IST-2004-027655

ESTRELLA

European project for Standardised Transparent Representations in order to Extend Legal Accessibility

Specific Targeted Research Project
Information Society Technologies

Deliverable 3.1

General XML format(s) for legal Sources

Due date of deliverable: 31 December 2006
Actual submission date: 22 January 2007

Start date of project: 1 January 2006 Duration: 30 months

CNIPA – MRIPA

FINAL

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Executive Summary

The ESTRELLA Project

ESTRELLA aims at developing and validating an open, standards-based platform allowing public administrations to develop and deploy comprehensive legal knowledge management solutions. The use of open-standard platforms allows users to be independent from proprietary products of particular vendors; similarly they facilitate the development of different software solutions implementing such standards as well as guaranteeing scalability and interoperability among the applications, allowing public stakeholders to freely choose among competing development environments, inference engines, and other tools. ESTRELLA will support, in an integrated way, both legal document management and legal knowledge systems, to provide a complete solution for improving the quality and efficiency of the determinative processes of public administration requiring the application of complex legislation and other legal sources.

To obtain these objectives different aspects will be taken into account: from the definition of a Legal Knowledge Interchange Format (LKIF) to represent legal knowledge, to the implementation of open-standards for representing legal sources, as well as the development of a prototype of an integrated Legal Sources Management System.

Scope and Objective of the Deliverable

This deliverable is included within the activities of Workpackage 3 (WP3) which deals with the definition of effective and explicit links between legal knowledge and original legal sources the knowledge is based on.

This is one of the main pre-conditions and central requirement of automated systems based on representations of legal knowledge. Each argument in fact should be backed with plausible references to legal sources proving the validity of the argument itself.

Legal knowledge in ESTRELLA is represented by using a Legal Knowledge Interchange Format (LKIF), defined within the project as the main objective of WP1. It is built upon emerging XML-based standards of the Semantic Web, including RDF and OWL, and Application Programmer Interfaces (APIs) for interacting with LKIF legal knowledge systems. The LKIF is built on but go beyond this generic work to allow further kinds of legal knowledge to be modelled, including: meta-level rules for reasoning about rule priorities and exceptions, legal arguments, legal procedures, cases and case factors, values and principles.

In order to establish explicit links between legal sources and legal knowledge, an open, general, jurisdiction independent and language independent XML based representation language able to describe legal sources will be defined as one of the main objectives of WP3. This standard for legal sources will be defined so that it can easily be used and coupled to LKIF knowledge models of these sources. Moreover, another primary objective of WP3 is represented by the specification and development of a Legal Sources Management System based on this standard format.

This deliverable is addressed to analyse existing national standard formats for legal source representation.

This analysis is the background to further activities within the project, leading to the definition and management of a European XML standard for legal sources. In particular, within such further activities, an evaluation of existing XML based legal drafting tools for specific national standards will be carried on in order to implement a prototype of ESTRELLA Legal Sources Management System; similarly current Content Management Systems (CMS) used in a legal setting and appropriate auxiliary legislation defining the way legal sources should be managed and used will be analysed.

Approach, methods and activities

This deliverable reports a comparative analysis of the main standards for legislative documents at European and extra-European levels. The aim is to identify specificities and common features of such different formats. They will represent the ground from which the new open, general, jurisdiction independent and language independent XML based representation for legal sources will be defined. Such analysis is mainly based on official documentation of the considered legislative XML projects, as well as on the results of the related standard formats test on specific case-studies.

Summary of results and conclusions

Specific strengths and weaknesses of each standard are underlined in this deliverable, pointing out the characteristics which can be useful for the development of a new representation format for legal sources to be shared at European level.

From the comparison of different national standards some conclusions are drawn in order to provide guidelines for the definition of an ESTRELLA standard for legal sources. In particular some issues are identified as relevant for the new standard: the approach in defining the new schema, consisting in specific criteria of implementation (in terms of extendable layers of increasing degree of complexity) and the language with which expressing the syntax of the new schema; the support of multilingualism in order to take into account the European perspective of the new building schema; the document semantics, namely metadata and vocabulary of terms with which expressing the semantics; an ontology of the type of documents to be described; times and dynamic properties of the legal sources the new standards will deal with; finally the identification of such documents as well as their components.

Deliverable organisation

This deliverable is organised as follows.

In Section 1 the main activities within WP3, as well as partners involved, their specific tasks and efforts are shown.

In Section 2 an overview of the main existing national standards for legal documents is described.

In Section 3 a comparison among different national standards, the methodology used for such comparison, a qualitative analysis of them and some conclusions of this analysis are illustrated.

Section 4 discusses how existing experiences in representing legislative documents can be effectively used to derive a standard for legal sources fitting the project requirements.

Finally, in Section 5 some conclusions related to the comparison among national legal standards, including their main characteristics, useful for the European standard to be defined, will be discussed.

General XML format(s) for legal Sources

Deliverable D3.1

Caterina Lupo, Fabio Vitali and Enrico Francesconi
CNIPA – MRIPA

Monica Palmirani
University of Bologna

Radboud Winkels, Emile de Maat and Alexander Boer
University of Amsterdam

Paolo Mascellani
Consorzio Pisa Ricerche SCARL

Contents

ESTRELLA
p/a Faculty of Law
University of Amsterdam
PO Box 1030
1000BA Amsterdam
The Netherlands

tel: +31 20 525 3485/3490
fax: +31 20 525 3495
http://www.estrellaproject.org

Corresponding author:
Caterina Lupo
tel: +39-06-85264262
lupo@cnipa.it
http://www.cnipa.it

Chapter 1

Workpackage 3 in the ESTRELLA project

1.1 Objectives

The primary aim of this WP is to specify an open, general, jurisdiction independent and language independent XML based representation language for legal sources that can easily be used and coupled to LKIF knowledge models of these sources. Furthermore, this WP is addressed to specify and build a prototype of a Legal Content Management System based on this representation format, using Open Source components where possible. As a subordinate aim of the WP the validation of MetaLex XML as an interchange standard will be taken into account.

A central requirement of automated systems based on representations of legal knowledge is an explicit link to the original legal sources the knowledge is based on. Preferably this link should be isomorphic: each argument should be backed with plausible references to legal sources proving the validity of the argument. This is not a trivial task. Legal sources come in many forms and formats, from different jurisdictions using different legal systems, in different languages, with different internal structures, etc. Legal sources are not self-contained entities but are meaningfully related to each other in one jurisdiction by for instance temporal relations, and relations of priority and relative legislative competence. Each jurisdiction has separate rules for assignment of identity, for versioning, for establishment of validity and applicability etc. Jurisdiction specific rules are captured in an XML based representation language specifically designed for configuring a Content Management System, so that it can use the appropriate regime for managing legal sources from that jurisdiction. The legal sources themselves are stored in a single Content Management System architecture that is general enough to respect the legal content management requirements of multiple jurisdictions, while allowing manipulation of all legal sources in a single XML language designed for that purpose: MetaLex XML (`http://www.metalex.nl`). The Content Management System can use existing mature national XML based services for valid legislation like the Norme-in-Rete service in Italy (http://www.normeinrete.it), or AWB in the Netherlands (`http://wetten.overheid.nl`), and translate other XML languages to the appropriate XML language for internal manipulation.

1.2 Partners and tasks

Partners of the project specifically involved in this WP (according to the project identification number) are:

01 – UVA - University of Amsterdam (NL)

02 – UNIBO - University of Bologna (IT)

06 – RuleBurst - RuleBurst (EUROPE) Limited (UK)

07 – KI - knowledgeTools International Gmbh

09 – SOGEI - Società Generale d'Informatica S.p.A. (IT)

10 – MRIPA - Ministro per le riforme e le innovazioni nella Pubblica Amministrazione (IT)

12 – CUB - Budapesti Corvinus Egyetem - Budapest (HU)

15 – CPR - Consorzio Pisa Ricerche (IT)

Partners in this WP are involved according to the following person-months alloction schema:

Participant id:	01	02	06	07	09	10	12	15
Person-months per participant:	4	3	2	2	2	5	9	6

The Italian National Centre for ICT in the Public Administration (CNIPA), on behalf of MRIPA (the Italian Ministry for Reforms and Innovations in the Public Administration), is the WP leader.

Partners, according to their specific experiences, are differently involved into the two main specific areas of interest of this WP, which are the definition of the legal standards and the analysis of legal document CMSs as well as the development of a prototype of the ESTRELLA Legal Sources Management System.

In particular in the WP the following job allocation has been adopted:

1. MRIPA, UVA, UNIBO, RuleBurst and KI are involved in the analysis of existing solutions in Europe for representing legal sources, as well as in the definition of an open, general, jurisdiction independent and language independent XML based representation standard for legal sources that can easily be used and coupled to LKIF knowledge models of these sources as defined in WP1.

   Along with such XML standard analysis and specification, the analysis of current CMSs used in a legal setting and appropriate auxiliary legislation, defining the way legal sources should be managed and used, is carried out.

   Partners specifically involved in this activity are MRIPA and SOGEI.

2. UVA, UNIBO, CUB, CPR are involved in the specifications and development of the Legal Content Management System based on this representation format.

Chapter 2

Representing legal document throughout XML languages

2.1 Current main available European XML standards

Currently in Europe different national initiatives have introduced standards for legal source description as “MetaLex” (Section 2.2) and SDU BWB (Section 2.3) in the Netherlands, “LexDania” in Denmark (Section 2.4), the “NormeinRete” project in Italy (Section 2.5), “eLaw” project in Austria (Section 2.7.4), “CHLexML” in Switzerland (Section 2.7.1).

Such initiatives mainly promoted by national institutions, involving also research institutes and universities, have established XML standards for legislation in the related countries, as well as schemes for legal document identification.

Out of the European context, another important initiative, the “AKOMA NTOSO” project (Section 2.6), coordinated by UNDESA (United Nation Department of Economic and Social Affairs), has been launched by the pan-African Parliaments organization. It aims at the definition of an XML based document format for legislative and parliamentary documents in African Parliaments.

Another initiative in legislative standardization, named “EnAct” (Section 2.7.2), involve Tasmania government, Canada government, some federal state of US and New Zealand government.

Similarly in US an interesting initiative, named “Legal-RDF”, has recently proposed a standard for legal documents based on XHTML (Section 2.7.3).

Other similar initiatives in legal domain are working at international level to define non-proprietary standards for legal documents (Section 2.7). For example LegalXML has created a forum to define standards for the electronic exchange of legal data; other initiatives are mainly devoted to the development of common standards across websites for legal contents, as “Legal and Advice Sectors Metadata Scheme” (LAMS) in UK, “Justice Sector Metadata Scheme” (JSMS) designed for the use of organisations in New South Wales and UK Metadata Framework (UKMF).

These experiences are mature enough to represent a knowledge background for promoting and developing a common XML language for legal sources to be shared at European level.

Here below the main features of the standards defined within the related national projects are illustrated.

2.2 MetaLex

2.2.1 History

MetaLex has been developed as part of the E-POWER project. This project was aimed at the use of ICT to support citizens and governments in dealing with an increasing number of regulations. European citizens and enterprises are more and more confronted with rules and regulations. This affects various aspects of their daily business. Regulations come from international, European, national and local authorities. Despite attempts at harmonization and de-regulation, the size and complexity of the potentially relevant body of 'law' continuously increases. This is a problem for administrations too, legislative and executive bodies alike. The process of drafting consistent and coherent legislation is getting more complicated, as is that of upholding and applying valid law. ICT has the potential of supporting both the government and citizens in dealing with this increasing body of law.

A precondition for the tools that were developed in E-POWER was the electronic availability of legal sources in a structured and standard format. MetaLex was developed to fit this need. In addition, it allows for exchange and comparison of legal documents from different sources (such as publishers). It provides a generic and easily extensible framework for the XML encoding of the structure and contents of written public decisions and public legal documents of a general and regulatory nature. MetaLex poses only minimal requirements on the structure of documents.

Currently, MetaLex is used by the Dutch Tax and Customs Administration, Be Value, the Belgian Public Centers for Welfare and others.

2.2.2 Brief description of language main features

2.2.2.1 Strengths

An important strength of MetaLex is that there are few restrictions on its use. It is independent of both jurisdiction and language. In addition, MetaLex XML can be freely mingled with other XML schemes. This means that MetaLex does not interfere with e.g. the proprietary XML formats being used by publishers, and can therefore be implemented within existing organisations without causing problems.

MetaLex XML has a strict division of containers (elements that contain other containers or blocks in a defined sequence), blocks (elements occurring within containers, containing mixed text and inline elements) and inline elements (elements that occur within text). This results in a clear structure with less chance of errors.

MetaLex is based on several existing standards of the World Wide Web Consortium: XML, RDF, RDF(S), OWL, XLink, XHTML, Web naming and addressing (URI), XSL, and XML Base. The link with RDF and OWL is especially important, since it allows for a decentralised storage of knowledge: no demands are made on the contents of a single file. By requiring every element to have an ID, MetaLex supports URIs on a very detailed level (it is possible to link to very specific elements of the text), but it avoids commitment to a complicated syntax and semantics for URIs.

In addition, MetaLex can be combined with the Geography Markup Language

(GML) to link the regulation to specific (spatial) areas, which opens the way to visualisation of the jurisdiction (of parts of) legislation on a map. It also allows access to legislation through maps.

Finally, work is in progress to make MetaLex a CEN/ISSS norm.

2.2.2.2 Weaknesses

The main weakness of MetaLex is its lack of software support, which is true for most legal XML formats. Of course, regular XML tools can be used to create MetaLex XML files, but no MetaLex-specific tools or editors exist. Providing this software support is the most important challenge MetaLex faces. There are some old editors that fell into disuse, like the E-POWER workbench, that writes MetaLex. New MetaLex aware editors are under development at the Leibniz Center and Be Value.

2.2.3 Complete language description

2.2.3.1 Overall design principles

MetaLex is based on a number of design principles. It is the aim of MetaLex to separate primary and secondary text in a legal source. In addition, the official text by the legislator is separated from text created by an annotator.

Next, MetaLex is designed to be independent of jurisdiction. The professional user of legislation today has to take notice of regulations from several legislators (for instance municipality, water authority, and provincial, national, EU, and international governments, for a civil servant in the Netherlands), and special-purpose software to support decision making processes is affected by - and may have to manipulate - legislation that conforms to many different standards for legislative drafting and is delivered in as many different formats. To be truly useful for these users and these tools, a language should be wholly independent of jurisdiction. MetaLex is therefore limited to the few features that regulatory documents from these different jurisdictions share. This leads to a simple and generic but also rather 'trivial' structure that does not necessarily meet specific requirements of potential users. To allow for these specific needs, MetaLex has been structured in such a way that it is possible to add custom extensions to the schema; MetaLex makes the easy things easy, but the hard things possible.

The next the design principle is compliance with open standards and proposals of the World Wide Web Consortium and other standardization bodies that are supported by standard or Open Source software. This reduces the effort of implementation and learning curve for XML developers and increases the usefulness of MetaLex documents. For this reason, the MetaLex standard is specified using W3C's XML schema and RDF, and supports features from standards such as (X)HTML and XML Linking Language (XLink).

Finally, a consequence of the increasing global presence of supranational legislators like the EU is a growing need to separate regulations as such from the specific authorized translations in which they are available. We have to accept as an axiom that for instance a citation in a French text to an international treaty can be resolved to the English translation of that treaty without a change in meaning for users who prefer English. Legislation is increasingly available in multiple authorized translations as a service to immigrants and - in the case of for instance fiscal regulations - potential immigrants. Therefore, a final design principle is to design MetaLex with multilingual regulations and differences between the main European languages in mind.

2.2.3.2 Representation of document structure

Partition structure In MetaLex, a regulation is split into articles. Articles are self-contained discourses in the sense that they can be read and understood without reference to nearby articles to resolve anaphoric references.

Articles can be grouped into parts (which themselves can be grouped into parts. Thus, a regulation can consist of several layers of parts, which bottom out in articles. The parts themselves do not contain any text (sentences); only the articles contain text. Each part may contain a title and/or index, consisting of a type (such as chapter) and an index ("1", "1bis", "α").

A basic article consists of one of more sentences. More complex articles may not only be also contain lists and/or subparts (such as members) which themselves can contain subparts, lists and sentences.

Actual parts of the text are tagged using the textversion-element. This element allows for a language-attribute (lang) that denotes the language in which the text enclosed by the textversion-tags is written. This way, it is possible to have several version of the same text in different languages in the same document.

Text that is part of the actual regulation appear in Textversion-elements that are part of Sentence-elements and SentenceFragment-elements and SentenceFragmentSubPart-elements. Text that is not part of the actual regulation appears in Textversion-elements that are encapsulated by Annotation or Appendix elements (see below).

Granularity of mark-up The granularity of mark-up differs for different section of the document. The actual rules and norms in the regulation are split up into full sentences, or, if it is a sentence containing a list, in several sub sentences.

All other parts of the regulation, such as introduction, conclusion and appendices, as well as text that is not part of the regulation (annotation, see below) may appear as a whole in a MetaLex document. The user may split the text into as many parts as he finds appropriate.

Quote part of text Quotes from other documents can be marked using the quote element.

Table management There is no specific support for tables in MetaLex. However, since MetaLex may be mingled with any other type of XML, HTML tables (for example) can be used for tables.

Annexes MetaLex allows for a regulation to have zero or more annexes or appendices at the end of the document. Contrary to the actual regulation, the appendices consist only of text; it is not structured in any way.

Other features Publishers often add helpful commentaries and annotations to a regulation they publish. In order to make MetaLex compatible with not only the "pure" regulations but with these "embellished" regulations as well, MetaLex allows for annotations in the document.

Any MetaLex element that contains part of the text of the regulation can also contain zero or more annotations. Each annotation itself consists of text and further annotations.

2.2.4 Identification of documents and document parts

MetaLex does not prescribe an URI for documents. Instead, users are free to use their own URI definitions. However, users are required to define an `xml:base` attribute for their MetaLex documents; although the format of URIs is not prescribed, URIs must be used.

2.2.5 Knowledge management

2.2.5.1 General meta-information

MetaLex elements contain three groups of meta-information. First of all, a MetaLex element contains three meta-information elements on the creation of the legislation. These are:

1. author: the legislator that is responsible for creating the legislation, for instance "the government", "government and parliament", "the Crown", "the Minister of X", etc.

2. authority: the competence the legislator has used to create or change this legislation.

3. procedure: the type of procedure that resulted in this legislation.

Next are two attributes concerning the editor of the document:

1. editor: the editor that is responsible for creating and managing this MetaLex encoding of the legislation.

2. editor-resolver: this is an URI that combined with xlink:href should result in an URI (for this document) that can be resolved. This is necessary since the URI of the document is an identifier for the document that is not required to be resolvable through generic HTTP.

Finally, two attributes are available to connect the regulation with an area (spatial entity) in which the rules in the regulation can be applied.

1. region: the spatial entity. This will be an URI referring to coordinates in a coordinate reference system if MetaLex is combined with GML.

2. region-resolver: a resolver for the URI in the region attribute.

2.2.5.2 Document classification

MetaLex does not provide any attributes for document classification. Regulations and legal decisions are separated because they use a different MetaLex XML schema, which have a different root element.

2.2.5.3 References management

MetaLex XML allows for two types of references, namely:

1. references to documents, such as “The Rome Statute, Article 1”
2. references by name to a known person (“the King”), a body (“the Government”), (the territory of the Kingdom of the Netherlands in Europe) or concept defined in a regulation.

The references can be given through an URI as well as an URL. A date can also be added to the reference to keep track of the version being referenced.

2.2.6 Time management

MetaLex keeps track of dates and periods that are relevant to the lifecycle of legislation. First of all, the date of publication is tracked. Next, the period in which the legislation is active (that is, during which the legislation may be applied) can be marked, by noting the date of enactment and the date of repeal.

To deal with more complicated aspects, two more periods can be marked in MetaLex. Legislation can be retroactive, which means that it can be applied to facts that occurred before the legislation was published. An example of such legislation: In 2001 there is a tax provision which states that the premium payments to be made in 2001-2005 for certain capital insurance policies closed in and meeting certain conditions in the period 1993-1998 are tax deductible.

The period in which facts must have occurred in order for this legislation to be applicable is called the efficacy period.

Similarly, legislation may have an effect that reaches past its own active period. The four periods which MetaLex keeps track of are shown in the picture below: In

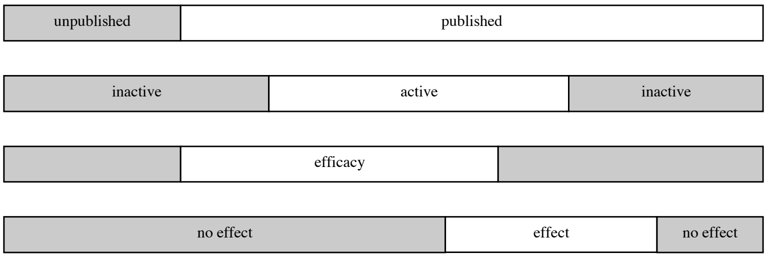

Figure 2.1: Metalex four periods

order to keep track of these periods, each MetaLex element has several attributes:

- Publication date
  - date-publication: the date the regulation was published.
- Enactment date
  - date-enacted: the date the regulation was enacted.
  - duration-to-enactment: the amount of time between the publication date and the enactment date. Used when the enactment date is not given (i.e. a law that will be enacted three months after publication).
- Date of repeal
  - date-repealed: the date on which the regulation was repealed.
  - duration-active: the amount of time between the date of repeal and the enactment date.
- Efficacy period
  - date-start-efficacy: the starting date of the efficacy period.
  - date-end-efficacy: the ending date of the efficacy period.
  - duration-efficacy: the amount of time between the starting date and the ending date of the efficacy period.
- Effect period
  - date-start-effect: the starting date of the effect period.
  - date-end-effect: the starting date of the effect period.

For version control, a final date is added:

- Version
  - date-version: the date of the serialization of the XML document.

MetaLex does not provide an attribute to keep track of all modification dates. Instead, only the most recent modification date is stored (in the date-version attribute). In a consolidated regulation, this means that different parts may have different dates, as some parts of it may have been modified, while others are still original.

In addition, it is possible to keep several versions of a text in the same MetaLex document, each of them marked with a different value for the date-version attribute (and probably different values for the other date attributes as well).

2.2.7 Support for visualisation and text formatting

An MetaLex element can easily be transformed to a human-readable text with a familiar layout by application of an XSL transformation. XSL sheets for translation to XHTML, RTF and PDF are included with MetaLex. It preserves the identifiers of the elements so that each element has a valid URI by which it can be referenced. References to other documents are embedded as hyperlinks.

2.2.8 Native support or extensibility for integration with other kind of knowledge representation standards

It is possible to make external statements in RDF refer to the right part of a MetaLex document, because each element has an identifier. Regulations in MetaLex XML can be transformed to MetaLex RDF using a XSL transformation [Boer et al., 2003]. The MetaLex RDF is defined in an OWL schema, which brings with it description logic validators (such as Pellet, FaCT++ and Racer) as well as expert system rule engines (like JESS/SWRL/RIF).

2.3 SDU BWB

In this section it is briefly described the Dutch SDU BWB standard. This XML format is currently used for encoding the laws in the Dutch Basiswettenbestand (BWB) database, that is a large database containing almost all Dutch laws and decisions. The standard is based on a DTD originally developed by SDU publishers, and now maintained by the Dutch government.

2.3.1 Representation of document structure

2.3.1.1 Partition structure

A law or decision is divided in intul, introduction, the actual text, closure and appendices. The structure of the actual text is given through the following regular expressions:

$$\begin{array}{lll}
\text{regulation} & \rightarrow & \text{book+}|\text{part+}|(\text{chapter}|\text{title}|\text{department}|\text{paragraph}|\text{article})^{*} \\
\text{book} & \rightarrow & (\text{chapter}|\text{title}|\text{department}|\text{paragraph})^{*} \\
\text{part} & \rightarrow & (\text{chapter}|\text{title}|\text{department}|\text{paragraph}|\text{article})^{*} \\
\text{title} & \rightarrow & (\text{chapter}|\text{department}|\text{paragraph})^{*} \\
\text{department} & \rightarrow & (\text{chapter}|\text{paragraph}|\text{article})^{*} \\
\text{chapter} & \rightarrow & (\text{title}|\text{department}|\text{paragraph}|\text{article})^{*} \\
\text{paragraph} & \rightarrow & (\text{department}|\text{subparagraph}|\text{article})^{*} \\
\text{subparagraph} & \rightarrow & (\text{article}|\text{subparagraph})^{*} \\
\text{article} & \rightarrow & (\text{member})^{*}
\end{array}$$

All these structure elements can contain actual text. This text comes before any subparts. The only exception to this is the article, which can mix text and members. Instead of the structure given above, a regulation may also consist directly of the actual text (without any intervening layers) or it may contain a single article (which is a separate element from a "regular" article).

A regulation is structured similar to a law or decision, but it cannot contain books a parts. However, it may contain (at top level) modification articles, which contain modification to existing laws. These modification articles can themselves contain law text, which may be structured as is given above.

A circulaire is composed, like the other documents, from an intul, an introduction, the actual text, closure and (possibly) appendices. However, the actual text is composed differently than it is within the other documents. In a circulaire, text can be unstructured, divided into articles or follow a more layered structure. This layered structure is less formal than it is within the other documents, and therefore, a more abstract approach is chosen. A circulaire can be split into divisions, which can themselves consists of divisions. A division differs from the other structure elements in that it need not be called a division in the actual text, while a book should also be named a book in the actual text.

The fourth structure, the treaty, consists of an intul, an introduction, the actual text and a closure. The actual text may be structured like a law (the first structure) or similar to a circulaire (the third structure, which is more loose).

2.3.1.2 Granularity of mark-up

BWB XML divides the text into individual paragraphs. Actual parts of text are included in paragraph (different from the paragraph structure element) and list elements. These appear in articles, members, divisions and in the structuretext element (which may be included at the start of any other structure element. If no structure is present in the document, paragraph and list elements can appear directly below a regulation-text, circulaire-text or treaty-text element.

2.3.1.3 Quote part of text

There are two different ways of quoting text using BWB XML:

1. Quoting a legal text (using the wetcitaat element). The text that appears in this element can be structured into articles and members using other XML members.

2. Quoting other texts (using the aanhaling element). This text is not structured.

In both cases, attributes are provided to link to the quoted document.

2.3.1.4 Table management

BWB XML supports tables based on a simplified version of the CALS Table Model.

2.3.1.5 Annexes

Laws, decisions, regulations and circulaires may contain appendices. An appendix consists of text that may be divided using division elements. It may also contain a closure.

2.3.2 Identification of documents and document parts

Each structure element has an ID, which can be used for identification. For use in the web based system, two other IDs added:

- Base-ID: An identifier for a specific part of a regulation. It encompasses all versions of that part.

- Version-ID: An identifier for a specific version of a specific part of a regulation.

2.3.3 Knowledge management

2.3.3.1 General meta-information

The general metadata includes:

- The publication in which the legislation was published.
- The location where it was created, and the date on which it was created.
- The name of the ministry (or other body) that is the main responsible for the legislation.
- The official title of the document, any other titles it may have and any abbreviations of these titles.
- An overview of the parliamentary history of the regulation.
- References to those laws that empower the regulation.
- In case the document is a translation of another document: a reference to the original and the name of the translator.
- In case of a treaty: the names of the different parties that take part in the treaty.

2.3.3.2 Document classification

Documents are classified as one out of 20 different types (including laws, decisions, Orders of Council, EU Guidelines and treaties).

2.3.3.3 References management

Other documents may be referenced using a extref element. The extref contains a URI for the referenced sources (split into a URI for the entire document and an anchor within that document), as well as a description and a status indicator: active, sleeping (does not yet exist) or repealed (does not exist anymore).

For references located within the document itself, an intref element is used (which does not have a URI for an external document).

References may be grouped together in a extref-group (or a intref group).

2.3.4 Time management

In BWB XML, the entire history of each article is tracked in relation to modifications to the text of the article, though only the latest version of the text is included in the document. For each modification, the source document of that modification is listed, along with the nature of the modification. These can be:

- Original (first) text
- Renumbering

- Modification
- Renumbering and modification
- Repeal (of this specific element)
- Repeal (of the entire document)
- Re-introduction
- Translation
- New

The source document is identified by its publication type, year of publication and date of publication. The sign date is also included in the meta data. Next, the date of enactment is given, together with the publication in which this date has been published (for this publication, publication type, year of publication and date of publication and sign date are included as well).

The metadata includes the current source of the text (including the source of the enactment date), as well as a history (containing all sources that have modified this text in some way, including the last).

2.3.5 Support for visualisation and text formatting

BWB XML provides tags with information on how a text should be presented to the user. Available elements are:

- Emphasis: for text that should be in italics, bold, smallcaps etc. The exact font modifcations are listed in an attribute.
- Superscript
- Subscript
- Underlined (for formulas)
- Overlined (for formulas)

2.3.6 Other metadata

BWB XML allows for several kinds of editor's notes: comments, markers for official/unofficial information and markers for tentative/active information.

2.4 LexDania

The LexDania project is defining a national Danish system for the creation and interchange of legislative documentation. LexDania was initiated by the Danish Ministry of Science, Technology and Innovation then continued by the Retsinformation (Ministry of Justice) and the Folketinget (Danish Parliament). The work was conducted in two phases. In a first phase, a research on international activities and an investigation in other national standards and projects was done. In second phase, the development followed using the basic data model / methodology of the General Danish Public Information Online (OIO) XML strategy. This strategy consists in choosing a set of central types and elements (standards if possible), creating sets of "building blocks" for national use, (re-)using building blocks to create specific legislative schemas.

The project is focusing on developing a system of schemas for the systematic creation and maintenance of document type and application schemas. The system has an unique approach to building schemas. A structure of stratified layers is used to incrementally construct the schemas from functional features - rather than document characteristics. The structure is accompanied by a methodology explaining ways of constructing schemas to assure consistent and compatible schemas.

2.4.1 Brief description of language main features

XML Schema LexDania project uses extensively the XML Schema Definition Language (XSD) [Tucker, 2004b] [Tucker, 2004f]. XSD has been chosen because it is considered an ideal language for constructing schemas in a systematic way due to its ability to declare data types and to modify them in subsequent schemas (providing a controllable yet flexible environment). The choice of XSD has been also compared with its compatibility (interchange) with international law/ projects, and the common goals/ tools/ discussions/ experience in XSD.

The system methodology The standards (XML, XSD, RDF, and DC) are general tools and include facilities for almost every need. Therefore, LexDania has discipline and restrictions to achieve a robust and maintainable system. The framework is accompanied by a methodology (see: Guidelines for Writing Omni, DocType and Application Schemas) that imposes rules for writing schemas (XSD constraints) – but without limiting functionality. The methodology supports the creation, management, and maintenance of many schemas for both domains and applications.

The functional approach Having a common and consistent functional approach provides several advantages. The meta-schema provides syntactic details for all schemas, thus these numerous and repeated details only need to be designed once but can be re-used many times. This approach also positively influences maintenance, documentation and the "learning-curve". All meta-schema components have a common syntax, congruent linking, addressing, container structures, etc. that enables exciting possibilities for future developments, such as automated information fragment interchange. The intermediary level of schemas, provide functionality for individual domain specific needs, and thereby supporting the re-use of a common syntax by separating out the physical implementation details.

2.4.1.1 Strenghts

The most significative contribution of the LexDania schema definition seems to be its modularity which allows to derive specific schemas for particular kind of documents from the same common root.

This characteristic of extensibility follows a well-grounded inheritance mechanisms of elements and data types according to an object-oriented development philosophy. The inheritance mechanism allows to increment the semantics and the details of the data types for each layer in LexDania schema system.

During the design and implementation phases of the project, LexDania reached consensus on some XML "best practices". These are described in [Tucker, 2004e]. Applying these practices across a broad range of domains within the legislative domain is possible; this framework make it easier writing schemas as well as their modifying and maintaining.

2.4.1.2 Weaknesses

Including restrictions as regards data types increases control over document contents but, as a drawback, it can represent a burden for the drafting and validation activities.

2.4.2 Complete language technical description

2.4.2.1 Overall design principles

To systematise and simplify the writing of many diverse schemas, a general system framework based on gradual refinement of functionality, inheritance and re-use of data types and components have been developed. The framework is stratified, with each layer consisting of schemas. A system with 3-layers has been implemented: a meta-schema in the first layer, omni-schemas in the second and application schemas in the third (Figure 2.2). (The number of levels is dependent upon the complexity of the system being supported – they consider 3 levels as the optimal number for the LexDania needs). Each schema layer, rather than encoding document semantics, contributes a specific "functionality". The system functionality is divided so that the basic level defines the syntax; the intermediary layers define domain specifics; and the outer layer defines the document and application specific features and functions.

In the LexDania implementation, the fundamental syntactic details are defined in a single schema `LexDania.xsd` - sometimes referred as the meta-schema. The intention is that the data types of the LexDania meta-schema will be used as the basis for defining specific legislative document schemas (using the XSL inheritance constructions). In other words, the schema `LexDania.xsd` defines data types that are used as templates for (i) the content (ii) the structure of all legislative documentation (iii) addressing and (iv) referencing scheme. (The metadata section is supported by a separate RDF schema for importing and defining vocabularies.)

As already pointed out, the LexDania schema system consists of three layers of functional schemas. The project's data model and methodology encourage and control the incremental development. Further layers build upon the data types and declarations gradually refining and enhancing them with environmental, domain, and application semantics. The end result is the layer of semantically-rich explicit application/ document-type schemas (see Figure 2.2).

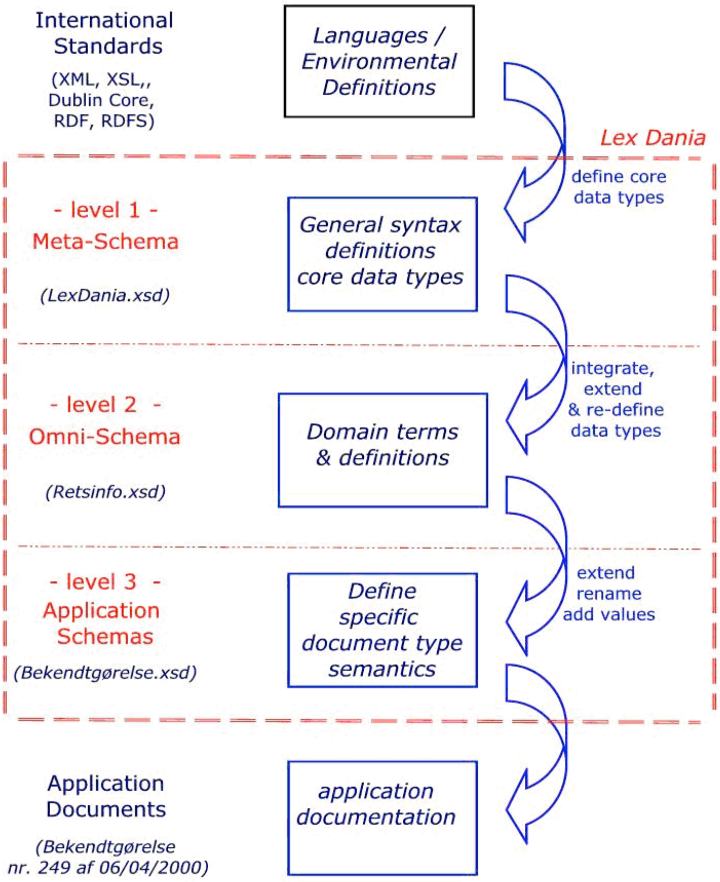
International Standards
(XML, XSL,, Dublin Core, RDF, RDFS)
Languages / Environmental Definitions
Lex Dania
define core data types
- level 1 -
Meta-Schema
(LexDania.xsd)
General syntax definitions core data types
integrate, extend & re-define data types
- level 2 -
Omni-Schema
(Retsinfo.xsd)
Domain terms & definitions
extend rename add values
- level 3 -
Application Schemas
(Bekendtgørelse.xsd)
Define specific document type semantics
Application Documents
(Bekendtgørelse nr. 249 af 06/04/2000)
application documentation

Figure 2.2: Overall design structure

2.4.3 Representation of document structure

2.4.3.1 Partition structure

In the LexDania implementation, the fundamental syntactic details of the basic level are defined in a single schema - referred also as the meta-schema [Tucker, 2004e]. This schema, `LexDania.xsd`, defines data types that are used as templates for (i) the content (ii) structure of all legislative documentation (iii) common addressing (iv) referencing scheme.

1. Content: contains basic content data type core definitions. They use the Cara_type data type to create Content elements.

2. Structure: contains general strucuture components or so-called containers. Arca_type (In Latin Arca means “container”) data type is used to create structure elements (p29 guidelines).

3. Addressing: contains addressing / references to data type core definitions.

4. References: contains references and reference structures for LexDania constructs.

There is also a metadata schema supported by a schema (initially envisaged to be written in RDF) for importing and defining metadata vocabularies.

2.4.3.2 Granularity of mark up

The granularity of mark-up is refined through the hierarchy of schemas. It is defined and fixed at the application-schema level.

2.4.3.3 Quote part of text

LexDania standard uses the Cara_type data type to create text elements. Cara_type is a "place holder" and the definitions of text elements is at mercy of the application schema writer. Cara_type is a non-empty sequence of characters (chardata) and there is no mark-up allowed in Cara_type elements, so there is no mixed content. Cara_type has, as all content containers, a number of "standard" attributes, ID, lang, etc.

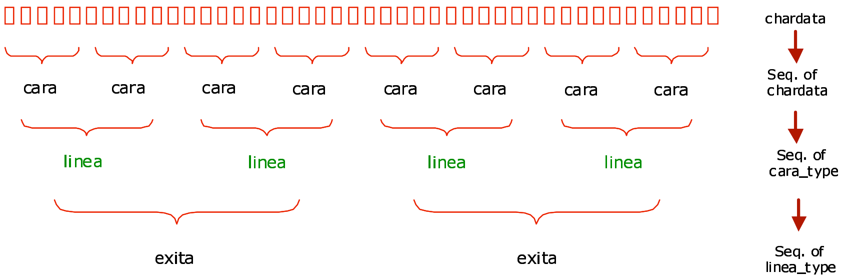

Figure 2.3: The progression of content complexity from simple sequences of chardata to sequences of more complex content

Others related data types are:

- Linea_type to deal with phrases, sentence fragments, sentences, lines of text, etc. It is a non-empty sequence of Cara_type elements.
- Exita_type to deal with blocks of text, paragraphs, verses, etc. It is is a non-empty sequence of Linea_type, Caudex_type, Mensa_type, Figura_type, and/or Epistola_type etc.
- Caudex_type to deal with lists. It is a non-empty sequence of Exita_type
- Epistola_type (notes)
- Mensa_type (tables)
- Figura_type (image)
- Articulus_type (point, item, sequence of blocks of text)

2.4.3.4 Table management

LexDania standard uses the Mensa_type data type to create tables. Mensa_type is "place holder" and the definitions of tables is at mercy of the application schema writer.

2.4.3.5 Other features

Notes:

1. LexDania does not use namespaces; this choice is motivated by the fact that it makes the definitions and declarations of the schemas and data types easier to read and to work with.
2. There is no mixed content (elements that contains elements and texts), because it is considered to make programming easier.
3. LexDania uses attributes for Common (global) modifiers, otherwise elements are used.
4. In general, Types, attributes, elements are in Latin, documentation is usually in English and examples are in Danish.

2.4.4 Identification of documents and document parts

2.4.4.1 URI definition for the whole document

LexDania uses URN to identify documents[1] [Tucker, 2004c], [Tucker, 2004a].

2.4.4.2 Version identification

Concerning the versions of schema, a simple versioning method to track version history has been chosen (during the development stages a schema versioning is used). The versioning is part of the component `id` such that the full component `id` becomes:

```
id = "componentName_versionNo-appName-LexDania"
```

e.g.:

```
id = "Title_type-Retsinfo-LexDania"
```

If needed, this method could be extended to include both application and component versions by adding version numbers as suffixes. For example, the following would identify the title component:

```
id = "Title_type_2-Retsinfo_3.1-LexDania_9.3"
```

The identity of this component is:

- version 2 of Title_type that is defined in:

[1] The document "LexDania Documentation Universal addresses" does not explain much about the way the URI is constructed

- version 3.1 of the omni-schema Retsinfo, that originates from:

- version 9.3 of the meta-schema LexDania.

Concerning versions of documents: amendments to acts are always passed as the very content of an act or as part of another act. Consolidated acts (a new version of an act and its later amendments) are made on the initiative of the ministry administrating the act in question. All 4 documents below are separately officially published:

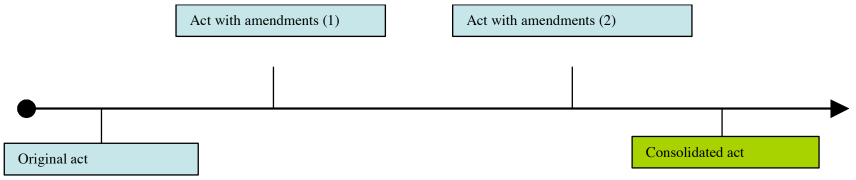

Figure 2.4: LexDania document versions

Exact mark up of different dates of the coming into force and validity enables us to (in principle) show automatically generated versions of documents with the exact valid content on a chosen date. How small units one can change by acts of amendments? Probably not all kinds of amendments as they occur today can be handled, e.g. "*2. In §11, §36 and §125 substitute "100.000 DKK" with "500.000 DKK"*".

2.4.5 Knowledge management

2.4.5.1 General meta-information

Metadata have been designed starting with the top-down considerations imposed by the internationally accepted standards and practices but weight is also put on the bottom-up issues - the existing (legacy) systems and documentation. The design approach from both the top and bottom aims to be internationally compatible but also to assure a sensible, down-to-earth, end-user format. Doing so, five levels of refinement for metadata have been identified (see Figure 2.5) [Tucker, 2004d].

Level 1 This level is the international level of specifications. For LexDania, the most important actors in the metadata arena are the activities of RDF (at the definition level) and Dublin Core (at the vocabulary level). All definitions, vocabulary and metadata, should directly use or be based as much as possible on these two activities. Adhering to these accepted trends will enable a robust and generally understood set of metadata terms as well as a migration path that a significant number of others will also be implementing.

Level 2 This level refers to the meta-schema. A specific schema concerning metadata has been created. According the documentation, the definitions from RDF and Dublin Core should have been imported in this schema. And, any new basic vocabulary should have been defined in RDF.

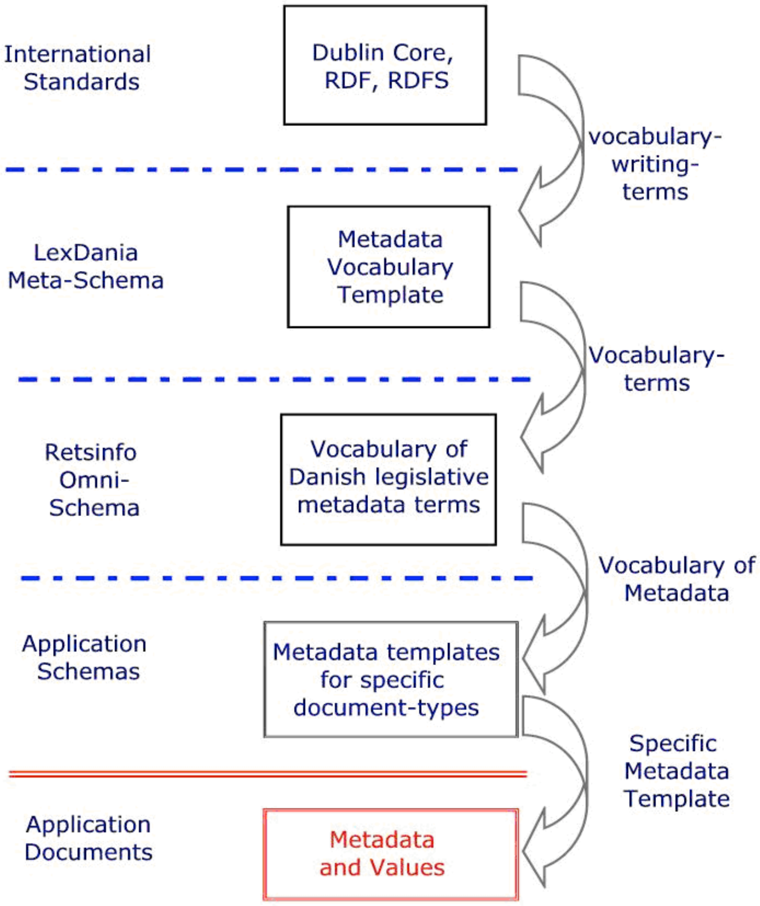

Figure 2.5: The levels of information about data or metadata.

Level 3 This level refers to the Omni-schema. This level is recommended to be used to map from RDF to XML (XSL). According the documentation, this should be accomplished by defining the LexDania vocabulary terms in RDF but providing an implementation in XSL.

Level 4 This level refers to the document-type, i.e. specific application factors. Document-type level schemas should import the data types an elements defined at the Omni-schema level. It is the intention that definitions from the Omni-schema level will completely satisfy the needs of the document-type schema. If new metadata terms are needed then it is recommended that the Omni-schema level be used to define them.

Level 5 At the end-user level, where the instance documents are produced, managed, and disseminated the use of XML format for metadata is recommended. There are several grounds for this recommendation, but practically the documentation tools and formats that are used today (and in the foreseeable future) are all XML (and XSL) oriented. If an RDF metadata solution were recommended then a new set of tools would be needed as well as maintenance, updating, training, etc. As well, this would introduce a new file format type and extra management of RDF schemas. However, restrictions that will allow us to convert the metadata to RDF are recommended, e.g., no nested logical structures. This approach will also assure legacy application compatibility and allow to remain within a time and economic budget (relevant to implementation).

2.4.5.2 References management

LexDania choices and discussion of linking constructs are described in a paper in the LexDania Documentation titled Introduction to Concepts and Status of XML Linking. After investigating the state of affairs in linking LexDania decided to implement a solution that was rather simple but robust - that of the proposed XHTML 2.0. Therefore LexDania have an attribute (ligio) at the meta-schema layer that functions as an `href` attribute. This is implemented on all containers. Addressing is divided into two functional groups - internal ID – IDREF attributes and external ligio attributes that should only be used with URNs.

2.4.6 Time management

Time management is part of the metadata in the top of each document (application schema). Two types of data are considered: those that refer to periods of time an those that correspond to events.

2.4.6.1 Efficacy, modification and in force date

Dates referring to a period of time (enforcement, validity etc.) are placed in elements by reference to an id for a “start date” and an “end date”. Concerning modifications, see Paragraph “Version identification”.

2.4.6.2 Events management

Single “physical” actions (signature, promulgation) are placed directly in elements in the source of the document. Example concerning the whole document:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="ACT M.Retsinfo_1.0.xsl" ?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="ACT M.Retsinfo.LexDania_1.0.xsd">

<Meta id="ThisDocument">
    <DocumentTitle>Act on annual accounts</DocumentTitle>
    <DocumentId>AD000903</DocumentId>
    <Number>448</Number>
    <Year>2001</Year>
    <DateOfPublication>2001-06-08</DateOfPublication>
    <AnnouncedIn>Lovtidende A</AnnouncedIn>
    <Ministry>Ministry of Economic and Business Affairs</Ministry>
    <PlaceOfSignature>Given on Christiansborg Castle,</PlaceOfSignature>
    <DateOfSignature>2001-06-07</DateOfSignature>
    <Seal>Under Our Royal Hand and Seal</Seal>
    <DateEffective id="Effective_ThisDocument"/>
    <StartDate REFid="Effective_ThisDocument">2001-06-09</StartDate>
    <DocumentType>ACT MAIN</DocumentType>
    <Rank>A</Rank>
</Meta>
```

2.5 NormeinRete

2.5.1 Introduction

The project NormeinRete [Lupo and Batini, 2003] (translation: Norms on the Net), or NIR, started in 1999 with the leadership of CNIPA together with the Italian Ministry of Justice, and it gathers several Italian public institutions and research organizations.

During the last years NormeinRete has reached the following results:

- a web portal (`www.normeinrete.it`) providing a unique access point for searching the Italian legislative corpus. It offers search and retrieval services operating on all Italian laws since 1904, and utilities for automated hyperlinking. The entire project documentations and other information related to the project are also available (in Italian). The portal includes e-learning facilities, a software download section to deliver open source utilities developed by the project team, and a best-practices section to encourage experiences re-use, in order to create a virtual space for knowledge sharing within the Public administrations community.

- a standard for XML representation of legal documents. DTDs and XML Schemas for Italian legislation have been defined; these schemas can represent metadata and all the significant information useful to automate legislative documents life-cycle management. Moreover, the availability of XML documents marked-up according to shared formats allows to provide advanced search and retrieval functions operating on heterogeneous data bases effectively.

- A standard for persistent identification of legal documents, compliant with the IETF Uniform Resource Name [Moats and Sollins, 1997], and an infrastructure for identifiers resolution and management.

The NormeinRete standards have been issued as Technical Norms by Italian Authority for information technology in the Public Administration (AIPA – now CNIPA), and published on the Italian Official Journal [AIPA, 2001, AIPA, 2002].

Nowadays, both standards have being adopted, in addition to the majority of Italian public administration, by a growing number of private operators. For example, the whole legislative documents collection of Italian Supreme Court of Cassation, the most wide and prestigious national collection in this field, adopt NormeinRete standards.

2.5.2 Main features

NormeinRete is the result of several years of collaboration between technology and law scholars, software developers, public administrators both at the national and regional levels. An open venue exist for discussion and feature requests exist and is listening to anyone who cares to propose an integration and amendment to the existing schemas.

2.5.2.1 Strengths

Arguably, the greatest strength of the NormeInRete project is the separation of the technical aspects of creating a shared standard from the related efforts of creating software that handles it. This means that in NormeInRete the characteristics of the software are derived rather than driving the characteristics of the standard, which allows multiple players to contribute software that are compatible with the standards and a healthy competition to exist.

This approach, albeit slower and more difficult than the classical approach of document data formats being the subproduct of an implemented software, has the advantages of keeping the actual details o the data independent of the software being used and of fostering interoperability at all levels.

Another important strength of the NormeinRete approach is that the data format is not meant to legislative offices only, but to all individuals and organizations dealing with legislative documents, both in the public and private sectors: creators, publishers, readers.

2.5.2.2 Weaknesses

NormeInRete exists, as a published data format, since 1999. In these years it has evolved considerably from a simple document format aimed at national legislation into a flexible tools that can be used to map and describe all the technical nuances of the complex and convoluted legislative system that is in place in Italy.

Even trends in how laws are drafted change over time, and Italy has a legislative tradition spanning over 140 years of history, including a kingdom, a dictatorship, a temporary secession and a republic, as well as documents spanning from national, to international, regional and local levels. Furthermore, after an initial stage of strict and careful design, NormeinRete has further grown out of aggregation, and once simple structures have evolved into rather complex content models out of the need to take care to individual exceptions found here and there in the complex body of the existing legislation.

For this reason, the full set of schemas constituting NormeInRete is huge and can be frightening at first. Many elements are in place for dealing with tiny special cases that have little impact on the majority of legislation, yet are relevant for specialized applications. The whole discussion about provisions can be considered of interest to scholars only, and possibly to implementers of specialized applications (such as generators of current texts), but are surely excessive for many simpler needs such as publishing or storage.

To partially overcome this problem, the Simplified schema, which is considerably simpler to learn and use, should be adequate for the needs of most users, yet still is fully compatible with the other, more complete schemas.

2.5.2.3 Open challenges

NormeInRete is now in its maturity as far as approved acts and documents are concerned. Several tools exist that correctly handle its format. Major databases are already converted into this format and are regularly used by public administrators and casual users alike.

Further evolutions of the current schema will without doubts include dealing with the actual drafting process and with further formalization of the semantic aspects of the documents. Thus the two main evolutions that can be foreseen in the future are:

- "Disegni di legge" (bills) and the logging and description of the complex process leading from the initial draft to the enactment of an act.

- Ontologies, or the expression of the metadata section of NormeinRete documents according to a formalized ontology language such as OWL.

2.5.3 Complete language technical description

2.5.3.1 Overall design principles

Schema organization The most evident aspect of the NormeInRete DTD is the parallel support of the documents that follow the drafting rules expressed in the "circolare" of 20th April 2001, and of the documents that, having been written earlier or by institutions that are not bound to it, may present some differences [Megale and Vitali, 2001].

The drafting rules contained in the above-mentioned "circolare" express constraints about the order and names of the hierarchical parts, as well as regarding their titling and numbering. Norms have always followed most of these constraints, but some of them have been formalized recently, and not all documents are compliant to them.

In order to describe both kinds of documents, the working group produced two different schemas, called "Complete" (or "Strict") and "Flexible" (or "Loose"). The main characteristic of these schemas is their reciprocal compatibility: the Complete Schema does not describe structures different from the Flexible Schema, but only adds more constraints to the acceptable structures. Consequently, all the documents that are valid according to the Complete Schema are also valid according to the Flexible Schema; besides, all the documents that were drafted according to the rules of the "circolare" but were created earlier can still use the Complete Schema.

The organization of the schema is based on the complex interactions of 14 different files. Of these, seven contains entity definitions for individual characters and are taken directly from HTML. The remaining seven provide an interconnected organization that help Loose and Strict schemas. In fact, two files (`nirstrict` and `nirloose`) contain the content models and rules that make up the difference between the Complete and the Flexible schemas, while the remaining five contain the whole vocabulary and content models that is shared by the two schemas. They are organized as follows:

- `nirstrict`, `nirloose`: the content models that differ between the two schemas, as well as the definition of the main document types.
- `Tipi`: the definition of the simple and complex types used throughout the schemas.
- `Globali`: the definition of global attributes and generic elements used throughout the schemas.
- `Testo`: the definition of HTML-derived elements for styles, blocks, tables and forms as found in legislative documents and void of legislation-specific meaning.
- `Norme`: the exact and complete set of legislation-specific elements to be used because of their specific meaning.
- `Meta`: the set of metadata elements, as contained in the metadata sections in all NormeinRete documents.

Since it was felt that a 14-file structure was too difficult to be approachable by individuals needing to learn NormeInRete for the simplest of purposes, an additional schema was added, called “Simplified” or “Light”.

The Simplified Schema provides a simplification of the vocabulary and a subset of the available content models. It is guaranteed that all documents that are valid against the Simplified Schema are also valid against the Flexible schema, but the Simplified schema is meant to be used for simpler document sets (e.g. local acts and regional documents) that do not need the full flexibility and vocabulary made available through the two main schemas. On the other hand, oftentimes such simpler documents do not follow strictly the rules imposed by the “circolare”, and for this reason a few stricter rules expressed in the Complete schema are not enforced in the Simplified.

Schema design approaches Design patterns are distillation of common wisdom in organizing the parts and the constraints of a schema. NormeinRete uses some well-known patterns:

- *Universal Root*: one generic root element contains all elements describing the document types. The NormeinRete schema covers seventeen different document types covering national, regional norms and open documents. The naïve solution would be to create seventeen different schemas, one for each document type. This would mean activating seventeen different processes for editing, marking up, validating, converting and displaying these documents, all of which driven by information contained within the document itself, and unavailable in advance. By creating a single root element containing the actual document elements, on the other hand, we have a single schema describing correctly and completely all document types.

- *Consistent Element Set*: many elements share similar or identical internal structures (in XML lingo, they have the same content model). As mentioned in the previous section, all elements rely on a very limited set of types of content models, and all document elements share just two content models (a strictly hierarchical structure and a loosely hierarchical structure). The advantage of this approach is that the proliferation of element names does not detract from the overall clarity and simplicity of the overall schema.

- *Generic document & role attribute*: besides named elements, the NormeinRete schema also provides for a generic element for each of the five main types of content model. These are elements meant to be used for markup that fits the content model but for which no specific element was provided. A naming attribute is provided for explicitly naming the element. A similar approach is taken in XHTML with the `<div>` and `<span>` elements, which are generic elements qualified by the "class" attribute. It is also a characteristic of the NormeinRete schema that all named elements are equivalent, and can be substituted to the corresponding generic element with their name as the value of the name attribute. Thus, by definition, `<p>` is in NormeinRete the same as `<blocco name="p">`, etc.

- *Reuse Document Types* (partial): many NormeinRete element types mean exactly the same as corresponding elements in XHTML. For this reason, rather than inventing new names that need to be learnt from scratch, correctly understood and actually used when marking documents, only to have them translated into the corresponding XHTML terms for display, we have decided to just use the XHTML name, saving time in explanation, documentation, learning and usage. Therefore, for instance, a table in NormeinRete uses the tags `<table>`, `<tr>`, `<td>` and `<th>` that were introduced in HTML, a paragraph is `<p>`, bold and italic are simply `<b>` and `<i>` respectively.

2.5.3.2 Representation of document structure

Partition structure All NormeInRete documents share the same root element `<NIR>`, under which the specific document type is selected. The single root element follows a specific design pattern "Universal root" aimed at better identification of the root and separation of namespace and schema declaration (available in the root) and meaningful attributes (available in the document type element).

There are currently seventeen document types, organized in three categories: national norms, other norms, generic documents. These are further structured as to follow two different structure organization: the `docarticolato` (strongly hierarchical) and the `semiarticolato` (loosely hierarchical).

Both structures allow a hierarchy of partitions: within the stricter set of rules of he strongly hierarchical document, we can find a eight-level partition: book, part, title, heading, section, paragraph, article and clause: within clauses we can find block elements or lists. Loosely hierarchical structures, on the other hand, beside the eight-level partition allow a flat sequence of paragraphs and a set of generic constructs for non-standard hierarchies.

Granularity of mark up The markup in NormeInRete can describe all aspects of the document, from the main structures down to the smallest part of the clauses. Inline elements and markers (empty elements) can be added anywhere. Generic elements can be added to take care of any fragment or structure of text that needs to be marked up and for which no pre-defined element has been defined.

Quote part of text NormeInRete allows the specification of quoted texts anywhere where plain text is allowed. This means that in any place (but mostly within clauses) the element `<virgolette>`, that contains the quoted text, can be added. Furthermore, the role of such quoted text (amended text, removed text, newly inserted text, etc.) is not specified in the clause, but in the metadata section.

Table management A simplified HTML table model is used: table elements contain table row elements, which in turn contain table cells and table headers. A simplified list of attributes is also used, in order to avoid the over-specification of rendering options. Structural attributes, though, are fully available (e.g., rowspan and colspan).

Annexes Annexes are called `<annessi>` in Norme In Rete. All types of documents can have attachments. The attachment section of NormeinRete is a list of individual `<annesso>` elements, each of which is either an empty element containing a reference to the actual referenced document. Each annex is introduced by heading elements, such as title, name and prefix, that introduce the annex while not being part of it.

Other features: generic elements Norme In Rete strongly supports the idea of using semantically rich terms whenever a semantically justifiable text fragment exists in the document. Yet, it is still possible that users in daily work will find the need for more elements than currently provided.

Generic elements come to aid in this respect. Whenever a new semantic is needed to describe a text fragment, a generic element of the appropriate content model is used instead, and the correct label is specified in the name attribute.

2.5.3.3 Identification of documents and document parts

Identifiers are systematically used in NormeinRete. All NormeinRete elements allow an identifier. Many relevant elements and sections require it. Identifiers are the main way to identify fragments and parts of the document in an unambiguous form. They can be used in document references (e.g. links and amendment commands) as a precise pointer to the actual part of the document mentioned (as opposed to simply referring to a document as a whole). Also internal links need to use identifiers. The schema does not explicitly provide a syntax for identifiers, which is described here in human readable format.

URI definition for the whole document Norme In Rete uses unique and persistent identifiers to refer to normative documents. They rely on URN (Uniform Resource Names) [Spinosa, 2001], in order to be up-to-date with Internet current best practices, and to allow easy navigation among documents, creating a global hypertext regardless of the local storage of actual documents, yet capturing all relevant relations between documents. URN are associated to documents through a formal schema properly defined and representing:

- any document (be it of normative, administrative or jurisprudential nature);
- emanated by any kind of authority (national, international, regional or local);
- in any moment in time (past, present and future).

The adopted approach is compatible with the citation praxis, and it is based on meaningful aspects of the document, and represent formally:

- the emanating authority;
- the nature of the document;
- date and number, if present;
- the annex, if appropriate;
- the version, if appropriate;

The following are a few examples:

National act	`urn:nir:stato:legge:1997-05-15;127`
Regional act	`urn:nir:regione.lazio:legge:2001-03-15;27`
Decree of the President	`urn:nir:presidente.repubblica:decreto:2001-09-22;345`
Decree of ministry	`urn:nir:ministero.finanze:decreto:1999-12-21;537`
Attachment of a law	`urn:nir:stato:legge:1982-12-31;979:allegato.a;sostanze.nocive`

Version identification NormeinRete deals both with formally approved acts from formal bodies, and with versions, which are editorial works generating amended texts out of original texts and amendment instructions. Versions are not identified by a sequential number, but rather by the enactment date of the corresponding amended text. This date is placed at the end of the URN and it is connected to the main URN by a "@" character, as in the following example:

Updated text of a king's decree (originally enacted before 1948)	`urn:nir:stato:regio.decreto:1941-01-30;12@1998-02-19`

element	**id**
Clause 2 of article 12	art12-com2
Third section (sezione) of heading (capo) IV of second book (libro)	lib2-cap4-sez3
Number 5 of item c of article 142 ter	art142ter-let3-num5
Third paragraph of first annex	ann1-blc3

ID definition for document part, granularity and composition rules Identifiers are composed by juxtaposing subidentifiers of the path needed to access them. Legal documents provide explicit global numbering for sections and articles, and local numbering for hierarchical subparts of them. For instance, all parts in different sections are numbered starting each time from 1, so “part 1” is not sufficient to clearly identify the actual part, while “article 12” clearly points to a single and well-specified element.

Each subidentifier is composed of the three letters, plus a number identifying its position within the overall list of similarly named elements. Blocks are all named “blcXX”, regardless of their actual names, and inlines are all named “inlXX”, regardless of their actual names. An exception to this are references “rifXX” and amendments “modXX”.

The following is a table with some examples of identifiers: Identifiers *never change* even if and when the elements get officially renumbered. Insertions may add a substring at the end of the identifier. So if an amendment creates articles “12 bis” and “12 ter” between articles 12 and 13, then the relevant identifiers will be “art12bis” and “art12ter”.

Structures quoted within the `<virgolette>` elements add the identifier of the relevant mod before their “natural” identifiers. So for instance if clause 3 of article 15 has an amendment that adds article 4 bis to a different act, the identifier of the `<virgolette>` element that contains the new article will be “art15-com3-mod1-art4bis”. Of course, automatic systems that create current versions of texts will remove the prefix belonging to the amendment law and will only keep the “art4bis” identifier in the final result.

2.5.3.4 Knowledge management

The meta section contains all the meta-information that needs or can be added to the actual content of the document. As a rule, all editorial content (i.e. content added by the editorial process out of Parliament rooms) needs to be placed in the meta section, except for markup and note references. Vice versa, all actual content of the document needs to have a place outside of the meta section in the appropriate content sections.

Meta elements are divided in six subsections:

- **Descrittori** (*descriptors*): i.e., a set of meta information providing info about the document and its publication and edition details, including its official promulgation date, its official URI, and so on.

- **Inquadramento** (*framing*): the framing elements provide technical information meant to properly characterize and describe the role, status and relevance of the document with respect to the overall legislation.

- **Ciclodivita** (*lifecycle*): the lifecycle element provides information about the events that the document has undergone, and references to the documents that have caused these events.

- **Lavoripreparatori** (*drafting process*): the official set of documents and texts being created that accompany the official text and describe the drafting process.

- **Redazionale** (*editors' notes*): all notes and addition that the editors feel appropriate to add to comment, explain and frame the content of the document.

- **Proprietari**o (*proprietary metadata*): a place to store all other metadata that the editors or the applications used by specific editors may require to add to the document.

General meta-information The `<descrittori>` section contains a set of meta information providing info about the document and its publication and edition details, including its official promulgation date, its official URI, and so on.

Document classification A section for keywords is allowed in the descriptor section of the metadata.

Document preservation No document preservation information elements are currently in place but some will be soon.

References management Within the lifecycle section of the metadata, a full list of references are allowed to external documents that impact on the current document. References are better explained in the Paragraph "Events management".

Other metadata The section `<redazionale>` contains the text of the editorial notes that might be produced to comment and expand the actual text of the document. Note references inside the text point to notes contained here.

The section `<proprietary>` allows any additional metadata to be specified in any order and vocabulary (provided it uses a different namespace than NormeIn-Rete). Proprietary metadata can be used within a specific document management system to specify additional information useful for internal search and document management that is not worth standardizing and imposing across all NormeinRete implementations.

2.5.3.5 Time management

Efficacy, modification and in force date NormeinRete allows the specification of enactment and in force dates [Palmirani, 2005] on each individual fragment of document through the use of the global attributes `iniziovigore`, `finevigore`, `inizioef-ficacia` and `fineefficacia`. The status attribute further specifies the current status of the fragment. Furthermore, an `entratainvigore` element is present in the descriptor section of the metadata, to record the first enactment date of the document as a whole.

Events management NormeinRete includes a sophisticated mechanism to keep track of the life cycle and evolution in time of a legislative document. This is particularly useful for acts that are amended and modified in time, while maintaining their fundamental nature.

The management of evolution of a document makes two very important assumptions:

- Amendments and events in the life cycle of a document (including original approval, final repeal and any other event affecting its presence in the law system or its content) happen in precise moments in time that can be determined objectively (albeit possibly with difficulty) and attributed a specific date.
- Amendments and events in the life cycle are due to the enactment of a specific, individual document that can be objectively traced back and identified with an URI. If two different documents affect the same act on the same date, then these must be counted as two different and separate events on the amended act.

Handling events in NormeinRete centers around the `<ciclodivita>` element in the meta section. This contains two containers, `<eventi>` and `<relazioni>`, used to list the dates of all the events affecting a document, and the references to the URIs of all the documents generating these events. Each reference is provided with a required identifier, which is used by the event list to specify which document is responsible for which events. These elements must appear in all documents that have undergone two or more events (i.e., all acts except the ones that still have no amendments).

Documents in NormeinRete are organized in three main categories, as specified in the tipo attribute of the `<NIR>` element:

- **Originale** (*Original Version*): this value reflects the fact that the content on the document is exactly the content that has been formally and explicitly approved by the relevant authority, with no amendments applied.
- **Monovigente** (*Single Version*): this value reflects the fact that the content of the document is an editorially modified version of the original act, according to one or more subsequent amendment acts. These amendment acts and the enactment dates of the amendments must be all present in the `<ciclodivita>` element. Individual additions and deletions are not necessarily marked in the content.
- **Multivigente** (*multiple versions*): this value reflects the fact that the content of the document is the juxtaposition of fragments belonging to two or more different versions of the same act, each fragment marked as belonging to one or many of these versions. Thus in a Multiple Versions act there could be two or more copies of article 2, each associated to the date it started enactment and ended enactment.

The <ciclodivita> element is a required element for all Single Version and Multiple Version documents, and must be complete up to the enactment date of the latest document referenced in the <ciclodivita> element (i.e., there can potentially be subsequent amendments non included in a Single Version or Multi Version document, but all intermediate amendments must be correctly listed and referenced, even if they play no part to the displayed content). Original Version documents need not have the <ciclodivita> element, but surely can have it if the editors decide so.

In case a Multiple Versions document is being generated, each element and text fragment may be associated an enactment or in force specification through the means of the four attributes: iniziovigore, finevigore, inizioefficacia and fineefficacia, end. Each fragment (a whole element if appropriate, otherwise a newly inserted <span> or <inline> element if no exact containing element exists) use these attribute to specify their nature.

The start and end attributes contain an IDREF to the ID of the event that has marked the beginning or the end of the enactment of the fragment. A start attribute with no end attribute marks a fragment that has appeared in an amendment and still exists in the latest recorded version of the document. An end attribute with no start attribute mark a fragment that was part of the original document but has been repealed before or at the latest recorded version of the document. The status attribute records the type of amendment of the fragment. The value "omissis" can only be used by private editors that want to display only part of the whole document. In this case, the structure must be complete anyway, but the actual content can be removed is the status="omissis" attribute is present.

2.5.3.6 Support for visualization and text formatting

NormeinRete uses a number of HTML elements for text fragments whose purpose is mainly presentation-oriented. These include paragraphs, lists, images, tables, and so on.

Only a strict subset of the HTML language has been chosen, and no additional element should be added. In particular, headings (<H1>, <H2> and so forth) cannot be used in NormeinRete document, since they enforce a flat organization of sections, which is against the fundamentally hierarchical nature of NormeinRete documents. All HTML elements have exactly the same nature and role as they have in HTML documents.

All HTML elements (and, in fact, all NormeinRete elements as well) can be optionally enriched with standard HTML core attributes allowing CSS styles with precise presentation instructions to be associated to them. The class and style attributes can be used as in HTML for external or internal CSS rules, liberally and without limitations on both HTML and NormeinRete elements.

2.5.3.7 Native support or extensibility for integration with other kind of knowledge representation standards

NormeinRete provides a rich set of provisions to describe and characterize the meaning of most fragments of a legislative document [Biagioli, 1997]. In fact, there exist 61 elements in NormeinRete that describe semantically the most common types of provisions that are present in acts. These provision elements are organized in five main categories:

- **Motivazioni** (*motivations*) describing the types of the statements that are placed at the beginning of decrees and other types of legislative documents (except acts) that introduce and justify the need and the opportunity of the document.
- **Regole** (*rules*) describing the set of normative rules that are present within a document and their arguments.
- **Modifiche attive** (*active amendments*) describing the set of amendments that are expressed by the document, and their effects on amended documents.
- **Modifiche passive** (*passive amendments*) describing the set of amendments that are expressed elsewhere and affect the document.
- **Comunicazioni** (*communications*) describing further notices that affect the interpretation and power of an act without being a formal amendment.

Each provision element includes a number of attributes and sub-elements that are used to express and detail the parameters of the provision itself. Of particular importance is the time-related element `<dsp:termine>`, that allows the specification of the interval in time in which the provision is in effect. Also, provisions are connected to the text they refer to by means of a `<dsp:pos>` element that either contains the text or points to it via an XPointer.

Ontology, semantics, rules The set of metadata elements connected to a legislative document in NormeinRete is immediately convertible into an RDF document instance. In fact, it is foreseen that a future development of the metadata section of a NormeinRete document becomes an externalized fragment fully expressed in RDF.

Similarly, to this date there is no formal ontology (in OWL or any other ontology language) of the metadata elements allowed in a NormeinRete document. Nonetheless, such ontology is immediately derivable from a simple analysis of the metadata section of the NormeinRete DTD, and is planned to be available soon.

2.6 AKOMA NTOSO

2.6.1 Introduction

In 2004 and 2005, the UNITED NATIONS Department for Economic and Social Affairs (UN/DESA) project “Strengthening Parliaments’ Information Systems in Africa” has aimed at empowering legislatures to better fulfill their democratic functions, using ICTs to increase the quality of parliamentary services, facilitate the work of parliamentarians and create new ways to promote the access of civil society to parliamentary processes. In the first stage, the project has been actively supported by the NormeinRete community, and has been influenced by NormeinRete project results.

A strategic role in this project is played by the AKOMA NTOSO[2] (Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies) framework, a set of guidelines for e-Parliament services in a Pan-African context. The framework addresses information content and recommends technical policies and specifications for building and connecting Parliament information systems across Africa. In particular, the AKOMA NTOSO framework proposes an XML document schema providing sophisticated description possibilities for several Parliamentary document types (including bills, acts and parliamentary records, etc.), therefore fostering easier implementation of Parliamentary Information systems and interoperability across African Parliaments, ultimately allowing open access to Parliamentary information.

AKOMA NTOSO was developed as a necessary foundation for the development of a comprehensive Parliamentary Information System (PIS). The goal of the Parliamentary Information System is to maximize the operational efficiency and effectiveness of National Legislatures by implementing a solution which provides secure, reliable, and timely collection, storage, access, and transmission of information. The aim is to equip Parliaments with a solution that fosters accessibility, transparency and accountability of Parliaments by exploiting open source multi-platform applications based on open standards and available in multiple human languages.

Individual country Parliaments are meant to use the guidance provided by the AKOMA NTOSO Framework to supplement their national e-Government initiatives with a Pan-African dimension and thus enable Pan-African interoperability of Parliaments. Thus the AKOMA NTOSO framework is meant to supplement, rather than replace, national interoperability guidance that may exist, and to add a pan-African dimension to them.

The AKOMA NTOSO Framework reaches three main objectives:

- to define a common standard for data interchange between parliaments;
- to define the specifications for a base document model on which parliamentary systems can be built;
- to define an easy mechanism for citation and cross referencing of data between parliaments.

The AKOMA NTOSO framework aims at providing two basic types of interoperability: semantic interoperability is concerned with ensuring that the precise meaning of exchanged information is understandable by any person or application receiving the data; technical interoperability is aimed at ensuring that all AKOMA NTOSO-related applications, systems, interfaces are based on a shared core of technologies, languages and technical assumptions easing data interchange, data access and reuse of acquired competencies and tools. AKOMA NTOSO ensures technical interoperability by enforcing the use of open standards and open document formats.

[2] The “Akoma Ntoso” (“linked hearts” in English) symbol is used by the Akan people of West Africa to represent understanding and agreement

2.6.2 Main features

The current version of the standard describes two different but connected families of schemas:

- **AKOMA NTOSO General Schema**: a vocabulary and minimal set of constraints that all AKOMA NTOSO documents must comply to.

- **AKOMA NTOSO Detailed Schemas**: a set of stricter schemas. They provide more constraints over the same vocabulary of elements to enforce the rules of specific document types in specific African Parliaments. It is a requirement of AKOMA NTOSO that all documents satisfying one of the Detailed Schemas also satisfy the General Schema.

So far, only the general schema is complete and operative. Detailed schemas are expected to be delivered when individual countries join the project and determine the required modifications to be brought to the general schema.

2.6.2.1 Strengths

AKOMA NTOSO provides rich and full support for the semantic description of a large set of elements in legislative documents. The set of allowed elements is extensive and complete, although the complexity of their use is fairly reduced by the systematic use of content patterns, that simplify the comprehension of the structures and the generation of software tools. A full support for documents evolving in time is also a feature that needs to be stressed among the advantages of the language.

2.6.2.2 Weaknesses

By design, AKOMA NTOSO provides a very limited set of constraints on the allowed values in legislative documents. This implies that a large number of “badly formed” documents can in theory be accepted by the AKOMA NTOSO language. On the other hand, the assumption behind the descriptive role of the language justifies this by saying that, if such a badly formed act is approved by a Parliament, it is not the job of an XML DTD or schema to reject such document; on the other hand, if no such badly formed act is ever found out, then this potentiality of danger does not eventually produce a problem and thus should be ignored as a weakness.

2.6.2.3 Open challenges

Arguably, AKOMA NTOSO is still in an initial stage. The set of metadata features is not complete yet, there is very limited support for provisions, and only limited support for the thesauri that could be used for fruitful search and classification of documents. We expect future versions of the standard to address these points.

2.6.3 Complete language technical description

2.6.3.1 Overall design principles

The AKOMA NTOSO model deals with a rather complex situation of five document types and several African countries, by creating two classes of document types, the Generic Schema (GS) and several Detailed Schemas (DSs) that provide support for differences in document types. Interoperability across these schemas is granted by a generalized approach that maintains full descriptions of the element while unifying and limiting in scopes the structures.

This is obtained through the systematic use of patterns. Patterns are the abstraction and distillation of past experiences in designing and resolving design problems. They are general and widely applicable guidelines for approaching and justifying design issues that often occur in XML-based projects.

We distinguish between patterns in content models (a restriction of content models to the ones that are actually useful) and patterns in schema design (guidelines on how to make a schema more modular, flexible and understandable by users). Both types of patterns are well known and well established in the literature, although by different experts in different ways. For patterns in content models we rely on [Vitali et al., 2005], while the authoritative resource for patterns in schema design is `http://www.xmlpatterns.com/`.

Patterns in content model The AKOMA NTOSO 1.0 Schema uses systematically five of the seven patterns described in [7]. This means that all content models and complex types used in the schema follow precisely the form of the relevant pattern, and all elements can be simply described and treated according to their pattern rather than individually.

These patterns are:

- the *hierarchy*: a hierarchy is a set of arbitrarily deep nested sections with title and numbering. Each level of the nesting can contain either more nested sections or blocks. No text is allowed directly inside the hierarchy, but only within the appropriate block element (or, of course, titles and numbering).
- The *blocks*: a block is a container of text or structures that is organized vertically on the display (i.e., has paragraph breaks) and can contain either substructures or text. Most blocks in AKOMA NTOSO are based on the HTML language.
- The *inlines*: an inline element is an element placed within a mixed model element that identifies some text fragments as relevant for some reason. There are both semantically relevant inlines and presentation oriented inlines. There is but one content model using inlines (and markers), which means that all mixed model elements (i.e., those that allow both text and elements) also allow the a repeatable selection of all inline elements.
- The *markers*: markers are content-less elements that are scattered here and there in the document and are meaningful for their names as well as their attributes. Markers are also known in literature as empty elements or milestones.

There are two main families of markers in the AKOMA NTOSO schema: placeholders in the text content (e.g., note references) that can appear in any position that also has text, and metadata elements that only appear in some subsection of the `<meta>` section.

- The *containers*: containers are sequences of specific elements, some of which can be optional. Containers are all different from each other (as the actual list of contained elements vary), and so there is no single container content model, but rather a number of content models that share the record pattern.

The following is an example of a hierarchy of sections:

```
<clauses>
    <chapter id="chap2">
        <num>Chapter 2</num>
        <title>Traditional communities and ...</title>
        <paragraph id="chap2-para2">
            <num>2</num>
            <title>Recognition of traditional ...</title>
            <clause id="cla1">
                <num>1</num>
                <p>A community may be recognised as ...</p>
            </clause>
            ...
        </paragraph>
        ...
    </chapter>
    ...
</clauses>
```

Patterns in schema design Design patterns are distillation of common wisdom in organizing the parts and the constraints of a schema. AKOMA NTOSO refers systematically to patterns listed in [9]. Whenever there has been a design choice to be made that was not immediately obvious and naturally acceptable, a relevant pattern has been sought and properly used. In fact, a large number of patterns from [9] have been used, but only a few of them need to be explicitly described and explained:

- *Universal Root*: one generic root element contains all elements describing the document types. The AKOMA NTOSO schema covers five different document types: acts, bills, parliamentary debate records, parliamentary order papers, and miscellaneous parliamentary documents. The naïve solution would be to create five different schemas, one for each document type. This would mean activating five different processes for editing, marking up, validating, converting and displaying these documents, all of which driven by information contained within the document itself, and unavailable in advance. By creating a single root element containing the actual document elements, on the other hand, we have a single schema describing correctly and completely all document types.

- *Consistent Element Set*: many elements share similar or identical internal structures (in XML lingo, they have the same content model). As mentioned in the previous section, all elements rely on five types of content models only, and all document elements share just two content models (a strictly hierarchical structure and a loosely hierarchical structure). The advantage of this approach is that the proliferation of element names does not detract from the overall clarity and simplicity of the overall schema.
- *Generic document & role attribute*: Besides named elements, the AKOMA NTOSO schema also provides for a generic element for each of the five main types of content model. These are elements called just like the corresponding pattern (thus they are called `<hierarchy>`, `<block>`, `<inline>`, `<marker>` and `<container>`), and are meant to be used for markup that fits the content model but for which no specific element was provided. A `"name"` attribute is provided for explicitly naming the element. A similar approach is taken in XHTML with the `<div>` and `<span>` elements, which are generic elements qualified by the `"class"` attribute. It is also a characteristic of the AKOMA NTOSO schema that all named elements are equivalent, and can be substituted to the corresponding generic element with their name as the value of the name attribute. Thus, by definition, `<p>` is in AKOMA NTOSO the same as `<block name="p">`, `<part>` is the same as `<hcontainer name="part">`, `<act>` is the same as `<container name="act">`, etc.
- *Reuse Document Types* (partial): many AKOMA NTOSO element types mean exactly the same as corresponding elements in XHTML. For this reason, rather than inventing new names that need to be learnt from scratch, correctly understood and actually used when marking documents, only to have them translated into the corresponding XHTML terms for display, we have decided to just use the XHTML name, saving time in explanation, documentation, learning and usage. Therefore, for instance, a table in AKOMA NTOSO uses the tags `<table>`, `<tr>`, `<td>` and `<th>` that were introduced in HTML, a paragraph is `<p>`, bold and italic are simply `<b>` and `<i>` respectively.

2.6.3.2 Representation of document structure

Partition structure All AKOMA NTOSO documents share the same root element `<akomantoso>`, under which the specific document type is selected. The single root element follows a specific design pattern “Universal root” aimed at better identification of the root and separation of namespace and schema declaration (available in the root) and meaningful attributes (available in the document type element). There are currently five document types: acts, bills, reports, minutes, and generic documents. These correspond to three different structures: the hierarchical structure, the debate structure, and the open structure.

All structures allow a hierarchy of partitions:

- the hierarchical structure (used for acts and bills) has a six-level partition: section, part, paragraph, chapter, article and clause: within clause you can find block elements or lists.

- The debate structure (used for minutes) has a multiple level hierarchy of generic structures called subdivisions. At the leaf level, subdivisions can contain speeches and questions.

- The open structure can finally contain both the debate hierarchy and the act hierarchy, as well as generic block elements and other containers.

Granularity of mark up The markup in AKOMA NTOSO can describe any aspect of the document, from the main structures down to the smallest part of the clauses. Inline elements and markers (empty elements) can be added anywhere. Generic elements (one for each pattern of content model) can be added to take care of any fragment or structure of text that needs to be marked up and for which no pre-defined element has been defined).

Quote part of text Similarly to NormeInRete, AKOMA NTOSO allow the specification of quoted texts anywhere where plain text is allowed. This means that in any place (but mostly within clauses) the element QuotedText can be added. Furthermore, the role of the quoted text (amended text, removed text, newly inserted text, etc.) is not specified in the clause, but in the metadata section. This is again similar to what happens in NormeInRete.

Table management A simplified HTML table model is used: table elements contain table row elements, which in turn contain table cells and table headers. A simplified list of attributes is also used, in order to avoid the over-specification of rendering options. Structural attributes, though, are fully available (e.g., rowspan and colspan).

Annexes Annexes are called attachments in AKOMA NTOSO. Al types of documents can have attachments: acts and bills, debates and generic documents. The attachment section of AKOMA NTOSO is a list of individual ¡attachment¿ elements, each of which is an empty element containing a reference to the actual referenced document.

Thus in AKOMA NTOSO all attachments are external documents that are explicitly referenced in the main document.

Other features: generic elements AKOMA NTOSO strongly supports the idea of using semantically rich terms whenever a semantically justifiable text fragment exists in the document. This means that it is possible that users of AKOMA NTOSO in daily work will find the need for more elements than currently provided.

Generic elements come to aid in this respect. Whenever a new semantic is needed to describe a text fragment, a generic element of the appropriate content model is used instead, and the correct label is specified in the name attribute.

It is strongly discouraged to use presentation-oriented elements (such as b, i, etc.) elements to emphasize fragments that do have a semantic justification for being emphasized. Also, each text fragment need to be enclosed within the appropriate generic element according to its position and content model, which is the reason for there being five generic elements (one for each content model pattern).

Finally, an explicit equivalence is provided between named elements and generic elements: all named elements are just generic elements in disguise, the value of the name attribute having been upgraded to being the full element name. Therefore, for instance, `<section>` is absolutely equivalent to `<hcontainer name="section">`, or `<noteref>` is equivalent to `<marker name="noteref">`.

This is turn means that it is possible to reverse the approach, and, after a revision process, officially enrich the AKOMA NTOSO language with new elements that have been used in the past as values for the name attribute of generic elements.

2.6.3.3 Identification of documents and document parts

Identifiers are systematically used in AKOMA NTOSO. All AKOMA NTOSO elements allow an identifier. Many relevant elements and sections require it. Identifiers are the main way to identify fragments and parts of the document in an unambiguous form. They can be used in document references (e.g. links and amendment commands) as a precise pointer to the actual part of the document mentioned (as opposed to simply referring to a document as a whole). Also internal links need to use identifiers. The schema does not explicitly provide a syntax for identifiers, which is described here in human readable format.

URI definition for the whole document All AKOMA NTOSO resources are identified by a unique name. Although the naming convention of AKOMA NTOSO is not fixed or final, at the current stage of evolution the naming mechanism for whole documents is as follows. All names consist of pieces separated by slashes: a names starts with a domain name designating both the originating country and the authoritative source, followed the name of the type of the document, followed by an option issuing authority, followed by a dating/numbering mechanism, followed by the issuing authority (if relevant) and a language coding information.

URI	**Description**
`www.sierra-leone.org/act/2-2004/en`	Sierra Leone enacted Legislation. Act number 2 of 2004. English version.
`http://www.parliament.gov.na/bill/19-2003/1/en`	Namibia Bill number 19 of 2003, first stage, English version
`http://www.parliament.gov.za/bill/12-2005/2/MinistryOfAgriculture/en`	South Africa. Bill 12 of 2005 (second stage), issued by the Ministry Of Agriculture. English version.

Version identification At the moment AKOMA NTOSO only deals with formally approved acts from formal bodies, so it does not deal with versione, which are editorial works on amended texts. Thus no version identification is currently provided.

ID definition for document part, granularity and composition rules Identifiers are composed by juxtaposing subidentifiers of the path needed to access them. Legal documents provide explicit global numbering for sections and articles, and local numbering for hierarchical subparts of them. For instance, all parts in different sections are numbered starting each time from 1, so “part 1” is not sufficient to clearly identify the actual part, while “article 12” clearly points to a single and well-specified element.

Each subidentifier is composed of the three letters, plus a number identifying its position within the overall list of similarly named elements. Blocks are all named “blkXX”, regardless of their actual names, and inlines are all named “inlXX”, regardless of their actual names. An exception to this are references “refXX” and amendments “modXX”.

The following is a table with some examples of identifiers:

Element	**Identifie**	**Example**	**Identifier of example**
<section>	secXX	Section 2 of this act	sec02
<part>	prtXX	Part 1 of section 2 of this act	sec02-prt01
<paragraph>	parXX	Paragraph 3 of part 1 of section 2 of this act	sec02-prt01-par03
<chapter>	chpXX	Chapter 5 of paragraph 3 of part 1 of section 2 of this act	sec02-prt01-par03-chp05
<article>	artXX	Article 12	art12
<clause>	claXX	Clause 3 of article 12	art12-cla03
<li>	itmXX	item ”c” of clause 3 of article 12	art12-cla03-itm03
<p>	blkXX	Third paragraph of clause 3 of article 12	art12-cla03-blk03

Identifiers **never change** even if and when the elements get officially renumbered. Insertions may add a character at the end of the identifier. So if an amendment creates an article ”12/a” or ”12 bis” between articles 12 and 13, then the relevant identifies will be ”art12a” in both cases.

Structures within the <quotedStructure> elements add the relevant mod identifier before their “natural” identifiers. So for instance if clause 3 of article 15 has an amendment that adds article 4/a to a different act, the identifier of the <quotedStructure> element that contains the new article will be “art15-cla03-mod01-art04a”. Of course, automatic systems that create current versions of texts will remove the prefix belonging to the amendment law and will only keep the “art04a” identifier in the final result.

2.6.3.4 Knowledge management

The meta section contains all the meta-information that needs or can be added to the actual content of the document. As a rule, all editorial content (i.e. content added by the editorial process out of Parliament rooms) need to be placed in the meta section, except for markup and note references. Vice versa, all actual content of the document needs to have a place outside of the meta section in the appropriate content sections.

The current version of AKOMA NTOSO only contains a limited number of metadata elements. A larger number is expected to be provided as soon as the global architecture for the document exchange becomes clearer.

Meta elements are divided in four subsections:

- **Descriptors**: i.e., a set of meta information providing info about the document and its publication and edition details, including its official promulgation date, its official URI, and so on.

- **Lifecycle**: the lifecycle element provides information about the events that the document has undergone, and references to the documents that have caused these events.

The development of the meta section is not finished yet. For instance, support for Dublin Core metadata is currently imperfect (there are semantic equivalences between Dublin Core elements and AKOMA NTOSO elements, but they are not complete nor officially described as equivalent).

General meta-information The `<descriptor>` section contains a set of meta information providing info about the document and its publication and edition details, including its official promulgation date, its official URI, and so on.

Document classification A section for keywords is allowed in the descriptor section of the metadata.

Document preservation No document preservation information elements are currently foreseen.

References management Within the lifecycle section of the metadata, a full list of references are allowed to external documents that impact on the current document. References are better explained in the following Paragraph *Events management.*

Other metadata The section `<notes>` contains the text of the editorial notes that might be produced to comment and expand the actual text of the document. Note references inside the text point to notes contained here.

The section `<proprietary>` allows any additional metadata to be specified in any order and vocabulary (provided it uses a different namespace than AKOMA-NTOSO). Proprietary metadata can be used within a specific document management system to specify additional information useful for internal search and document management that is not worth standardizing and imposing across all AKOMA NTOSO implementations.

2.6.3.5 Time management

Efficacy, modification and in force date AKOMA NTOSO allow the specification of enactment dates on each individual fragment of document through the use of the global attributes start, end and status. Furthermore, an `<enactmentDate>` element is present in the descriptor section of the metadata, to record the first enactment date of the document as a whole.

Events management AKOMA NTOSO 1.0 includes a sophisticated mechanism to keep track of the life cycle and evolution in time of a legislative document. This is particularly useful for acts that are amended and modified in time, while maintaining their fundamental nature.

The management of evolution of a document makes two very important assumptions:

- Amendments and events in the life cycle of a document (including original approval, final repeal and any other event affecting its presence in the law system or its content) happen in precise moments in time that can be determined objectively (albeit with difficulty) and attributed a specific date.

- Amendments and events in the life cycle are due to the enactment of a specific, individual document that can be objectively traced back and identified with an URI. If two different documents affect the same act on the same date, then these must be counted as two different and separate events on the amended act.

Handling events in AKOMA NTOSO centers around the `<lifecycle>` element in the meta section. This contains two containers, `<events>` and `<references>`, used to list the dates of all the events affecting a document, and the references to the URIs of all the documents generating these events. Each reference is provided with a required identifier, which is used by the event list to specify which document is responsible for which events. These elements must appear in all documents that have undergone two or more events (i.e., all acts except the ones that still have no amendments).

Documents in AKOMA NTOSO are organized in three main categories, as specified in the contain attribute of the document type element:

- **OriginalVersion**: this value reflects the fact that the content on the document is exactly the content that has been formally and explicitly approved by the relevant authority, with no amendments applied.

- **SingleVersion**: this value reflects the fact that the content of the document is an editorially modified version of the original act, according to one or more subsequent amendment acts. These amendment acts and the enactment dates of the amendments **must** be all present in the `<lifecycle>` element. Individual additions and deletions are not necessarily marked in the content.

- **MultipleVersions**: this value reflects the fact that the content of the document is the juxtaposition of fragments belonging to two or more different versions of the same act, each fragment marked as belonging to one or many of these versions. Thus in a MultipleVersions act there could be two or more copies of article 2, each associated to the date it started enactment and ended enactment.

The `<lifecycle>` element is a required element for all SingleVersion and MultipleVersion documents, and must be complete up to the enactment date of the latest document referenced in the `<lifecycle>` element (i.e., there can potentially be subsequent amendments non included in a SingleVersion or MultiVersion document, but all intermediate amendments must be correctly listed and referenced, even if they play no part to the displayed content). OriginalVersion documents need not have the `<lifecycle>` element, but surely can have it if the editors decide so.

In case a MultipleVersions document is being generated, each element and text fragment may be associated an enactment specification through the means of the three enactment attributes: start, end and status. Each fragment (a whole element if appropriate, otherwise a newly inserted `<span>` or `<inline>` element if no exact containing element exists) use these attribute to specify their nature.

The start and end attributes contain an IDREF to the ID of the event that has marked the beginning or the end of the enactment of the fragment. A start attribute with no end attribute marks a fragment that has appeared in an amendment and still exists in the latest recorded version of the document. An end attribute with no start attribute mark a fragment that was part of the original document but has been repealed before or at the latest recorded version of the document. The status attribute records the type of amendment of the fragment. The value `"omissis"` can only be used by private editors that want to display only part of the whole document. In this case, the structure must be complete anyway, but the actual content can be removed is the `status="omissis"` attribute is present.

2.6.3.6 Support for visualization and text formatting

AKOMA NTOSO uses a number of HTML elements for text fragments whose purpose is mainly presentation-oriented. These include paragraphs, lists, images, tables, and so on. Furthermore, as mentioned, even HTML elements have been made into the AKOMA-NTOSO namespace, so as to simplify the namespace management.

Only a strict subset of the HTML language has been chosen, and no additional element should be added. In particular, headings (`<H1>`, `<H2>` and so forth) cannot be used in AKOMA NTOSO document, since they enforce a flat organization of sections, which is against the fundamentally hierarchical nature of AKOMA NTOSO documents. All HTML elements have exactly the same nature and role as they have in HTML documents, with one exception: `<div>` is a generic container rather than a generic block as in HTML. This is due to the fact that a generic block already exist (`<p>`), and that in many automatically produced HTML documents (e.g., Open Office and MS Word), the `<div>` element is in fact used as a section separator (i.e., a container) rather than a paragraph.

All HTML elements (and, in fact, all AKOMA NTOSO elements as well) can be optionally enriched with standard HTML core attributes allowing CSS styles with precise presentation instructions to be associated to them. The class and style attributes can be used as in HTML for external or internal CSS rules, liberally and without limitations on both HTML and AKOMA NTOSO elements.

2.6.3.7 Native support or extensibility for integration with other kind of knowledge representation standards

There is a plan to introduce full support for the semantic description of provisions within acts in AKOMA NTOSO. These provisions will initially allow full description of the temporal evolution of amended documents, and thus will provide more information about the role that each amending provisions has on the amended text. At the moment, nonetheless, no formal provision mechanism is provided.

Ontology, semantics, rules It is forseen that the provision mechanism is expressed in OWL language. At the moment, no such mechanism is in place.

2.7 Other XML standards in legal domain

2.7.1 CHLexML

The activities on standardisation in legal domain, with particular attention to multilingual issues in the Swiss Confederation, are coordinated by COPIUR "Coordination Office for the Electronic Publication of Legal Data Federal Office of Justice" in Bern, Switzerland (`http://www.rechtsinformation.admin.ch/copiur/index.html`).

Copiur started its work in October 1998. Initially attached to the Federal Chancellery, it later joined the Service of "Legal Data Processing and Computer Law" (Rechtsinformatik und Informatikrecht/Informatique juridique, droit et informatique/Informatica giuridica, diritto dell'informatica) at the Federal Office of Justice. The Service initiates and leads informatics and organizational projects of national importance in areas like registers, electronic exchange of legal documents and electronic publication of legislative data. Copiur evaluates and promotes new information technologies in the legislative field. It deals principally with the elaboration of uniform norms, standards, and information structures. Its principal objective is to harmonize federal, cantonal as well as private sector publications, in order to give the public a rich, uniformly presented and simply accessible online-offer of legislative data. Copiur also represents the federal administration at the appropriate national and international coordination bodies.

COPIUR currently deals with two relevant projects.

The first one is CHLexML (`www.chlexml.ch`), or the establishment of a comprehensive XML schema to be used for the publication of all (federal, cantonal and even communal) legislative acts. The schema has been finalised by a working group of the Swiss association for juridical informatics led by Copiur and the Federal Chancellery. After an internal review, it will be adopted at the 5th seminar on law and informatics organised by Copiur on 30 June 1st July 2005. The schema CHLexML will then be submitted to cantonal chancelleries and other interested players for consultation and to the eCH (the E-Government standards setting association) for normalisation.

The second project is named LexGo (www.lexgo.ch). Its main purpose is to align the 27 classification systems (all different) used by the 27 (federal and cantonal) systematic collections of law in order to simplify and improve search results. Based on the common systematic for the classification of legal acts elaborated by the Institute of Federalism of Fribourg, LexGo has created 27 matrix tables (Konkordanztabellen, tableaux de concordance) aligning all cantonal and federal legislative acts to the common systematic. Thanks also to a database of accurate links, LexGo allows users to find, for example, all federal and cantonal norms related to a given subject.

2.7.2 EnAct

2.7.2.1 Introduction

EnAct is a legislation drafting, management and delivery system that has been built to enable the Tasmanian Government to provide improved legislation information services to the community [Arnold-Moore et al., 2002]. EnAct provides the community with a facility that enables cost effective public access to reliable, up-to-date, searchable consolidated Tasmanian legislation. The “point-in-time” capability allows users to search and browse the consolidated database as it was at any time since 1 February 1997. Tasmania achieved these goals by automating much of the legislative drafting and consolidation process.

2.7.2.2 Main feature

The EnAct repository makes use of the Structured Information Manager (SIM) to store SGML fragments and associated metadata directly. Acts are stored in SIM as fragments with a timestamp marking the time interval (start and end time) over which the fragment or table of contents is valid. The SIM repository incorporates an SGML parser allowing sophisticated indexing based on the logical structure of the SGML fragments. Time point searching is also supported. When a time point is specified, a filter is applied to the database leaving only those fragments and tables of content that were valid at the specified time. This allows a snapshot of the database to be searched and browsed as if it were the entire database. The SIM web server uses the same SGML parser. This allows HTML to be generated dynamically from the fragment repository for delivering Web pages. This means that the same repository can be viewed in a number of different ways. In the current public site one can view a fragment with or without history notes. A single fragment can be viewed on one side with a table of contents on the other, or with the table of contents and all of the fragments united together into a single document. All cross-references are activated as hypertext links. Because all hypertext links are activated using queries in the database, it is just as easy to do reverse hypertext linksi.e. show a list of all fragments that refer to this fragment or this Act. When history notes are displayed, those that refer to amending Acts on the system are also hypertext links to those Acts. The table of contents can be viewed either as a conventional section number and headnote or with a section number and list of links to successive versions of the corresponding fragments. This provides a quick overview of the history of a provision. A new feature is a previous and next version button for each fragment.

In addition to the public web site, which delivers HTML to user's browsers, the Printing Authority of Tasmania (PAT) also uses the same repository to generate "on-demand" authorized reprints. A member of the public can contact the PAT shop and request a reprint of a particular Act on a specified day. They search the repository to find the relevant table of contents and fragments at that time point, which are joined together into a single SGML document that is translated into Rich Text Format using SIM formatters. This RTF representation is then passed to high-speed printers for paper output. Each authorized version is preceded by a certificate of authenticity from the Chief Parliamentary Counsel.

At the end of the document two tables appear. The first shows the name and the time of commencement of all amending Acts applied to that Act since it was enacted. The second lists each provision that has been amended and how it was amended. The database that is shared by the public Web gateway and the private Printing Authority gateway is not the only repository in the EnAct system. A working database is also kept in the OPC. This database contains everything in the public repository, but also contains the politically sensitive draft Bills in preparation, and all of the workflow information about the status of Bills and other draft legislation. Data is periodically migrated from the production database to the publication database providing a level of protection from intrusion and preserving the integrity of the data on the public repository.

2.7.2.3 The drafting environment

In the past, any consolidation has been a laborious manual process. Drafters or clerical staff were required to handle the text of an Amending Act and go through it, section by section, applying each amendment to one or more Principal Acts. In the past drafters composed amending legislation directly, combining the process of deriving the effect of the amendments with the text describing them. The Tasmanian government presented a different approach. The drafters would mark amendments directly on a consolidation of the Principal using strike-through and underline markings familiar to many lawyers. Amendment wordings for those markings would then be generated automatically.

These markings are then captured in an internal (SGML) representation of the changes called a Change Description Document (CDD). These changes are then used to generate amendment wordings, which are appended to a stub or substantive Bill [Arnold-Moore, 1997]. This process is managed by a workflow enactment service that keeps the CDD and generated amendments together so that, if and when the amendment Act commences, the amendments can then be automatically applied to the principle to generate new fragments in the historical repository.

2.7.3 Legal RDF

2.7.3.1 Introduction

Legal RDF is a non-profit organization that is sponsored by legal firms, software companies, and other stakeholders interested in software tools that leverage the Semantic Web.

Today within the legal community, the Internet has had its greatest impact improving the process of rendering legal services by government and legal firms. The Internet provides e-mail for faster and more certain communications; electronic document filing for faster and more certain document submissions; and websites for disseminating information concerning one's legal services, one's legal and administrative staff, and one's publicly-available documents.

To date, the content of legal documents has been represented as either a simple stream of text, or as a non-interpretable PDF image. The next, inevitable, step is to identify the type and meaning of the content in legal documents, thereby exposing this information to the numerous reasoning tools emerging from the Semantic Web community. Advancing in this direction, the legal community will lower its costs; improve the quality of its services; and create an environment conducive to mass-customization of legal products.

The Semantic Web is a disruptive technology in this sense for the legal industry. Firms can grow significantly by fielding products that cater to the needs of clients who ordinarily would not pursue legal advice. Reasoning-based software offers opportunities to provide these clients a level of service with an acceptable level of risk.

Therefore the strategy of the Legal-RDF community is to construct two databases – a comprehensive open-source ontology that is then applied in structured descriptions of statutory and administrative codes. These databases are then leveraged by the Semantic Web community to create the reasoning software envisioned for orderly industry growth.

Specifically underlying this strategy is the cognition that functional requirements that apply to many legal documents (e.g., to contracts and wills) are nearly identical to those in legal statutes. Benefits will powerfully ripple through the entire community if relevant software is applicable equally to both domains. Consequently, development of a contracts-related ontology, separate from or preceding a statutes-related ontology, is undesirable from economic, social, and legal perspectives.

2.7.3.2 Brief description of the language main features

Legal-RDF proposes to use a qualified XHTML standard to annotate legal documents [McClure, 2006] by defining a global attribute that can be embedded onto any XHTML element - such as the `<span>` element - during annotation of a block of narrative text, or phrase, word, or numeric text, occurring in the XHTML document. According to Legal-RDF this will reduce costs and difficulties that accrue when an organization replaces its current technologies.

An example of simple annotation within an XHTML document is:

```
<span about="somePublisher" x:property="PublisherTitle.1.eng"> THE
```

```
COUNCIL OF THE EUROPEAN UNION</span>
```

Moreover for the purpose of introducing reasoning facilities for legal documents, Legal-RDF proposes to adhere to the Resource Description Framework (RDF), so that deployment of reasoning wizard-bots to desktops of annotators and authors can be possible on a wide basis.

This is a possible translation in RDF of the previous example:

```
<rdf:Description about="somePublisher">
   <has>
       <PublisherTitle rdf:ID="PublisherTitle.1"
             eng="THE COUNCIL OF THE EUROPEAN UNION"/>
   </has>
</rdf:Description>
```

Legal-RDF defines a small set of predicates - just the tenses of the verbs *to-have* and *to-be* - that starkly contrasts with typical RDF predicate-naming practice, which is to concatenate a verb with a noun, for instance, *hasTitle*. Several advantages occur using the Legal-RDF approach of defining predicates that are verbs-only. Chief among these is that object-property naming practices adopted by international standards organizations (typically required to be concatenated nouns or qualified nouns only) can be accommodated without change. Secondarily, the Legal-RDF approach supports block/phrase/word name-selection menus, a key tool needed by annotators and authors of legal documents. Typical RDF naming practices (such as *hasTitle*) would be more difficult to accommodate with cascading menu implementations.

The following example shows that the previous one is composed of two RDF statements. Explicit statement specifications allow linguistic models to reflect the statement's context, voice, aspect, *et al* to be recorded and used in semantic analyses.

```
<rdf:Statement>
   <rdf:subject resource="somePublisher"/>
   <rdf:predicate resource="has"/>
   <rdf:object resource="PublisherTitle.1"/>
</rdf:Statement> <rdf:Statement>
   <rdf:subject>
       <PublisherTitle rdf:ID="PublisherTitle.1"/>
   </rdf:subject>
   <rdf:predicate resource="eng"/>
   <rdf:object>
       <rdf:Literal value="COUNCIL OF THE EUROPEAN UNION"/>
   </rdf:object>
</rdf:Statement>
```

The following example represents an optimization of these statements. Use of a `<verb>` property for a *Statement* that is distinct from the `<predicate>` property has eliminated the now-redundant *Statement* and *PublisherTitle* resources. The `<predicate>` property remains a pointer to an *ObjectProperty* instance, but one whose name is a noun phrase that concatenates the names of the classes that are referenced by the *domain* and *range* properties of the <predicate> property. Another type of noun-phrase typically represented by noun-oriented predicates, are those whose name concatenates an adjective, adverb, or participle with a noun, *e.g., FirstArticle*.

```
<rdf:Statement xml:id="PublisherTitle.1.eng">
   <lgl:model resource="someURI"/>
   <rdf:subject resource="somePublisher"/>
   <lgl:verb resource="has"/>
   <rdf:predicate resource="PublisherTitle"/>
   <rdf:object>
       <rdf:Literal value="COUNCIL OF THE EUROPEAN UNION"/>
   </rdf:object>
</rdf:Statement>
```

As can be seen, noun-oriented predicates contrast markedly with typical RDF predicate-naming practice (*e.g., hasTitle*). When predicate-verbs can be separately specified, verb tenses can be semantically exploited to easily express, for instance, that the publisher in in the previous example <had> a different title. Semantic distinctions between <has>, <had>, <willHave>, <mayHave> and <mustHave> can be easily developed, as well as negatory expressions such as <mayHaveNot>.

Legal-RDF's <model> has a mechanism for associating resources with named sub-graphs of nodes in a universal directed acyclic graph, a common feature in network schemas.

Annotation of narrative legal documents requires a number of vocabularies to be minimally available to semantic wizard-bots and to users during name-selection menu navigation. These vocabularies are each framed as a model that contains class- and property-name definitions.

In particular Legal-RDF identify the following fourteen (14) vocabularies: Semantics Model, Linguistics Model, Statistics Model, Economics Model, Politics Model, Aspects Model, Document Model, Actors Model, Roles Model, Scenes Model, Props Model, Themes Model, Dramas Model, Scripts Model.

For Semantics Model the RDF's Statement model is refined for the needs of annotating legal documents found in managerial, judicial, electoral, and personal contexts. Several legally-significant types of statements have been identified, for example Condition, Definition, Fact, Hypothesis, Negation, Permission, Prohibition, Requirement, and Warning, that are found within most legal documents.

The Linguistic Model is aimed to facilitate (and automate) the annotation process as much as possible, allowing to grammatically parse blocks of text to sentences, sentences to clauses, clauses to phrases, and phrases to individual words and other types of tokens.

The Statistics Model allows to annotate documents as regards (1) names of statistical quantities and calculated expressions including base types for percentages and rates (2) units of measurement applicable as a quantity or to a quantity and (3) numeric datatypes standardized by W3 XML Schema and used as the foundation for all other classes defined in the model.

The Economics Model defines an extensive vocabulary of economic measures including currency-based flow rates. These measures can be categorized into 18 types of currency amounts, from `CapitalAmount` and `ChargeAmount` through `WealthAmount` and `WithdrawalAmount`.

The Political Model describes the political and legal context(s) in which a document is to be interpreted.

The Document Model needs to adopt the grammatical paragraph model that is being introduced by the forthcoming XHTML, version 2.0. Among its many problems, the current XHTML version lacks the regular structure of document divisions, sub-divisions, sections, sub-sections, paragraphs, lines, and tables now being enforced by XHTML, version 2.0.

The Aspect Model is aimed at separating concrete classes from 'mix-in' classes. Mixin-classes identify the capabilities and the states that can be associated with new classes added-in by a user to a foundation of classes.

The Actors Model contains concrete classes for individuals and groups of persons, companies, governments, and organizations. Actors perform roles in the context of many performances, refining their behavior according to their context and experience and the scripts applicable to the performances. The Role is the general category for vocabulary terms that denote work-roles (occupations), legal roles, and family-life roles.

The Scene Model is the general category for locations (and eras), that is, time and place.

The Prop is the general category for products and for types of legal property. The Product hierarchy can be derived from the North American Industrial Product Classification. Property includes personal and real property.

The Themes Model allows the incorporation of topic maps that are applicable to a certain industry or activity. These topics all concern conceptual non-concrete classes. For instance, Justice and Injustice classes are subtypes of the Theme class, meaning that instances of 'Injustice' can be identified and described using the normal RDF mechanisms.

The Dramas Model provides the framework for describing discrete acts and open-ended activities that occur during a performance. This category is for definitions of Act and Activity classes as subclasses of the Event class.

The Scripts Model contains definitions of specific forms that are completed manually or by machine. A Script is a set of recorded statements. A Scripts model addresses the kind and content of statements that are to be present in instances of a completed form, are optional, and are disallowed.

2.7.4 eLaw

2.7.4.1 Introduction

The eLaw Project aims at a reform of legal text production, creating one continuous electronic production channel with a uniform layout prepared on the same electronic text basis from draft to publication (promulgation) on the Internet. The workflow system includes government bills, committee reports, legal enactments of the Nationalrat and decisions of the Bundesrat. The eLaw workflow system supports the electronic production of all committee reports, of legal enactments of the Nationalrat and decisions of the Bundesrat. Parliament returns consolidated electronic texts of legal enctments to the Federal Chancellery, ready for publication. The authentic electronic publication on the Internet (since 2004) is available for everybody free of charge.

The eLaw (e-Recht "Electronic Law") project aims at creating one continuous electronic production channel from the invitation to comment on draft legislation to promulgation (on the Internet). As a result, it is only required to enter amendments to the text during the legislative stages (for example by a committee, or in the plenary of the Nationalrat). As the first result of the project, texts of laws on paper are to be replaced by electronic texts, that is to say, printed government bills, committee reports and other parliamentary printed matter will cease to exist. Technology will make it possible to draw up texts which can be queried electronically while all stages can be tracked in a fully transparent process. Primarily for the purpose of cost-cutting, the texts of legislation were to be given a uniform layout and were to be prepared on the same electronic text basis from draft to publication in the Federal Law Gazette on the Internet. As a result, the Federal Chancellery sends government bills to Parliament, Parliament returns the consolidated electronic version of the legal enactment adopted by the Nationalrat once parliamentary procedures have been completed. The State Printing Office (Wiener Zeitung) is no longer necessary.

2.7.4.2 General background

To translate eLaw into reality, two projects were launched by the Administration of Parliament in view of the complex task on hand and the brief period available by decision of the federal government (trial operations were to start as early as on 1 September 2001):

- the "Implementing E-Law" project (in April 2001) to ensure one continuous electronic channel for the legislative procedure in the Nationalrat and the Bundesrat, as well as
- the "Roll-out Plan for laptops' to be used by Members of Parliament" (in December 2002).

2.7.4.3 Policy context and strategy

The re-design of the legislative procedure for the ministries was formally adopted by resolution of the Austrian Federal Government of 6 June 2001. The Conference of Presidents of the Nationalrat also advocated the implementation of the eLaw project in 2001. However, at the same time the Presidents called for better IT equipment for the Members of Parliament. The objectives to be met in the reform of the legislative process were defined as follows:

1. building up on existing databases
2. ensuring that the high quality requirements for parliamentary business will be fulfilled

3. taking into account the separation of powers between government and parliament
4. considering the principle of true costs (no passing on of costs or tasks from the government to parliament)
5. minimization of the total costs of parliamentary business
6. considering the special working conditions of parliament.

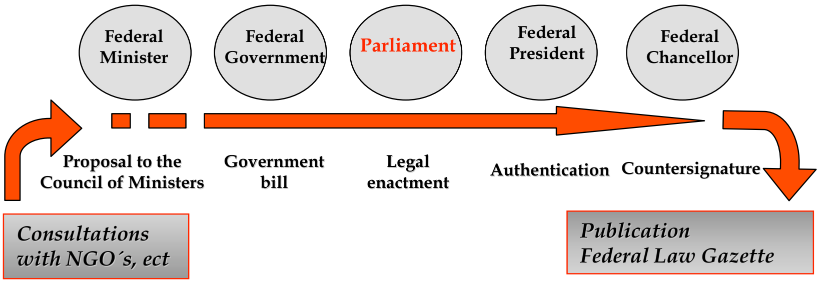

Figure 2.6: Austrian federal legislation (simplified organizational view)

2.7.4.4 Specific objectives

The basic ideas of the eLaw project are:

- To provide an electronic workflow for producing legal texts beginning with the draft bill and ending with the ePublication of the Federal Law Gazette (e.g. law, regulation, announcement, treaty)
- To replace printed legal texts by digitally signed electronic documents
- Official publication of the Austrian Federal Law Gazette in the Internet

On the basis of this main ideas, the eLaw project aims to obtain the following objectives:

- Publishing and archiving of legislation documents (draft bill, government bill) in the Austrian Legal Information System (RIS) - `http://www.ris.bka.gv.at/`
- Documents leaving workflow will be signed electronically on XML basis
- Easier administration of different versions of documents
- Implemention of a standard layout of the Federal Law Gazette
- Support for legislative bodies

The technical and organisational solution chosen is characterised by a high degree of user-friendliness as

- the Austrian parliament builds up on existing databases (that is to say, the new e-Legislation process was integrated into the database application "Parliamentary Business")
- an independent workflow has been established to account for the special features of parliamentary procedure and the separation of powers.

The exchange of documents with the federal administration, where a separate workflow is in place, is based on compatible formats and an independent interface, and detailed procedures have been developed for the exchange of data between parliament and government.

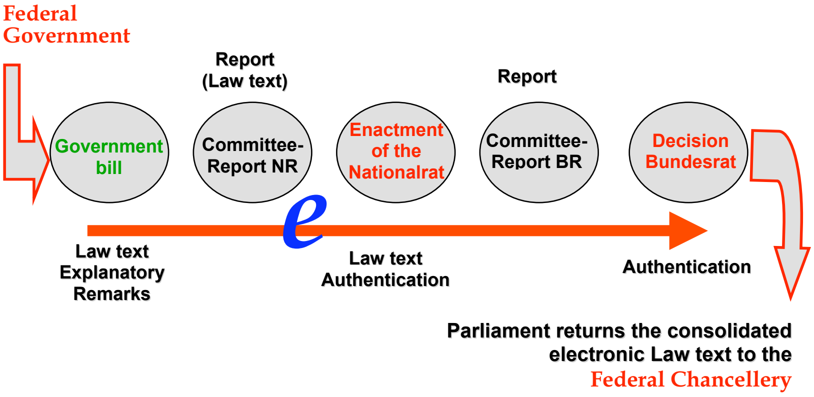

Figure 2.7: Parliament Electronic Workflow

The system offers the people involved in the legislative process not only electronic information but also participation opportunities; in this context, it has to be mentioned that a special upload mask was created for Members of Parliament and their assistants to enter the electronic versions of motions into the system. In a competence center established in Parliament, which has meanwhile successfully completed know-how transfer from the State Printing Office and taken up co-operation with the Federal Chancellery, staff supports the rapporteurs of the committees and the committee secretaries of the Parliamentary Administration in preparing the committee reports as well as the staff members of the Parliamentary Administration responsible for executing the legal enactments of the Nationalrat, and is in charge of quality management and the layout of legislative documents as well as covering the need for additional labour in peak times. The electronic exchange of data between government and parliament takes concrete shape as follows: the Federal Chancellery sends government bills to Parliament, Parliament returns the consolidated electronic version of any legal enactment adopted by the Nationalrat once parliamentary procedures have been completed.

2.7.4.5 Implementation main features

The project implementation is based on two main components:

1. Workflow procedures;

2. MS Word templates and macros.

Workflow The legislative process of a law is divided into different sub-processes:

- A draft of a bill is prepared by a Ministry
- The Ministry sends the draft bill for internal consultation (expert's opinion) to different interest groups (e.g. trade unions, chamber of commerce)
- The draft bill can also be put into the Austrian Legal Information System (RIS)
- Decision of the Council of Ministers (the weekly meeting of the Austrian federal ministers)
- The draft bill becomes a government bill which must be put into the RIS
- The government bill is transferred to Parliament which runs an independent system
- The government bill is debated by Parliament
- The decision of Parliament is transferred back to the Federal Chancellery
- Signing of the law by the Federal President and counter-signing by Federal Chancellor on paper
- Server based electronic signature by the Federal Chancellery
- Official publication of the Federal Law Gazette in RIS `http://ris1.bka.gv.at/authentic/index.aspx`

MS Word templates and macros In order to be exchange within the workflow process, documents are written in MS Word supported by macros. All documents are structured with respect to different legal categories (formats). The correct use of the special templates are necessary for the conversion of the documents to XML (65 templates for paragraphs and 11 templates for illustrations are used). The XML produced by this conversion is a presentation oriented standard, basically a qualified XHTML able to identify different parts of the documents, in order to associate them with specific presentation stylesheets. Some additional information are provided in terms of metadata, as the title, the date of publication, the number of the Gazette.

2.7.4.6 Brief description of language main features

In the way RIS is designed, categories are important for the production of documents (inputting documents into RIS representing a big document store). Search fields are important for queries (retrieval of documents from RIS, from the document store). The user, when making an online search, uses an enquiry screen with search fields. The legal categories are embedded in the document and visible only after having retrieved it. In most cases, a search field corresponds to a legal category, but does not necessarily match it precisely. However, matching is close enough to allow for concentrating on search fields which are easily visible. The following survey therefore lists the most important search fields for the RIS application "Federal Law". Other RIS applications utilise the same or similar search fields [Lachmayer and Hoffmann, 2005].

Suchworte (search words) The search field "Suchworte" was placed before the other search fields as practical experience has shown that many users base the search on search words. Searches in RIS are character-oriented, which means that the right sequence of characters is imperative. Some elasticity is achieved in that words may also be truncated (masked), i.e. by putting an asterisk at the beginning or end of the search word.

Kurztitel/Abkrzung (short title/abbreviation) The categories "Kurztitel" and "Abkürzung" were put together into a single search field, containing either all legal abbreviations and short titles as they appear in the text of the respective law, or - in those cases where this information is missing - a synthetic short title which is added during the time when the document is processed. It is not necessary to enter the full short title - some words from the short title separated by blanks are sufficient.

Paragraph (section) If the "Paragraph" (§) is known, it is entered in the corresponding search field. A single figure may be entered. It is also possible to search all sections within a range, e.g.: $0 \leq \S \leq 50$. Such a range makes sense where the result of a search would yield more than 300 hits, because the sorting function is only guaranteed up to a maximum of 300 hits.

Artikel (article) The search field "Artikel" essentially corresponds to the search field "Paragraph". In the corresponding category "Artikel", all articles are expressed by Arabic numerals even though they may appear in Roman numerals in the law itself.

Anhang (annex) Some laws have annexes. The entry of a numeral into this search field selects an Annex, independent from its numbering (e.g. A, B or C). If the numbering is not clear, separate information is displayed in the §0-document in the category "Anmerkung".

Typ (type) Search field "Typ" selects the type of law, e.g. federal constitutional laws, federal laws, announcements, treaties (multi- and bilateral).

Kundmachungsorgan (promulgation organ) For browsing through the text of a law, which is a typical search in everyday practice, it is sufficient to enter the number of the Federal Law Gazette in the field "Kundmachungsorgan". If the number of an amending law is entered, the system will display all documents affected by that amending law.

Index The "Index" search field allows for the entry of the index of the current version of the federal law. In contrast to the printed index, which uses six numerals for federal laws, only four numerals may be entered, allowing for the search by subjects (one digit), main groups (two digits) and sub-groups (four digits) but not by laws (min. six digits).

Unterzeichnungsdatum (date of signature) The date of signature search field only exists in the §0-document of state treaties. Entry requires the YYYYMMDD format. Wildcards allow e.g. searching for all state treaties signed in a particular year (e.g. 1993 by entering 1993*).

Fassung vom (version of) The system always suggests the date of the current version. Using the search field "Fassung vom" the user may manually change this suggestion:

- To another date.
- To an empty date (by deleting the suggestion). In this case, all documents (enacted as well as suspended ones) will be found.

The following search fields are available in the §0-document only. This is a synthetic document which does not exist in "Federal Law". Its purpose is to store the following specific metadata essential for the administration of federal laws:

Fundstelle (source) This category allows for the entry of the number of the Federal Law Gazette (BGBl.Nr.) or of its predecessor as a promulgation organ, e.g. the Imperial Law Gazette (RGBl.Nr.). If the number of an amending law is entered, the system will display all documents affected by that amending law.

IDAT and ADAT (enacting and suspending dates of a law) The IDAT and ADAT categories allow for entering a date in the YYYYMMDD format. An ADAT (suspending date) of 99999999 means that this legal text is still valid.

Beachte (note) This text in this category is not intended for interpreting the legal text but for e.g. indicating that the legal norm has either been substantively repealed or otherwise rendered inapplicable. An other example is the indication of a "applicability period" which might differ from IDAT and ADAT as frequently is the case with tax laws.

Langtitel (long title) This category contains both the full title of the Federal Law and its "Fundstelle" in the Federal Law Gazette. According to the new procedure applicable since 1990, the long title is supplemented by the parliamentary documentation and, since Austria's accession to the European Union, with the Celex number.

Änderungen (modifications) This category contains the so-called “idF-List” including all sources which amend the legal norm in a formal manner. These are not only amendments of the same type of legal source but also corrections of printing errors (DFB) and rulings of the Constitutional Court (VfGH), which modify the legal source by means of repeal. Apart from the Federal Law Gazette No. (BGBl.Nr.) this information is supplemented with a reference to the parliamentary documentation and, since Austrias accession to the European Union, with the Celex number.

Anmerkungen (notes) This category is free text for any purpose.

Erfassungsstichtag (date of listing) Not every listed legal norm has a date of listing.

As stated in the beginning, legal categories are important for document production. Later on in this paper this process will also be called “ex ante”. Search fields are relevant for document retrieval, later on called “ex post”. It is important to note that the number of categories usually is larger than the number of search fields, e.g. in RIS there is no specific search field for keywords. However, keywords may be defined during document production, filling a specific category. As a consequence, document retrieval on keywords is always possible, but only by using the full text retrieval mode.

2.7.4.7 Strengths

The approach proposed by the eLaw project is mainly devoted to electronic workflow implementation. Being addressed to workflow and presentation, such a schema has been conceived to have a low impact on different information systems of different offices. This is probably the main strength of the project, which has led to the definition of a rather simple schema, with respect to others as the MetaLex and the NormeinRete one.

2.7.4.8 Weaknesses

The simplicity of the eLaw schema represent also a point of weakness: using this schema the possibility of implementing advanced reasoning functions on norms is strongly limited.

2.7.5 Legal XML

2.7.5.1 Introduction

The LegalXML mission is to develop open, non proprietary standards for legal documents and associated applications. LegalXML is a collection of standards developed by different Technical Committees, covering a wide spectrum of legal materials. To date, the Court Filing Workgroup is the first and only workgroup to publish a ”proposed” standard specification. Other fields of application, like the Legislative Documents Workgroup having similar goals as ESTRELLA WP3 have still no appropriate results available at the moment.

2.7.5.2 Context from which XML language started

Formed in 1998, LegalXML is a collaboration of attorneys, court administrators and IT staff, academics, and companies. Early LegalXML work focused on electronic filing of court documents. The group's first specifications, "*Court Filing 1.0 and 1.1*", "*Court Document 1.1*", and "*Query and Response 1.0*" addressed key areas of concern for attorneys and managers of court case records.

LegalXML joined OASIS (the Organization for Advancement of Structured Information Systems) in 2002. Much of the work done in LegalXML touches on the concerns of national organizations and associations outside OASIS. Such work is presented to those groups for consideration for their adoption as business and technical standards. Specifications from the Electronic Court Filing Technical Committee (ECFTC) are submitted for review and have been adopted by the Joint Technology Committee (JTC) of the Conference of State Court Administrators (COSCA) and National Association for Court Management (NACM).

2.7.5.3 Main features

LegalXML should produce standards for electronic court filing, court documents, legal citations, transcripts, criminal justice intelligence systems, and others.

LegalXML divides it's "domains" vertically and horizontally into various "subdomains". For instance, vertical subdomains include, but are not limited to, Court Filings, Transcripts, Judicial Decisions, Public Law, Private Law and Publications. Horizontal subdomains include Citations, General Vocabulary, and Logical Document Structure (e.g. root elements, tables, paragraphs). Horizontal subdomains cut across vertical subdomains. For example, citations will be found in Court Filings, Case Law, Public Law and Private Law documents. Legal XML seeks to harmonize and coordinate the various horizontal and vertical subdomains within the larger legal community.

LegalXML has subcategories of specifications and respective Technical Committees (TCs)[3]:

- **LegalXML Electronic Court Filing**: XML standards to create legal documents and to transmit legal documents between an attorney, party or self-represented litigant and a court or other user of legal documents.

- **LegalXML eContracts**: markup of contract documents to enable the efficient creation, maintenance, management, exchange, and publication of contract documents and contract terms.

- **LegalXML eNotary**: an agreed set of technical requirements to govern self-proving electronic legal information.

- **LegalXML Integrated Justice**: XML standards for exchanging data among justice system branches and agencies.

[3] Current LegalXML Technical Committees, `http://www.legalxml.org/committees/index.shtml`

- **LegalXML Legislative Documents**: XML standards for the markup of legislative documents and a system of simple citation capability for non-legislative documents (e.g. newspaper articles).This TC closed and reopened recently due to lack of interest and resources

- **LegalXML Online Dispute Resolution (OdrXML)**: XML standards for the markup of information and documents used in online dispute resolution systems.

- **LegalXML Legal Transcripts**: XML standards for the syntax to represent legal transcript documents either as stand-alone structured content or as part of other legal records.

- **LegalXML Subscriber Data Handover Interface (SDHI)**: XML standards for the production of consistent Subscriber Data Handover Interface (SDHI), by telecommunication or Internet service providers, concerning a subscriber or communications identifier (e.g., a telephone number) in response to an XML structured request which includes, when necessary, authorization from a judicial, public safety, or law enforcement authority.

Unfortunately these committees has not proposed any draft specifications yet. Some does not even have any published documents, others are drafting requirements.

The only exception is the **Electronic Court Filing TC**, which developed the descendant of the early Court Filing documents, the current version is the LegalXML Electronic Court Filing 3.0 draft specification. The document defines procedures for court filing and document metadata, but does not include any document structure description concerning a legal document.

All of the aforementioned documents are planned to use the following data model as building blocks: **Global Justice XML Data Model (Global JXDM)**[4]: XML standards that enable the justice and public safety community to effectively share information at all levels - laying the foundation for local, state, and national justice interoperability. It provides the semantics and structure of common data elements and types required to exchange information within the justice and public safety communities.

As LegalXML is not yet capable of representing legal documents in detailed structure, it is not possible to evaluate it as a standard for such task.

2.7.6 LAMS

The "Legal and Advice Sectors Metadata Scheme" (LAMS[5]) initiative, developed by the Lord Chancellor's Department in UK as part of the Community Legal Service (CLS) launched in April 2000, aims at promoting common standards across Internet sites developed by organisations in the Legal and Advice Sectors.

It is specifically concerned with the standardisation of websites holding information on legal matters to the extent that they should classify information according

[4] Global Justice XML Data Model (Global JXDM), `http://www.it.ojp.gov/gjxdm`

[5] `www.lcd.gov.uk/consult/meta/metafr.htm`

to a common framework. The proposals are intended to deliver benefits to both providers and users of these websites.

LAMS is conform closely to the standard metadata element set of “Simple” Dublin Core. This gives the greatest possibility of gaining the advantages offered by the adoption of an existing metadata scheme.

2.7.7 JSMS

Justice Sector Metadata Scheme (JSMS) is a development of the AGLS metadata scheme, and it is designed for the use of organisations in New South Wales, Australia, which are publishing legal materials on the Internet. JSMS makes some minor qualifications to the “Simple” Dublin Core metadata elements adopted by AGLS. It does not use any of the AGLS-specific elements but does add several of its own.

2.7.8 UKMF

UK Metadata Framework (UKMF) aims at describing all resources within the government sector so that policy-makers have access to the resources on a particular policy issue, regardless of the department to which those resources belong.

Chapter 3

A comparison among normative XML representations

3.1 Methodology Presentation

3.1.1 Scope and Principles

The scope of this chapter is to present a synoptic table for comparing normative XML models existing in the state of the art. Most of them come from European State Members, while others come from extra-EU members such as Switzerland, Australia, New Zeeland, Canada, African countries.

This comparative analysis aids to fulfil two sub-objectives of ESTRELLA work-package 3:

- to underline the weakness and the straightness in the existing state of the art;
- to define a map of the main European and non-European methods of mark-up applied to the legal resources in order to implement the interchange meta-model. This analysis can provide important input to the other deliverables in order to guarantee the transposition from others national standards to ESTRELLA.

We based our analysis on the following criteria:

- we evaluate content, presentation and metadata levels separately;
- we are interested in modelling the structure of the document rather than to define a data-model of the document (document model vs. data model);
- the document model should be independent of the presentation (e.g. typographical tag) and procedural (e.g. application procedure) levels;

3.1.2 Sample Selection

Considering the principles illustrated in the previous Paragraph, we compare the following six XML standards described in Chapter 2: CHlexML, LexDania, MetaLex, NormeinRete, AKOMA NTOSO, EnAct. We included CHlexML and EnAct in order to extend the comparison to some relevant non EU experiences. Furthermore, in the case of Dutch standards we have limited the analysis to MetaLex for it is a more general and promising standard.

We have excluded by our analysis any XML schema or model that aims, as main goal, to model the legal document for presentational purpose or only with an ontological approach. For example, we do not take into account the Legal-RDF[1]. In fact, this standard provides a method for enriching the existing HTML/PDF files with an ontological framework transforming them in XHTML[2]. In this approach everything is considered as annotation from the editorial staff and for this reason matter of the metadata. We have excluded this approach in this comparison for the following two reasons:

- it is totally oriented to an ontological mark-up also of the content;
- it is based on XHTML, even if some important inputs come from the categorization of the metadata classes: actors (legal entities, groups, and organizations), dramas (legal and illegal acts and activities -events-), roles (legal, commercial, public, and private roles), props (legal intellectual and commercial products), scenes (location and time expressions), scripts (for presentational layer), themes (economy, justice, politics, etc).

Moreover, we do not consider in our analysis FORMEX[3], that is the European XML standard for marking-up the European Official Journal. Despite it is very interesting especially because it derives from SGML experience, it mainly addresses presentation aspects since it is targeted to publication.

CronoLex [de Andrés Rivero and Gómez Skarmeta, 2006],[Marín et al., 2005], a Spanish DTD model for legislation act, was not included in sample because it is essentially a transposition in Spanish of a sub-set of NormeinRete and currently it is not adopted to any national or regional institutions.

3.1.3 Comparison Criteria

Here below are described the six criteria that have been chosen for comparing the expressivity of the different solutions.

1. Architecture and approach of the model
   (a) Scope of the model: we considered the scope of the model in term of kind of documents (law, debates, case, etc.) or domain of application (normative, administrative, juridical, etc.).
   (b) DTD or XML-schema: we underlined the technology used for creating the grammar model.

[1] Hypergrove Engineering publishes the first working draft for Legal XHTML, a grammar for legal documents encoded using XHTML 2.0 and the Resource Description Framework, and now is managed by Legal-RDF no-profit organisation. http://www.hypergrove.com/legalrdf.org/

[2] John McClure, Tagging Legal Text, Hypergrove Engineering, February 28 2006, http://www.hypergrove.com/legalrdf.org/LegalMarkup.html

[3] http://formex.publications.eu.int/

   (c) Prescription versus description: we analysed if the model is simply descriptive or if it proposes some mechanisms in order to ensuring compliance with respect to legal drafting rules.

   (d) Pattern versus free listing of elements: we analysed if the schema is built using patterns or not.

   (e) Content model containment versus free content model: we considered if the content model definition is limited to few categories or if it is completely free.

2. Structural elements and normative references

   (a) Content: we considered the part of the model for representing the structure of the document and the normative references. We emphasize also the naming convention used for preserving the persistence of the references to the internal or external resources (URI, URN, URL).

   (b) Presentation: we analysed mark-up elements for representing objects such as table, images, etc.

3. Metadata

   (a) General metadata: we considered the presence of structures inside of the model dedicated to the general metadata concerning the whole document.

   (b) Metadata inline: we evaluated the presence of structure for expressing the metadata in the place where it is connected to the part.

4. Temporal elements and versioning

   (a) General temporal elements: we evaluated the presence of elements or attributes for expressing the temporal entities concerning the enter in force, the date of publication, the date of delivery of the document, or any static temporal entities connected to the document.

   (b) Temporal elements inline: we evaluated the presence of elements or attributes for expressing the temporal entities concerning the principle of enter in force, efficacy, applicability of each disposition o fragment of the text. These kind of elements are fundamental for managing the amendments of the document over time. Other temporal elements concerning obligations, rights, permissions, etc. were considered.

   (c) Versioning: we evaluated the different modalities permitted for managing versioning of the same document over time. The norms change over the time so it is essential to manage the mechanism of versioning. We define multiversioning when it is possible to include in the same document different versions of the same text, monoversioning when it is possible to mange the versioning linked to the physic document (one version, one document).

5. Semantic web and Ontology connection

   (a) Ontology: compatibility with external ontology, possibility to refer external RDF models.

6. Special functions

   (a) Multilingual elements: we analysed if the model object of this comparison includes elements for managing multilingual both for each text fragment and for the entire document as a particular expression of the same document.

   (b) Geographical elements: we analysed if the model includes elements for representing space and geographical object connected also with the norms.

These criteria will be used for the comparison analysis.
Other parameters should be considered for the objective of ESTRELLA project:

1. openness, extensibility, and interoperability;

2. suitability for logic description, logic reasoning, and semantic web.

3.2 Comparative tables

Tables 3.5-3.6 show, for comparison, the most relevant features of some of the main existing legislative XML models.

MetaLex	
Architecture and approach of the model	**It is used by some Dutch ministries but it is not a national standard**. It models several legal documents: law, case law, doctrine. There is one main XML-Schema (Metalex.xsd), which is the combination of two partial schemas (MetalexStd.xsd and MetalexRef.xsd). In addition, there is a separate schema for decisions (MetalexDec.xsd) and for OWL representations (metalex.owl).
Structural elements and normative references	Structure is defined in recursive way. It preserves the definition of the meaning of the part of the text. Hierarchical structure is included. References are made through URIs Annexes may be plain text or XML, but need not be structured. Tables and images are managed. Quoted text is managed.
Metadata	Metadata for the document and metadata inline are defined. Annotation
Temporal elements & versioning	Nine times are managed for any part of the text: • Publication • Repeal • Version • Enact • Interval-efficacy • Interval-effect • Duration-enactment • Duration-Active • Duration-efficacy MetaLex keeps track of the latest modification date of each structure element. Versioning is managed with namespace in order to have multiversioning document
Semantic web and Ontology connection	Very smart mechanism for link ontology based on RDF or OWL.
Special functions	Include multilingual mechanisms and geographic metadata for managing the jurisdiction. It is possible to localize tags

Table 3.1: Metalex

LexDania	
Architecture and approach of the model	**Adopted by the Denmark Parliament. Official Standard.** One meta-schema with 41 sub-derived XML-Schemas. One for specific typology of legal document: bill, act, minute, debate, order of work, order of day, etc.
Structural elements and normative references	Structure is defined on typology elements and not on semantic meaning in the text. The hierarchy changes on the basis of the typology of the document. The elements are very complex and nested. The naming convention declared for the resources is URN but the composition is similar to an internal codification (e.g. `AT002281`). Also the normative references have the same structure but with more significant sub-parts. (e.g. part21, section 2 → `ACCN_P21_S2`)
Metadata	One specific XML Schema is dedicated to the metadata. A lot of metadata are modelled for the parliament acts and processes but without a categorisation
Temporal elements & versioning	Times are managed by means of two dates: date of enter in force - *Diesvalens*, date of efficacy - *Diescoactu*. The main modificatory provisions are managed.
Semantic web and Ontology connection	No connection with RDF or OWL
Special functions	Simple mechanism for managing multilinguism

Table 3.2: **LexDania**

NormeinRete	
Architecture and approach of the model	Defined by a Ordinance of the Italian Government in 2002. 3 DTDs and XML-Schemata: • basic for didactical scope and simple documents • loose for legal documents not compliant with current Italian drafting rules • strict for legal documents fully compliant with current Italian drafting rules The core vocabulary is the same in all the three schemata. All the national legislative and normative documents are covered by the standard (national, regional, local level)
Structural elements and normative references	Structure is based on the semantic meaning of the part of the text. Annexes are completely managed. Tables and presentational basic elements are managed using the namespace of HTML. Nested list are managed but with a limitation of semantic. Name convention is defined with a specific technical norm URN, and the normative references are managed by the same mechanism.
Metadata	Metadata are divided in general metadata for the whole document and inline metadata for the place-on. General metadata include the lifecycle tracking and all the main events occurred to the document are included. InlineMeta includes several metadata such as: • Analysis provisions metadata concerning the deontic, • Modificatiory provisions for managing the point-in-time.
Temporal elements & versioning	Three main times are managed: in force(interval), efficacy (interval), date of application (instant not interval). Possibility to express duration (interval) of the modification, if the modification is negative, if it is conditioned. Versioning is managed in two way: multiversioning or monoversioning
Semantic web and Ontology connection	No connection with RDF or OWL
Special functions	Very simple mechanism for marking up multilingual contents.

Table 3.3: NormeinRete

AKOMA NTOSO	
Architecture and approach of the model	**Adopted by the UNDESA as African legal resources standard.** One XML-schema and DTD only for didactical reasons for managing act, bill, parliament documents such as debates, minutes, order of day.
Structural elements and normative references	Structure is modelled on semantic meaning of the part of the text. There are: • hierarchy structure • semi-structural document • free-document Tables, list and annexes are managed. Naming convention of the document is based on PURL as well as the normative references. Quoted text is modelled. Using of pattern in the content model and in the XML-schema design.
Metadata	A general metadata block is defined for managing all the metadata of the document. A sophisticated ontology of the legal document is implemented: work, expression, manifestation, item. Lifecycle events are annotated for each main event occurred in the document. Analysis provisions' metadata concerning the modifications. Metadata inline.
Temporal elements & versioning	Date of enter in force for all the document is included, as well as the date of publication, adoption. Three main interval times are managed for each partition or text fragment or modificatory provision: in force, efficacy, application.
Semantic web and Ontology connection	Mechanism for connect the schema with any ontology by URI.
Special functions	Very simple mechanism for labelling the multilingual content.

Table 3.4: AKOMA NTOSO

CHLexML	
Architecture and approach of the model	**It is under approval by the Government as national standard.** Only one schema for all types of legislative texts.
Structural elements and normative references	Structure is modelled on semantic meaning of the part of the text. General structure: • Metadata • Contents • Tail • Comments No predefined structure or hierarchy. There are a lot of very different hierarchy trees which can be found in the legislation of the Confederation and 26 cantons. No "quoted text" tag. Table, images, annexes are managed. Naming convention is based on URN[Khaled et al., 2004] URN includes versions,languages, annexes. URN example: `urn:chlex:ch;bund:1982-12-31;979`
Metadata	Annotation are included. References with external resources are managed as links; there is not ontology management. RDF namespace is included. e.g. `http://www.admin.ch/ch/d/sr/220/a962.html`
Temporal elements & versioning	Three dates only at document level: • In force • Efficacy • Modification date Some modificatory provisions for the entire document: adoption, votation, publication, approval, ratification, force, lastChange, abrogation
Semantic web and Ontology connection	None, it is only possible to include the RDF namespace

Special functions	Simple mechanism for managing unlimited mutlilinguism
	`<normSubjects>`
	`<normSubject lang="en">tax VAT reduction</normSubject>`
	`<normSubject lang="de">MwSt Steuern Reduktion</normSubject>`
	`<normSubject lang="fr">TVA rduction impts</normSubject>`
	`</normSubjects>`

Table 3.5: CHLexML

<table>
<tr><th colspan="2">EnAct</th></tr>
<tr><td>Architecture and approach of the model</td><td><b>Used in official way in the legal resources collections of Tasmania government, Canada government, some federal state of US, New Zealand government.</b>
One DTD for each main kind of document: bills, act, regulations.</td></tr>
<tr><td>Structural elements and normative references</td><td>Structure is modelled on semantic meaning of the part of the text following the Anglo-American traditions.
Mostly it is managed as a list of elements very detailed but without mandatory or restricted schema hierarchy. A simply list of element in alternative sequence.
No pattern or content model definition.

```
<!ELEMENT     altEl --
( schedule | appendix | chapter | part | division |
c.chapter | subdivision | section | subsection |
definition | paragraph | subparagraph | subsubparagraph |
subsubsubparagraph | text | s.part | s.division |
s.subdivision | s.clause | subheading | s.subclause |
table | tgroup+ | row+ | image | equation | graphic |
oath | charge | penalty | heading | headNote | note |
title | longTitle)
-(altEl)>
```

</td></tr>
<tr><td>Metadata</td><td>All the metadata are included as attribute or elements in the DTD.
There isn't division between content and metadata.</td></tr>
<tr><td>Temporal elements & versioning</td><td>Two main data are managed:
• Commence date
• Valid date
Other processes data are managed in the bill:
• GazetteDate
• upperHouse
• lowerHouse
• assentDate
• reservedDate
• proclaimDate
There is a very simple mechanism for managing the textual modifications basically on the base of the Anglo-American traditions where the modifications are very well structured and textual.</td></tr>
<tr><td>Semantic web and Ontology connection</td><td>No references to the external OWL or RDF resources</td></tr>
<tr><td>Special functions</td><td>No multilingual.
Very limited geographical applicability in national legislation.</td></tr>
</table>

Table 3.6: EnAct

3.3 Conclusion

3.3.1 Summary of the comparison analysis

1. **Scope**: all the models analysed represent legislative documents at least at national level. AKOMA NTOSO, LexDania and EnAct also manage documents produced during the norms life cycle process (e.g., parliament proposals, preparatory works, etc.) MetaLex standard manages the case laws and the doctrine.

2. **DTD vs. XML-Schema**: several models are based on XML Schema standard (MetaLex, AKOMA NTOSO, LexDania, CHLexML) that guarantees data typing, possibility to define restriction rules or extensions and in general hierarchy among classes. EnAct is implemented by means of DTD. NormeinRete standard has both XML Schema version and a DTD version. Therefore, XML Schema can be considered the preferred choice for the project.

3. **Structure**: several models (NormeinRete, EnAct) define **structure** elements to manage the semantic meaning of the the legislative text part (e.g. title, part, section, etc.). Others (LexDania, CHLexML) define general neutral text elements (e.g. char, para, parText, etc.). Some others, such as MetaLex and AKOMA NTOSO, maintain the hierarchy of the document structure ("*procreativity*"); they are flexible, generic and recursive. Moreover AKOMA NTOSO is based on pattern content models and XML Schema design. The **recursive approach** and the use of **patterns** are strongly recommended for improving interoperability and interchanging.

4. **URI**: some models use URL (e.g., LexDania), some are based on URI/URN (NormeinRete, CHLexML, MetaLex), AKOMA NTOSO is based on Permanent URL. The main pillar emerging from the analysis is to define naming convention on the basis of significant and enduring properties of the document, in order to make it permanent, stable, automatically derivable, independent to any interpretation. Also the normative references should be based on this mechanism in order to enable the cross-relationship among documents belonging to different data sources. URL model should be not admitted in the ESTRELLA model.

5. **Metadata**: all models include metadata at document level for representing document status, document information (authors, data of markup, etc.) or simply annotations. Metadata can be mixed with the content (EnAct, CHLexML) or can be included in a different position with respect to the text. In some cases metadata are well structured and categorised (MetaLex, AKOMA NTOSO, NormeinRete), in others are a simply list of fields (EnAct, LexDania). Therefore a metadata model is needed for managing: functional analysis of the norms (modifications provisions, functional provisions, permission, obligation, rights, etc.), preservation metadata (archivistic domain), processes metadata (workflow), publication/editorial metadata; classification metadata (thesauri).

6. **Temporal entities**: all models include temporal entity definition. Most of them are defined at document level and include static dates (publication, adoption, delivery, enter in force date). Few schemas manage the temporal entities with an inlinemeta mechanism (MetaLex, AKOMA NTOSO, NormeinRete) linked to a document partition (e.g. article). Only few standards handle this kind of information with finest granularity inside the fragment of the text (MetaLex, AKOMA NTOSO, NormeinRete). The dynamic temporal entities (interval) should be managed in the fragment level in order to handle three kind of axes: timeline of the come into force, timeline of the efficacy, timeline of the application of norm. AKOMA NTOSO and MetaLex are now the unique models that permits this mechanism.

7. **Versioning over time**: some schemas include mechanism for managing the versioning of the document over the time (CHLexML, LexDania). Other schemas permit the versioning of each part of text (NormeinRete, AKOMA NTOSO, MetaLex, EnAct) and allow the multi-versioning model. Nevertheless only AKOMA NTOSO and NormeinRete are able to track the modificative provisions in the modifier text and also in the modified text with specific metadata in order to track any actions done the document. Thus, it is possible to know the whole lifecycle of the act, not only as macro information (document level) but also in the place where the modification occurred and with all the information for rolling back to the document responsible of the action. In MetaLex, it is possible to include several versions of the same document, and by doing so keep track of the entire lifecycle of an act. MetaLex and Akoma Ntoso support similar versioning model. CHLexML, NormeinRete and AKOMA NTOSO use a special specification to extend the URI of the legal resource, that includes the versioning information of the main URI-act; this guarantees the correct and coherent cross-references in case of mono-versioning approach.

8. **Semantic approach**: apart the general metadata of the document, that each model analysed include in some way, MetaLex includes a specific mechanism dedicated to modelling ontology. AKOMA NTOSO permits to connect through reference model any external ontology with the internal part of the text. In the other schema the semantic level is managed with metadata inside the model.

9. **Multilingual functions**: MetaLex is the most expressive model for managing multilingualism according to two point of view: a) it is possible to include multi-language portion of text inside of the same document using the multi-versioning approach and it is possible to extract the mono-language document; b) it is possible to customise the element naming in a language using a `substitutionGroup` function. All the others schemas permit to manage the attribute `lang` for each fragment of text and someone (CHLexML, NormeinRete, AKOMA NTOSO) permits also with the URI to identify the different expressions of the same document in different languages.

10. **Geographic elements**: Only MetaLex contains elements for mark-up geographic scope connected with the textual fragment in a very sophisticated way, in order to infer the legal jurisdiction regarding a place. Other standards such as NormeinRete, CHLexML, EnAct include some simple elements for defining the geo-spatial scope of the entire document. Finally AKOMA NTOSO connects each part of the text with a possible geo-spatial ontology but currently none of them has been implemented for this aim.

3.3.2 Final considerations an synoptic table

All the standards analysed include some characteristics useful for the ESTRELLA model:

1. for the flexibility of the structure see AKOMA NTOSO and MetaLex;
2. for the procreativity of the legal drafting rules see NormeinRete;
3. for the naming convention see CHLexML, NormeinRete, AKOMA NTOSO;
4. for the richness of metadata related the parliament act and workflow see LexDania;
5. for the precision and simplicity of the temporal model see NormeinRete, AKOMA NTOSO, EnAct;
6. for the richness of the analysis of the provisions see NormeinRete, LexDania, EnAct;
7. for the ontology mechanism oriented to the semantic web see AKOMA NTOSO, MetaLex;
8. for the multilingual function see MetaLex;
9. for the geo-spatial function see MetaLex.

The considerations above reported are summarized in the Table 3.7.

	MetaLex	LexDania	NormeinRete	AKOMA NTOSO	CHLexML	EnAtc
1. Structure clean and reiterated	X			X		
2. Prescriptive			X			
3. Naming Convention URI/URN		X	X	X	X	
4. Parliament Act Workflow		X				X
5. Temporal model			X	X		X
6. Provision/analysis		X	X			X
7. Ontology	X			X		
8. Multilingualism in the tag	X					
9. Geo-spatiality linked to the norms	X					

Table 3.7: Synoptic table of the analysed XML formats

Chapter 4

Fitting ESTRELLA requirements

In this section we discuss how the diverse European experiences in analyzing, describing and publishing legislative documents in XML can be used to derive a useful guideline for integration and interoperability across all different domains. In the following section we will first discuss on the relationship between existing experiences and the output of the ESTRELLA effort, and then discuss what should be the ideal characteristics of the XML standard that is proposed by the ESTRELLA project.

4.1 A new XML model, adapt an existing one or integration of them?

The previous sections have shown the complexity and variety of needs, approaches, solutions and syntaxes used throughout Europe for marking up legislative documents. It is clear from even a cursory analysis that every single aspect of the markup is implemented wildly differently in these standards. For instance, NormeInRete uses DTDs, while all others rely on XML Schema; NormeInRete and LexDania use URNs, while MetaLex opts to have users use their own URIs; MetaLex emphasizes reliance with RDF and OWL, NormeInRete and Akoma Ntoso strives for full respect of the published text, and LexDania strongly uses XML Schema datatypes. Thus it is our impression that plainly adopting one solution to the exclusion of all others ends up being impractical and presumptuous and severely risks of being ignored by the supporters of all the rejected ones. On the other hand, it is also preposterous and unjustified to admit the impossibility to reconcile the different experiences and advantages of the existing standards, and to propose to plainly invent a new standard. Such solution would dissatisfy all proponents, and would end up being ignored to even a greater extent than the adoption of one existing standard. This reflection goes to show that we believe that the best option available is to pick and choose the best ideas (or, for that matter, the solutions shared by the most) and try to integrate them, adapting them into a standard that allows each existing standard as a direct sub-construct immediately and automatically obtainable through the use of appropriate applications. In the following section we outline how such integration and adaptation can be designed and what are the main characteristics of such language.

4.2 XML language for ESTRELLA

4.3 Basic principles

The basic principles that we propose are in many respect a choice of the best and most convincing principles of the discussed European standards. They are presented here in no particular order or precedence, and each of them should be clearly lead to the justifying reasons and to all the existing European standards that adopt it.

- **Full reliance on existing standards** from W3C and other relevant standardization organizations: XML, XML Schema, XML Namespaces, OWL and RDF, URIs, etc. must be taken advantage of to their fullest expressive power.
- **Strong distinction between content, presentation and metadata**. Content is defined as all and only the text that has been formally approved by the emanating body. Presentation is defined as the typographical and stylistic choices that have been adopted for the official publication or any other publication and that does not derive from precise and explicit approval from the emanating body. Metadata is defined as all the additional contributions provided by the editors and publishers to make the content better suitable for interpretation and use. Metadata include explicit structuring in terms of XML markup (which implies that all XML markup is considered editorial interpretation of the structural, semantic and jurisprudential aspects of the content) and explicit additional data that is not present in the emanated content, such as publication date and details, emanating body if not explicitly contained in the text, etc.
- **Systematic and required naming policies**, both at the document level and at all levels of the internal structure. This means proposing and adopting one (or a family of) URI schemas (which can be URNs as in NormeInRete or LexDania, or permanent URLs as in Akoma Ntoso) to identify documents, and proposing and adopting one ID/IDREF mechanism (as proposed, strangely enough, by all the examined standards) for every individual structure within the document regardless of their role and importance in the document.
- **Strong distinction between data types and element names**: as LexDania and Akoma Ntoso have taught, there is a larger variety of names than of content models: many differently named structures, such as all elements in the hierarchical structure of parts, have really the same internal structure. So it is possible to allow a large variety of names (basically one for every conceivable structure present in the examine documents) while maintaining a fairly limited and strict variety of data types. Akoma Ntoso and MetaLex propose a set of five classes of content types that can be adopted: containers, hierarchical containers, paragraphs, inline elements and empty elements (or markers).
- **Extensive but constrained extensibility and genericity**: the number of different structural names that can be found in our legislative documents is incredibly high. It is probably impossible to determine all of them and list all

of them before the start of the actual examination and mark-up. Therefore a mechanism for extensibility has to be foreseen and included at the very beginning of the design of the integrated standard. Yet extensibility has not to be mistaken for anarchy and complete freedom to recombine and customize the standard. Akoma Ntoso and NormeInRete have adopted a three-layered model for extensibility that we believe should be adopted in the integrated standard:

  - A specific container for all editor-specific metadata that do not impact on the actual document content (the `proprietary` element in Akoma Ntoso). Virtually anything is allowed in this element.
  - Elements of foreign namespaces for content that is not immediately contemplated in the standard. For instance, it would be too much work to propose a mathematical notation within a standard for legislative documents, yet in some acts some mathematical formulae appear. Appropriate behavior in these cases is to adopt a standard and widely known notation (e.g., MathML) and allow elements of such notation to appear (in their own namespace) in the document.
  - Unforeseen semantic and structural elements will appear that were not discussed and approved in the official standard. In this case, the adoption of generic elements seem reasonable and appropriate: a generic element constrain the structure to one of the approved and known content types, yet allows the name to be specified freely.

- **Self-containment**: a fundamental requirement if we have to consider a long life for marked up documents is that they do not have to rely on specific software systems or the availability of specific databases to be understandable. In particular, every document should contain all the required information for identification, classification, presentation, analysis. This self-containment guarantees that even when a document is received in isolation, and when the content management system they were generated for have stopped working, the document will still be inspectable, understandable and re-usable without problems.

- **Strong and formally sound ontological organization of metadata**: all metadata elements need to be considered as properties of instances of classes associated to actual content. So a full ontology of classes and an extensive list of properties associated to each class need to be devised and a syntax to express these properties needs to be determined. As MetaLex suggests, a full OWL schema needs to be invented and individual properties are expressed in RDF within the XML document.

- **One big and modular schema rather than many independent ones**. Experiences within the NormeInRete project bring to the conclusion that modularity should happen within schema documents, rather than outside of them. That is to say, the standard should end up with one all-encompassing schema, rather than many independent ones for each document class. The advantages of such organization are many: there is only one source of documentation for all document types; the schema can be associated to the document before even knowing the document class; promotion of documents (e.g., a bill becoming an act) and demotion (e.g., a decree incorrectly classified as an act) can happen within the same schema; common structures can be more easily shared across document types, etc.

4.3.1 Mark-up meaningful text elements

As mentioned, we believe that all text elements should systematically be constrained within five content types:

- the *containers*: containers are sequences of specific elements, some of which can be optional. Containers are all different from each other (as the actual list of contained elements vary), but they all share the idea of linear structure of same-level content elements.

- The *hierarchical* elements: a hierarchy is a set of arbitrarily deep nested sections with title and numbering. Each level of the nesting can contain either more nested sections or paragraphs. No text is allowed directly inside the hierarchy, but only within the appropriate paragraph element (or, of course, titles and numbering).

- The *paragraphs*: a paragraph is a container of text or structures that is organized vertically on the display (i.e., has paragraph breaks) and can contain either substructures or text.

- The *inline elements*: an inline element is an element placed within a mixed model element that identifies some text fragments as relevant for some reason. There are both semantically relevant inlines and presentation oriented inlines. There is but one content model using inlines and markers, which means that all mixed model elements (i.e., those that allow both text and elements) also allow the a repeatable selection of all inline elements.

- The *empty elements* (or *markers*): markers are content-less elements that are scattered here and there in the document and are meaningful for their names as well as their attributes. Markers are also known in literature as milestones. There are two main families of markers: placeholders in the text content (e.g., note references) that can appear in any position that also has text, and metadata elements.

Generic elements also are of great importance. Besides named elements, the the new schema should provide for a generic element for each of the five main content types .These are elements called just like the corresponding pattern (e.g., `<container>`, `<hierarchy>`, `<paragraph>`, `<inline>`, and `<marker>`), and are meant to be used for markup that fits the content model but for which no explicitly-named element was provided. A 'name' attribute should be provided for explicitly naming the element. Finally, all metadata elements should appear within a separate section, rather than intermixed with the document content, and expressed as empty elements, with their values placed in appropriate attributes. This allows very simple techniques to separate content from meta-content, as the content is all in the text elements of the document, while meta-content is all in the markup and the markup attributes of the document.

4.3.2 Identify single text elements

All elements of the legislative document in XML format need to be individually identifiable and referenceable. This allows hypertext links, version consolidation and textual analysis to work across all documents. Elements in legislative documents will either be:

- *individuals*: at most one instance of them is present. Examples are the preamble, the conclusion, etc... Their id is simply composed of (a shortened version of) their name. For instance, the id for the preamble could be simply "pre".
- *unnumbered repeatables*: zero, one or many instances of them without explicit numbers in the text. Examples include paragraphs, references, quoted structures, etc. Their id is composed of (a shortened version of) their name followed by the global position of the element within the list of such elements. For instance, the id for the 23rd paragraph should be something like "para0023".
- *globally numbered repeatables*: zero, one or many instances of them are present, and are accompanied in the text by a ordering value such as a number or a letter, which starts at 1 at the beginning of the document. Examples include articles and attachments. Their id is composed of (a shortened version of) their name followed by the explicit ordering value exactly in the format it was written in the original text. For instance, the id for article 12 could be "art12", for attachment iii could be "attiii".
- *locally numbered repeatables*: zero, one or many instances of them are present, and are accompanied in the text by a ordering value such as a number or a letter which re-starts at 1 at the beginning of the containing element. This is the case of sections, parts, list items, etc. Their id is composed of a sequence of the id of the locally-numbered containing elements that are necessary to disambiguate the ordering value of the element, plus (a shortened version of) their name followed by the explicit ordering value exactly in the format it was written in the original text. For instance, the id of part 2 of section iii of book IV could be "bookIV-sectiii-pt2", while for list item ii within list item 3 within list item c within article 13 the id could be "art13-itemc-item3-itemii".

4.3.3 Knowledge representation

The markup of a legislative document must also be accompanied by a rich and complete set of metadata that can be used for more than mere on-screen display of the document. Additional applications of metadata include identification of documents, classification of normative content, consolidation of current and intermediate version of a legislative document, workflow description and summary, preservation management, juridical analysis for scholarly and comparative studies, and many more. Most of the data required for doing such tasks is either non present in the document, or expertly deduced from a semantic analysis of the text content, which means that a domain expert must be in charge of such deductions.

Therefore all knowledge representation must provide support for a large number of metadata elements. These metadata elements, in turn, refer to one and only one of several relevant classes of the ontology of the system. Important classes of such ontology include:

- A *normative act* or *source of law* or *work of law* is the abstract collector associated to the set of provisions that can be described and named as a single entity, that was originally created by its originator (e.g. a Parliament) in a single creative process, and that is referred to in legal reasoning. In a way it is whatever abstract binding connects all different expressions or versions of a document providing normative content.

- An *expression*, or *version*, is (one of) the realization(s) of a normative act in a specific collection of actual sentences and words and punctuation and (where appropriate) presentation choices. For instance, each consolidation of a national act is an expression of that normative act. For instance, the English, Dutch, Italian and German versions of an European act are different expressions of the same normative act. The content of a legislative document is usually a property of each individual expressions or versions of such document.

- A *manifestation* is one of the physical or electronic embodiments of an expression of a normative act. Thus, a specific XML representation, a PDF file (as generated by printing into PDF a specific Word file with a specific PDF distiller), a printed booklet, all represent different manifestations of the same version of the same normative act.

- An *item* is one specific exemplar of a manifestation: a very specific copy of a booklet, a file stored on a very specific computer in a very specific location, etc.

Other relevant classes to be considered when dealing with the knowledge representation of legislative documents are:

- the *normative system* (or the *Law* with capital L) is the whole set of legal provisions that can be applied in any given moment to any given event of real life within any given context (e.g. a country or a region).

- The *content* or *text* of the norm is the set of words, punctuation and (where relevant) presentation that were chosen by the author of an expression for conveying a specific meaning. Note that the idea of content applies to expressions, not works.

- The *components*: the content of a document is usually further organized hierarchically in components, among which we can identify a main document (the only required component), zero or more attachments, workflow and presentation-related documents, and further data fragments and associated texts of various nature and purposes.

- The *folder* or *document collector* is an abstract container of all documents (or precisely, works) that are relevant to the analysis of a piece of legislation. A folder may contain all the different bills from which an act originated, all the different amendment acts that were emanated to modify it, all the official minutes that were drafted during its discussion, etc. The folder thus acts as the connector to bind together a number of different documents that were relevant for a piece of legislation to be enacted.

- The *agent* is the party (a person, an organization) that is relevant for the description of the workflow (or a step thereof) and that must be described in the appropriate workflow description metadata. For instance, the parliament or one of the government ministers could be the agent of an act, while an editor is to be considered the agent of a specific manifestation of such act, and so on.

- *Time records* are important for the determination of the temporal logic implied by the workflow of an act: enactment, in force periods, efficacy, amendments and their effects, retro-activity and ultra-activity, and so on are all metadata that are dependent on the exact identification of a time period.

- *Places* are important as well. In MetaLex the correct identification of places and time are fundamental for the correct description of the efficacy of a piece of legislation.

- *People*, roles and organizations are important as well, as legslation affects roles and organizations and are discussed and proposed by individuals acting within a role for an organization.

4.4 Content management system

Among the basic rules of all systems supporting legislative processes is the complete separation between the data and the system to produce, manage, deliver them. This is important for ensuring that the pieces of legislation marked up and converted to XML will exist and thrive after and beyond the useful life of the applications that created them. As such, the content management system should make no assumption on the data format, and the data format should never assume a specific feature in the content management system to be present. This principle notwithstanding, in this section we detail a few features that the content management system for legislative documents should provide. There are four very separate functions that the content management system must provide:

- *editing*: the task of marking up and generating the XML manifestations of legislative documents. Also, the task of generating the appropriate metadata and the correct organization of user interface and task support according to the expertise of the editorial team.

- *Workflow support*: the task of enabling, verifying and certifying the procedure followed for the generation of a work, an expression, a manifestation or even an item related to a piece of legislation. Also, the task of connecting and clustering different works, expressions, manifestations and items according to their roles within the drafting, discussing, emanating, amending, consolidating and publishing tasks.

- *Consolidation*: the task of creating the editorial expression of an amended document, starting from the original document and all the relevant amending documents. Also, the task of generating all derivative documents and metadata (such as editorial footnotes, lists of amendments, parallel displays, inline comparative versions, etc.) that are connected to the creation and management of consolidated versions of legislative documents and that are necessary or useful for their management and understanding.

- *Publication*: the task of delivering legislative documents to the general public, on the web, in print, and possibly on an arbitrary number of user devices. Also, where appropriate, the support of hyperlinking features for explicit legislative references, editorial footnotes and all types of document references as expressed in the metadata.

Of course, these four functions need not be provided exactly by the same system, and can in fact be considered as characterizing four very different and independent systems. Furthermore, very clearly the users of each function will vary greatly in computer practice as well as in legislative expertise: for instance, while we can expect consolidating expert to be fairly proficient in legislative techniques, we cannot expect them to be fully competent in XML-related issues. Analogously, one should design the editor to be fully and fruitfully usable by clerical help not familiar with legislative issues, and certainly absolutely unaware of the underlying XML technologies. Finally, users of the publication platform should be considered absolutely lacking both computer and legislative competencies.

Chapter 5

Conclusion

In this deliverable an analysis of the main national standards for legal sources, developed within as many national projects, has been carried out. In particular their main strengths and weaknesses have been underlined, pointing out the characteristics which can be useful for the development of a new representation format for legal sources to be shared at European level.

¿From the comparison of different national standards some conclusions can be drawn in order to provide guidelines for the definition of an ESTRELLA standard for legal sources.

Standard definition The comparison results gave evidence of the fact that, for the level of maturity reached by the project and the characteristics of the defined XML language, the MetaLex standard seems to be the main candidate to represent the basis which the new format can be built upon.

The use of XML Schema (XSD) in MetaLex provides a well-grounded approach to deal with data types and extensibility in terms of an object-oriented inheritance mechanism. This characteristic is common to the LexDania project, natively defined using the XSD standard, and to the NormeinRete project which has produced an XML Schema version of its standard, starting from a DTD version of the standard itself.

The approach to building schemas adopted by all these projects should be taken into account. They all share an approach based on the definition of a basic schema to be extended in terms of data types and elements, making widely use of schemas inclusion and inheritance mechanism. This way a structure of stratified layers can be used to incrementally construct the schemas from functional features. This approach provides a well-established methodology to effectively use schemas, as well as to extend them, assuring consistency and compatibility to the new schema extensions. As well, the resulting schemas will be constructed on common components enabling easier interchange and reduced maintenance effort.

In particular the LexDania concept of having three tiered data-model (i. a meta-schema, ii. a set of domain-specific omni-schemas and, iii. application-specific legislative schemas) could be properly taken into account in building the new schema.

For the aims of harmonising different standards, the experience of AKOMA NTOSO seems to be an effective paradigm. It was born in 2004 taking into account the best practices at European level, thus representing an effective compendium of the main valuable features of previous projects, especially of NormeinRete and MetaLex.

While the AKOMA NTOSO methodology of integrating experiences has to be considered for the new standard, it cannot be used as such for ESTRELLA. This is due to the fact that it provides a very limited set of constraints specifically designed to cope with frequent badly-formed documents, thus providing low support to the improvement of the legislative production quality.

Multiligualism Considering the European perspective of the new building schema, the support of multilingualism will be a key-feature of the new standard. MetaLex natively supports multilingualism in two distinct ways:

- Through localization of XML elements;
- By providing the means to maintain multiple language versions of the same document in one file (`xml:lang` attribute);

Localization is achieved by a language-specific schema extension in the standard document schema. This extension maps the element names to the names in a target language, as in the following example:

```
<xsd:element name="regeling" type="Regulation"
substitutionGroup="Regulation"/>
```

The native support of multilingualism in XSD and the wide implementation of it within the MetaLex project is another important reason to choose MetaLex as the basis for the new standard format.

Semantics: Metadata and Vocabularies The definition of the new standard must be accompanied by the definition of a rich set of metadata, referring to specific semantic models.

Metadata should include, at least, identification of documents, classification of normative content, consolidation of current and intermediate version of a legislative document, workflow description and summary, preservation management, juridical analysis for scholarly and comparative studies.

Harmonization of the rich set of metadata on the acts supported by MetaLex, LexDania, NormeinRete and AKOMA NTOSO is expected in defining the new standard.

Moreover, the semantic model of provisions, allowing the description of the meaning and the role of each fragment of a regulation, supported by the NormeinRete project could be taken into account.

The model of provisions has been specifically promoted in the NormeinRete project: a “provision-centric” view of legal order in fact has been considered of primary importance to define strategies and tools for the upkeep of legal systems and to provide facilities to access norms.

The inability to obtain an analytical/systematic vision of a legal order necessarily creates obstacles to its knowledge and upkeep. Therefore a more analytical unit of reference was identified in order to have a more organic view of the legal system. According to this point of view a legislative text may be seen as a vehicle that contains and transports rules, or *provisions*, and the legal order as a set of rules rather than of laws. This perspective, inspired by analytic legal philosophy, permits to perceive the rules as the true bricks in the legal system, and the laws as purely temporary events.

The model of provisions, described in Section 2.5 represents a middle layer of abstraction, bridging the gap between legal texts and legal knowledge. Along with the representation of the text structure, this semantic model seems to fit well the ESTRELLA requirements of establishing explicit links between legal knowledge and legal sources the knowledge is based on. This model could effectively support the activities of WP1, related to the specification and development of a Legal Knowledge Interchange Format (LKIF), as well as those of WP3 where a representation language for legal sources, that can easily be used and coupled to LKIF knowledge models of these sources, is foreseen.

The use of the provision model to describe the semantics of legislative texts gives the possibility of developing different applications [Biagioli and Francesconi, 2005]. A corpus of laws and regulations entirely qualified according to the model of provisions allows to develop advanced search and retrieval services for legislative documents. It can describe the modifications included in a legislative text, thus paving the way to the automatic construction of the consolidated text.

Moreover it allows to perform analyses concerning the coherency of the legal system.

Along with semantic models, the new schema should also provide reference to terms vocabulary for defining and encoding management information, effectively implementing these semantic models.

As regards the language to be used in representing metadata and knowledge within the documents, that has a wide impact in distributing legislation to the public, the use of RDF/OWL schema seems to be the best choice in order to be compliant to the W3C effort to construct a Semantic Web based on RDF.

Document ontology Semantics and its expression on metadata elements, should refer to one and only one of several relevant classes of the ontology of the system.

An ontology of legal sources is being defined within the activities of the CEN/ISSS Workshop Metalex. This vision has been widely discussed in Section 4 and it should be taken into account for the new standard.

Time management The mechanism for time management established by the NormeinRete project, dealing with an integrated management of the whole lifecycle of the acts, represents an important feature that the new standard can inherit and it should be taken into account. As discussed in Section 2.5 it is obtained including in a document a table of all the important events that the related act, or part of it, underwent. Such events, in terms of points in time or time intervals, can be assigned to each element. This mechanism allows to represents the history of a document or document parts, the time properties of them, as well as different categories of documents: Original version, Single version, Multiple version.

On the other end the time model established by MetaLex, which keeps track of dates and periods that are relevant to the lifecycle of legislation is an important starting point to give a semantics, on the legislative point of view, to the variety of times which can be managed by the NormeinRete mechanism.

Harmonisation, or at least inclusion in the new standard, of the different events considered by the other main European project, whose variety is due to the differences in the related legal systems, should be considered.

Identification of documents and document parts An important issue in legal source management is represented by document identification.

The enormous growth and dominance of the Internet and in particular the World Wide Web has made Uniform Resource Location (URL) almost synonymous with an identifier of an entity (or document). However, such identifier cannot be used as an effective reference point since it lacks stability and persistence. The use of hypertext links on the Web based on URLs in fact allows to express references, providing also an effective retrieval systems, but do not appear to be suitable for wide-scale use in the law. References based on physical locations, expressed through URLs, in fact presents the following well-known problems:

- difficulty in knowing the location of the cited resource;
- loss of validity over time of the document locations;
- impossibility of describing references to the resources not published yet.

To overcome these problems identifiers to electronic entities must be based on the logical content, and not on the physical location of the resource. This is the case of Uniform Resource Names (URNs) a mechanism for a unique and intelligent identifier. URNs are particularly recommended in legal domain where documents abound in references, since they are conceived for providing unique, unambiguous and lasting identifiers of network resources, independently of their physical locations.

Uniqueness is the essential attribute of an identifier, which must be unambiguous in the defined namespace. Moreover identifiers should be *intelligent*, which, in this context, means *human readable*, built for ready interpretation outside the identifier scheme and able to derive meaningful information. The expression *human readable* in this case refers to a property allowing the construction of a URN on the basis of the textual citation of a document. This property is particularly desirable in a distributed environment, since it guarantees the possibility of linking in a stable way unavailable documents, without making reference, for example, to on-line services providing "repository-bound" identifiers to be associated to such documents. Moreover it allows to develop software tools implementing automatic hyperlinking of legal sources on the basis of the textual citations of the acts.

An example of intelligent identifier is the Serial Item and Contribution Identifier (SICI) for a periodical's article, which contains sub-strings denoting elements such as the ISSN, the first of the title of the article, issue, volume, date of publication, page number, etc. Another example is the CELEX number, the unique identifier for European legislation (directives). Both the rules for accessing European legislation using the CELEX number and the rules to construct the CELEX (or Eur-lex) number from formal identification parameters identifying the act (the issuing authority, the type of the act, the publication date, the identifier number, etc.) are known.

Nowadays in the legal domain the use of URN is growing.

The project examined in this report have different modalities to identify legal sources. MetaLex does not prescribe an URI for documents, therefore users are free to use their own URI definitions and any document part can be linked to by extending the chosen URI. On the other hand LexDania, NormeinRete and AKOMA NTOSO prescribe a specific URI syntax to identify legal sources, the first two project in terms of a URN standards, while AKOMA NTOSO in terms of a PURL standard.

The definition of a specific URI syntax to be shared at European level to identify legal documents seems to be the most useful solution for identification of documents and document parts. It guarantees the solution of the well-known problems previously discussed. Moreover a URI standard can be used to represent relationships among legal documents and, as discussed, allows to implement the automatic hyperlinking of legal sources on the basis of the textual citations of the acts.

In this context the experience of the NormeinRete project seems to be the best practice. As discussed, the NormeinRete uniform name standard assigned to a legal document depends on the characteristics of the document itself, therefore it is independent from the availability of the document, from its physical location and the access mode. Only the significant details of the document and the knowledge of the URN syntax are necessary to the identification of the act. Other projects examined in this report have defined URN standards on the basis of the NormeinRete experience, as the LexDania identification standard.

A URN identification mechanism is based on a schema for assigning names capable of representing unambiguously any legal measure, issued by any authority at any time (past, present and future) and on a resolution mechanism (RDS: Resolver Discovery Service) able to retrieve the corresponding object. This is according to the standard defined within IETF (Internet Engineering Task Force) by the URN Working Group.

For NormeinRete documents the URN syntax has been defined according to RFC 2141 URN Syntax [Moats and Sollins, 1997], moreover a resolution mechanism able to associate a document URN to one or more physical locations has been developed.

The adoption of a shared URI standard (in terms of URN or PURL) at European level could in fact represent a problem, since a common agreement should be obtained in the identification of sources of law coming from different countries which may have different legal systems and different practices in identification and classification of legal sources.

An alternative to the use of a URI standard can be the adoption of “repository-bound” identifier, given by a specific service provider.

Both these two solutions have benefits and drawbacks. The use of a URI standard allows interoperability among information systems to be federated. Moreover it represents an open standard which can be implemented without being linked to proprietary technologies. However, as discussed, it might result unfeasible because it could be difficult to come into an agreement on the formal parameters of identification.

As regards possible variations of the URI standard, the PURL standard has the advantage of being a URL, that guarantees to be effective without an infrastructure at a service provider level implementing a resolution system. Therefore such a mechanism can work even within basic on-line information systems. However, to guarantee the property of “permanence” for a PURL, a resolution mechanism at the data provider level should be implemented in terms of a hierarchy of directories following the taxonomy of documents identification formal parameters and, if necessary, a redirection mechanism at each virtual location of a given document which points to its actual physical location. It is questionable if it is better a resolution system for a URN schema, based on a database where the associations between a URN and its URLs are maintained, or the one for a PURL scheme, based on redirection mechanisms distributed in a permanent hierarchy of directories.

On the other hand, the use of identifiers given by a service provider facilitates the feasibility of a common identification scheme to be shared among different legal systems at European level, but interoperability among information systems is reduced, unless they agree on using the services associated to such a proprietary identification scheme.

Bibliography

[AIPA, 2001] AIPA (2001). Definizione delle regole per l'assegnazione dei nomi uniformi ai documenti giuridici. Circolare n. AIPA/CR/35. In Italian.

[AIPA, 2002] AIPA (2002). Formato per la Rappresentazione Elettronica dei Provvedimenti Normativi tramite il Linguaggio di Marcatura XML. Circolare n. AIPA/CR/40 (in Italian), `http://www.cnipa.gov.it/site/\_contentfiles/00127500/127544\_CR\_40\_2002.pdf`.

[Arnold-Moore et al., 2002] Arnold-Moore, T., Clemes, J., and Tadd, M. (2002). Connected to the law: Tasmanian legislation using enact. Technical report, TeraText.

[Arnold-Moore, 1997] Arnold-Moore, T. (1997). Automatic generation of amendment legislation. In *Proceedings of the International Conference of Artificial Intelligence and Law.*

[Biagioli and Francesconi, 2005] Biagioli, C. and Francesconi, E. (2005). A semantics-based visual framework for planning a new bill. In *Proceedings of the Jurix Conference: Legal Knowledge and Information Systems*, pages 103–104.

[Biagioli, 1997] Biagioli, C. (1997). Towards a legal rules functional micro-ontology. In *Proceedings of workshop LEGONT '97.*

[Boer et al., 2002] Boer, A., Hoekstra, R., and Winkels, R. (2002). Metalex: Legislation in xml. In *Proceedings of JURIX 2002: Legal Knowledge and Information System*, pages 1–10.

[Boer et al., 2003] Boer, A., Winkels, R., Hoekstra, R., and van Engers, T. (2003). Knowledge management for legislative drafting in an international setting. In *Proceedings of JURIX 2003: Legal Knowledge and Information System*, pages 91–100.

[de Andrés Rivero and Gómez Skarmeta, 2006] de Andrés Rivero, J. and Gómez Skarmeta, A. F. (Fiesole, Firenze, Italy, June 2006). Cronolex. a system for a dynamic representation of laws. In *Proceedings of the V Legislative XML Workshop.*

[Khaled et al., 2004] Khaled, O. A., Chabbi, H., and Ramalho, M. (2004). *ÉTUDE URN.* University of Applied Sciences of Western Switzerland, EIA-FR.

[Lachmayer and Hoffmann, 2005] Lachmayer, F. and Hoffmann, H. (2005). From legal categories towards legal ontologies. In *Proceedings of the International Workshop on Legal Ontologies and Artificial Intelligence Techniques*, pages 63–69.

[Lupo and Batini, 2003] Lupo, C. and Batini, C. (2003). A federative approach to laws access by citizens: The "NormeinRete" system. In Traunmuller, R., editor, *Proceedings of Second International Conference Electronic Government (EGOV)*, pages 413–416. Springer-Verlag.

[Marín et al., 2005] Marín, R. H., Hernández, J. L., and Delgado, J. J. I. (Granada, Spain, May 2005). Cronolex: a computing representation of law dynamics. In *XXII World congress of Philosophy of Law and Socialphilosophy.*

[McClure, 2006] McClure, J. (2006). Legal-rdf vocabularies, requirements & design rationale. In *Proceedings of the V Legislative XML Workshop.* To appear.

[Megale and Vitali, 2001] Megale, F. and Vitali, F. (2001). I dtd dei documenti di norme in rete. *Informatica e Diritto*, 1:167–231.

[Moats and Sollins, 1997] Moats, R. and Sollins, K. R. (1997). URN syntax. Technical Report RFC 2141, Internet Engineering Task Force (IETF).

[Palmirani, 2005] Palmirani, M. (2005). Time model in normative information system. In *Post-proceedings of the ICAIL Workshop on the Role of Legal Knowledge in e-Government.*

[Spinosa, 2001] Spinosa, P. (2001). Identification of legal documents through urns (uniform resource names). In *Proceedings of the EuroWeb 2001, The Web in Public Administration.*

[Tucker, 2004a] Tucker, H. (2004a). *LexDania Documentation. Universal addresses.*

[Tucker, 2004b] Tucker, H. (2004b). *LexDania - Documentation. Guidelines for Writing Omni- and DocType and Application Schemas.*

[Tucker, 2004c] Tucker, H. (2004c). *LexDania - Documentation. Introduction to Concepts and Status of XML Linking.*

[Tucker, 2004d] Tucker, H. (2004d). *LexDania - Documentation. Metadata.*

[Tucker, 2004e] Tucker, H. (2004e). *LexDania - Documentation. XML and XSL Best Practices for Writing Schemas.*

[Tucker, 2004f] Tucker, H. (2004f). *Lex Dania - White Paper. A System of XML Schemas for Danish Legislative Documentation.*

[Vitali et al., 2005] Vitali, F., Iorio, A. D., and Gubellini, D. (2005). Design patterns for document substructures. In *Extreme Markup 2005 Conference.* Montreal, 1-5 August 2005, `http://www.mulberrytech.com/Extreme/Proceedings/xslfo-pdf/2005/Vitali01/EML2005Vitali01.pdf`.