the development of HTML applications for a(ny) kind of document. Conversely, the lack of (a standard, domain-specific) validation is not a problem at all from our point of view since the development of an ad-hoc validator should be considered an integral part in the development of a new standard.

In fact, HTML should not even be really considered an alternative to XML: indeed, the traditional HTML serialization has long been complemented by one based on XML and called XHTML. Thanks to the latter, it is possible to use XML schemas for validation or to reuse existing schemas for different concerns. In this regard, it is worth to mention the use of XAdES [54] to digitally sign an XHTML document without relying on transport-level security, which is widespread on the Web.

Thus, the question is not about whether or not to use XML but rather about whether or not to adopt HTML as a general document model, which has become the standard for document representation on the web. In our opinion, the answer to this question is definitely affirmative. As such, we developed LegalHTML as an extension of HTML to represent the structure of a legal act, as in Akoma Ntoso or other similar models. Indeed, we do not aim to improve the coverage of the legal domain compared to existing standards, rather to increase interoperability and extensibility using (well-accepted) standards for content and data representation on the (semantic) web, streamlining the production workflow and improving content fruition through a rich representation that unifies semantic and presentation concerns.

A unified solution certainly simplifies the production workflow, avoiding the need for different formats and document instances at each stage of the workflow. In most cases, first drafts are edited using generic tools (e.g., Microsoft Word) possibly with dedicated extensions (e.g., the collection of Word templates used at the European Union called LegisWrite). This phase is then followed by a refinement phase, and then by the production of various formats, including semantic formats (e.g., Formex and CoV for the European Union and Akoma Ntoso in other jurisdictions), usually based on XML, and other formats for visualization (e.g., PDF and HTML). The advantage of LegalHTML is to combine formal, structural, and semantic representation with the possibility to support any typographical need for presentation. This has the potential to greatly simplify the production workflow, as the same document instance is first drafted and incrementally refined, taking advantage of its dual nature depending on the context.

Publication and fruition of legal content is benefited as well, as everything is incorporated into a single document, or more precisely, a single file, supporting not only "machine readability" but also "machine understandability" thanks to semantic annotations about the structure of the document itself and references to external entities (e.g., organizations, signatories, and people in general and the roles they play, the scope of the document, etc.).

Table 1
Matching extension mechanisms found in HTML to our semantic layers. Each row is associated with an extension mechanism (in the first column), while the values in the subsequent columns indicate whether that mechanism is not applicable (N), applicable (Y), or possibly applicable (P) but inconvenient for each of the 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

4. Approach

The first step toward semantic representation of legal acts in HTML is to clearly identify the different types of semantics to be addressed. As such, we introduced three semantic layers that address different needs:

- document semantics, further distinguished into
  - global information: document metadata
  - structure: organization of the document in different constituent parts
- external domain knowledge (i.e., non-document classes in Akoma Ntoso)

We then matched (see Table 1) the "extensibility mechanisms [provided by HTML] that can be used for adding semantics in a safe manner" [55] with these semantic layers, considering a match only if a mechanism not only theoretically supports the needs of a given layer, but also does so in a convenient and concise manner.

Concerning the structure of a legal act, LegalHTML combines applicable HTML elements to purposely defined *custom elements*. As there are two types of custom elements: *customized built-in elements*, inheriting and extending the semantics of existing HTML elements, and *autonomous custom elements*, minting completely new elements and their semantics, we defined the following policy for their adoption in LegalHTML:

- *Customized built-in elements* (`<div is="lh-recital">...</div>`) are used to represent the structural elements of a legal act.
- *Autonomous custom elements* (`<lh-version id="art_2">...</lh-version>`) are used for control code (e.g., related to consolidation), which is in any case beyond the document semantics defined by HTML.

LegalHTML uses different mechanisms to represent metadata within HTML documents, which allow metadata to be extracted into an RDF graph using off-the-shelf extractors. Indeed, LegalHTML is accompanied by an ontology vocabulary (see Fig. 1) for representing metadata and semantic annotations, by extending and combining existing vocabularies, such as the ELI (European Legislation Identifier) ontology [32].

Most metadata (e.g., that related to passive modifications) is stored within a *script* element in Turtle syntax:

```
<script type="text/turtle">
<![CDATA[
<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 ;
[…]
]]>
</script>
```

A detailed discussion of this consolidation-related metadata is postponed to Section 5.

This mechanism is far more concise than `meta` elements, but these are used in a few cases where metadata needs to be readily available to an HTML processor without the need to extract it into an RDF graph (which is still a supported option). Accordingly, the document type should be represented as in the following snippet:

Interestingly, LegalHTML does not prescribe a typology of legal acts, instead allowing the use of a controlled vocabulary chosen by the user, such as the Resource type Named Authority List [56] in the example. The advantage is to decouple LegalHTML from the peculiarities of a legal tradition, opening it up for reuse in different jurisdictions without compromising the integrity of the language.

Finally, RDFa can be used to annotate some metadata inline when it naturally occurs within the text of the legal content. In fact, this use case

```
<meta about="" property="eli:type_document"
resource="http://publications.europa.eu/resource/authority/resource-type/REG"/>
```

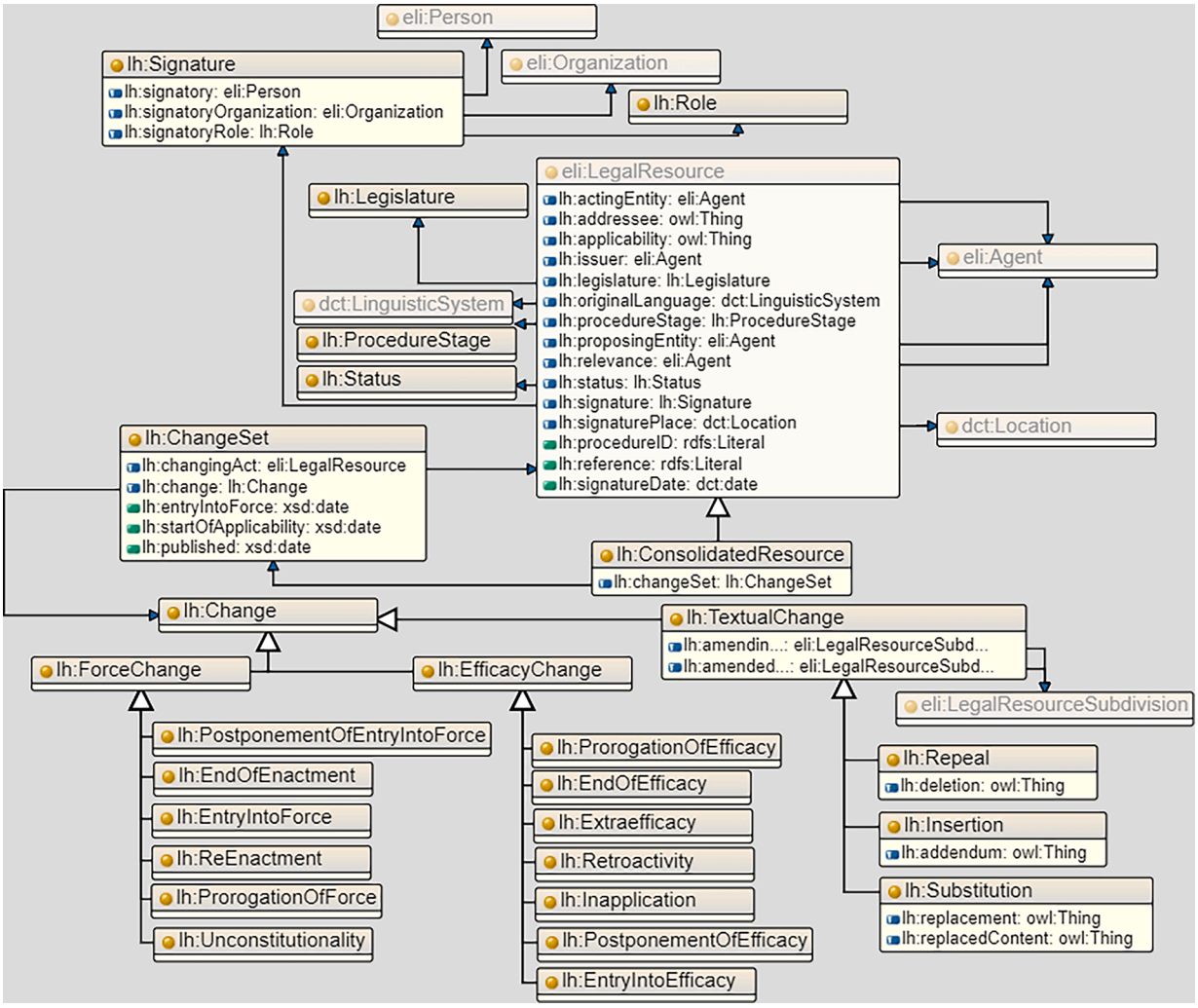

Fig. 1. The LegalHTML ontology.

overlaps with the annotation of domain knowledge. As an example, we mention the need to annotate the signatory of a legal act, including the *signature place*, the *signature date* as well as individual *signatures*, made by a *signatory* who plays a certain *role* for an *organization*.

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

Again, we used authority resources of the European Union for places [57], corporate bodies [58], and roles [59]. In the absence of such a resource for people, we used DBpedia [60] for them.

5. Consolidation

LegalHTML makes it possible to represent a legal act together with subsequent amendments in the same document as a consolidated resource. Portions of the act subject to change are marked with the element *lh-cons*, while individual versions are annotated with *lh-version*. The following HTML fragment encodes the different versions of the title of Commission Decision 2008/589/EC of 12 June 2008:

```
<lh-cons>
    <lh-version id="pfc_1.tit_1">
      <h1 is="lh-effective-title">[…] 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">[…] salmon and cod stocks in the Baltic Sea</h1>
    </lh-version>
</lh-cons>
```

The content of an element *lh-version* can be consolidated as well, according to a tree-based (recursive) multiversion consolidation model. This model follows the only natural constraint that a modification cannot change a text that has already been deleted by the modification itself or by any of its dependencies. This does not hinder the possibility of a subsequent modification entering into force earlier and, say, modifying the original fragment in a different way.

Fig. 2 reports an excerpt of the metadata describing the consolidation of the Commission Decision 2008/589/EC of 12 June 2008 considered in the previous example.

The LegalHTML ontology that has been already mentioned is then used to describe (inside a *script* element) the *passive modifications* of a legal document, describing each content fragment with multidimensional information such as entry-into-force, entry-into-efficacy, or postponement of entry-into-force.

First, the document is defined as an `lh:ConsolidatedResource`, which is a kind of `eli:LegalResouce` that consolidates (i.e., combines) a base act together with subsequent changes by amendments and corrigenda. The consolidated resource is linked to the modifying documents via the property `eli:consolidates`, while the property `lh:changeSet` connects it to the `lh:ChangeSet` resources.

An `lh:ChangeSet` describes changes (`lh:Change`) that are introduced by the same modifying act (`lh:changingAct`), and thus share the same date of publication (`lh:published`), entry into force (`lh:entryIntoForce`) and start of application (`lh:startOfApplicability`). The property `lh:change` relates the change set to individual changes.

These changes address the analysis of the lifecycle of the legal resource and can be further classified depending on whether they are related to the *force* (`lh:ForceChange`), *efficacy* (`lh:EfficacyChange`), or the textual content (`lh:TextualChange`) of the act. In fact, also the property `lh:change` is specialized depending on the type of the referenced change, as `lh:forceChange`, `lh:efficacyChange`, and `lh:textualChange`, respectively. Each category of changes is further specialized into different classes (with specific properties) to cover different cases within the category. Textual changes, for example, affect a subdivision of the act under analysis (`lh:amendedText`), which is identified using a URI (e.g., based on thee ELI naming convention), following the amendment contained in a subdivision (`lh:amendingText`) of the modifying act. This kind of change is further specialized into `lh:Repeal`, `lh:Insertion`, and `lh:Substitution`, to represent the deletion, insertion, and substitution of normative text, respectively. With each subclass having specific properties, let us discuss in more detail the class `lh:Substitution`. In this case, the property `lh:replacedContent` is the technical identifier of the HTML element within the HTML document replaced. The property `lh:replacement` holds, instead, the technical identifier of the HTML element that replaced the former. These technical identifiers are usually given as relative URLs with a fragment identifier equal to the HTML `id` of the affected elements. In the provided example, it can be seen that `change_1` has replaced the element `pfc_1.tit_1` with the element `dec_impl/2012/262/pfc_1.tit_1`. The containing change set, in turn, tells when this change was published, entered into force, and become applicable.

This information makes it possible to reconstruct a view of the act at any point in time, considering not only the applicable provisions but also other constraints, such as the publication status, as in the following use case concerning corrigenda.

```
<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. 2. Example of metadata about consolidation.

Unlike most types of legal acts, corrigenda are usually retroactive in that they apply from the date of publication of the corrected act. Thus, the text in effect on a given date may include content that has not yet been published.

Let us consider the following scenario, also depicted in Fig. 3:

- a Basic act (B) published on 01/01/2020 and entered into force on 20/01/2020,
- modified by modifier (M1) published on 01/01/2022 and applicable on 20/01/2022, and
- corrected by the corrigendum (C1) published on 01/04/2022 and applicable on 01/01/2020 (the day when the error occurred).

Let us say that someone is interested in the legal provisions on 01/03/2022. In principle, these should include B + M1 + C1.

But in a dispute in order to prove their good will, someone may be reasoning that they were not aware about mistake later corrected by a

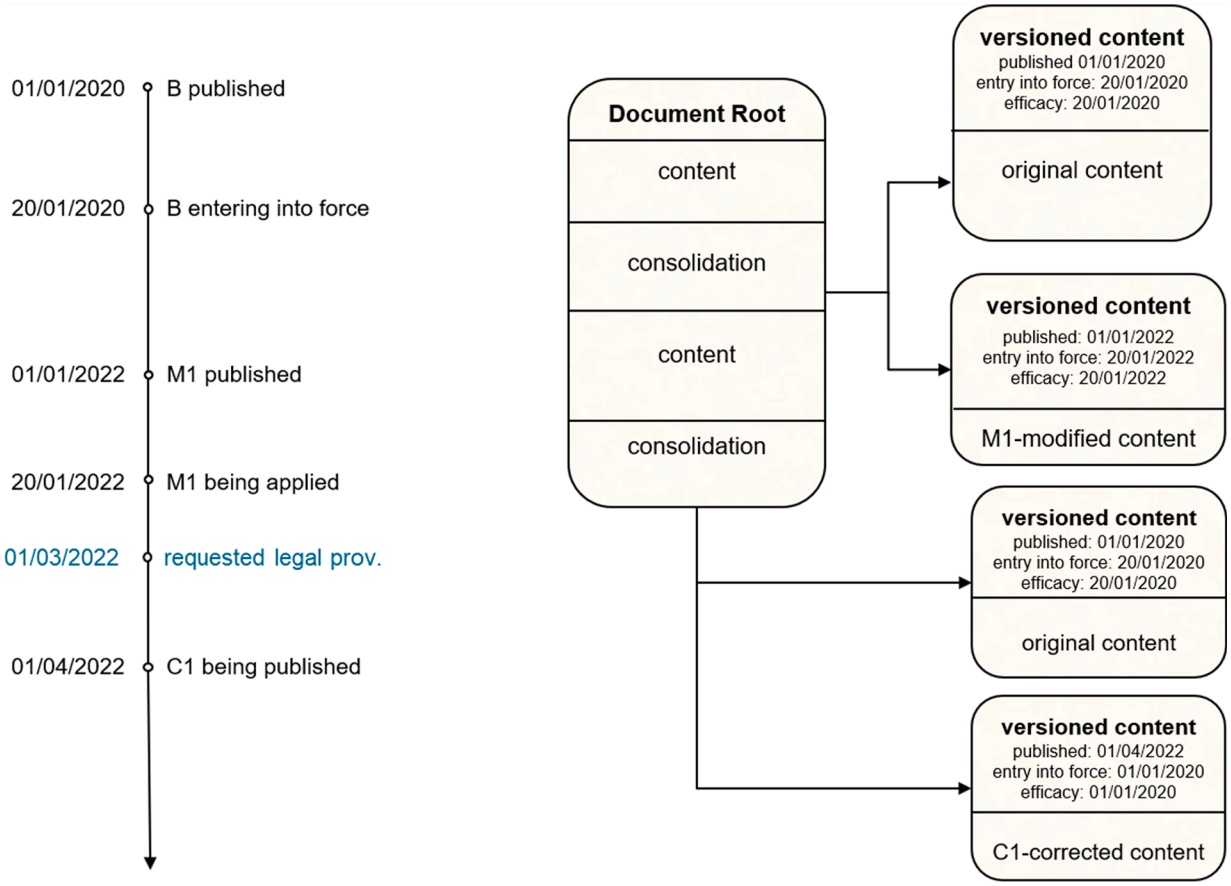

Fig. 3. Use case concerning a (retroactive) corrigendum. Dates are formatted as dd/mm/yyyy: for example, 01/03/2022 is March 1, 2022.

corrigendum, because it was not published on 01/03/2022. Thus, if one wants to generate a document for 01/03/2022 based on the date of publication, it should include B + M1 only.

However, if the date of interest is 15/01/2022 and the user is interested in applicable provisions, the output should be B + C1.

These different views can be computed using a recursive algorithm on the tree-based multiversion model just described. Let us consider a consolidated resource and let *ChangeSet* be the change sets affecting this resource. Let us define an inclusion criterion $Inc \subseteq ChangeSet \times Date$, e. g. in the scenario about good will, the change sets applying at any given time that are also published. Given a date $d \in Date$, it is then possible to construct a view of a document at time d that only reflects changesets cs such that $(cs, d) \in Inc$ (i.e. they satisfy the inclusion criterion). Starting from the document root, given a node n,

- If n is an 1h-cons , then the contained `1h-version` elements are considered. Let v be each of these elements, find the change sets $ChangeSet^v = \{cs^v \mid cs^v \ relates\_to \ v \ \wedge \ (cs^v, d) \in Inc\}$ that are connected to it and satisfy the inclusion criterion.
  - If $ChangeSet^v = \emptyset$ or if there exists a change set that deleted or replaces v, then v is discarded;
  - otherwise, the algorithm proceeds recursively on the content of v.
- Otherwise (n is not an 1h-cons), it is included in the generated view, and the algorithm must proceed recursively on each child of n

6. Evaluation

In this section we present and discuss the various evaluation and validation steps that we took towards the finalization of LegalHTML.

6.1. Discussion with stakeholders, law experts and legal document representation experts

Both the initial study proving the feasibility of LegalHTML and the subsequent finalization of its specifications were supported by legal experts on our side, interviews with staff (several groups with different competencies and duties) from the Publications Office concerning all aspects and steps of the production and publication workflow, and finally various steps of feedback provided by legal document experts from OP.

6.2. Simplification of the production workflow

We noted in the introduction that LegalHTML was proposed to address the complexity of the production workflow at the funding organization, and in particular to limit the proliferation of formats at the various stages, and then the consequent need for conversions between stages. Fig. 4 gives a possibly simplified but highly representative overview of the current workflow in the leftmost column (labeled "current") and shows two alternatives in the subsequent columns labeled as "short-term" and "long-term" adoption of LegalHTML, respectively.

In the current workflow, LegisWrite supports the first editing and proofreading of content. LegisWrite consists of a collection of styles and macros for use with Microsoft Word, which enable to annotate the structural parts of a legal document, but still perform editing in a visual, document-oriented environment. At this stage, a conversion from Word to Formex (or, in the future, Akoma Ntoso) is required, to produce an initial semantic representation of the legal texts. Editing of semantic content is then conducted using an XML editor or, in case of Akoma Ntoso, using EdiT. A futher conversion is then necessary to produce several dissemination files (e.g. PDF and HTML).

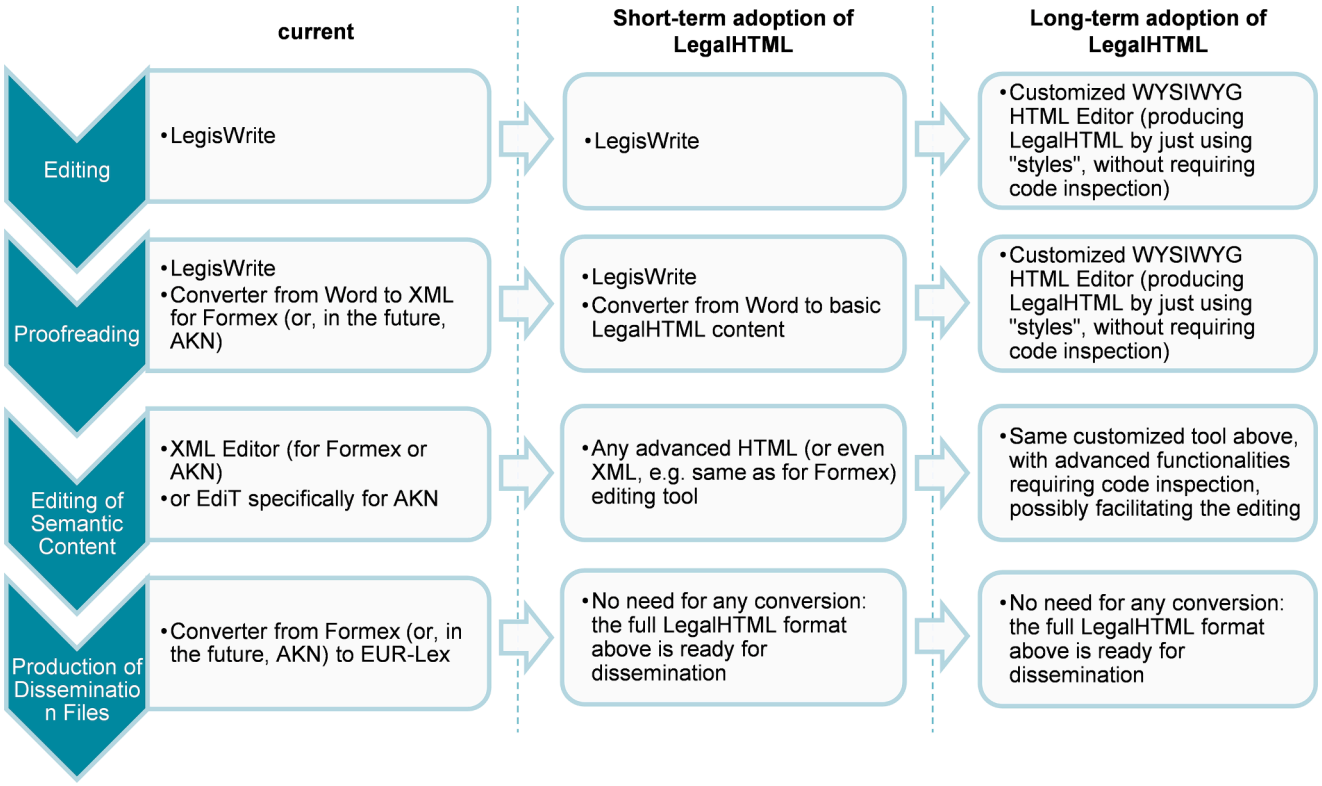

Fig. 4. Simplification of the production workflow determined by the adoption of LegalHTML.

The short-term plan for LegalHTML adoption maintains the use of LegisWrite in the early stages of the workflow, so that content editors, reviewers, and business users in general do not need to change the way they work. Still, this plan simplifies the workflow a bit, since there is no more need for conversions after the initial generation of LegalHTML from Word. Instead, the same document is incrementally enriched throughout the rest of the workflow, using any advanced HTML editor or, if XHTML is chosen, XML editors, even the same one used for Formex (allowing people continuing to work as they are accustomed to).

The long-term plan is to use LegalHTML from initial content editing through to dissemination with the support of customized HTML editors. Content editors and proofreaders would use a WYSIWYG editor, to do their work visually, using only "styles", without any need to look at code – just as they already do with LegisWrite. Subsequent semantic editing requires the customized HTML editor to support low-level code inspection and editing.

The long-term plan definitely simplifies the production workflow:

- The same document instance evolves throughout the various stages without the need for conversions between formats.
- People can use the same customized HTML editor at each stage using different capabilities depending on their role (e.g., WYSIWYG mode vs LegalHTML code editing)

The workflow simplification shown here is on the one hand specific to the case presented by the Publications Office of the EU; on the other hand, it is representative of the many similar processes being managed in other organizations and companies publishing legal acts, whether for public dissemination or for services dedicated to legal professionals. This invariance in format proliferation is only the tip of the iceberg, concealing a deeper problem: the "conversions" we mentioned in the earlier paragraphs are never really carrying all the information, and each new stage requires new information to be injected (and other to be lost). For instance, Microsoft Word is handy for drafting but, as such, can only be converted in an initial draft of the Formex document, which requires further information to be completed. The same holds for the conversion from Formex to HTML, which is partially automatically generated but requires further editing in order to replicate the various document setting that are to be expressed in the Official Journal, such as – at least – the positioning of tables, figures and other graphical assets. The asperity of maintaining (e.g. because of corrigenda, or for producing consolidated versions following amendments) the acts is then related also to the fragmentation of the act in different documents each containing original information, which makes it impossible to intervene on a single artifact, but rather to keep aligned the different existing manifestations.

6.3. *Coverage of the legal domain*

The main goal of LegalHTML is to achieve a more efficient and standard-compliant representation of legal texts, which implies a simplification at the workflow level. Although we consider this goal to be achieved (see Section 6.2), it is equally important that LegalHTML is on par with existing formats in terms of representation capabilities, at least to the extent required by the funding organization.

Once again, continuous discussion with stakeholders and experts (see Section 6.1) was valuable in this regard as well, allowing us to notice things that we had been overlooking, and to elicit important use cases, such as the multi-version consolidation model (see Section 5).

While the initial specification of LegalHTML was developed in an ad hoc basis from a few sample documents, the finalization effort was carried out in a systematic manner to ensure the coverage of the organization's needs related to the representation of legal texts. To this end, we have developed a mapping[2] between LegalHTML and the IMFC Common Vocabulary (CoV) [61]. This is an inter-institutional standard

[2] The LegalHTML mapping to COV is reachable, together with other documentation, from the following page: https://op.europa.eu/web/eu-vocabularies/legalhtml

at the European Union that reflects the agreement among business users, including lawyers, policy makers and proofreaders, on the structural parts of a legal text (each of which is properly defined) and how they can be used/nested (by specifying business rules). The mapping between LegalHTML and CoV ensures the conceptual validity of the former, both in terms of coverage and soundness. On the one hand, the ability to express CoV concepts in LegalHTML means that it does not miss structural elements of interest. On the other hand, the fact that LegalHTML can be interpreted back in terms of the CoV model guarantees that the representation is meaningful, conforming to the definitions and the rules that were specified by CoV.

Meant to guide the development of technical standards for the representation of legal texts, CoV has also informed the development of the Akoma Ntoso extension for the European Union (AKN4EU), for which a mapping similar to ours has been developed. As such, although no explicit mapping between LegalHTML and Akoma Ntoso exists, one is implicitly defined by the composition of these two mappings using CoV as pivot. Obviously, this would not be a one-to-one mapping that allows a straightforward conversion between LegalHTML and Akoma Ntoso, firstly because the mentioned mappings to CoV are not one-to-one either. In fact, both mappings share similar characteristics, determined by the gap between CoV as a conceptual model and either LegalHTML or Akoma Ntoso as technical standards. CoV focuses on concrete concepts that related to the surface organization of a legal text, with a particular emphasis on the various subdivisions of the legal text that can be the target of in-text references (e.g., recitals, citations, articles, paragraphs and various higher subdivisions). For example, the concepts of "part", "title" (not to be confused with the document title), "chapter", and "section" are individually defined as different levels of *higher subdivision* of an act or annex, while the latter concept is only mentioned in the definition of the former. Conversely, LegalHTML represents this concept per se as an *element*, while the specific instances are distinguished by attributes.

LegalHTML does not prescribe specific higher subdivisions, the values of various other attributes, nor does it define an ontology for the semantic annotation of legal texts. Accordingly, it leaves the choice of these aspects to the end users of the language, who should adopt third-party resources that reflect a specific legal tradition, without violating the integrity of the language and its neutrality with respect to different legal traditions. Conversely, enforcing a particular choice in LegalHTML would have imposed a particular view on the semantics of legal texts, possibly grounded in a specific legal tradition, forcing the establishment of a mapping when targeting a different legal tradition, which adopts a different conceptual model, let alone a different terminology.

It may also be the case that the same concept in CoV is mapped to different elements in technical standards depending on the context. For example, the notion of "point" in CoV expresses both an item in an enumeration and the basic unit of the provisions in an annex, while these usages are decoupled into separate modeling elements in both LegalHTML and Akoma Ntoso.

Another reference for the design of LegalHTML was Formex, a standard used by the Publications Office of the European Union to exchange information with contractors in its production and publication workflow. This is another technical standard that, like Akoma Ntoso, helped us evaluate our design, especially from a structural perspective. This is of particular interest because CoV, with its focus on the needs of business users, is rather shallow form a structural perspective.

For example, consider that the representation of recitals in CoV (using an XML rendering of it) is as follows:

```
<introduction_to_the_recitals>Whereas:</introduction_to_the_recitals>
<recital><numbering>(1)</numbering>In order to ensure legal certainty and harmonised […]
streamlined.</recital>
<recital><numbering>(2)</numbering>In order to facilitate implementation of […]
training.</recital>
```

The recitals are simply listed one after the other, while within each recital the number is annotated as such, but not the content or the recital. In addition, the list of recitals is introduced by a formula, without the formula and the recitals being enclosed in a single conceptual element.

The excerpt below clearly shows that Akoma Ntoso does add these missing pieces.

```
<recitals>
  <intro>
    <p>Whereas:</p>
  </intro>
  <recital>
    <num>(1)</num>
    <p>In order to ensure legal certainty and harmonised [...] streamlined.</p>
  </recital>
  <recital>
    <num>(2)</num>
    <p>In order to facilitate implementation of [...] training.</p>
  </recital>
  [...]
 <recitals>
```

Formex supports a congruent representation, shown below, with the addition of another level to explicitly say with the <NP> element that these recitals (called <CONSID>) are actually numbered points.

```
<GR.CONSID>
  <GR.CONSID.INIT>Whereas:</GR.CONSID.INIT>
  <CONSID>
    <NP>
      <NO.P>(1)</NO.P>
      <TXT>In order to ensure legal certainty and harmonised [...] streamlined.</TXT>
    </NP>
  </CONSID>
  <CONSID>
    <NP>
      <NO.P>(2)</NO.P>
      <TXT>In order to facilitate implementation of [...] training.</TXT>
    </NP>
  </CONSID>
  [...]
<GR.CONSID>
```

The analysis of Akoma Ntoso with respect to this and other cases validates our stance on introducing additional concepts beyond those defined by CoV to improve the clarity of the representation. For example, the markup around the content of a recital (on the right of the numbering) allows it to be easily referenced (e.g., through an XPath expression such as `//recitals/recital[1]/p` in the case of Akoma Ntoso).

There are also cases in which the three specs, CoV, Akoma Ntoso and Formex, disagree with each other from a terminological and structural perspective, motivating us to find a unifying solution. In the EU legal

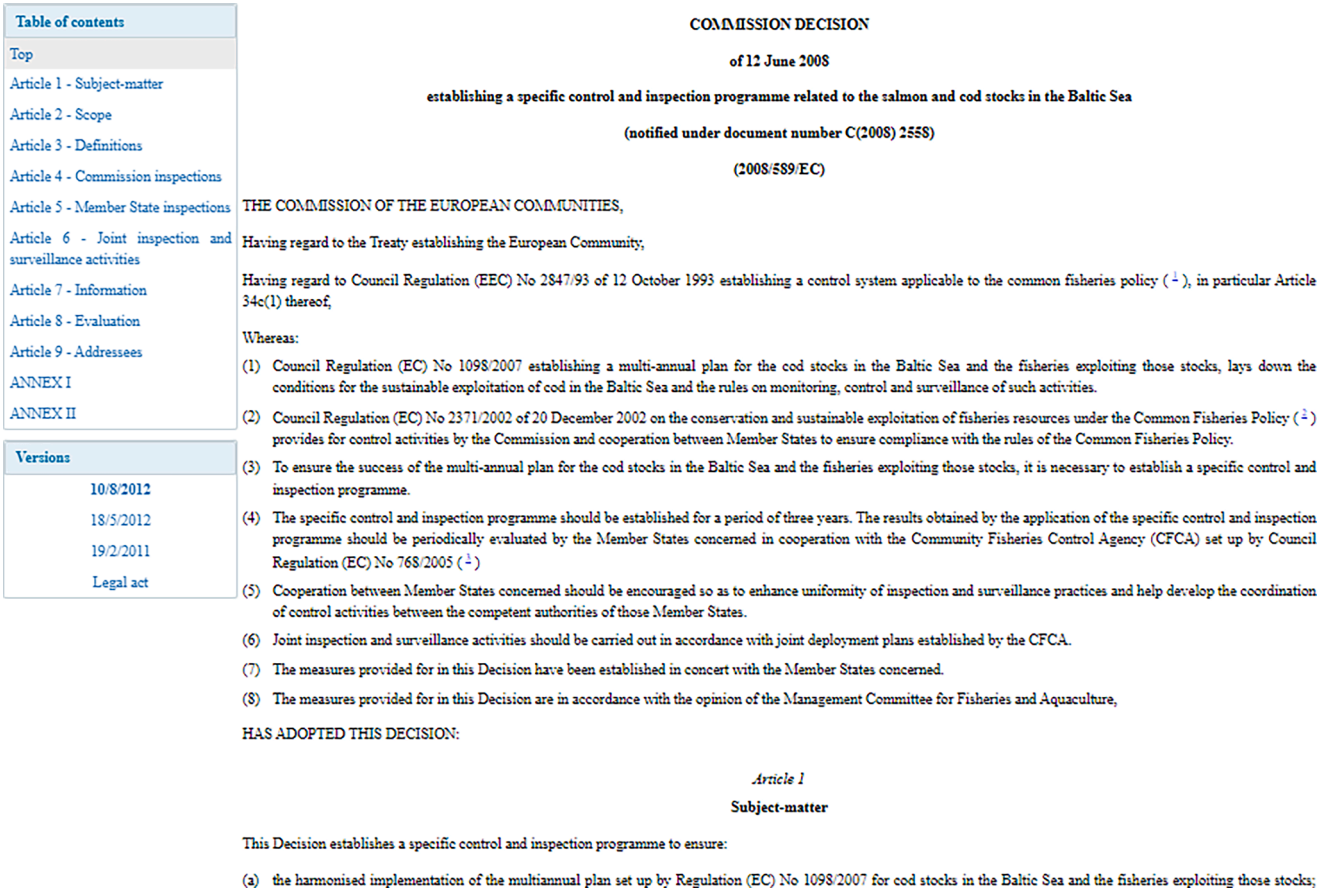

Table of contents

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. 5. Proof-of-concept of a Commission Decision in LegalHTML. Also available online at: https://art.uniroma2.it/legalhtml/examples/OJ/L_2008190EN.01001101/L_2008190EN.01001101.xhtml

tradition, the citations in the preamble of an act are introduced by a solemn formula such as "THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION", which states the acting entities of the document. In Formex, this is marked up as a `<PREAMBLE.INIT>`, while CoV and Akoma Ntoso (actually, AKN4EU) directly mark it as a formula denoting the acting entity. LegalHTML somehow mixes these two approaches with the introduction of a dedicated `lh-preamble-init` section, whose content can be annotated with an ontology to represent the acting entities. We believe that our solution has several advantages:

- It is more *general* in that it accounts for an introduction section that can provide for different content in different legal traditions.
- It is more precise as the acting entities are marked individually, unlike Akoma Ntoso and CoV, which treat them as a single solemn formula.
- It is more *semantically explicit* in that the entities are annotated with respect to a dataset that provides identifiers for the acting entities.

So far, we have presented differences that hint more at the different philosophies that drove the development of each of the presented standards. On purpose, we did not present a comparison table concerning the differences in coverage of the legal domain as, besides their implementation differences, all of the technical standards (thus excluding the conceptual model CoV) are roughly equivalent. Furthermore, it becomes also relative to talk about coverage when, for instance, Akoma Ntoso has been fragmented into a plethora of different dialects (e.g., the aforementioned AKN4EU, AKN4UN for the United Nations and so forth) to cover different legal traditions. The real contributions of LegalHTML, with respect to its predecessors, are more aimed at the possibility to unite under a single model the different aspects of structure, semantics and representation and having done that by finely reusing all modern (Semantic) Web standards. From a mere technical point of view, the approach promoted by LegalHTML has no drawback with respect to traditional solutions based on XML simply because, by relying on the XHTML serialization of HTML, LegalHTML is itself XML, and is thus – again, from the mere point of view of the conceptual and technological solution – subsuming the other solutions. On the other hand, on a more ground basis, being still in its infancy, LegalHTML might (potentially) lack some features that are particularly specific to some legal traditions and that have been not taken into account when developing it in the EU context; nonetheless, as already mentioned in the case of Akoma Ntoso, legal traditions each have their own demands that have been often met through strong customizations and dialects. In this sense, the strongly extensible approach promoted by LegalHTML can easily take into account for new structural elements (by declaring them as extensions of the language) while the intrinsically open nature of Web Ontologies allows for easily adopting new semantic models or extending existing ones according to the representation needs of each domain.

The discussion so far has skipped over LegalRuleML, which is a companion to Akoma Ntoso within the LegalXML working group. The reason is that LegalHTML must be compared to Akoma Ntoso, while LegalRuleML deals with an orthogonal and thus complementary aspect, namely the interpretation of the norms within a legal text and their representation as rules using a defeasible logic. In fact, LegalRuleML can also be used in conjunction with LegalHTML as well, since the rules are represented separately from the source legal document. The only connection between the two is in the forms of references (e.g., using fragment identifiers also supported by HTML) to the position in the source document where a rule (or parts of it) is derived.

6.4. Coverage of the legal document base

The development of CoV was the result of the "interinstitutional collaborative analysis of commonly agreed example documents" [61]. These are 47 documents that are considered to be representative of the 22 different "document types" used by the institutions of the European Union. In fact, every definition and business rule specified by CoV references (parts of) one of these documents, providing an example for the definition/rule, but also motivating it in the first place.

We complemented the intentional analysis done with the mapping between LegalHTML and CoV with a more extensional analysis consisting of mapping each of the 47 documents mentioned above. Since these are considered by an inter-institutional committee of the European Union to be representative of the entire document base, we argue that this small-scale experiment is strong evidence that indeed all documents can be represented in LegalHTML.

7. Implementation

The specifications of LegalHTML are complete and made available using a variety of channels, including a persistent URL and deposit on Zenodo. They were complemented by a proof-of-concept implementation of the necessary support files (i.e., CSS stylesheet and JavaScript file) to support the proper visualization (and interactions) in the browser (see Fig. 5).

The proof-of-concept implementation mimics the look-and-feel of EUR-Lex without pretending a pixel-perfect match. Additionally, it supports "active documents", which support:

- the production of a table of content with clickable links to the corresponding subdivisions of the document
- the generation of back-links from footnotes to in-text references
- point-in-time visualization (see Section 5)

8. Conclusion

We proposed LegalHTML to unify the formal, structural, and semantic representation of legal acts with the need for human friendly rendering, all within a single document. This unified model simplifies the production workflow as a single document instance is first drafted and then enriched at various stages of the process. Publication and fruition benefit as well, since the same document can be used for human viewing, while also supporting other use cases that require machine readability and understandability.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

download links have been provided in the article.

Acknowledgement

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 Publications Office of the European Union. A special acknowledgement goes 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 (Ashok Hariharan, Zilvinas Bubnys, Maria Kardami, Christian Marien and Tamas Schlemmer, among the many)

References

[1] Stellato A, Fiorelli M, et al. LegalHTML: a representation language for legal acts. In: Pesquita C, et al., editors. The semantic web (lecture notes in computer science, LNCS). Springer, Cham; 2023. p. 520–37. https://doi.org/10.1007/978-3-031-33455-9_31. vol. 13870.

[2] Lupo C, Vitali F, Francesconi E, Palmirani M, Winkels R, de Maat E, Boer A, Mascellani P. Deliverable 3.1: general XML format(s) for legal sources. Estrella Project 2007 [Online]. Available, http://www.estrellaproject.org/doc/D3.1-General-XML-formats-For-Legal-Sources.pdf.

[3] Arnold-Moore T, Clemes J. Connected to the law: tasmanian legislation using EnAct. J Inf Law Technol 2000;2000(1) [Online]. Available, http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2000_1/arnold/.

[4] Francesconi E. The "Norme in Rete" project: standards and tools for italian legislation. Int J Legal Inf 2006;34(2):358–76. https://doi.org/10.1017/S0731126500001517.

[5] Istituto Poligrafico e Zecca dello Stato S.p.A. Normattiva. [Online]. Available: https://www.normattiva.it/.

[6] Petersen KE. Experiences with "Lex Dania Live. Front Artif Intell Applic 2011;236:69–76. https://doi.org/10.3233/978-1-60750-988-2-69.

[7] W. Engeljehringer. (2006) The Austrian E-LAW project (E-LAW). [Online]. Available: https://joinup.ec.europa.eu/collection/justice-law-and-security/document/austrian-e-law-project-e-law.

[8] eJustice.CH. (2018) CHLexML. [Online]. Available: https://www.ejustice.ch/it/CHLexML.html.

[9] Tullo C. Online access to UK legislation: strategy and structure. Front Artif Intell Applic 2011;236:21–32. https://doi.org/10.3233/978-1-60750-988-2-21.

[10] K. Toyama, D. Saito, Y. Sekine, Y. Ogawa, T. Kakuta, T. Kimura, and Y. Matsuura, "Design and development of Japanese law translation database system," in *Proceedings of Law via the Internet 2011*, Hong Kong, 8-10 June 2011, 2011, p. 12. [Online]. Available: https://www.hklii.hk/conference/paper/1C2.pdf.

[11] Japanese Law. [Online]. Available: https://www.japaneselawtranslation.go.jp/.

[12] e-LAWS. [Online]. Available: https://elaws.e-gov.go.jp/.

[13] EU Vocabularies. [Online]. Available: https://op.europa.eu/en/web/eu-vocabularies/formex.

[14] Breedstraet E. EUR-Lex: towards a common legal portal. Front Artif Intell Applic 2011;236:15–20. https://doi.org/10.3233/978-1-60750-988-2-15.

[15] Office of the Law Revision Counsel. [Online]. Available: https://uscode.house.gov/.

[16] U.S. Government Publishing Office. United States Legislative Markup (USLM) XML schema. [Online]. Available: https://github.com/usgpo/uslm.

[17] U.S. Government Publishing Office. Bill DTD. [Online]. Available: https://github.com/usgpo/bill-dtd.

[18] Library of Congress. Congress.gov. [Online]. Available: https://www.congress.gov/.

[19] Library of Congress. [Online]. Available: https://www.loc.gov/.

[20] U.S. Government Publishing Office. GovInfo. [Online]. Available: https://www.govinfo.gov/.

[21] U.S. Government Publishing Office | America Informed. [Online]. Available: https://www.gpo.gov/.

[22] Legal XML. [Online]. Available: http://www.legalxml.org/.

[23] LEXML. [Online]. Available: http://www.lexml.de/mission_english.htm.

[24] M. Muller, "Legal RDF dictionary," in *XML Europe* 2002, *Barcelona, May, 2002*, 2002. [Online]. Available: http://www.lexml.de/legal_rdf_dictionary_barcelona.htm.

[25] McClure J. Legal-rdf vocabularies, requirements & design rationale. In: Proceedings of the V Legislative XML Workshop; 2006.

[26] CEN MetaLex: Open XML interchange format for legal and legislative resources. [Online]. Available: http://www.metalex.eu/.

[27] Boer A, Hoekstra R, Winkels R. METALex: legislation in XML. In: Bench-Capon TJM, Daskalopulu A, Winkels R, editors. Legal knowledge and information systems: Jurix 2002. IOS Press; 2002.

[28] Palmirani M, Vitali F. Akoma-ntoso for legal documents. Legislative XML for the semantic web. Law, governance and technology series. Dordrecht; 2011. p. 75–100. vol. 4.

[29] Akoma Ntoso. [Online]. Available: http://www.akomantoso.org/.

[30] Vitali F, Palmirani M. Akoma Ntoso: flexibility and customization to meet different legal traditions. In: Proceedings of the Symposium on Markup; 2019. https://doi.org/10.4242/BalisageVol24.Palmirani01. July 29, 2019.

[31] Gen K, Akira N, Makoto M, Yasuhiro O, Tomohiro O, Katsuhiko T. Applying the Akoma Ntoso XML schema to Japanese legislation. J Law Inf Sci 2016;24(2):49–70.

[32] About ELI. [Online]. Available: https://eur-lex.europa.eu/eli-register/about.html.

[33] IFLA, *Functional requirements for bibliographic records.*, 1998, vol. 19. [Online]. Available: https://www.ifla.org/publications/ifla-series-on-bibliographic-control-19.

[34] B. Adida and M. Birbeck. (2007, October) World wide web consortium - web standards. [Online]. Available: http://www.w3.org/TR/xhtml-rdfa-primer/.

[35] Schneider JM, Rehm G, Montiel-Ponsoda E, Rodríguez-Doncel V, Martín-Chozas P, Navas-Loro M, Kaltenböck M, Revenko A, Karampatakis S, Sageder C, Gracia J, Maganza F, Kernerman I, Lonke D, Lagzdins A, Gil JB, Verhoeven P, Diaz EG, Boil Ballesteros D. Lynx: a knowledge-based AI service platform for content processing, enrichment and analysis for the legal domain. Inf Syst 2022;106:101966. https://doi.org/10.1016/j.is.2021.101966.

[36] Rodriguez-Doncel V, Montiel-Ponsoda E. Lynx: towards a legal knowledge graph for multilingual Europe. Law Context 2020;37(1):175–8. https://doi.org/10.26826/law-in-context.v37i1.129.

[37] Athan T, Boley H, G G, P M, Paschke A, W A. OASIS LegalRuleML. In: Fourteenth International Conference on Artificial Intelligence and Law; 2013. p. 3–12.

[38] J. Mangiafico. (2015, January) Legislative data challenges, one year later. [Online]. Available: https://blogs.loc.gov/law/2015/01/legislative-data-challenges-one-year-later/.

[39] 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, editors. Web engineering. ICWE 2017. (Lecture notes in computer science). Springer, Cham; 2017. p. 469–81. https://doi.org/10.1007/978-3-319-60131-1_33. vol. 10360.

[40] Peroni S, Osborne F, Di Iorio A, Nuzzolese AG, Poggi F, Vitali F, Motta E. Research articles in simplified HTML: a web-first format for HTML-based scholarly articles. PeerJ Comput Sci 2017;3:e132. https://doi.org/10.7717/peerj-cs.132.

[41] ReSpec. [Online]. Available: https://respec.org/.

[42] Solid. [Online]. Available: https://solid.mit.edu/.

[43] Whitehead EJ, Wiggins M. WebDAV: IETF standard for collaborative authoring on the Web. IEEE Internet Comput 1998;2(5):34–40. https://doi.org/10.1109/4236.722228.

[44] McKiernan EC, Bourne PE, Brown CT, Buck S, Kenall A, Lin J, McDougall D, Nosek BA, Ram K, Soderberg CK, Spies JR, Thaney K, Updegrove A, Woo KH, Ya T. Point of view: how open science helps researchers succeed. eLife July 2016;5. https://doi.org/10.7554/eLife.16800.

[45] RELAX NG. [Online]. Available: https://relaxng.org/.

[46] Specref. [Online]. Available: https://www.specref.org/.

[47] GitHub: Where the world builds software. [Online]. Available: https://github.com/.

[48] Bitbucket. [Online]. Available: https://bitbucket.org/.

[49] The One DevOps Platform | GitLab. [Online]. Available: https://gitlab.com/.

[50] L. Cervone, M. Palmirani, and F. Vitali. What it is | Akoma Ntoso. [Online]. Available: http://www.akomantoso.org/?page_id=25.

[51] A. Hariharan. (2019, October) AkomaNtoso.io - A resource on learning and using the Akoma Ntoso schema. [Online]. Available: https://akomantoso.io/faq/why-is-akoma-ntoso-in-xml-and-not-html-or-json-or-pdf/.

[52] MDN contributors. (2022, Sep.) MDN web docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/Semantics.

[53] WHATWG. HTML living standard. [Online]. Available: https://html.spec.whatwg.org/multipage/dom.html#semantics-2.

[54] J. C. Cruellas, G. Karlinger, D. Pinkas, and J. Ross. (2003, February) world wide web consortium - web standards. [Online]. Available: http://www.w3.org/TR/XAdES/.

[55] WHATWG. (2022, December) HTML - living standard. [Online]. Available: https://html.spec.whatwg.org/multipage/introduction.html#extensibility.

[56] Publications Office of the European Union. (2011) Resource type named authority list. [Online]. Available: http://data.europa.eu/88u/dataset/resource-type.

[57] Publications Office of the European Union. (2009) Place named authority list. [Online]. Available: http://data.europa.eu/88u/dataset/place.

[58] Publications Office of the European Union. (2011) Corporate body named authority list. [Online]. Available: http://data.europa.eu/88u/dataset/corporate-body.

[59] Publications Office of the European Union. (2009) Role named authority list. [Online]. Available: http://data.europa.eu/88u/dataset/role.

[60] Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, Hellmann S, Morsey M, van Kleef P, Auer S, Bizer C. DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web 2015;6(2):167–95. https://doi.org/10.3233/SW-140134.

[61] Publications Office of the European Union. (2021) EU vocabularies. [Online]. Available: https://op.europa.eu/en/web/eu-vocabularies/cov.