



**LEOS**

**LEGISLATIVE EDITING OPEN SOFTWARE**

Final Results

**JOINING UP GOVERNMENTS**



**This report / paper was prepared for the ISA programme by:**

TRASYS SA<sup>1</sup>

**Disclaimer**

The views expressed in this document are purely those of the writer and may not, in any circumstances, be interpreted as stating an official position of the European Commission. The European Commission does not guarantee the accuracy of the information included in this study, nor does it accept any responsibility for any use thereof. Reference herein to any specific products, specifications, processes, or services by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute nor imply its endorsement, recommendation, or favouring by the European Commission.

This paper can be downloaded from the ISA website:

<http://ec.europa.eu/isa>

---

<sup>1</sup> TRASYS is a member of the STRATIQO consortium

## Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>9</b>
1.1	CONTEXT.....	9
1.2	AIM OF THE DOCUMENT.....	9
1.3	METHODOLOGY.....	10
1.3.1	<i>Collection of information.....</i>	<i>10</i>
1.3.2	<i>Content of the Final Results .....</i>	<i>11</i>
<b>2</b>	<b>SOLUTIONS OVERVIEW .....</b>	<b>13</b>
2.1	INTRODUCTION.....	13
2.2	ANALYSED SOLUTIONS OVERVIEW .....	14
2.3	TOOLS & TECHNOLOGIES USED .....	17
<b>3</b>	<b>ANALYSIS OF “AS IS” OF EXISTING PRACTICES.....</b>	<b>21</b>
3.1	USAGE AND USERS OF THE SOLUTIONS .....	21
3.2	TYPE OF TEXT PRODUCED .....	22
3.3	LEVEL OF CONSTRAINT ON USER’S INPUT, LEVEL OF STRUCTURE OF INFORMATION, INTEROPERABILITY.....	23
3.4	ANALYSIS OF FUNCTIONALITY .....	25
3.4.1	<i>Editing .....</i>	<i>25</i>
3.4.2	<i>Checking.....</i>	<i>28</i>
3.4.3	<i>Metadata, archiving, signing, importing .....</i>	<i>29</i>
3.4.4	<i>Formats.....</i>	<i>30</i>
3.4.5	<i>Content .....</i>	<i>32</i>
3.4.6	<i>Multilingualism.....</i>	<i>32</i>
3.4.7	<i>Other features.....</i>	<i>33</i>
3.5	RE-USABILITY OF THE SOLUTION.....	35
3.6	ADVANTAGES AND DISADVANTAGES .....	36
3.7	FUTURE DEVELOPMENTS .....	40
3.8	CONCLUSIONS .....	42
<b>4</b>	<b>ONGOING EFFORTS AND OUTCOMES FOR COMMON FRAMEWORKS AND STANDARDS FOR STRUCTURE OF LEGAL TEXTS .....</b>	<b>45</b>
4.1	AN EMERGING INTERNATIONAL STANDARD: AKOMA NTOSO .....	45
4.1.1	<i>Introduction.....</i>	<i>45</i>
4.1.2	<i>The purpose of Akoma Ntoso.....</i>	<i>46</i>
4.1.3	<i>Design issues of Akoma Ntoso.....</i>	<i>46</i>
4.1.4	<i>Document types supported.....</i>	<i>48</i>
4.1.5	<i>Structures shared across document types.....</i>	<i>51</i>
4.1.6	<i>Special features of Akoma Ntoso .....</i>	<i>52</i>
4.1.7	<i>The naming convention.....</i>	<i>53</i>
4.1.8	<i>Schemas and subschemas .....</i>	<i>53</i>
4.1.9	<i>Conclusion.....</i>	<i>53</i>

4.2	AN INTEROPERABILITY STANDARD: CEN/METALEX .....	54
4.3	AKOMA NTOSO AND METALEX/CEN – A COMPARISON .....	54
4.4	SURVEY ON THE USE OF XML IN PARLIAMENTS .....	56
4.4.1	<i>XML at the European parliament</i> .....	57
4.4.2	<i>Challenges for using XML</i> .....	57
4.5	THE PUBLICATION OFFICE XML SCHEMA: FORMEX .....	58
4.6	STUDIES ON THE USE OF XML IN EU MEMBER STATES AND ON THE NEED FOR COMMON STRUCTURES.....	59
4.6.1	<i>Outcome of the working group European Forum of Official Gazettes on the Use of XML for the production and distribution of the official gazettes (2006)</i> .....	59
4.6.2	<i>Survey on the EU Parliaments initiatives on common standards for digital data and documents (IPEX - 2011)</i> .....	59
4.7	IPEX (INTERPARLIAMENTARY EU INFORMATION EXCHANGE) XML.....	60
4.8	NATIONAL SCHEMAS .....	62
<b>5</b>	<b>CONCLUSION OF THE INITIAL ANALYSIS .....</b>	<b>63</b>
<b>6</b>	<b>CASE STUDIES .....</b>	<b>65</b>
	<b>CASE STUDY 1: ANALYSIS OF STRUCTURE MANAGEMENT IN TOOLS BASED ON WORD PROCESSORS .....</b>	<b>66</b>
6.1	E NORM/X NORM CASE STUDY – A FRAMEWORK FOR APPLYING STRUCTURES .....	66
6.1.1	<i>Introduction</i> .....	66
6.1.2	<i>Applying the eNorm structure</i> .....	68
6.1.3	<i>Benefits and trade-offs</i> .....	74
6.1.4	<i>eNorm: conclusion</i> .....	79
6.1.5	<i>Annex: eNorm DQC Criteria (Document Quality Check)</i> .....	79
6.2	LEX DANIA CASE STUDY – AN APPROACH TO IMPOSING A STRUCTURE .....	83
6.2.1	<i>Introduction</i> .....	83
6.2.2	<i>Lex Dania schema and user interaction</i> .....	84
6.2.3	<i>Lex Dania editor: benefits and trade-offs</i> .....	91
6.2.4	<i>Lex Dania conclusion</i> .....	93
	CASE STUDY 1 CONCLUSION.....	94
	<b>CASE STUDY 2: “XMLISATION” .....</b>	<b>95</b>
6.3	SDU XMLISATION PROCESS .....	96
6.3.1	<i>Overview of the publication workflow of legislative documents</i> .....	96
6.3.2	<i>Converting to XML</i> .....	98
6.3.3	<i>The need for XMLisation after drafting</i> .....	99
6.3.4	<i>Developing the XMLisation heuristics</i> .....	100
6.3.5	<i>Efficiency of the XMLisation process</i> .....	100
6.4	THE XMLISATION PROCESS FOR EUROPEAN PARLIAMENT AMENDMENTS .....	101
6.4.1	<i>Overview of the amendment workflow</i> .....	101
6.4.2	<i>Converting to XML</i> .....	102
6.4.3	<i>Benefits and efficiency</i> .....	106
6.5	THE MARK-UP OF A DOCUMENT WITH THE NORMA EDITOR.....	107
6.5.1	<i>Introduction</i> .....	107

6.5.2	<i>Converting to XML</i> .....	107
6.5.3	<i>Re-usability</i> .....	110
6.6	THE MARKUP OF A DOCUMENT WITH XMLLEGESEDITOR.....	111
6.6.1	<i>Introduction</i> .....	111
6.6.2	<i>Converting to XML</i> .....	113
6.6.3	<i>Efficiency - accuracy evaluation</i> .....	116
6.6.4	<i>Re-usability</i> .....	117
	CASE STUDY 2 CONCLUSION.....	120
	<b>CASE STUDY 3: LEGISLATIVE EDITOR USING OPEN OFFICE.....</b>	<b>121</b>
6.7	BUNGENI, AN OPEN SOURCE DESKTOP EDITOR FOR LEGISLATIVE MARK-UP BASED ON OPEN OFFICE .....	121
6.7.1	<i>Applying the structure</i> .....	121
6.7.2	<i>User-guidance for applying the structure</i> .....	124
6.7.3	<i>Technical aspects</i> .....	125
6.7.4	<i>Benefits and trade-offs</i> .....	132
	CASE STUDY 3 CONCLUSION.....	135
	<b>CASE STUDY 4: LEGISLATIVE PROCESS SUPPORTED BY OPEN SOURCE DOCUMENT MANAGEMENT AND OPEN SOURCE WORD PROCESSOR.....</b>	<b>137</b>
6.8	S.O.L.O.N II, OPEN SOURCE WORD PROCESSOR AND DOCUMENT MANAGEMENT SYSTEM SUPPORT THE LEGISLATIVE PROCESS. ....	137
6.8.1	<i>Description of the solution</i> .....	137
6.8.2	<i>Users and change management</i> .....	143
6.8.3	<i>Benefits and needed improvements of the new solution</i> .....	143
	CASE STUDY 4 CONCLUSION.....	144
	<b>CASE STUDY 5: DESKTOP XML EDITOR CUSTOM DEVELOPED FOR LEGAL TEXTS ...</b>	<b>145</b>
6.9	VEXPRO, A DESKTOP XML EDITOR FOR EDITING LEGISLATIVE TEXTS .....	145
6.9.1	<i>Rational for an XML editor</i> .....	145
6.9.2	<i>Users</i> .....	146
6.9.3	<i>User friendly editing in a complexe structure</i> .....	146
6.9.4	<i>Advantages and drawbacks</i> .....	150
6.9.5	<i>Re-usability</i> .....	151
	CASE STUDY 5 CONCLUSION.....	152
	<b>CASE STUDY 6: A WEB XML EDITOR BASED ON AN OFF-THE-SHELF SOLUTION.....</b>	<b>153</b>
6.10	WETSEEDITOR, A WEB XML EDITOR FOR AMENDMENTS .....	153
6.10.1	<i>User benefits</i> .....	153
6.10.2	<i>User-friendly editing in a complex structure</i> .....	156
6.10.3	<i>User involvement and change management</i> .....	165
6.10.4	<i>Benefits and trade-offs</i> .....	166
	CASE STUDY 6 CONCLUSION.....	167
7	<b>OVERALL CONCLUSIONS AND RECOMMENDATIONS .....</b>	<b>169</b>
7.1	CONCLUSIONS .....	169

7.1.1	<i>Solutions based on word processors.....</i>	<i>169</i>
7.1.2	<i>Solutions for user-friendly XML editing.....</i>	<i>170</i>
7.2	RECOMMENDATIONS .....	171
7.2.1	<i>The need for a structure.....</i>	<i>172</i>
7.2.2	<i>The need for divers tools .....</i>	<i>176</i>
<b>8</b>	<b>ANNEX 1: OTHER STUDIES ON LEGISLATIVE EDITING SOLUTIONS AND TOOLS</b>	<b>181</b>
8.1	FEATURES OF SYSTEMS FOR MANAGING BILLS IN PARLIAMENTS .....	181
8.2	FINAL REPORT OF THE WORKING GROUP “DATA CAPTURING AT THE SOURCE” - EUROPEAN FORUM OF OFFICIAL GAZETTES .....	182
8.3	FINAL REPORT OF THE WORKING GROUP ON XML - EUROPEAN FORUM OF OFFICIAL GAZETTES 184	
<b>9</b>	<b>ANNEXE 2: THE AIM OF OPEN LEGISLATIVE DOCUMENT STANDARDS .....</b>	<b>185</b>
<b>10</b>	<b>ANNEXE 3: ADVANTAGES TO THE USE OF OPEN STANDARDS IN PARLIAMENTS....</b>	<b>187</b>

## Figures

FIGURE 1 XMLEGES.IT SCREENSHOT	26
FIGURE 2 BUNGENI-EDITOR SCREENSHOT	43
FIGURE 3 WETSEEDITOR.NL SCREENSHOT	44
FIGURE 4 AT4AM.EP SCREENSHOT	44
FIGURE 5 IPEX XML SCRUTINY STRUCTURE	61
FIGURE 6 IPEX XML – SCRUTINY STRUCTURE ILLUSTRATION	61
FIGURE 7 ENORM – SCREENSHOT OF REFERENCE TEXTS AND DIRECT LINKS TO THE HANDBOOK OF THE LEGAL FORM	68
FIGURE 8 ENORM SCREENSHOT SHOWING THE GRANULARITY OF THE DOCUMENT STRUCTURE	70
FIGURE 9 ENORM SCREENSHOT – STRUCTURE ELEMENTS AND DRAFTING PHASE	71
FIGURE 10 ENORM SCREENSHOT - SYNOPSES	71
FIGURE 11 ENORM SCREENSHOT: EDITING IN A WORD PROCESSOR INTERFACE	74
FIGURE 12 ENORM SCREENSHOT - ANCHOR LINKS	75
FIGURE 13 ENORM SCREENSHOT: STRUCTURE VIEW	76
FIGURE 14 LEX DANIA EDITOR INTERFACE SCREENSHOT	83
FIGURE 15 LEX DANIA XML SCHEMA STRUCTURE	85
FIGURE 16 STRUCTURE OF AN AMENDING ACT IN LEX DANIA	86
FIGURE 17 STRUCTURE OF AN ANNEXE IN LEX DANIA	87
FIGURE 18 LEX DANIA EDITOR – VIEW DOCUMENT WITHOUT MARK-UP TAGS	90
FIGURE 19 SCREENSHOT NL OFFICIAL PUBLICATIONS OF 29/02/2012	97
FIGURE 20 SCREENSHOT OF ONE OFFICIAL PUBLICATION – WEB VERSION	97
FIGURE 21 SCREENSHOT OF ONE OFFICIAL PUBLICATION – PDF VERSION (AUTHENTIC VERSION)	97
FIGURE 22 STANDARD TRANSFORMATION – EC AND EP DOCUMENTS	103
FIGURE 23 SCREENSHOT NORMA EDITOR DOCUMENT STRUCTURE	109
FIGURE 24 XMLEGESEDITOR SCREENSHOT	112
FIGURE 25 XMLEGES SCREENSHOT: IMPORTING A DOCUMENT	113
FIGURE 27 XMLEGES SCREENSHOT – NOTES SECTION	114
FIGURE 26 XMLEGES SCREENSHOT – STRUCTURE ERROR	114
FIGURE 28 XMLEGES SCREENSHOT IDENTIFICATION OF STRUCTURE ISSUES	116
FIGURE 29 XSMLEGESEDITOR ARCHITECTURE	117
FIGURE 30 BUNGENI EDITOR SCREENSHOT USER INTERFACE	122
FIGURE 32 BUNGENI SCREENSHOT EDITING FUNCTIONS	123
FIGURE 33BUNGENI SCREENSHOT MARK-UP FUNCTIONS	124
FIGURE 33 SCREENSHOT .ODT FILE CONTENT	126
FIGURE 34 BUNGENI SCREENSHOT OF THE SETTINGS DATABASE CONFIGURATION EDITOR	130
FIGURE 35 BUNGENI EDITOR ARCHITECTURE	131
FIGURE 36 BUNGENI EDITOR PROJECT ISSUE TRACKER	133
FIGURE 37 BUNGENI COLLABORATIVE DEVELOPMENT ENVIRONMENT SCREENSHOT	134
FIGURE 38 SOLON FILE MANAGEMENT SCREENSHOT	138
FIGURE 39 SOLON SCREENSHOT: CREATING A DOCUMENT	139
FIGURE 40 SOLON SCREENSHOT: TRANSPOSITION OF AN EU DIRECTIVE	139
FIGURE 41 SOLON SCREENSHOT USER INTERFACE – UPDATING A DOCUMENT	140
FIGURE 42 SOLON SCREENSHOT VERSION MANAGEMENT	140
FIGURE 43 SOLON SCREENSHOT UPLOAD OF A DOCUMENT	141
FIGURE 44 SOLON SCREENSHOT INVALID FILE	141
FIGURE 45 SOLON SCREENSHOT FOLDER METADATA	142
FIGURE 46 VEXPRO SCREENSHOT MENU BAR INTERFACE	147
FIGURE 47 VEXPRO SCREENSHOT - CREATING A DOCUMENT	148
FIGURE 48 VEXPRO SCREENSHOT METADATA VIEW	149
FIGURE 49 VEXPRO SCREENSHOT REFERENCE LINK VALIDATION	150

FIGURE 50 VEXPRO SCREENSHOT CONTEXTUAL GUIDANCE ELEMENT PROPOSAL	150
FIGURE 51 VEXPRO: GENERAL LOGIC AND ARCHITECTURE	151
FIGURE 52 WETSEEDITOR SCREENSHOT: CONSOLIDATED TEXT SHOWING AMENDMENTS	154
FIGURE 53 WETSEEDITOR SCREENSHOT: VIEW OF A BILL IN VARIOUS STAGES OF THE DRAFTING PROCESS	155
FIGURE 54 WETSEEDITOR: TECHNICAL MODEL FOR GENERATING AUTOMATIC FORMULATION OF AMENDING INSTRUCTIONS	156
FIGURE 55 WETSEEDITOR SCREENSHOT: HOMEPAGE	157
FIGURE 56 WETSEEDITOR SCREENSHOT: VIEW OF TEXT TO MODIFY	158
FIGURE 57 WETSEEDITOR SCREENSHOT: EDITING A CHAPTER	158
FIGURE 58 WETSEEDITOR SCREENSHOT: AMENDING A TEXT IN THE EDITOR	159
FIGURE 59 WETSEEDITOR SCREENSHOT: VIEW OF THE TEXT OF LAW WITH THE CHANGES MADE.	160
FIGURE 60 WETSEEDITOR SCREENSHOT: AUTOMATIC FORMULATION OF THE AMENDMENT	161
FIGURE 61 WETSEEDITOR SCREENSHOT: VIEW OF THE INITIAL TEXT, THE COMPLETE NEW TEXT, AND THE EDITORS	161
FIGURE 62 AT4AM SCREENSHOT: EDITABLE PARAGRAPHS	162
FIGURE 63 AT4AM SCREENSHOT: DOUBLE COLUMN AMENDING	163
FIGURE 64 AT4AM SCREENSHOT: AMENDMENT GENERATED	164

## Tables

TABLE 2-1 SOLUTIONS OVERVIEW	16
TABLE 2-2 OVERVIEW - TOOLS AND TECHNOLOGIES	19
TABLE 3-1 USAGE AND USERS	22
TABLE 3-2 TYPE OF TEXT PRODUCED	23
TABLE 3-3 LEVEL OF CONSTRAINT, INFORMATION STRUCTURE, INTEROPERABILITY	24
TABLE 3-4 EDITING AND VERSION MANAGEMENT FUNCTIONALITY	27
TABLE 3-5 CHECKING FUNCTIONALITY	28
TABLE 3-6 METADATA, ARCHIVING, SIGNING, IMPORTING FUNCTIONALITY	29
TABLE 3-7 FORMATS USED	31
TABLE 3-8 TYPES OF CONTENT	32
TABLE 3-9 MULTILINGUALISM SUPPORTED	33
TABLE 3-10 OTHER FEATURES	34
TABLE 3-11 RE-USABILITY	35
TABLE 3-12 ADVANTAGES OF THE SOLUTIONS	37
TABLE 3-13 DISADVANTAGES OF THE SOLUTIONS	39
TABLE 3-14 FUTURE DEVELOPMENTS	41



# 1 INTRODUCTION

## 1.1 CONTEXT

Drafting a new legislation is a complex process, involving many actors. In the case of legal texts,

- There is no overall view of the way Member states address that question, with a view to share best practices
- The EU and national legislations are more and more interconnected (for instance when Member states implement EU directives or when National Parliaments issue opinions on the Commission proposals) and Commission legal texts could be re-used in these entities
- All administrations are facing the question of reconciling the freedom associated with Office automation tools and the necessity to produce structured documents at the end of the process;
- the creation step is only the first one in a process leading to the publication and the entry into force of the laws;
- texts might need to be reviewed by other internal or external entities, using different techniques and tools;
- the texts have to respect certain presentation rules and canvasses;
- there are XML (eXtensible Markup Language) standardisation activities aiming at defining a generic model for a law (like MetaLex);

Therefore, a **better understanding** of the way public administrations address the issue of drafting their legislations is useful, with a view to identify **best practices, improvements and areas where common efforts and developments are possible**, a study is made of the existing situation and planned/ongoing projects in Member States and EU administrations.

The aim of this study is to identify trends of existing practices and planned initiatives in Member States regarding legislative editing software, addressing formats, tools, content, overall functionality, as well as the structure of the document with a description of the level of granularity with which the data entry is requested/ possible. The analysis also includes an overview of on-going standardisation efforts (Metalex for instance) to define a common framework for the structure of legal texts.

## 1.2 AIM OF THE DOCUMENT

This document D2.2 Final Analysis Results is produced by TRASYS in the scope of the specific contract 004422 of framework contract DI/6772 in between STRATIQO and DIGIT, and Task 2: Feedback analysis. It presents analysis results covering initial findings on the “As Is” of usage methods and tools for editing legislative software, and lessons learned in terms of adequacy

and efficiency of these tools (deliverable D2.1 Initial Analysis Results – included in this document in **sections 2 to 5**). In **section 6** it presents a series of six case studies analysing some of the solutions identified in the initial analysis, their benefits and trade-offs, re-usability, as well as aspects regarding user-guidance and user friendliness. **Section 7** presents overall conclusions and recommendations.

## 1.3 METHODOLOGY

### 1.3.1 Collection of information

The mapping of the As Is situation is not intended to be an exhaustive inventory, but is based on a voluntary “share and re-use” approach among Member States and European Institutions. In order to provide a picture including a large range of actors and reflecting practices and methods of various areas in the complex legislative drafting process, the approach for identifying stakeholders involves is broad and open, using various networks and identification methods, described in D1.1 List of Contacted Stakeholders. The study involves stakeholders of legislative editing software with various profiles and from various organisations.

The list and short description of the solutions analysed are presented in section 2.1.

Information regarding the solutions is gathered through a questionnaire. The elaboration of the questionnaire, its draft and final versions, is the scope of task 1 of this study, and is described in D1.2 Final Questionnaire. The questionnaire is divided in 4 main sections:

- Description of type of input and the format used
- Description of the structure/fields of information input
- Description of the tool
- Advantages and disadvantages of the current solution, future plans.

The questionnaire used **open questions**, which allowed collecting a wider range of input than if the questions had been closed (and pre-defined). This however implies that in the summary tables presented below, some features may not be mentioned as available in a solution even if they are. They may be implicit or not mentioned because not identified as characterizing the most the solution. The tables in this report need to be read as **descriptive**, and not as an exhaustive taking-stock of functionalities, features and technologies.

Additional information regarding standards and other expert initiatives in the field of legislative text is done through desk search and workshop organisations with experts.

The case studies are elaborated based on interviews conducted with specific experts, developers and project managers of the solutions, desk research of existing documentation such as user manuals and technical documentation, and conducting of workshops about functionality of the tools and existing standards. Guidelines for interview questionnaires and workshop agenda were created by TRASYS, in collaboration with DIGIT, based on the specific topics which the team wanted to address for each case study.

The recommendations are elaborated based on findings from the initial analysis, the case studies, and expert consultation through a workshop organised by TRASYS and DIGIT at the European Commission on 24/02.

## 1.3.2 Content of the Final Results

### 1.3.2.1 Initial Analysis

**Section 2** presents the different types of solutions for legislative drafting, an overview of the tools analysed in this study, and which tools and technologies are used.

**Section 3** provides a detailed analysis of the data provided by the respondents in the questionnaires. This leads to analysing the solutions through a mapping of various characteristics:

- Type of text produced
- Level of constraint on user's input, level of structure of information, interoperability
- Analysis of Functionality (Editing, Checking, Metadata, archiving, signing, importing  
Formats, Content, Multilingualism )
- Re-usability of the solution
- Advantages and disadvantages
- Future developments

**Section 4** presents and analyses ongoing efforts and outcomes for common frameworks and standards for structure of legal texts. It presents an emerging international standard (the document oriented XML standard Akoma Ntoso), as well as an interoperability standard CEN/MetaLex, and a comparison between both. It also presents briefly the results of the use of XML in parliaments from the World eParliament Report 2010, the XML schema used in the Publication Office, the results of existing studies on the use of XML in EU Member States and on the need for common structures. The section ends with an overview of the IPEX schema and of National schemas used in Europe.

**Section 5** presents the conclusions of the Initial Analysis.

### 1.3.2.2 Final Analysis

**Section 6** presents a series of case studies, analysing benefits and trade-offs of different types of solutions, functional and technical aspects of user guidance, as well as re-usability of the solutions.

- Case study 1 covers an analysis of structure management in two tools based on word processors: the eNorm/xNorm case study analyses a framework for applying structures and the Lex Dania case study analyses an approach to imposing a structure. The benefits and trade-offs are also analysed.
- Case study 2 addresses XMLisation (or mark-up) methods and the need for them. It details the Dutch publication XMLisation process at Sdu publishers. It presents the XMLisation

process for European Parliament amendments. It explains the way Norma Editor and xmLeges – two desktop tools - provide mark-up to a legal text.

- Case study 3 analyses Bungeni, an open source legislative editor based on the word processor Open Office, and exporting to Akoma Ntoso XML standard. It covers how the tool applies the structure, the user guidance provided, the technical aspects of adding document types,
- Case study 4 presents SOLONII, the new French system for managing the process of legislative texts from creation to publication, based on Open Office and an open source document management system Nuxeo.
- Case study 5 covers vexpro, an open source desktop XML editor developed for legislative drafting in Estonia.
- Case study 6 presents Wetseditor, a Web XML editor used in the publishing process in the Netherlands, and introduced recently in Parliament for amendments, and covers the user benefits, as well as change management aspects.

**Section 7** draws conclusions from the overall study and provides recommendations regarding the next phase of the LEOS project.

## 2 SOLUTIONS OVERVIEW

### 2.1 INTRODUCTION

#### Legislative drafting tools - requirements

Understanding what requirements a legislative drafting tool should meet – “ideally” – can be described in part as follows:

- User friendly interface and possibility to be used by general user (legislative author) without specialize IT knowledge
- Structuring of legal documents according to drafting rules and possibility to check of formal and legislative validity.
- Structure (schema) based on standards to allow maximum of interoperability.
- Covering needs of legislative process from initial draft to promulgation.
- Change management including drafting in consolidated version and (semi)automatic consolidation legislative texts.

Other existing studies have similar conclusions, regarding requirements for the successful implementation of **systems covering the entire process from drafting through to publication<sup>2</sup>**:

- Techniques that are easy to understand and learn
- Text manipulation software that is easy to use
- Editors which can produce simple and difficult texts reliably in content and layout and which can be validated automatically
- It must allow both simple and complicated text with the same instruments to be worked upon (text, tables, graphics, pictures etc.)
- A general, well known data format that should be used in text process-systems by Parliament, Government officials and anybody who is to be consulted on the draft legislation.
- The method of operation and instruments to be used should be widely available ensuring that all people involved in the process can work with the same working technique and the same instruments.
- It must bring the same results both online and offline
- It must support structural elements within which the working documents can be turned into publications automatically (e.g. automatic conversion into XML)
- It may need to support more than one language so that the integrity of multilingual text issues can be guaranteed (content, structure and meaning)
- It must be capable of delivering outputs which end users can be secure in knowledge are authentic reproductions of the laws which have been passed by Parliament or adopted by Ministers.
- It should enable full version control of the drafts.

---

<sup>2</sup> See in Annexe, 8.2 Final Report of the Working Group “Data capturing at the source” - European Forum of Official Gazettes

A survey on systems for managing bills in parliament<sup>3</sup> identified a number of characteristics to be responsive to the needs of the members and staff of parliaments. Not all systems analysed had implemented these features in 2010. They include:

- Workflow and the ability to control versions
- Accommodations of all versions of bills
- Exchange and integration of documents and information (to have the complete legislative history of an act)
- Accommodation of bills with special formats.

### A first overview

A first overview of the solutions actually implemented in Member States and in the European Parliament (see Table 2-1 solutions overview) shows **a wide diversity in usage, coverage of needs and technology used**.

Some solutions – not all - provide fully integrated systems from the initial drafting to the publication. The drafting and publishing is linked to the use of a structure (XML schema) for some, others do not use such a schema for structuring information input. Some solutions are custom developments based on off-the-shelf software (office tools or XML editors), others are complete custom development.

Features of these solutions vary therefore in terms of functionalities, but also user-friendliness, efficiency and quality of the text, the need for additional resources to check texts after drafting, re-usability of data, etc.

## 2.2 ANALYSED SOLUTIONS OVERVIEW

[Tool. COUNTRY] <sup>4</sup>	Description
<b>Šablony.CZ</b>	Templates using MS Word macros (Šablony pro vypracování návrhu právního předpisu)
<b>PTJ.FI</b>	Use of MS Word document templates, which will in 2012 be replaced by a system of producing government's decision making documents by XML-editor. Parliament and Government work currently on different basis and with different tools. The plan is that from autumn 2012 they will be using common structures in document files and both systems will be integrated.
<b>.UK-N.le</b>	Customised templates in MS Word are currently used. However, the National assembly is in the process of procuring a new system.

<sup>3</sup> See Annex section 8.1 Features of systems for managing bills in parliaments- World eParliament Report 2010

<sup>4</sup> The country code simply refers to the country where the solution is in use. Other solutions may also be in use in the same country.

<b>BAS.UK-Sc</b>	Customised templates in MS Word are currently used.
<b>.LT</b>	<p>Word-processing software for the preparation and maintenance of the texts: MS Word (versions: 2000, 2003, 2007 and 2010).</p> <p>Application tool for input and maintenance of metadata and upload of text documents into the Oracle database with validation: the database application system TAIS (Teisės aktų informacinė sistema – Information System of Legal Acts – ISLA) based on Oracle Forms 10g.</p>
<b>.CH</b>	MS Word-based templates are used to draft the text. In a next phase, the text is reviewed and validated by jurists and typographers and language services. Metadata is stored in a database. The system provides a workflow.
<b>eNorm.DE<sup>5</sup></b>	eNorm is based on LegisWrite, adapted to new requirements. It provides support for compliance with formal legal and editorial standards in the preparation of draft laws and regulations and also automates certain routine tasks. The documents are created with eNorm in the legislative process from first draft to the discussion and dissemination of documentation. The software will gradually be developed. The final objective is to use open and structured data formats – XML - Xnorm. Some modules are open source <sup>6</sup> .
<b>AF.UK</b>	FrameMaker is a word processing package which can handle very large, structured documents (such as Bills) accurately. Customised plug-ins have been developed.
<b>eRecht.AT</b>	Legal texts pass through a continuous electronic production channel, from the initial draft of a bill to its authentic publication on the internet. The creation of electronic texts follows layout guidelines. MS Word-based templates facilitate the structuring of texts and the layout design for the Federal Ministries. 65 paragraph formats and 11 character formats are available. Specially defined toolbars help to assign the formats. Additional functions allow for a more comfortable editing.
<b>LexDania.DK</b>	Lex Dania editor is a .NET-based add-in for Microsoft Word which helps and guides the user to write LexDania xml documents in an easy and user friendly way.

<sup>5</sup> [http://www.enorm.bund.de/enid/bf6706c781b1ed5a8ae52382a46c5c24,0/eNorm/Projekt\\_32.html](http://www.enorm.bund.de/enid/bf6706c781b1ed5a8ae52382a46c5c24,0/eNorm/Projekt_32.html)

<sup>6</sup> [http://www.bmj.de/DE/Recht/Justizverwaltung/JustizariatRechtsinformationenGeheimchutz/eNorm/\\_doc/Projekt\\_eNorm\\_Open-Source.html](http://www.bmj.de/DE/Recht/Justizverwaltung/JustizariatRechtsinformationenGeheimchutz/eNorm/_doc/Projekt_eNorm_Open-Source.html)

<b>LWB.IE</b>	The Legislative Workbench (LWB), is a set of business applications that support the running of legislatures. LWB includes full document and workflow management, automated amendment engrossment and true in-context and point in time archiving. LWB uses .odt. They are about to install the new version of the system (plan to go live on 1 January 2012), and replied in this context.
<b>LWB.UK-Wa</b>	Id above
<b>VexPro.EE</b>	Since June 1, 2010 the State Gazette is not printed on paper any more. On November 2, 2010 a new system version of the State Gazette was made available. The system only publishes XML format documents. XML VexPro tool was introduced at the same time. This friendly tool enables the user to create XML documents without knowing XML. So far, the VexPro tool is used for creating consolidated acts of law.
<b>xmLegesEditor.IT</b>	xmLeges <sup>7</sup> : XMLLeges Editor is a visual XML editor developed by ITTIG-CNR (Institute of Legal Information Theory and Techniques, Italian National Research Council). It provides an integrated solution for legal texts drafting. The current version is able to produce documents according to NIR DTD (XMLSchema), URN and Metadata legislative standards. The core platform can be easily adapted to implement different xml standards. The environment has the aspect of a common text-editor; it doesn't require any technical knowledge about standards, since it hides all the xml details to the final user.
<b>Wetseditor.NL</b>	Wetseditor is a solution for editing law texts based on the OP XML schema, using a web based interaction. It is currently used in publication and for automatically generating changes in law texts, and is in a pilot phase at Parliament for text drafting. It is custom made, based on the Xophus XML editor.
<b>AT4AM.EP</b>	AT4AM produces amendments at the European Parliament, with a web based user interaction and an XML-based content management.

**Table 2-1 solutions overview**

<sup>7</sup> <http://www.xmlleges.org>



## 2.3 TOOLS & TECHNOLOGIES USED

Legislative drafting tools are grouped in **three main categories**<sup>8</sup>:

### ● Text editors based on word processors

These editors use word processors already on the market (Microsoft Word, Open Office, etc.) use internal format (e.g., ODT, DOC, DOCX, etc.), with custom development such as add-ons, macros etc...

Additionally, they can provide semantic check of the document, convert and validate it according to the chosen XML schema and import and export XML files.

### ● Editors based on a native XML editor

These editors are guided by the rules of the standard XML schema and can at all times check for compliance with such rules. These editors are based on validated XML that can be open source or proprietary.

### ● Web-based editors

There are text editors on the Web. The idea of writing a text, sharing it with others, and handling plural Web applications is encouraging and has led some developers to design software for the XML mark-up of legislative texts, too.

For solutions using XML, the back-office tools include the following:

- Specialized text editors for XML mark-up
- Converters for converting documents coming from legacy systems or different standards using
- Databases, for storing the marked up texts and for managing and verifying each document's different versions
- Web-integrated platforms, for managing parliamentary workflow

The front-office tools are the following:

- URI/Link Resolver, enabling citizens and external applications to connect to Parliament's digital document resources.
- XML consultation websites, with texts having semantic search forms.

A **mapping of tools and technologies** used in the solutions analysed (see Table 2-2) shows the three types of tools are in use by the different stakeholders participating in this study:

---

<sup>8</sup> Monica Palmirani - CIRSFID – University of Bologna, Law Faculty LEOS workshop - Summer School LEX September 2011

## ● Text editors based on word processors:

Not using XML: (or only for export): Šablony.CZ, PTJ.FI, UK-N.le, BAS.UK-Sc, .LT, .CH, eNorm.DE\*

Šablony.CZ
PTJ.FI
UK-N.le
BAS.UK-Sc
.LT
.CH
eNorm.DE

\* Note: eNorm documents can be validated, imported/exported in the XML schema xNorm from beginning 2012 (after the initial analysis was finished). See the xNorm/eNorm case study in this document. Before that date, xNorm, eNorm only exported in XML, but there is a high degree of structure in eNorm, which makes it different from the other type 1 tools.

Using XML: AF.UK eRecht.AT LexDania.DK LWB.IE LWB.UK-Wa

AF.UK
eRecht.AT
LexDania.DK
LWB.IE
LWB.UK-Wa

AF.UK is a template-based authoring and publishing solution for unstructured, structured and XML/DITA content.

## ● Editors based on a native XML editor:

VexPro.EE xmLegesEditor.IT  
Wetseditor.NL

VexPro.EE
xmLegesEditor.IT
Wetseditor.NL

## ● Web-based editors:

AT4AM.EP Wetseditor.NL.

Wetseditor.NL
AT4AM.EP

	Sablony.CZ	PTJ.FI	UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	Wetseditor.NL	A T4A MEP
Tools & technologies used																
MSWord	X	X	X	X	X	X	X		X	X						
OOo / Libre Office											X	X				
Framemaker								X								
add-in/ plugins / macros	X	X	X	X		X	X	X	X	X	X	X				
templates	X	X	X	X	X	X	X		X	X	X	X				
editing in XML (*)								X	X	X	X	X	X	X	X	X
XML editor													X	X	X	
WEB							X		X	X					X	X
Workflow						X			X		X	X				
Databases					X	X	X			X	X	X	X		X	
Oracle Application Server (Forms)					X											
SVN											X	X				
batch jobs						X										
XSL													X	X		
workbench: files, editor,											X	X				
OLE					X											
Legacy content parsers: editor plug-in														X		
Supported standards																
DTD							X				X	X		X		
Schema							X	X	X	X			X	X	X	
Akoma-Ntoso																X
Metalex														X		X
URN:LEX														X		

(\*) excl. export to XML.

**Table 2-2 Overview - tools and technologies**

For the purpose of this study and analysis of the solutions, the tools can be classified according to **three categories** (color codes in the table light grey, medium grey and dark grey):

- The first category (light grey) covers solutions using text editors based on word processors and which do not use XML (one solution exports to XML and supports a structure).
- The second category (medium grey) covers solutions using text editors based on word processors and which use XML (import and export) and a schema/DTD. The solution based on Adobe Framemaker (AF.UK) is slightly different but is included in this category rather than in the next category.
- The third category (dark grey) covers other solutions using XML (XML editors, Web editors).

### Comments

- Almost 70% of the solutions are based on word processors (MS Word, Open Office).
- Only 2 countries use a solution based on Open Office (the same solution).
- Almost 70% of the solutions use XML.
- All the solutions using XML use a standard schema or DTD.
- 25% use a web interface at some point in time.



- 25% implement a workflow.
- All the solutions based on word processors use custom made templates.

### 3 ANALYSIS OF “AS IS” OF EXISTING PRACTICES

This section analysis more in detail the different solutions, based on replies collected through the questionnaire. The questions were formulated as open questions. Therefore, in the summary tables presented below, some features may not be mentioned as available in a solution even if they are. They may be implicit or not mentioned because not identified as characterizing the most the solution. The tables need **to be read as descriptive**, and not as an exhaustive taking-stock of functionalities, features and technologies.

Reminder:

- Type 1 solutions: using text editors based on word processors and which do not use XML . (or only export for publishing).
- Type 2 solutions: using text editors based on word processors and which use XML for editing.
- Type 3 solutions: other solutions using XML (XML editors, Web editors).

#### 3.1 USAGE AND USERS OF THE SOLUTIONS

This section maps and analyses, for each solution/type of solution, data regarding their usage and users. See Table 3-1.

##### Usage

- Most of the solutions analysed are used in Parliament (90%), many are used by Government (80%). Half are used in Publications.
- 70% of the solutions are used in both Parliament and Government.
- Nearly 40% of the solutions are used in Parliament, Government and Publications. None of them are type 3 solutions; this could be linked to the fact that they are rather newly implemented and are being in pilot phases of implementation.

##### Number of users, years of use

The data shows that:

- There tends to be more users for solutions which have been in use for a long time
- Solutions from Type 1 have been in use for a long time (from 10 to 25 years)

*Note:* Wetseditor.NL is in pilot phase in the Parliament, but in full use in publications.

##### Users, expertise needed for using the tool

- All users need basic IT literacy skills and knowledge of drafting of legal texts.
- Almost half of the solutions are qualified as needing advanced training (more than half a day/ one day). It is not explicit that solutions from type 2 and type 3 need more training/expertise.

However, other studies show that level of training/expertise is high (see section 4.4.2 Challenges for using XML) – but this could be linked to the need for change management, more than specific expertise, as users have been used to a specific interface.

- Among the solutions needing advanced training, some solutions need specific expertise (up-markers for LexDania.DK, specially trained users of the database application software for .LT).
- All solutions are used by public officials taking part in normative or law formulation, and/or secretary staff/clerks.
- 2 solutions are actually used only by clerks/administrative staff. These are Type 1 solutions.

Users																
Lawyers / Legislative specialists/ Public officials taking part in normative or law formulation	1		1	1	1	1		1	1	1	1	1	1	1	1	1
Secretary staff/ Assistants / Clerks	1	1		1		1	1	1	1	1		1				
Typographers						1										
Linguists						1										
Specially trained users of the database application software					1											
Translators												1				
Up-markers / Pre-press										1						
MEPs																1
Secretariats of parliamentary committees /political groups						1										1
Specialised users in the preparation of consolidated texts													1			
Anyone importing existing texts and complying to schema														1		
Expertise																
all users - short training provided	1		1	1	1	1	1		1					1	1	1
Specific expertise - advanced training		1			1		1	1			1	1	1			
Years of use																
	10		12	12	17	12			7	2		0,2	1		1	2

**Table 3-1 Usage and users**

## 3.2 TYPE OF TEXT PRODUCED

This section maps for each solution/type of solution what type of text is produced. See Table 3-2.

The respondents to the questionnaires provided a wide range of responses, from a very detailed list of types of documents to an overall “legal texts and other texts” or “all types of legislative and normative texts”. Types of legislative and normative texts are linked to a country’s legislation, therefore aggregating/ categorising the type of documents for a further analysis was not done. The list of types of documents in the table below is descriptive and therefore informative.

What comes out of the questionnaires is often a solution which *can* be used for many types of documents, but which is *actually not used* by many different administrations, such as eNorm, or Wetseditor (used in publications, and will be implemented “up” the complete drafting workflow

process). Some solutions are very specific, such as AT4AM.EP which covers amendments of 3 types of texts.

	Šablony.CZ	PTJ.FI	.UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.LAT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xml.LegesEditor.IT	WetseEditor.NL	AT4AM.EP
<b>Type of text produced</b>																
Law text	1					1	1		1	1		1		1		
Report of law text						1			1							
Primary legislation			1													
Secondary legislation (regulation, ordinance)	1						1			1						
Government document proposal	1						1									
Regulatory Impact Assessment report	1															
consultation papers/ communications/ notifications						1										
government proposal to Parliament, communication, report, decree, resolution, statement		1														
all types of legislative and normative texts					1											
Parliament Bills				1												
amendments				1				1								
parliamentary communication							1									
Union communication		1														
act		1												1		
treaty		1					1									
decree of the President on the Republic		1														
Normative text documentation							1									
all parliamentary documents connected with legislation										1						
ombudsman										1						
legal texts and other texts															1	
comments															1	
Consolidated texts of the acts of law, regulations and orders, resolutions													1			
Consolidated texts							1									
Amendments on proposals made by EC																1
Amendments on European Council Position																1
Amendments on EP non legislative reports																1
Amendments and amendment lists											1	1		1		
Changes in law texts															1	
Bill and Documents, Order Papers, Journals											1					
all documents associated with the introduction and passage of Bills								1								
hansard														1		

**Table 3-2 Type of text produced**

### 3.3 LEVEL OF CONSTRAINT ON USER'S INPUT, LEVEL OF STRUCTURE OF INFORMATION, INTEROPERABILITY

This section maps for each solution/type of solution, how the structuring of information is managed. See Table 3-3.

#### Level of constraint imposed by the tool

The respondents qualified the level of constraint imposed on the user by the tool as either:

- low = the user/writer may enter text or provide other content in a rather free manner, and the legislation draft goes through a correction/validation phase after;
- high level of constraint = the ways of providing input are pre-defined and the writer needs to conform to strict rules.

As can be foreseen, all Type 1 solutions have a low or non-existent level of constraint on the data input. Some respondents in this category mentioned “high” when meaning that the data *should* be entered *following strict rules* linked to the templates, for the drafting to be of expected quality.

### Level of granularity of the information structure

The respondents qualified the granularity of the information structure as either:

- Non existing: the structure of the information is “flat”
- a low level of granularity: the predefined structure provides only fields dedicated to global chapters.
- a high level of granularity: the predefined structure allows identifying fields linked to sub-chapters or even numbers of paragraphs.

For type 2 and type 3 solutions, the level of granularity of the structure is high. For Type 1 solutions, almost half have also a high level of granularity.

For some tools, the identification of the different types of **fields** in the structure is **linked to the style** of the text which is encoded in that field. This is the case for all type 1 solutions.

	Šablony.CZ	PT.JFI	.UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	WetSeditor.NL	AT4AMEP
Level of Constraint																
None (0) Low (1) High (2)	1	0	0	1	0	1	1	2	2	2	2	2	2	2	2	2
Structure of information input																
Linked to type of document	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Granularity: None (0) Low (1) High (2)	1		2	1	0	2	2	2	2	2	2	2	2	2	2	2
Linked to style	X	X	X	X	X	X	X		X							
Interoperability																
Export to xml							X		X	X						
Re-use of information by another application							X			X	X	na			X	X
use of XML = potential re-use of all									X				X	X	X	
for solutions not using XML: Metadata in DB					X	X										

**Table 3-3 Level of constraint, information structure, interoperability**

### Interoperability

Respondents mentioned when or whether other applications re-used data entered using their solution.



For Type 1 solutions, this is the case only for eNorm, because the solution exports data to XML. For .LT and .CH, which do not use XML, metadata is exported to the database, but they have typographers and other specialised staff checking/correcting/encoding the data.

For Type 2 and Type 3 solutions, only 4 cases provide data actually re-used by other applications. One case does not need to as it is used also for publishing. The rest mention potential re-use of all information.

## 3.4 ANALYSIS OF FUNCTIONALITY

### 3.4.1 Editing

This section maps for each solution/type of solution, how functionality regarding Editing (including management of changes and different versions) is managed (see Table 3-4).

In general, editing is done from templates. The description of the VexPro.EE tool explains how editing is done in a Type 3 solution:

- Drafting rules based on legislative schemes are integrated into the tool and the template helps the user to select elements of composition (e.g. chapter, title, subtitle, section, paragraph, etc.) Schemas are structured and contain only elements in the proper hierarchy.
- A document is created as follows :
  1. an appropriate type of template is chosen – an empty template opens, with only a title template text. (User input is similar to conventional word processors) ;
  2. from the right panel, the user can choose various types of element templates to be used in the document. These templates help the user to work and visualize what the element looks like and provides the space for text. An element is chosen double clicking on the name and it is added into the template with element number and default text. The user then either types or copies the appropriate text of the document content;
  3. in another panel the whole structure of the document can be seen, and by placing the cursor in the right place on the workspace, the specific schema element in hierarchal structures can be followed at the bottom status bar.
- Elements can be moved to the right place in the document by dragging them in the structure panel, but only to where the elements belong depending on the type.
- Elements can also be deleted, including all the constituents, e.g. a section with all paragraphs, subsections and items, etc.

A screenshot of the XMLeges.IT editor illustrates editing, selecting elements in one panel, and typing/visualising the text in another; see Figure 1.

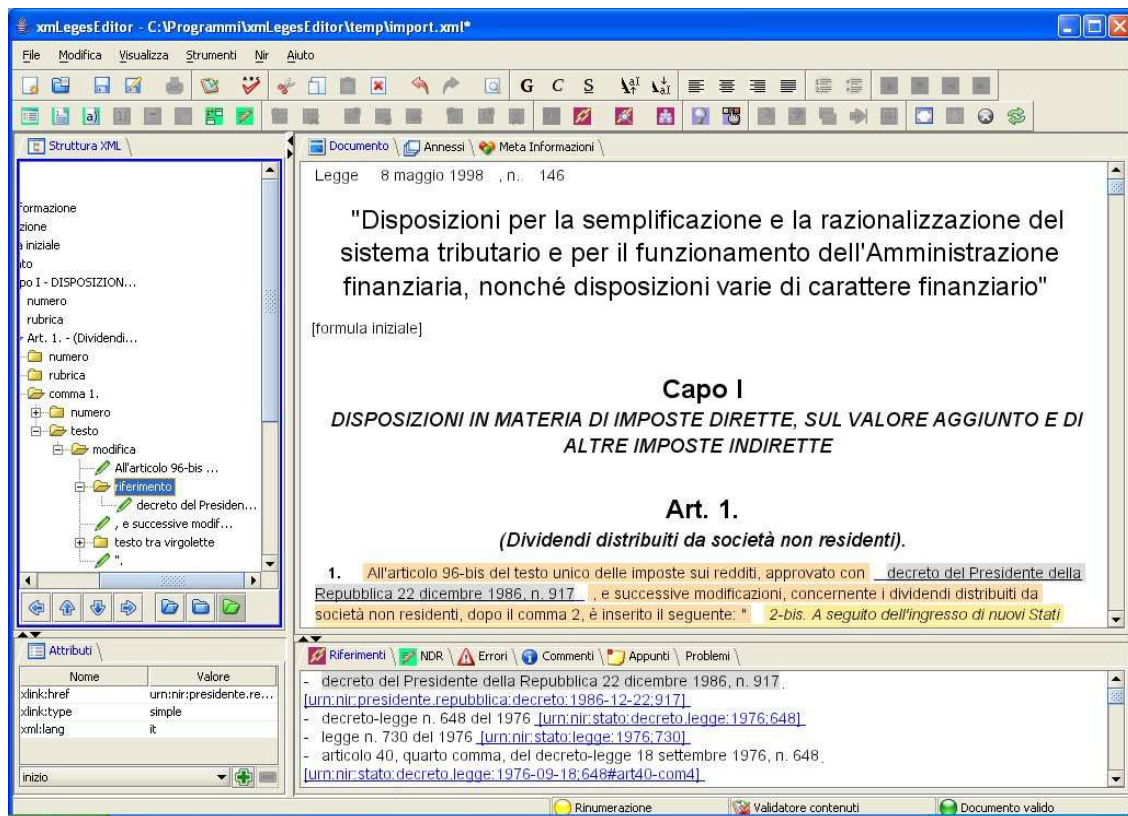


Figure 1 xmLeges.IT screenshot

Additional features provide more **user guidance** for only 3 of the solutions, these are interesting to highlight:

- LexDania.DK: a new document is started from a selection where the user can choose the document type. At that point it is possible to have user group specific document templates, so that the ministry can make their specific alterations within the limits given by the general guidelines/Lex Dania xml schemas. Then there is a form where the user is prompted for information to be used both as data in the document and metadata. And the cursor is then placed at the correct position in the xml structure.
- eNorm.DE: The programme proposes numerous recommendation texts or links to the online manual for drafting legal texts.
- XMLegesEditor.IT: Users are guided by the template itself, by automatic functions, by forms and wizards.

Following the **evolution of a change** in the document is addressed in different ways depending on the solutions.

- Only 3 of the Type 1 solutions follow change, using unsophisticated functionality (track changes of MS word, version control and links in between versions in a database).
- Type 2 and Type 3 solutions do not all provide this functionality, enhancement is foreseen in the future. The Web editors address currently only changes / amendments in law texts.
- Some of these applications focus actually on amendments, and provide advanced features. It is interesting to highlight the following functionality in the Wetseditor.NL tool:

- “You can load the existing law in the application, change the text and then generate a 'change text' in the database. The application also allows to see where each sentence of legislation came from (government, parliament, date, etc., enhancing the democratic process.”

Functionality	Sablon.CZ	PTJ.FI	.UK-NIe	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	WetseEditor.NL	AT4AMEP
<b>Editing</b>																
<b>Create a document</b>																
Create a document	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
From template	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
With user guidance							1			1				1		
<b>Update a document</b>																
Update a document	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1
With user guidance							1						1			1
<b>Track changes</b>																
"track changes" of MS Word						1	1	1								
version-control of the documents									1							
Automatic diffing/ synopsis (comparison between text and display of differences)							1									1
Consolidated version generated							1									
You can load the existing law in the application, change the text and then generate a 'change text' in the database. The application also allows to see where each sentence of legislation came from (government, parliament, date, etc.).															1	
Changes made by individual amendments can also be viewed using track changes, showing text proposed to be removed and text proposed to be added.											1	1				
Automatic parallel text creation of different versions of an amended text														1		
Comparison of different text versions of a bill and automatic view of the related parallel text														1		
Pdf and html export of the parallel text and the related amending text														1		
the final version of the document ("consolidated version") can be obtained. Similarly a multiversion document can be produced, containing document lifecycle information.														1		
Tortoise SVN used to track changes (browser of the version control system)											1	1				
Links between different versions of documents in database					1	1										
Not fully developed yet. [At present, all modified elements in a consolidated text are supplied with a note of updating, which reflects the date of publication of the modifying document [a direct link is also added] and the date of coming into force of the modification]													1			

**Table 3-4 Editing and version management functionality**

### 3.4.2 Checking

This section maps how the solutions or types of solutions check the compliance of the structure. See Table 3-5.

For all Type 2 and 3 solutions, the tool does a structure check. It uses a schema to validate the structure of the data encoded.

For Type 1 solutions, only 2 provide a structure check, one with partial automatic correction.

Functionality	Sablony.CZ	PT.JFI	.UK-NIe	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	Wetseditor.NL	A T4AM.EP
<b>Checking</b>																
<b>Document structure</b>																
<i>structure check by tool</i>	1						1	1	1	1	1	1	1	1	1	1
schema validation							1	1	1	1	1	1	1	1	1	1
Guidance - The Element Catalog : which elements are available at particular locations in the document								1								
functionality for a consolidated text: add amendment notation reuse													1			
possibility to navigate in the document (with highlighting of selected structural elements) and the regrouping of elements (e.g. whole sections with their entire content can be moved or copied into new documents/copied from existing documents)							1									
Recognition and mark up of known structures e.g. articles with subsection is automatically marked up when selected									1	1						
Relocate a single element or a group of elements together with their sub-elements in a structure tree													1			
detected errors can be partially automatically corrected							1									
validation check on demand, suggestion of correct solution	1						1									
<i>structure check by specialists</i>																
the document is validated by typographers and jurists and language services						1										
a posteriori validation after encoding of data by specialists					1											

**Table 3-5 Checking functionality**

Document presentation check is done indirectly through the document structure check when this one is available.

There is no tool which provides some form of legal support / checking of quality of content.

### 3.4.3 Metadata, archiving, signing, importing

This section maps if/how the solutions address metadata, archiving, signing and legacy management (importing of existing legislative or normative texts). See Table 3-6.

Functionality	Sablony.CZ	PTJ.FI	UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	WetseEditor.NL	A T4AM.EP
<b>Metadata</b>																
simple (1) complete (2)						2	2	1	2	1	1		2	2		1
<b>Archiving</b>																
Conversion to an archiving format			1			1		1						1		
Export to xml for archiving									1				1			
Archive of SVN repository											1	1				
Export to .PDF for archiving													1		1	
<b>Signing</b>																
Digital Signature									1	1						1
Unique identifier							1									
<b>Legacy / Importing</b>																
.doc									1							
.PDF									1							
.JPEG, TIFF									1							
Automatic conversion of legacy contents from MSWord, plain text and html formats														1		
Undo/redo function														1		
Automatic detection of the legal references and automatic hyperlinking														1		
Footnotes and attachments management														1		
Forms for metadata implementation														1		
Xml attributes management														1		
Automatic document URN construction														1		
Document export in pdf, html and rtf formats														1		
Specific focused views of meaningful document portions (list of references, notes, etc.).														1		

**Table 3-6 Metadata, archiving, signing, importing functionality**

- More than half the solutions manage metadata.
- Almost half of the solutions either export to a format which can be used for archiving, or convert to an archiving format of the legislative process.
- Only 3 solutions provide digital signing of the document.
- The legacy management was not part of the questions, however respondents mentioned that 2 solutions offered this feature. XmlLegesEditor.IT details the features in the table above.

### 3.4.4 Formats

This section maps which formats are used by the solutions for the draft legislation, the annex, the final version and the publishing of the legislation. See Table 3-7.

- 50% of the solutions can use .doc as a format for drafting.
- 30% of the solutions use only .doc as a format for drafting
- 60% of the solutions can use .docx, .odt, XML, or WordML as a format for drafting.
- Some tools can use up to 6 different formats for drafting.
- The solutions offer a wide range of format types for the annexe, the same as for drafting with additionally .JPEG and .xls.
- Formats for the annexes which are the most in use by the tools are .doc and .PDF.
- Formats for the final version (adopted) which are the most in use by the tools are .doc and .PDF.
- 50% of the solutions can use .docx, .odt or XML as a format for the final version.
- Some tools can use up to 4 different formats for the final version.
- 75% of the solutions use .PDF as publication format.
- 60% use HTML, one uses XHTML as publication format.
- For Type 3 solutions, whereas .XML is used for the drafting, several other formats are used in the other phases of the process.

Functionality	Sablon.CZ	PTJFI	.UK-NIe	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	Wetseditor.NL	AT4AMEP
<b>Formats</b>																
<b>Draft</b>	1	1	1	1	4	1	2	1	2	3	4	4	1	1	1	1
.doc	1	1	1	1	1	1	1		1							
.docx					1		1		1							
XML(WordML)										1						
.odt					1						1	1				
html										1						
.PDF					1					1	1	1				
XML											1	1	1	1	1	1
.FM								1								
.sxw											1	1				
<b>Annexe</b>	1	3	0	1	6	1	2	1	4	3	4	4	3	1	1	0
.doc	1	1		1	1	1	1		1							
.docx					1		1		1							
XML(WordML)										1						
.odt											1	1				
XML											1	1	1	1	1	
.PDF		1			1				1	1	1	1	1			
.xls		1			1				1							
.sxw											1	1				
html										1			1			
.JPEG					1											
.FM								1								
other					1											
<b>Final</b>	2	2	1	1	3	1	3	1	2	1	4	3	2	4	1	0
.doc	1	1		1	1	1	1		1							
.docx					1		1		1							
.odt											1	1				
XML							1				1	1	1	1	1	
.PDF		1	1		1					1	1	1		1		
.PDF/A	1															
.sxw											1					
.akt													1			
.FM								1								
XHTML														1		
.RTF														1		
<b>Publication</b>	4	2	2	0	3	2	2	2	3	3	2	2	2	2	1	0
html	1	1	1			1	1	1	1	1	1	1				
.PDF	1	1	1		1	1	1	1	1	1	1	1		1		
XML										1			1		1	
.rtf									1							
.doc	1				1											
.3B2	1															
.docx					1											
.akt													1			
XHTML														1		

Table 3-7 Formats used

### 3.4.5 Content

This section maps what type of content is supported by the solutions. See Table 3-8.

- Some Type 1 solutions support any type of content supported by MS Word.
- All solutions support text (of course) and tables.
- Most solutions support pictures.
- Half of the solutions support formulas.
- One solution uses .html container to support different types of content.

Functionality	Sablony.CZ	PTJFI	.UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	Wetseditor.NL	AT4M.LEP
<b>Content</b>																
Type of content	8	3	3	8	8	5	10	2	3	6	5	5	8	6	3	4
Text		1	1			1	1	1	1	1	1	1	1	1	1	1
Pictures		1				1	1		1	1	1	1	1	1		
Graphics							1									
Tables		1	1			1	1	1	1	1	1	1	1	1	1	1
Footnotes / endnotes / table notes						1				1			1	1		1
Footnotes							1									
Lists							1							1		
Formulas			1			1	1				1	1				
Typographical faces (bold, italic, underlined, subscript etc.)							1			1			1	1		1
Special characters e.g. Word joiner										1			1			
Legal act appendix (.PDF)													1			
header and footer							1				1	1				
html container													1			
annotations/ comments							1								1	
Any	8			8	8											

**Table 3-8 Types of content**

### 3.4.6 Multilingualism

This section points out which solutions support multilingualism, and how. See Table 3-9.



Multilingualism															
Support of parallel versions					1	1					1	1			
supports management of documents in all EU official languages															1
Functionality	Sablony.CZ	PTJFI	.UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	Wetseditor.NL
															AT4M.EP

**Table 3-9 Multilingualism supported**

- 5 solutions support multilingualism, 4 of them address this by supporting parallel versions. It is interesting to highlight the following:

- LWB: The system was designed for the production of translations at the end of the process. The XML document is broken into chunks and worked on separately by several translators before being reintegrated.
- The practice now, however, is to begin the process when the Bill is introduced. Then, following the reintegration of the Bill as translated, amendments made during the passage of the Bill are introduced and final edit carried out. The reason for this is to create more time in the process to allow the organisation to meet its obligations re the simultaneous publication of both language versions of the Acts on the passage of the enacted version

- VexPro.EE: Multilingualism will be supported in the future. Currently, the content is not translated at the same time of writing. Translations are made by using the Trados software which also allows to maintain the XML structure [at present not used, planned to be done in future].

### 3.4.7 Other features

An open question addressed “other” functionality which respondents used to highlight features worth mentioning. See Table 3-10.

Functionality	Sablon.CZ	PTJFI	UK-Nle	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LW.BIE	LW.B.UK-Wa	VexPro.EE	xml-egesEditor.IT	WetSeditor.NL	AT4All.EP
<b>Other</b>																
Internal references (internal links produce dynamic relationships between document parts )										1			1			1
internal cross-references are robust, enable navigation between the reference and its target back and forth, and can be updated automatically							1									
References to other legislation and metadata, links.										1			1			1
robust cross-referencing				1												
Internal references management and updating after text moving														1		
Form for external references and related URN construction														1		
auto-renumbering between different documents				1												
production of text in multiple component parts and then later assembly as a single Bill				1												
Table of content automatically generated							1									
Document tree describing the structure of the documents, with the possibility to edit the structure of the document combined with automatic renumbering of articles										1						
Transformation of documents from one state to the next e.g. passed bill to act										1						
Web service interface with parliament										1						1
Instant preview of documents										1			1	1		1
Add a comment							1						1			
Insert a reference, a note on the technical rules of legislative drafting, an explanatory note, an appendix, error correction or a signature (text field)													1			
Order of amendments																1
Rendering in WordML																1
Intelligent pasting preserving allowed styles and character/paragraph attributes.										1			1			
Automatic implementation of metadata (URI, publication date etc.);														1		
Integration and correction of automatic document tagging														1		
Automatic numbering of partitions													1	1		
The "structure check" allows moving whole blocks of content, which are automatically renumbered.							1									
Elements are automatically numbered													1			
Document sections management														1		
Find/replace														1		
do/undo							1									
templates contain hints about legislative drafting and links to the German legal guide							1									
quality check provides links to the appropriate action in the guide for identified errors							1									
Dates management														1		

**Table 3-10 Other features**

- Internal and external referencing are mentioned often;
- Automatic numbering is also mentioned often;
- Instant preview of documents is mentioned for the Type 3 solutions;

- Adding a comment is an interesting feature, considering e-democracy aspects.

### 3.5 RE-USABILITY OF THE SOLUTION

This section analyses the re-usability of the solution, taking into account which elements of the solution could be re-used (based on input from respondents) and the type of license (proprietary or open source). This leads to the Table 3-11 Re-usability.

	Šablony.CZ	PTJ.FI	UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor.IT	Wetseditor.NL	AT4A.MEP
Reusability of software	0	0	0	0	0	0	1	2	0	0	2	2	4	3	2	2
partly (2)																2
partly with customisation (1)							1							1		
completely with customisation (2)								2			2	2	2		2	
open source license (2)													2	2		

**Table 3-11 Re-usability**

Among the solutions available under an open source license,

- VexPro.ee solution can be re-used with customisation, according to the reply to the questionnaire
- Part of xmlLegeseditor.IT can be re-used, with customisation

The architecture of the xmlLegesEditor is organized into two software layers:

1. xmlLegesCore, which provides support to the management of generic XML documents based on both XMLSchema or DTD; and can be reused as a platform on which developing a customized solution for a different XML standard.
2. xmlLegesNIR, which provides specific functionalities able to implement the URN and XML Italian legislative standards defined within the NormeInrete (NIR) project.

This architecture allows you to adapt the system to deal with different schemas by developing a standard specific layer (layer 2), on a common platform (layer 1).

The interface is in Italian, and would need translation. User interface configuration settings are stored in an external XML file, therefore it can be adapted to different languages.

As regards the efforts needed, it depends on the complexity of the target standard to implement. However, whatever the standard is, only the "layer 2" of the architecture has to be developed, therefore this can be a good starting point.<sup>9</sup>

- The architecture of the LexDania editor can be re-used:

<sup>9</sup> For more detailed information <http://www.e-p-a-p.com/dlib/9788883980466/art17.pdf>

LexDania editor: The concept of having user-friendly schemas on top of very advanced schemas below can be re-used, and the general architecture of the editor as the editor is customised by schemas. So everybody could more or less use the editor if schemas, and user friendly schemas and the converter between the two different schemas are made.

### 3.6 ADVANTAGES AND DISADVANTAGES

This section presents the advantaged and disadvantages mentioned by the respondents to the open questions. See Table 3-12 and Table 3-13 and comments following.

	Sablonry.CZ	PTJ.FI	.UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.EE	xmlLegesEditor	Wetseditor.NL	A T4M.EP
Advantages																
Input speed relatively high				1												
Basic support to drafters	1															
Gives standardized appearance to texts	1															
Comfortable editing with known interface			1		1	1	1		1		1					
Easy to use			1	1									1		1	1
Has been in use in the ministries for a long time		1														
Fits needs						1										
Formatting is not constrained to pre-defined styles so innovative drafting approaches can be accommodated				1												
Freedom of drafting							1									
One single application for complete process						1			1			1				
Improved quality of law texts and efficiency of drafting							1			1		1				1
Strong control of documents										1						
Less support from technical staff (up-markers ...)										1						
Much better control and support of tables and pictures										1						
Assistance while creating the structure of the document													1			
Allows drafters to focus on content												1				1
Correct/ high level of structure								1							1	
Management of the XML validity of a generic XML document														1		
A priori validity of documents														1		
Automatic comparison of changes							1									
In-context amending tool											1	1				
Comprehensive and secure audit trail of changes to documents											1					
Automatically generating laws based on changes in the text															1	
Viewing where each sentence of a law text came from enhances democratic process															1	
Instant preview of documents										1						
WYSIWYG when drafting								1			1			1		
Production of draft legislation .doc and .PDF workflow reflecting business needs											1					
Metadata is part of the document													1			
Functionality to convert XML into PDF is provided													1			
Support provided for adding PDF, HTML, images in the composition of the XML document.													1			
Special tools / forms provided for creating a consolidated text													1			
Each clause and schedule is a separate file...making reordering easier to deal with.								1								
web based solution: easy to deploy and maintain.															1	
Documents and metadata stored in relational database that ensures accessibility, efficiency of searching and processing, and								1								
The centrally managed application system is accessible via the intranet and internet.								1								
Performance with large documents								1		1						
multilingualism						1										
open standards, open formats, no vendor lock-in											1					
XML, interoperability											1	1		1		
multiplatform tool (java-based solution)														1		

Table 3-12 Advantages of the solutions

### **In terms of advantages,**

#### **Type 1 solutions**

- Whereas they provide support to drafters and a “standardised appearance to texts”,
- they are easy to use, provide comfortable editing with a known interface, have been in use for a long time;
- and provide freedom of drafting, accommodate innovative drafting.

#### **Type 2 solutions**

- also provide comfortable editing with a known interface;
- provide improved quality and efficiency of drafting, allow drafters to focus on the content;
- require less support from additional technical staff (up-markers...);
- ensure strong control of documents, and a correct/ high level of structure;
- are appreciated for features such as in-context amending and comprehensive and secure audit trail of changes to documents;
- are appreciated when providing instant preview of documents / WYSIWYG when drafting;
- provide easy access to information;
- are using XML and are interoperable;
- based on .odt are seen as providing the advantages of using open standards and open formats, and avoiding vendor lock-in.

#### **Type 3 solutions**

- are also qualified as easy to use;
- improve the quality of law texts and efficiency of drafting , allow drafters to focus on content;
- provide assistance while creating the structure of the document, manage the validity of the XML, enable a correct/ high level of structure;
- allow a priori validity of documents;
- are using XML and are interoperable;
- are appreciated for features such as automatically generating laws based on changes in the text, and viewing where each sentence of a law text came from;
- are appreciated when providing WYSIWYG when drafting;
- are appreciated for making Metadata part of the document;
- based on Web solutions are seen as easy to deploy and maintain.

	Sablon.CZ	PTJFI	.UK-N.le	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.IE	xmlLegesEditor.IT	Wetseditor.NL	AT4AM.EP
Disadvantages																
difficult to update functionality						1										
not maintained/ outdated / old technology	1					1										
training needed						1										
no barrier-free web interface						1										
a posteriori validation (big effort by the typographers)	1					1										
no checking or validation		1		1												
structure not imposed							1									
no xml / no structuring	1		1		1	1										
no interoperability			1													
no change tracking	1			1	1											
Updated Documents do not contain change information.					1											
Automatic comparison of text (diffing) not completely efficient																1
no linking in between documents	1															
Limited functionality for managing larger documents,				1												
no digital signature	1				1	1										
Dependant on Word 2003 / obsolete proprietary format			1		1					1						
structural parts of documents are not accessible for search or direct reference					1											
the different language versions are linked at the document, not to the component part												1				
no WYSIWYG when drafting										1						
Fear of brackets/elements from the users										1						
conservative user community (adherence to paper artefacts such as line and page numbers to define amendments)											1					
high degree of training												1				
Different logic than a word-processor													1		1	
need an Internet connection															1	
Document created only one element at a time (slow work)													1			
The tool allows to work with only one document at a time.													1			
Need for an HTML container to import some data from other documents formats													1			
no workflow					1											
only used for primary legislation											1					
not used in the entire process											1					
customisation needed if re-used with another schema														1		
not fully XML compliant								1								
very large files								1								

Table 3-13 Disadvantages of the solutions

In terms of disadvantages,

#### Type 1 solutions

- Seem to have more disadvantages than advantages;
- Use of “old” technology, difficult to maintain;
- Require “a posteriori” validation; provide no checking or validation;

- Do not use XML; do not provide structuring ; or do not enforce structuring
- Do not use a “barrier free” Web interface;
- Are dependent on an obsolete proprietary format;

**Type 2 solutions**

- should provide WYSIWYG when drafting; users fear brackets / elements;
- face the challenge of needing to accommodate a conservative user community;
- need training.

Other disadvantages are specifically related to a missing functionality or an enhancement wished by the user of a specific solution.

**Type 3 solutions**

- Have a different logic than a word-processor;
- Allows to work on one document and one element at a time (mentioned by one respondent);
- Web solutions need an Internet connection.

### 3.7 FUTURE DEVELOPMENTS

This section maps the future developments planned for the solutions. See Table 3-14.



	Sablon.CZ	PTJFI	.UK-Nle	BAS.UK-Sc	.LT	.CH	eNorm.DE	AF.UK	eRecht.AT	LexDania.DK	LWB.IE	LWB.UK-Wa	VexPro.IEE	xmlLegesEditor.IT	Wetseditor.NL	AT4AM.EP
Future developments																
none (covers needs)									1							
change to a new system	1	1	1			1										
needs enhancement							1									
Support of Word 2010										1						
better interface							1									
xml schema	1	1					1									
Integrated tools for checking formatting and error				1												
better guidance / Wizards/Model documents or model structural parts of documents							1			1						
become fully XML compliant								1								
use of XML-based structure for major legislative documents.					1											
Use of XML throughout the complete process		1											1			
use of open standards and formats for document creation, storage and publication.					1											
Introduction of workflow functionality to fully cover entire legislative process and life of legal acts.					1											
be used by other stakeholders of the legislative process											1					
workflow	1						1									
Track changes				1						1						
Support of semi-automatic consolidation combined with automatic creation of parallel text (existing text parallel with amendment)										1						
Improve diffing																1
Introduction of digital signatures.					1											
Better tools for managing large documents				1												
Ability to output texts directly in different formats				1												
Importing documents created in other format, e.g. doc>html>xml.													1			
linking text at legislative component (parts, sections etc), not just at whole document level												1				
move to an "across the page" format											1					
Increase usability													1			
enhance interface to look more like Word															1	
To isolate legislation specific business logics from the tool as much as possible.													1			
Build an web version that functions while offline															1	
enhancement of functions related to the drafting of new bills and amendments handling														1		

**Table 3-14 Future developments**

In terms of future developments:

Only one solution covers all the needs, and will not have future developments (eRecht.AT.). It is a Type 2 solution.

**For Type 1 solutions:**

- 4 will not see any future developments because they will be replaced by a new system.
- 3 of them will be “developing an XML schema”; one “will use an XML based structure for major legal texts”; and one “plans integrated checking”.
- 1 will be using XML throughout the complete process, 3 will be developing workflow;
- Features will be enhanced, such as better guidance, better tools to manage large documents, digital signature, track changes, different export formats;
- 1 mentions changing to open formats and open standards.

**For Type 2 solutions:**

- Enhancements will be made such as support of Word 2010, better user guidance, track changes;
- 1 will be used in the entire legislation drafting/publishing process

**For Type 3 solutions**

- The XML editor which did not support importing / legacy management will include this in its future developments;
- 1 will be used in the entire legislation drafting/publishing process;
- 1 will enhance the interface “to look more like Word”; 1 will increase usability;
- For the Web solutions, 1 will build a version functioning off-line.

### 3.8 CONCLUSIONS

Conclusions of the questionnaire analysis show that:

- There is a move of all solutions towards using XML.
- XML schemas are developed for the Type 2 and Type 3 solutions.
- Type 3 solutions are not yet mature; Type 2 solutions are mature, and some cover all needs.

Generally speaking, **type 2 solutions** – which use word processors already on the market<sup>10</sup>,

- use internal format (e.g., ODT, DOC, DOCX, etc.)
- semantic check of the document
- convert and validate it according to the chosen XML schema.
- import and export XML files

Their main **disadvantage** is to miss validation during the drafting. On the other side, this means that the user can work without being interrupted by error messages.

Their main **advantages** are:

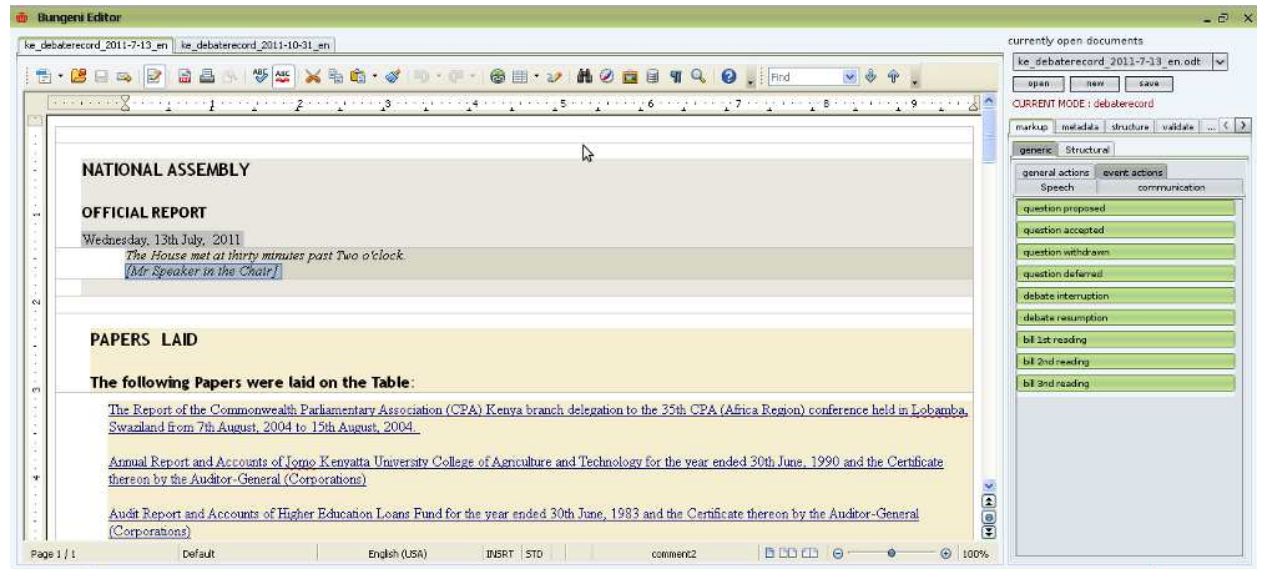
- use of an internal data format, which allows users to easily switch between different XML standards at the conversion stage;
- user-friendly interface;

---

<sup>10</sup> Monica Palmirani - CIRSFID – University of Bologna, Law Faculty – LEOS Workshop - Summer School LEX September 2011

- users are familiar with the word processing interface

Another example of this editor is the open source Bungeni-Editor<sup>11</sup> see Figure 2.



**Figure 2 Bungeni-Editor screenshot**

**As for type 3 solutions:**

**Editors based on a native XML editor** are guided by the rules of the standard XML schema and can at all times check for compliance with such rules. These editors are based on validated XML that can be open source (e.g., Java) or proprietary (e.g., Xophus).

Disadvantages are:

- a limitation of the end-user work (no freedom or innovation in drafting)
- the user must mark up following the logic on which the XML standard is based.

Advantages are:

- verifying compliance with the standard in any time

**Web-based editors** are text editors on the Web. However, there are several problems in the current state of the art.

Disadvantages are:

- Synchronization of the action on the Web
- dependent on the Internet and cannot work offline,
- synchronizing the files on the local network with those on the Web

<sup>11</sup> <http://code.google.com/p/bungeni-editor/>

Advantages are clear interfaces and ease of use, (see screenshots below), but these (known) solutions are currently used only for amendments – which do not implement extensive functionality.

Examples are presented in Figure 3 and Figure 4.

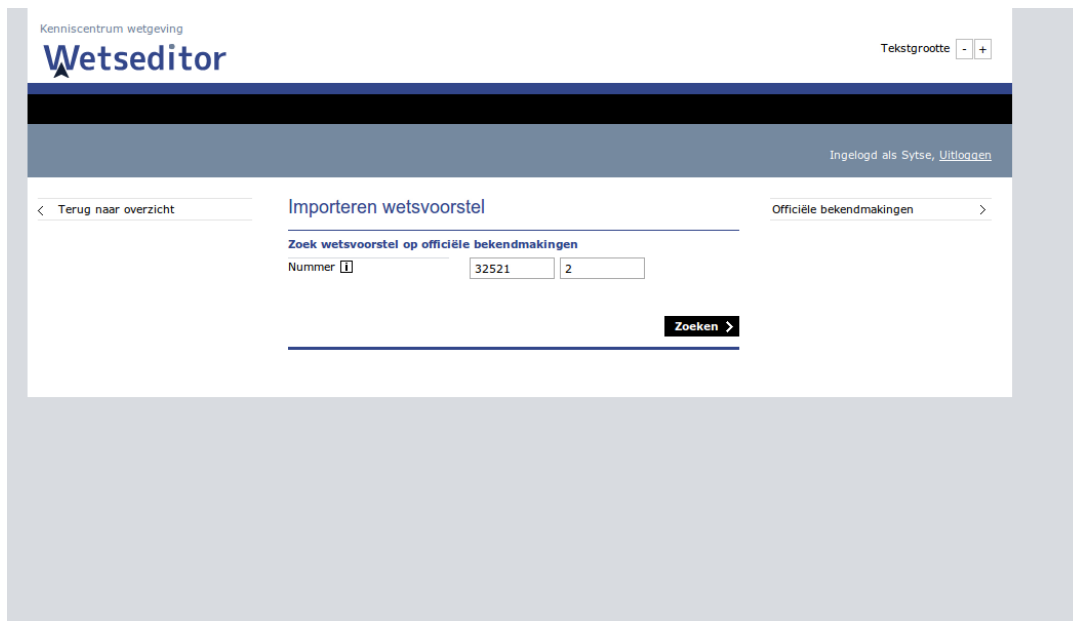


Figure 3 Wetseditor.NL screenshot

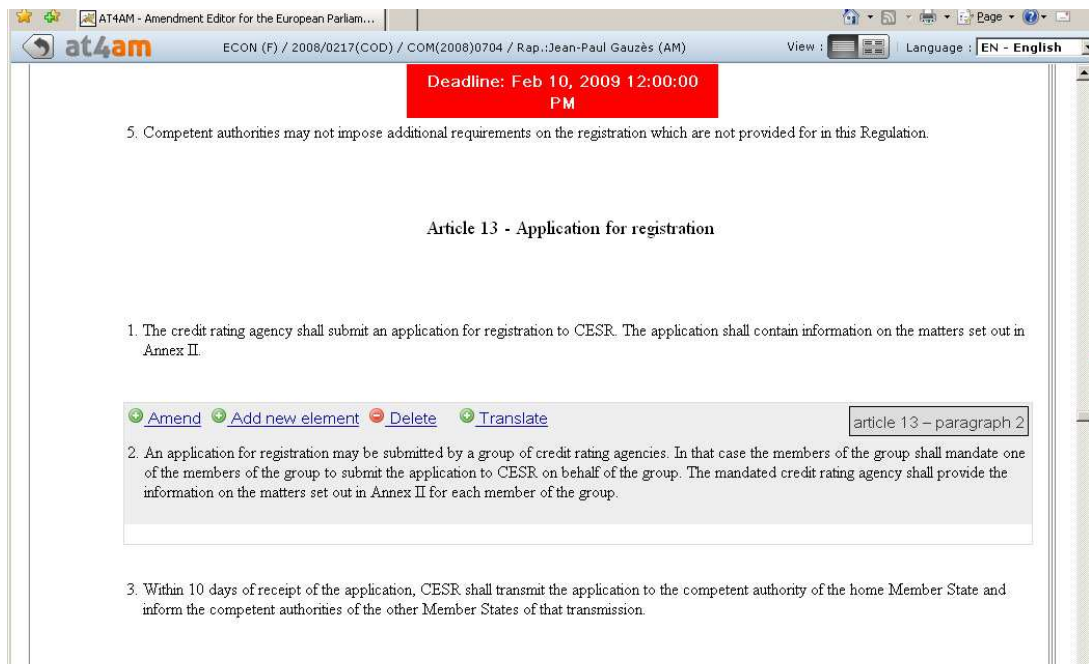


Figure 4 AT4AM.EP screenshot

## 4 ONGOING EFFORTS AND OUTCOMES FOR COMMON FRAMEWORKS AND STANDARDS FOR STRUCTURE OF LEGAL TEXTS

### 4.1 AN EMERGING INTERNATIONAL STANDARD: AKOMA NTOSO<sup>12</sup>

#### 4.1.1 Introduction

Akoma Ntoso has been developed in the context of a United Nation Department for Economic and Social Affairs (UN/DESA) project to support open access in African Parliaments and its maintenance is currently supported by the Africa i-Parliaments Action Plan of UN/DESA. It is now being adopted or entering as good practice in a number of other countries (e.g., Senate of Brazil and European Parliament) that aim to customise and to adapt Akoma Ntoso to their legal systems and purposes.

Akoma Ntoso is a set of simple, technology-neutral XML machine-readable descriptions of parliamentary, legislative and judiciary documents such as legislation, debate record, minutes, judgments, etc. that enables addition of descriptive structure (mark-up) to the content of parliamentary and legislative documents.

The main purpose of the Akoma Ntoso initiative is to develop a number of connected standards and languages and guidelines for parliamentary, legislative and judiciary documents, and specifically to:

- Define a common document format
- Define a common model for document interchange
- Define a common data schema
- Define a common metadata schema and ontology
- Define a common schema for citation and cross referencing

Akoma Ntoso is now under the process to become a Technical Committee in OASIS<sup>13</sup>.

The technical staff is currently developing a modularisation of the full Akoma Ntoso language aimed at extracting sub-schemas focused on particular types of documents:

- Akoma4Legislative for marking up the legislative document in any phase of its lifecycle: bill, act, amendments, report of the committee.
- Akoma4ParliamentaryDocuments for marking up the debates, report, order of day, motions, question-answer, minutes.
- Akoma4Justice for marking up judgments. Akoma4Justice defines how to mark up the header, the structure, the metadata, the citations to any kind of judiciary document (law, jurisprudence, judgments, articles, books, etc.) and also to define the permanent identifier (using the W3C URI standard) of these digital resources. It is also possible to qualify the citations by using a derivation of the Shepard classification system adopted, among others, by LexisNexis.

---

<sup>12</sup> <http://www.akomantoso.org>

<sup>13</sup> <http://lists.oasis-open.org/archives/legalxml-courtfilling/201110/pdf/HCZRqR9SSS.pdf>

A further description of the standard is detailed below:<sup>14</sup>

### 4.1.2 The purpose of Akoma Ntoso

- To associate specific descriptive labels (called mark-up) to legal documents and their content.
- Labels need to be associated to the whole document, to its main parts, to lesser structures (e.g., paragraphs), to individual words.
- Mark-up traditionally is divided in
  - Structural mark-up: to identify the parts of the documents and how they repeat, associate and contain others)
  - Semantic mark-up: to associate by way of labels interpretations to pieces of text (e.g.: title, date, emanating body, etc.)
  - Presentation mark-up: to associate specific visualization (e.g. typographical choices) to individual pieces of mark-up
- Furthermore, big importance is given to metadata, defined as every piece of information we want to associate to a document that does NOT belong to its content.

### 4.1.3 Design issues of Akoma Ntoso

- Short term concern
  - Being able to be used in many different ways: Multi-channel presentation, document workflow, document lifecycle, automatic point-in-time consolidation, semantic searches and classifications, etc.
- Long term concern
  - Being able to use the documents in 5, 10, 50 years' time. It means not relying on any software architecture, nor on the existence of specific resources, documents and manuals but the documents themselves which in turn means self-containment, self-description, rich set of metadata, expressive mark-up, etc.
- Attribution of contributions
  - Akoma Ntoso supports the very nature of legal documents, whose nature, integrity and validity must be ascertained with no possible doubt.
  - Thus anything that affects its integrity needs to be not rejected, but carefully marked as such.

---

<sup>14</sup> From "an introduction to the document types in akoma ntoso" F. Vitali <http://www.akomantoso.org/presentations>

- The main tool we have is the correct attribution of contributions.
- Documents have authors, that draft the content. For legislation, it is legislator, for judgments it is the judge, etc.
- Documents may have editors, that enrich the document with annotations and additional information. Editors also create the markup of a document.
- In Akoma Ntoso, the role of authors and editors is clearly identified, and their contribution (respectively, content and metadata) is NEVER mixed up.

#### ● Descriptiveness and prescriptiveness

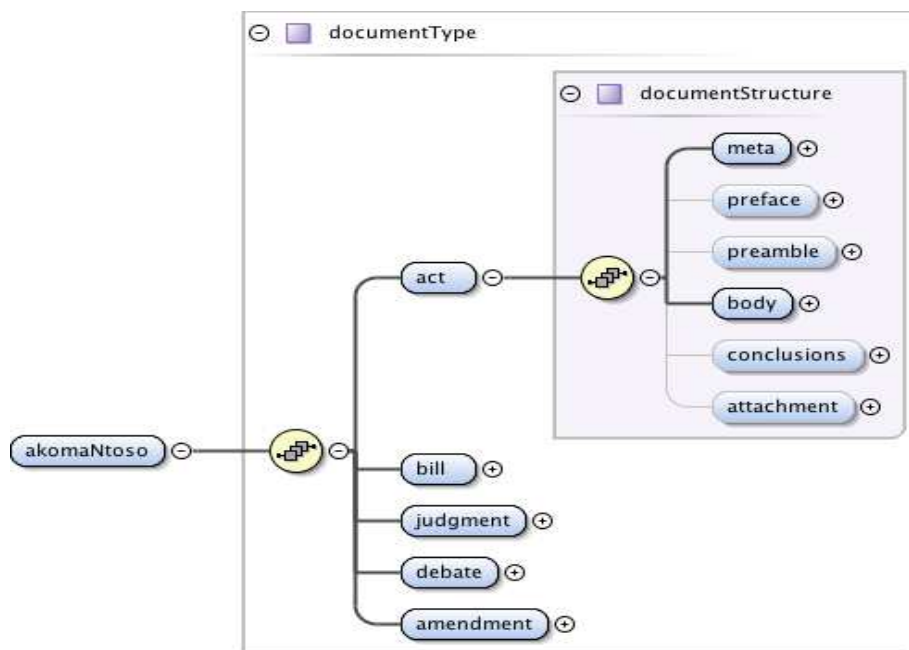
- A mark-up language resembles a language in that it has
  - a vocabulary (the list of labels you can associate to the parts of the document)
  - A grammar (the rules that let you combine the vocabulary and define correctness and incorrectness)
- A grammar gives power to its author. Power that can but should never be imposed excessively.
  - A prescriptive grammar rejects atypical constructs, and poses itself in a power position wrt them
  - A descriptive grammar tries to accommodate the widest collection of senseful constructs, and poses itself as their servant.
- Depending on the objective, Akoma Ntoso can be made
  - Prescriptive (e.g., in a drafting situation where you can and want to control the process of drafting)
  - Descriptive (e.g., in the salvaging of past collection of unchangeable documents)

#### ● Different uses, contexts, legal systems

- Most XML applications to legal systems have basically one purpose in mind: presentation.
- In Akoma Ntoso we consider also
  - Multi-channel presentation
  - Hypertext navigation (with static and dynamic links)
  - Sophisticated search
  - Automatic consolidation of amendments
- In different legal systems
  - Both civil law and common law countries

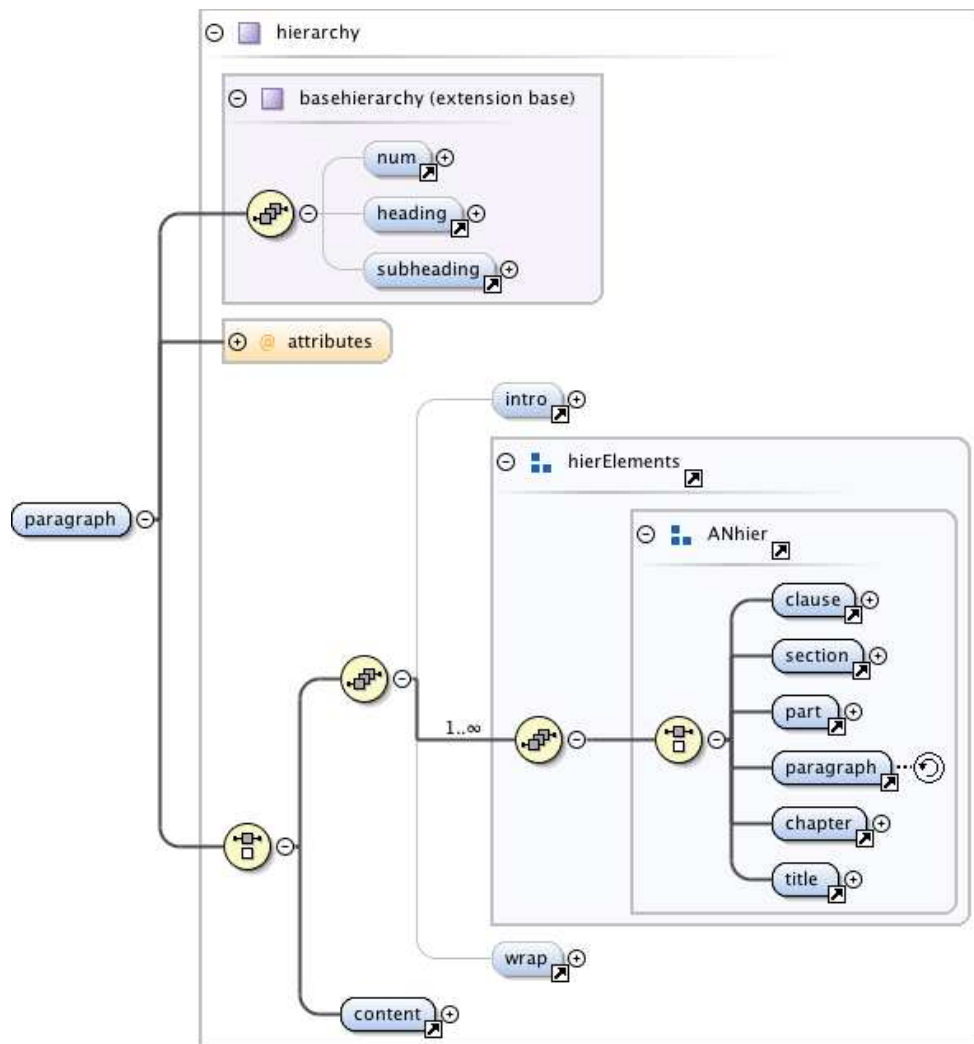
#### 4.1.4 Document types supported

- The schema supports the following types of documents:
  - Legislative documents (Bills, Acts)
  - Debates
  - Amendments
  - Judgments
  - Collective documents (Official gazettes, Amendment lists)
  - Generic documents
- The general structure of Akoma Ntoso documents :

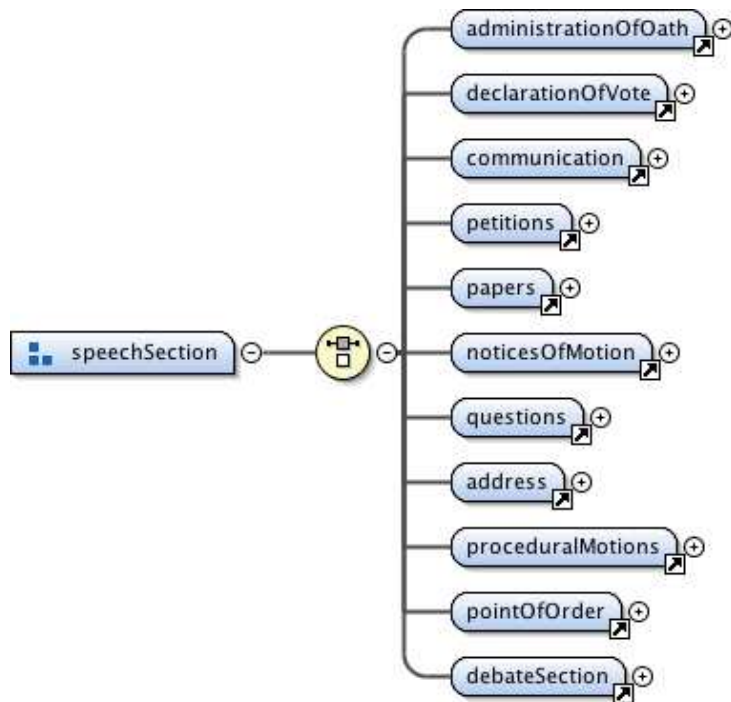




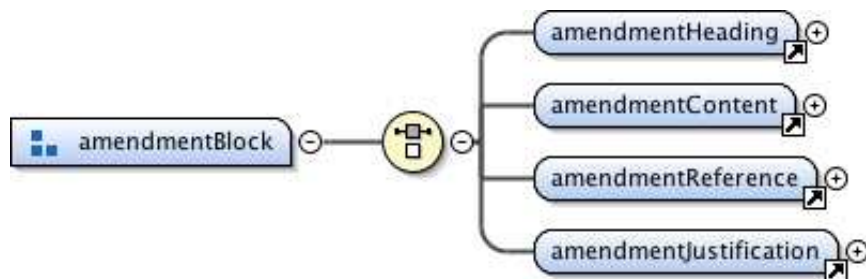
- The structure of legislative documents :



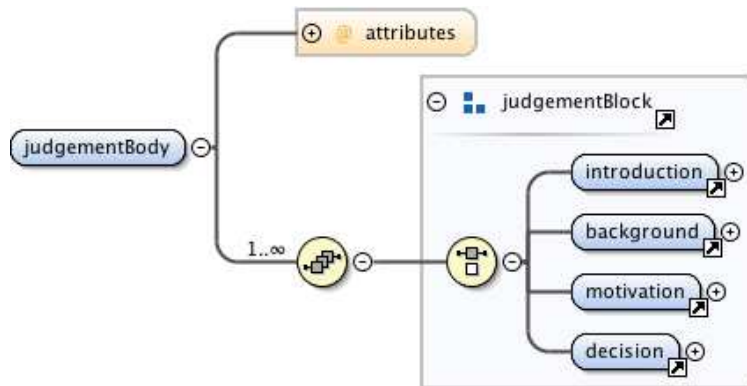
### ● Structure of debate documents



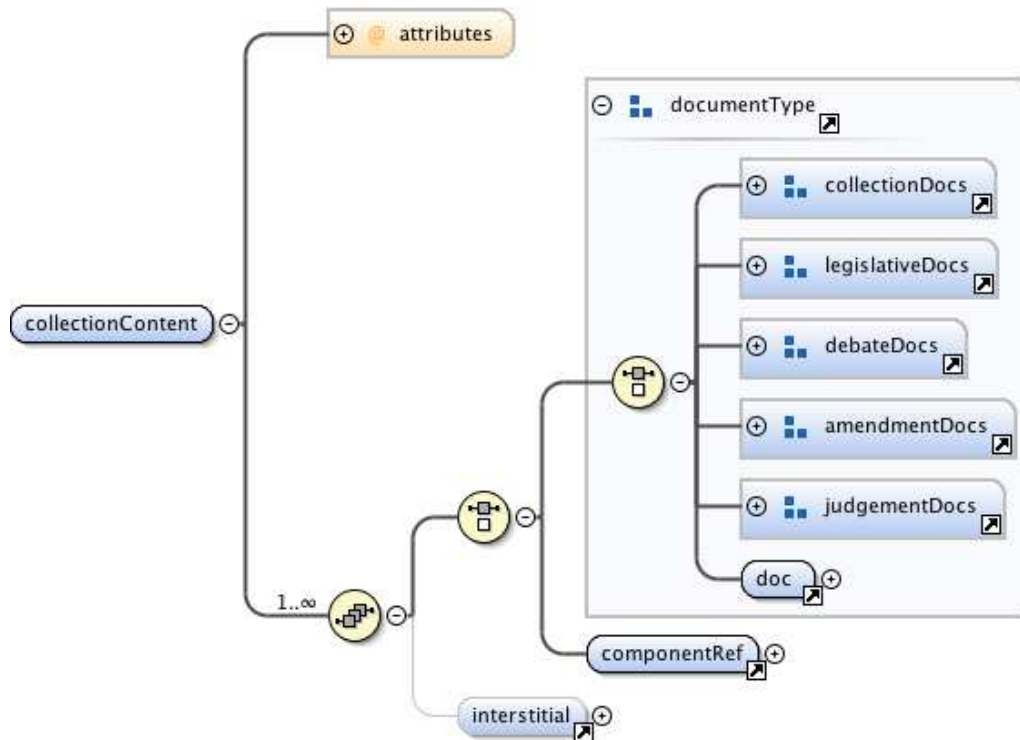
### ● Structure of amendment documents



- Structure of judgement documents



- Structure of collective documents



#### 4.1.5 Structures shared across document types

- Blocks and inlines

- Once the structure is described with all its parts and meanings, blocks and inlines are used to actually contain the text of the document
- Blocks are vertically separated containers of text (e.g., Word paragraphs).

- Inline are containers of text that do not break the line (e.g., styles within paragraphs)
- Akoma Ntoso has a rich collection of both.

#### ● References

- Explicit references to other legal document or their subparts.
- A static reference points to a specific version
- A dynamic reference points to a version that depends on the context within which the document is being sought.

#### ● Semantic elements

- Dates, individuals, concepts, events, documents need to be identified unambiguously to allow for sophisticated searches to work
- E.g. the person making a speech, the document being modified by the amending act, the validity date of an act.

#### ● Authorial vs. editorial footnotes

- In some document types the content is in the main flow, all notes are explanatory comments placed by editors. These are editor-generated text and therefore metadata.
- Sometimes the original content has text in a separate flow (sidenotes, footnotes, etc.) that is nonetheless authorial and authoritative.
- These MUST be distinguishable

#### ● Metadata: A large collection of information items that are not drafted by the author, but by drafting offices, editors, publishers, scholars, etc.

- Metadata for identification
- Metadata for classification
- Metadata for workflow and lifecycle support
- Metadata for legal analysis
- Metadata for entity classification

### 4.1.6 Special features of Akoma Ntoso

#### ● Modifications

- Managing the (semi-)automatic generation of consolidated version of the document off the original version and the sequence of modifying documents

#### ● Lifecycle and workflow

- Recording every step of the workflow which the document underwent, and giving specific relevance to those steps that generated new versions of the document (lifecycle events) and the generated document themselves

#### 4.1.7 The naming convention

- Possibly the most important feature of Akoma Ntoso is the mechanism to associate an identifiable name to each document, version of document, fragment of version of document regardless of type, country, age, etc.
- Through the naming convention it is possible to
  - Create the hypertext navigation between documents, both static and dynamic
  - Navigate through the maze of versions and variants of a document (across time, languages, uses, etc.)
  - Consolidate versions for point-in-time access to docs
  - Create a separation layers between the conceptual structure of the legal system and the physical storage on computers.

A tool used to separate conceptual names from physical names is available at: <http://akn.web.cs.unibo.it/>

#### 4.1.8 Schemas and subschemas

The full schema of Akoma Ntoso contains 229 elements and 79 attributes. While they are reasonably easy to maintain because they belong to five fundamental families called patterns, they can become rapidly confusing for the author or drafter of legal documents. Thus the full Akoma Ntoso has been modularized and it is possible to generate subschemas that are fully compatible with the complete schema, yet contain only those elements that are really useful for each individual situation.

A tool may help generating the right subschema: <http://akn.web.cs.unibo.it/aknssg>

#### 4.1.9 Conclusion

- Akoma Ntoso tries to provide for all kinds of legal documents.
- Basic structures and needs are already taken care of
- But experience has shown that individual countries do things differently all the time
- For this reason it is important to contribute with one's own experience and problem to the improvement of the whole standard

## 4.2 AN INTEROPERABILITY STANDARD: CEN/METALEX

CEN MetaLex is an interchange format, a lowest common denominator for other standards, intended not to replace jurisdiction-specific standards and vendor-specific formats in the publications process but to impose a standardized view on legal documents for the purposes of information exchange and interoperability in the context of software development.

To meet these requirements, MetaLex defines a mechanism for:

- schema extension,
- adding metadata,
- cross referencing,
- constructing compound documents
- and a basic naming convention.

## 4.3 AKOMA NTOSO AND METALEX/CEN – A COMPARISON<sup>15</sup>

CEN MetaLex constitutes a standard for sources of law, as established by the CEN Workshop on an Open XML Interchange Format for Legal and Legislative Resources (MetaLex) between 2006 and 2010. CEN MetaLex prescribes syntactic restrictions on XML documents and schemas, as defined by a XML Schema specification, and defines a semantics for XML document metadata. CEN MetaLex proposes itself as an interchange format, the lowest common denominator for other standards, intended not to replace jurisdiction-specific standards and vendor-specific formats in the publications process but to present a standardized view on legal documents for the purposes of information exchange and interoperability in the context of software development. To meet these requirements, MetaLex defines a mechanism for schema extension, for metadata addition and extraction, for cross referencing, for compound documents construction, and for implementation of a naming convention.

CEN Metalex, as such, is not born as an XML vocabulary to mark documents containing sources of law, but rather as an abstract mechanism to describe and label existing documents in any jurisdiction-specific format in a standardized way, as long as they adhere to some basic, very simple and abstract design principles. The purpose of CEN Metalex is not to mark up documents, but to enhance interchange, i.e., to allow documents which are marked up in any of the compliant vocabularies to be understood and used with little or no manual intervention in a software environment built around any other vocabulary. CEN Metalex does also provide a concrete XML vocabulary, but this vocabulary is meant more as a last resort for the interchange of difficult or problematic individual documents, than for the markup and management of a large scale collection of documents. The basic assumption of CEN Metalex is that interchange must be possible with properly edited and modified local, jurisdiction-specific schemas, but little or no change whatsoever in the actual document, that must remain as much as possible in their original jurisdiction-specific data format.

Akoma Ntoso, on the contrary, is a concrete XML vocabulary for the management of a large number of types of legal documents. Akoma Ntoso is fully compliant with CEN Metalex (in fact it was used by the CEN Workshop as the reference format for adherence to CEN Metalex), but it

---

<sup>15</sup> Monica Palmirani and Fabio Vitali, University of Bologna - 13 Nov. 2011

was designed with a concrete and specific attention to the needs of a drafting office of legal and legislative documents.

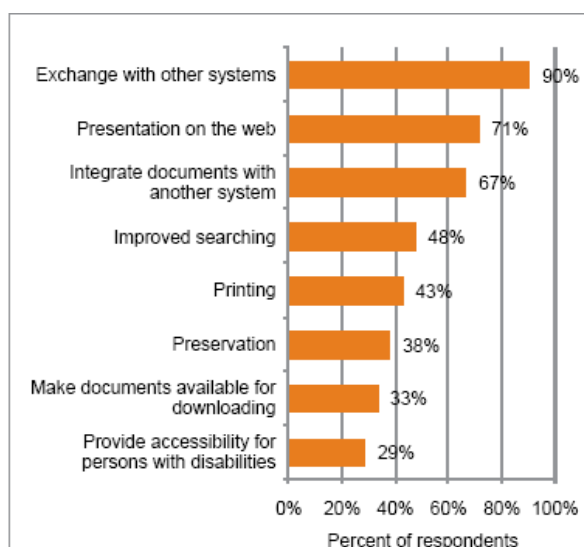
In particular, while CEN Metalex specifies compliancy requirements for the XML vocabulary, for the document metadata and for the naming convention, Akoma Ntoso defines an XML vocabulary, a metadata model and a naming convention that are compliant to these requirements, and still provides very concrete support for each individual and specific need of the document types that are described.

#### 4.4 SURVEY ON THE USE OF XML IN PARLIAMENTS<sup>16</sup>

The survey results published in the World e-Parliament Report 2010 suggest that the percentage of all parliaments that have implemented or are planning or considering implementing XML for bills has not increased significantly in the last two years.

34% of those that have a system for managing bills currently use XML. This represents 16% of the 134 parliaments responding to the 2009 survey. The comparable figure for the 2007 survey was 12%. While 16% represents a 30% increase over 12%, it still means that **fewer than 20%** of parliaments are using XML in document management systems for bills.

Parliaments that are currently using XML were asked **how it is being employed**. The results, shown in the figure below, highlight:



- exchanging documents with other systems (90%),
- presenting documents on the web (71%),
- integrating documents with another system (67%),
- improving searching (48%)
- printing (43%)
- preservation (38%).
- only 29% are using XML to provide accessibility

This list illustrates both the range and the value of the goals that XML supports. Future objectives will likely include the rendering of parliamentary information on mobile communication devices, increased support for accessibility by persons with disabilities, and more effective integration with new web technologies. The important point is that open standards such as XML offer greater flexibility for meeting both current and future needs for parliamentary document systems.

<sup>16</sup> World e-Parliament Report 2010



#### 4.4.1 XML at the European parliament

The main challenges that the e-Parliament Programme faces in rolling out XML are multilingualism and interoperability with partner institutions. Because of the high number of official languages, translation in the European Parliament happens on a massive scale. In 2006, 1.15 million pages were translated. To improve the efficiency of the process, it is imperative to ensure that XML mark-ups support the re-use of already translated information. Regarding interoperability, the European Parliament would eventually like to be able to exchange documents in XML format with the European Council, the European Commission, and national parliaments.

The European Parliament opted not to develop its own XML standard, but to comply with an emerging standard - Akoma Ntoso, and contribute to its further development.

#### 4.4.2 Challenges for using XML

World eParliament Report 2010<sup>17</sup>.

Implementation of XML poses a number of challenges. In the 2009 survey, parliaments using XML, or that have tried to use it, were asked which problems they had experienced. The figure below summarizes responses from two groups: those currently using XML (34 parliaments) and all those that identified at least one challenge on the list (59 parliaments). This latter group includes parliaments in the first group and those that are not currently using XML but have faced barriers in trying to implement it.

Challenges	Parliaments that identified at least one challenge (Total=59)	Parliaments currently using XML (Total=34)
Lack of staff knowledge and training	59%	26%
Lack of financial resources	44%	15%
Finding authoring / editing software	41%	26%
Complexity of using XML	34%	26%
Difficulty in developing a DTD or schema	34%	26%
Lack of management support	24%	12%
User resistance	14%	12%
Other	7%	9%
None of the above	N/A	29%

Several findings contained in the figure above are particularly informative. For the 34 parliaments currently using XML there was a relatively even distribution among **four** challenges, all identified by at least a quarter of the respondents:

- lack of staff knowledge and training,
- finding software for authoring and editing,
- the complexity of using XML, and
- difficulty in developing a Document Type Definition (DTD) or schema.

<sup>17</sup> <http://www.ictparliament.org/wepr2010>

It is interesting that the challenge checked by the most parliaments, however, was “None of the above”.

For the 59 parliaments that identified at least one item (whether they are currently using XML or not), the challenge mentioned by the most parliaments was the **lack of staff knowledge and training** (59%). The other obstacles mentioned by the most parliaments were the **lack of financial resources** (44%) and **finding authoring and editing software** (41%).

If legislatures currently using XML are removed from the combined group, the remaining 37 parliaments, which have presumably experienced challenges in trying to implement XML, cite the top two problems - staff knowledge and training and financial resources - even more frequently (70% and 57%, respectively).

#### 4.5 THE PUBLICATION OFFICE XML SCHEMA: FORMEX<sup>18</sup>

Formex is the XML schema implemented by the Publication Office, which describes the format for the exchange of data between the Publication Office and its contractors. In particular, it defines the logical mark-up for documents which are published in the different series of the Official Journal of the European Union.

The Office for Official Publications started the production of SGML instances for the documents published in the different OJ series in 1985. The specifications are known as Formex (Formalised exchange of electronic documents). The current version 3 entered into force on the 1st of April 1999.

For different reasons, SGML and SGML based grammars have not had the success as expected, because of the complexity of the grammar and the difficulty to develop tools supporting it. User organisations were involved to discuss the future of FORMEX, and a proposal was made to migrate to XML and to replace DTD with XML schema.

The Office for official publications decided to migrate the Formex specifications from SGML to XML. The version 4 is presented as an XML Schema grammar. At the same moment, the character set definition abandons the approach based on ISO 2022 and moves to Unicode (UTF-8). This has reduced problems with encoding of special characters and symbols of different EU languages.

The migration to XML also gave the opportunity of reviewing the existing specifications. Instead of about 1200 tags in Formex version 3, Formex version 4 consists of only about 260 tags.

The specifications consist of two parts:

- the physical specifications which contain information on the exchange of data, the construction of filenames and, in particular, on the character set; and
- the grammar for the mark-up based on XML Schema.

FORMEX is designed and enhanced to fit the publication needs of European legislation.

---

<sup>18</sup> <http://formex.publications.europa.eu/schema/>

## 4.6 STUDIES ON THE USE OF XML IN EU MEMBER STATES AND ON THE NEED FOR COMMON STRUCTURES

### 4.6.1 Outcome of the working group European Forum of Official Gazettes on the Use of XML for the production and distribution of the official gazettes (2006)<sup>19</sup>

The working group confirmed that there are important differences in the structure of the legislative documents of each country.

The conclusions are:

*“A common vocabulary on structure would not be successful from two different reasons;*

- 1. Important differences in legislative culture causing different structures in legislative documents.*
- 2. The advanced status of XML based projects in the different countries leaving no possibility to introduce a new schema on structure”*

However, a common vocabulary regarding metadata is proposed by the working group, while the analyses showed that metadata across legislative systems were highly congruent, and none of the existing metadata standards are appropriate for the juridical description of documents.

The working group has created a metadata glossary with agreed definitions, and from that a common XML Schema and DTD were written.

Two obvious benefits are stressed in the report. The common metadata can be used as an inspiration or a check list to ensure that the most typical subjects of legislative metadata are covered, and implementing the common metadata will facilitate the creation and maintenance of cross system portals giving access from the same website to multiple legal information systems.

This study was finalised in 2006. Today, there is no current usage of the common metadata schema identified.

However, additional information available today can provide some “nuance” to the conclusions of the study regarding a common structure with the IPEX study (see 4.6.2) about XML in parliaments and the development of the Akoma Ntoso standard which accommodates different types of legislative culture (common law and case law).

### 4.6.2 Survey on the EU Parliaments initiatives on common standards for digital data and documents<sup>20</sup> (IPEX - 2011)

The Survey has been prepared by the IPEX Central support on the basis of the replies provided by 31 parliamentary assemblies (out of 40) of 22 EU member States to a questionnaire sent to the IPEX Correspondents. One of the questions asked was about the use of (or the willingness to use) an open format (XML) for parliamentary documents and bill of law and the involvement

<sup>19</sup> 3rd meeting Ljubljana, September 14 – 16, 2006 <http://circa.europa.eu/irc/opoce/ojf/info/data/prod/data/pdf/Ljubljana2006-useOfXml.pdf>

<sup>20</sup> [http://www.ictparliament.org/attachements/XMLImplementation\\_Survey\\_common\\_standards.pdf](http://www.ictparliament.org/attachements/XMLImplementation_Survey_common_standards.pdf)

in any multi-country initiatives to exchange parliamentary documents or regulative instruments in open format (XML).

The Survey shows that 12 Parliaments/Chambers out of 31 uses XML for some parliamentary and/or legislative documents. The documents are marked-up at different stages of their production and from different departments in each Parliament/Chamber.

In 9 Parliament/Chambers both the metadata and the datacontent are structured in XML.

11 Parliament/Chambers made XML data public.

11 Parliaments/Chambers plan to introduce XML mark-up for parliamentary documents or for new and additional series of documents.

Most Parliaments use their own proprietary format. However 12 Parliaments/Chambers are expressly available or interested in making their XML documents compatible with XML documents produced by other Parliaments or institutions. Most of them require that a common standard is established to this end.

Some Parliaments/Chambers (7) underline that the only project for sharing parliamentary documents or information in open format (XML) with institutions of other countries is the IPEX XML project.

#### **4.7 IPEX (INTERPARLIAMENTARY EU INFORMATION EXCHANGE) XML**

IPEX, the InterParliamentary EU information eXchange<sup>21</sup>, is a platform for the mutual exchange of information between the national Parliaments and the European Parliament concerning issues related to the European Union, especially in light of the provisions of the Treaty of Lisbon.

The main part of IPEX is the Documents database which contains draft legislative proposals, consultation and information documents coming from the European Commission, parliamentary documents and information concerning the European Union. IPEX enables inter-parliamentary exchanges on legislative proposals from the European Institutions, and for this, the National parliamentary documents commenting on the proposed legislation are uploaded individually by each national Parliament.

There is a standard structured description (in XML) of each document uploaded by the National parliaments; it contains the scrutiny (comments on the legislation), the status of the scrutiny, reference to the documents and contact information. The XML feed contains the following information (see for structure Figure 5 IPEX XML scrutiny structure, and Figure 6 IPEX XML – scrutiny structure illustration<sup>22</sup>).

---

<sup>21</sup> <http://www.ipex.eu>

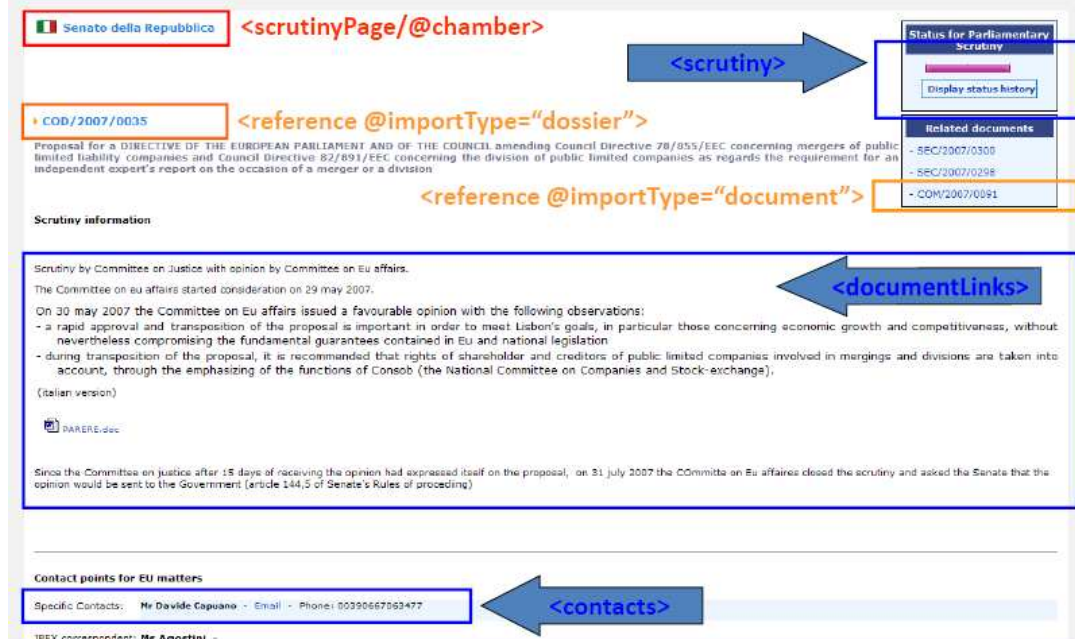
<sup>22</sup> [http://www.parlamento.it/documenti/repository/Parlamento/ECPRD/interventi/AracoStronatiCameraSenato-IPEX\\_ECPRD\\_ICT\\_2009.pdf](http://www.parlamento.it/documenti/repository/Parlamento/ECPRD/interventi/AracoStronatiCameraSenato-IPEX_ECPRD_ICT_2009.pdf)

## XML feed: <scrutinyPage/> example

```
<?xml version="1.0" encoding="UTF-8"?>
<scrutinyPage chamber="ITSEN"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="scrutinyPage.xsd">
  <reference importType="document">
    <code>COM</code>
    <year>2007</year>
    <number>0091</number>
  </reference>
  <scrutiny>
  <documentLinks>
  <usefulLinks>
  <contacts>
</scrutinyPage>
```

Figure 5 IPEX XML scrutiny structure

## XML feed: National Parliament scrutiny page



The screenshot illustrates the structure of the National Parliament scrutiny page, with various elements mapped to XML tags:

- Senato della Repubblica** is linked to `<scrutinyPage/@chamber>`.
- COD/2007/0035** is linked to `<reference @importType="dossier">`.
- The **Proposal for a DIRECTIVE OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL** is linked to `<reference @importType="document">`.
- The **Status for Parliamentary Scrutiny** section, including **Display status history** and **Related documents**, is linked to `<scrutiny>`.
- The **documentLinks** section, which lists related documents like SEC(2007)0300, SEC(2007)0298, and COM(2007)0091, is linked to `<documentLinks>`.
- The **Contact points for EU matters** section, including specific contacts like Mr Davide Capuano, is linked to `<contacts>`.

Figure 6 IPEX XML – scrutiny structure illustration

## 4.8 NATIONAL SCHEMAS

Several countries have developed a national standard for the XML schema used in drafting or in publication. Among them are:

In **Denmark**, Lex Dania xml<sup>23</sup> is an XML meta-schema used to compose other, more specific, XML schemas. In this manner, it provides a data-model for Danish Legislative Documentation. (see detailed description in the Case Study on Word processor based tools).

In **Estonia**, there are three schema definitions for legislative documents, available online.<sup>24</sup> There is one basic scheme in which all elements are included. Sub-circuits have been prepared on the basis of these and may have fewer elements. There different types of structures, with the biggest difference in metadata stages and statuses: act of law; act of law to modify an act of law already in force; a regulation; individual act to modify one already in force.

Legislation ML is used by the Houses of Oireachtas in **Ireland**, and DTDs are available online<sup>25</sup>.

In **Germany**, a new project, planned to end in January 2012, aims at defining xNorm: xml format for legislative texts. In terms of compliance to existing standards, xNorm will have to be XÖV (XML in public administrations) certified<sup>26</sup>. No compliance to other standards is required.

In **Italy** NIR (Norma in Rete)- translation: legislation on the Net) was initiated by CNIPA (Center for Information Technology in Public Administration) and the Ministry of Justice. NIR project produced a standard for XML representation of documents, DTDs and XML schemas were defined for Italian legislation. These schemas include structural information as well as administrative and semantic metadata providing meaningful information and to automate legislative documents in life cycle management. The NIR standards were issued as a technical standard by the Italian authority for Information Technology, and published in the Official Journal. NIR standard is adopted by several Italian public administrations and private operators.

In the **Netherlands**, a National schema used for publications exists (BWB). BWB stands for Dutch Basiswettenbestand – a large legal database containing almost all Dutch laws and decisions. BWB is the standard for encoding the laws in the database; it is based on a DTD originally developed by SDU publishers (see case study on “XMLisation”) and is now maintained by the Dutch government. This schema is used also by the Wetseditor solution.

---

<sup>23</sup> <https://www.retsinformation.dk/offentlig/xml/schemas/>

<sup>24</sup> [http://xmlr.eesti.ee/xml/schemas/oigusakt/tyviseadus\\_1\\_10.02.2010.xsd](http://xmlr.eesti.ee/xml/schemas/oigusakt/tyviseadus_1_10.02.2010.xsd)  
[http://xmlr.eesti.ee/xml/schemas/oigusakt/maarus\\_1\\_10.02.2010.xsd](http://xmlr.eesti.ee/xml/schemas/oigusakt/maarus_1_10.02.2010.xsd)  
[http://xmlr.eesti.ee/xml/schemas/oigusakt/yksikakt\\_1\\_10.02.2010.xsd](http://xmlr.eesti.ee/xml/schemas/oigusakt/yksikakt_1_10.02.2010.xsd)

<sup>25</sup> <http://www.oireachtas.ie/viewdoc.asp?fn=/documents/X-PrintSamples/document1.htm>

<sup>26</sup> From The Federal Office for Information Technology

[http://www.bit.bund.de/nn\\_387412/BIT/DE/Standards\\_Methoden/XOEV/Zertifizierungsstelle/node.html?\\_nnn=true](http://www.bit.bund.de/nn_387412/BIT/DE/Standards_Methoden/XOEV/Zertifizierungsstelle/node.html?_nnn=true)

## 5 CONCLUSION OF THE INITIAL ANALYSIS

The initial analysis shows that there is **a wide diversity in usage, coverage of needs and technology used**. Some solutions – not all - provide fully integrated systems from the initial drafting to the publication. The drafting and publishing is linked to the use of a structure (XML schema) for some, others do not use such a schema for structuring information input. Some solutions are custom developments based on off-the-shelf software (office tools or XML editors), others are complete custom development. Features of these solutions vary therefore in terms of functionalities, but also user-friendliness, efficiency and quality of the text, the need for additional resources to check texts after drafting, re-usability of data, etc.

In terms of tools and technology used, the analysis shows that almost 70% of the solutions are based on word processors (MS Word, Open Office). Only 2 countries<sup>27</sup> use a solution based on Open Office (the same solution). Almost 70% of the solutions use XML, and all the solutions using XML use a standard schema or DTD. 25% implement a workflow.

Solutions have been grouped into three categories:

- Type 1 solutions : using text editors based on word processors and which do not use XML .  
(or only export for publishing).
- Type 2 solutions : using text editors based on word processors and which use XML for editing.
- Type 3 solutions: other solutions using XML (XML editors, Web editors).

In terms of future developments, only one solution covers all the needs, and will not have future developments (eRecht.AT.). It is a **Type 2 solution**. The analysis shows that type 2 solutions are mature, and provide as main advantages a check and validation according to a predefined structure, and a familiar user interface. Moreover, as they use internal formats (.docx, .odt, ...), users can easily switch between different XML schemas (if needed). Their main “disadvantage” – sometime seen as advantage – is a lack of validation during drafting – and a degree of freedom for the user.

For **Type 1 solutions**, most will either be replaced by a new system or develop/use an XML schema (and move to a type 2 or Type 3 solution). 3 will be developing a workflow.

**Type 3 solutions** check for compliance with a pre-defined structure (and only one) at all times, while providing some user-friendly interface (e.g. WYSIWYG). Quality and efficiency of legislative drafting is improved. The (known) use of these tools is not wide-spread and implementations are currently limited to some features (ex: amendments management), or to some parts of the organisation (publications, parliament) and other pilots.

With regards standards and common frameworks, countries using XML have developed National schemas, but which do not comply with an international schema standard. However, initiatives for international standards exist. The Akoma Ntoso standard (used in parliaments across the world including the European Parliament) tries to provide for all kinds of legal

---

<sup>27</sup> The SOLON solution in France is based on Libre Office. At the time of the writing of this section of the report, the solution was being deployed. The final report will include this solution.

documents, and is at a stage where basic structures and needs are already taken care of, but needs to be improved in order to incorporate individual countries' needs.

A proposed approach for a next phase of the LEOS project is to work towards defining a common structure, studying the adequacy / needed adaptations of the international standard Akoma Ntoso, as well as setting up one or two proofs of concepts based on a Type 2 solution and a Type 3 solution which can support the common structure.



## 6 CASE STUDIES

- Case study 1 covers an analysis of structure management in two tools based on word processors:
  - the eNorm/xNorm case study analyses a framework for applying structures
  - the Lex Dania case study analyses an approach to imposing a structure.
- Case study 2 addresses XMLisation (or mark-up) methods and processes, the need for them, some desktop tools and their re-usability.
  - It details the Dutch publication XMLisation process at Sdu publishers.
  - It presents the XMLisation process needed for European Parliament amendments.
  - It explains the way Norma Editor – a tool based on Microsoft Word add-ons provide mark-up to a legal text.
  - It explains the way xmLeges – an open source desktop XML editor provides mark-up to of legal text.
- Case study 3 analyses Bungeni, an open source legislative editor based on the word processor Open Office, and exporting to Akoma Ntoso XML standard. It covers how the tool applies the structure, what the user guidance provided, the technical aspects of adding document types, and the re-usability of the tool.
- Case study 4 presents SOLON II, the new French system for managing the process of legislative texts from creation to publication, based on Open Office and an open source document management system Nuxeo. The study covers the change management aspects linked to the business process re)engineering and the roll-out of the project.
- Case study 5 covers Vexpro, an open source desktop XML editor developed for legislative drafting in Estonia.
- Case study 6 presents Wetseditor - a Web XML editor used in the publishing process in the Netherlands, and introduced recently in Parliament for amendments - and covers the user benefits, as well as change management aspects. It also briefly presents functionality for AT4AM the European Parliament amendment XML editing web tool.

## CASE STUDY 1: ANALYSIS OF STRUCTURE MANAGEMENT IN TOOLS BASED ON WORD PROCESSORS

This case study aims at analysing how a structure is managed in tools based on word processors. It focuses on how the tool guides the user in interacting with the structure, to which extent the structure is imposed, what level of drafting freedom the user has and how new templates/structures can be added. It addresses the benefits and trade-offs of this type of solution in general, as well as of the difference in approach each tool has.

The first case is eNorm, developed by the Federal Ministry of Justice in Germany, which counts over 1000 users at federal and regional levels. This solution, based on templates in MS Word, provides guidance for using a highly granular structure, and is currently adapting to a new National XML schema (xNorm). eNorm can be described as a framework for applying structures.

The second case is Lex Dania editor, developed by the Ministry of Justice in Denmark. The tool counts approximately 400 users. It allows editing XML (WordML) files using a word processor interface (Microsoft Word). The structure is based on a complex XML schema (Lex Dania XML) and related user friendly schemas. The case also details the Lex Dania schema and its strategy within a European context, as it was developed at the same time as current standards (CEN MetaLex, Akoma Ntoso). Lex Dania editor can be described as a tool to impose a structure.

### 6.1 ENORM/XNORM CASE STUDY – A FRAMEWORK FOR APPLYING STRUCTURES

*This case study is based on replies to the questionnaire, interview of the users of the tool from the Ministry of Justice, desk research on support documentation available on the web, as well as reply to a technical questionnaire from the developers of the tool (DlaLOGIKa).*

#### 6.1.1 Introduction

eNorm is being developed within the framework of the one-for-all service entitled "preparation of political and regulation decisions" by the Federal Ministry of Justice, as part of the initiative for better regulation in Germany. LegisWrite, a computer tool already used by EU institutions since 1996, served as a basis for the software. It was agreed between the European Commission and the Federal Ministry of Justice that this tool could be adapted to suit the needs of federal legislation.

eNorm is a framework for applying a structure using a word processor, allowing some freedom of drafting (no *a priori* validation), providing user guidance with user-known word-processor interfaces. Editing in eNorm is WYSIWYG, in the sense that it provides the user with an immediate view of the content of his text with the final layout while he is drafting. Structuring of legal texts is managed by the pre-defined styles; these formatting styles are then converted into elements.

*Feedback collected from the users of eNorm (questionnaire and interview at the Ministry of Justice) emphasised the user guidance functionality and user friendliness of the tool. The need to implement a new National XML standard for legal texts (xNorm) across the complete legislative system was also mentioned. This case study focuses on these two aspects, important to the users.*

In order to understand the advantages of this type of tool in terms of user friendliness and user's freedom in creating text and templates, this case study presents in a **first section**, the method for applying the eNorm structure both in terms of user guidance/functionality as well as in technical terms, considering aspects such as how the tool can support other structures, or how templates are created.

A **second section** explores the benefits and corresponding trade-offs of the tool, comprising user-friendliness and a non-imposed but detailed structure, as well as the needs for eNorm to adapt to evolving user needs, including the support of xNorm - a National open standard for structured data (an XML schema).

## 6.1.2 Applying the eNorm structure

### 6.1.2.1 Functional aspects: user guidance and other functionality

#### *A wide range of functionality*

Various eNorm functionality ensure the user can apply the structure needed in legal documents. The drafters are supported in the creation of bills by an automatic generation of correct legal structures. Formatting is automatically corrected, internal links produce dynamic relationships between document parts, elements are automatically numbered, there is automatic harmonisation of references within a draft bill, and tables of contents are created dynamically. eNorm allows to run an online validation of quotations of titles of other legal acts, ensuring assistance in the collection of legal citations. In addition, users can view consolidated version of other legal acts.

The section below details some functionality.

#### *Guidance in drafting a new document*

eNorm presents the user a series of document templates for new principle laws or for amending legislation. These templates present the compulsory components of a draft bill in the correct sequence. Functions ensure compliance with the requirements of legal drafting rules, e.g. guidance for insertion of new text passages and clear display of the overall structure, as well as a link to the Handbook on Formal Requirements for Drafting Legislation (see Figure 7 eNorm – screenshot of reference texts and direct links to the Handbook of the legal form).

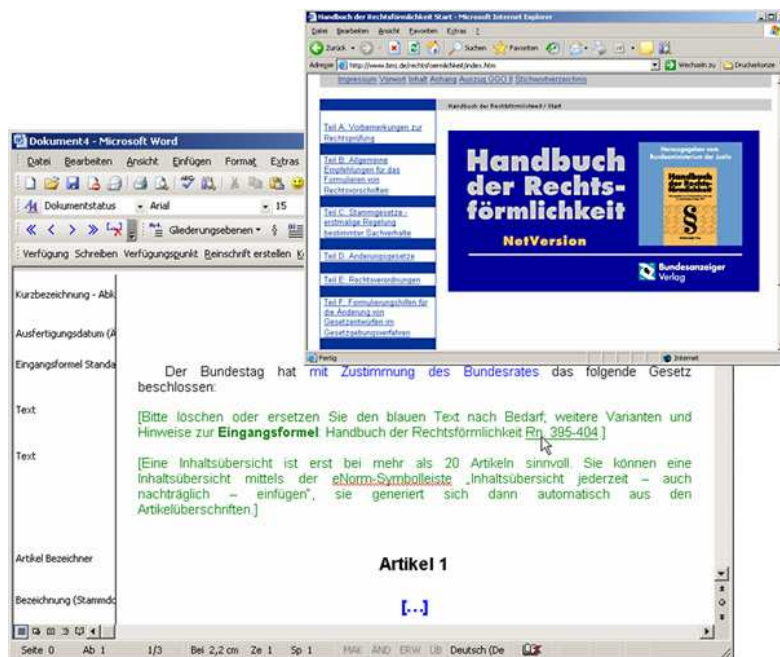


Figure 7 eNorm – screenshot of reference texts and direct links to the Handbook of the legal form

eNorm suggests formal and editorially necessary components of a law and provides concrete recommendations. The title on the front page is highlighted, and the suggestion is that the cover page should be no more than one page. Another example is the automatic insertion of the first sentence of a law (which is generally the same): "*The Bundestag has decided with the consent of the Bundesrat, the following law:*". Then, the § 1 is correctly positioned and predefined in the needed formats. The following labels and numberings (for example of paragraphs) are also automated, according to the legal requirement.

### ***Cross-referencing***

The internal cross-references (Binnenverweise) enable navigation between the reference and its target back and forth, and can be updated automatically. An interesting feature is the automated creation of a skeleton for the "Explanatory memorandum" counterpart in eNorm. In German laws, the explanations on the reasons for creating regulations and laws should be quite elaborate; it is possible to provide a reason for each individual legal part (e. g. numbered element). eNorm supports the creation of a skeleton of the explanation section, following the current structure of the document and enriching this skeleton section with the internal cross-reference mechanism (Binnenverweise). The explanations for the structural elements can be updated automatically, meaning that elements in the explanation section are rearranged if the corresponding structural elements in the legal text are rearranged. As the cross-reference mechanism offers navigation, going from a structural element to its explanatory statement and back is easy.

### ***Document quality check***

Examples for the semantic and formal error checks include (see complete list in annexe to this case):

- Consistent use of headings on article level.
- More legal paragraphs are contained in an article than recommended.
- Superfluous higher divisions (part, chapter) in case of sparse content.
- Too many sentences in enumerations, too many words in a sentence, too long words.
- Amendment statements are not verbalized correctly.
- Mistakes in amendments on amendments.
- Check on disallowed article headings, order of certain article headings.
- Presence of entry into force and correct positioning.
- Irregularities in titles, abbreviations.
- Irregularities in formatting of dates and numbers.
- Validity of citations of other legal texts (via online check in federal legal register) as mentioned above.
- Update and validity status of internal cross-references.

While LegisWrite's focus of the document quality check is rather on formal aspects, eNorm checks more *content* or *semantic* aspects, thereby assisting in improving the legislative quality by analysing compliance with many instructions of the German legal guide ("Handbuch der

Rechtsförmlichkeit" <http://hdr.bmj.de>). In addition, for most errors found during the quality check, there is a link to the appropriate instruction in the German legal guide.

## Document structure check

The structure information provided by eNorm allows a comprehensive auditing of the quality of the document, if the style guide of eNorm was applied. There are 250 styles in eNorm, the system applies the styles automatically, and the style is automatically repaired if the user made a change on it. Because of the high degree of structuration of eNorm documents (see Figure 8 eNorm screenshot showing the granularity of the document structure), the Structure Check can detect structural validity errors with high precision. This goes down to the level of individual paragraphs; in case of amendment statements, this even goes down to the level of quotation marks. The XML export of eNorm as well as the XML import and export of xNorm perform schema validation.

The Structure Check offers the possibility to navigate in the document (with highlighting of selected structural elements) and the regrouping of elements (e.g. whole sections with their entire content can be moved or copied into new documents/copied from existing documents). The Structure Check does not only check a valid structure, but also allows moving whole blocks of content, which are automatically renumbered.

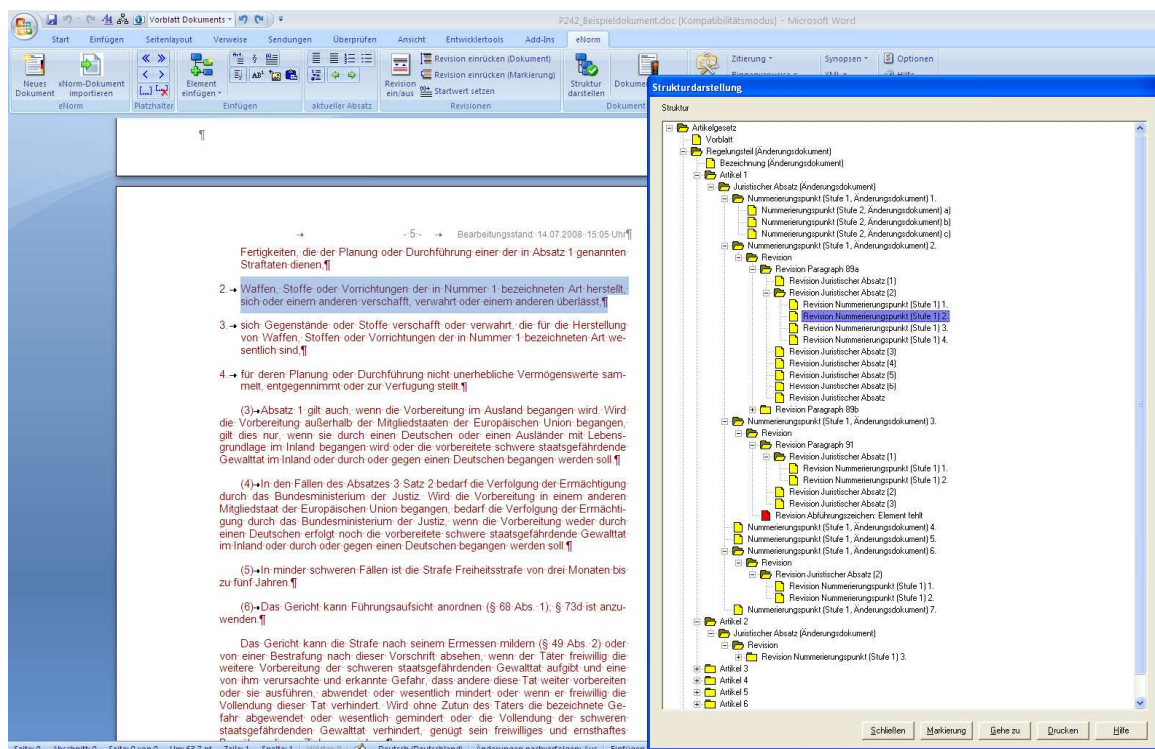
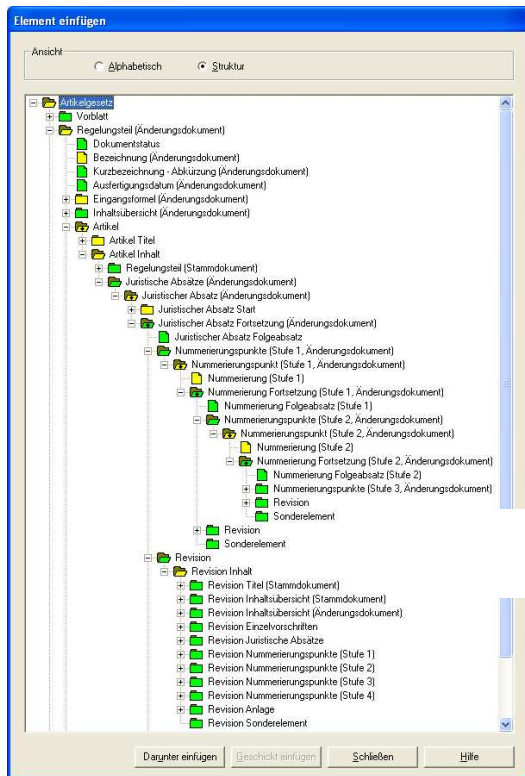


Figure 8 eNorm screenshot showing the granularity of the document structure





**Figure 9 eNorm screenshot – structure elements and drafting phase**

The high degree of granularity of the structure does not only support the user in generating well-structured documents, but also guides in determining the possibilities for continuing the drafting process by showing valid following elements to be completed (in green - see Figure 9 eNorm screenshot – structure elements and drafting phase)

The structure of a document can be viewed using eNorm's *Element einfügen* command, and in so doing, elements can be “smart insert”ed, meaning that the feasible insert positions are determined automatically by eNorm according to the concrete document structure and the abstract underlying schema.

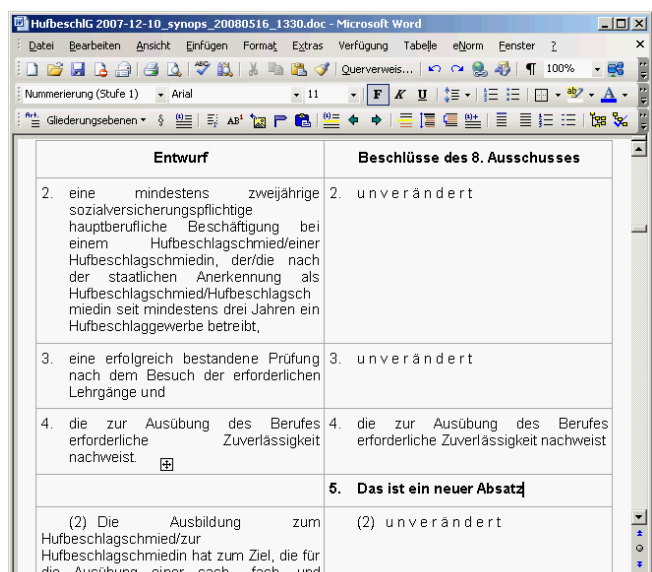
## Other functionality

### Track changes

Most eNorm commands have been extended in the past in order to be fully compatible with Word's “Tracked changes” – this extends to the Structure Check, the “Binnenverweise”, and so on. For some commands (e. g. XML export) the decision had been taken to ask the user to accept any pending “Tracked changes”.

### Consolidated version

Viewing the consolidated version of other legal acts (registered in a federal law database, accessed via a web service) from within eNorm is supported by specifying its title and



**Figure 10 eNorm Screenshot - Synopses**

the legal part intended to view (e.g. “article 5 paragraph 3”).

### *Synopses*

For the editing of different versions of a document, the synopsis feature is also useful: a two-column representation of the document can be created automatically, the user edits the right column as desired. From this so-called „working document“, the consolidated version can be generated, as well as an overview on the differences between the left and right column (i.e. the different version of the act).

### 6.1.2.2 Technical aspects

Up to the end of 2011, eNorm was working with Word documents and offered an export to XML only (used for publishing and archiving). eNorm was relying on a DTD<sup>28</sup> internally for structuring documents, but the users were never exposed to this structure, except through the corresponding eNorm styles in the templates. Compliance to the structure is achieved by applying correctly the styles when drafting.

End of 2011, (after the LEOS questionnaire had been submitted by the German ministry of justice), xNorm had been implemented. With xNorm, full input of and output towards XML (currently for base acts only – amending acts will be supported in a later stage) has been provided in compliance with the coming German XML standard for legislative documents (xNorm).

### ***Design principles***

LegisWrite/eNorm can be considered as a framework for applying structure to Word documents. The various customisations of LegisWrite for the Commission, for the Council (covering also various Council-internal document types apart from legislative documents) and for the ECB (e.g. press releases, speeches) and of eNorm for the German ministries, for the German Federal Parliament and for various Länder are implementations of the generic LegisWrite/eNorm approach.

Consequently, the major part of the LegisWrite/eNorm kernel is independent of any concrete document type and specific schema.

The main principles are as follows:

- A Word document template provides paragraph styles (and various other aspects such as page definitions, autotext entries, and so on).
- Per document template, there is an INI file comprising the following information among others:

---

<sup>28</sup> For historical reasons, parts of the definition are still in INI files, but in an equivalent representation that depicts classical XSD features like mandatory/optional elements, choices, cardinalities, dependencies, ... Transferring these to an XSD would be simple. In fact, the first versions of LegisWrite were using SGML as storage format (a mark-up language for documents oriented towards publishing). The SGML file format was replaced by the doc format for the sake of interoperability with other institutions, but the principle design and concept of structured document types based on underlying DTDs was kept and extended continuously.



- One or more document structure definitions; in fact, such a structure definition (DTD) is more or less a different representation of a schema.
  - A mapping of leaf elements to paragraph styles (actually, a paragraph style can have a different semantic depending on its context).
- Per document template, there is an XML file containing one or more so-called state machine definitions. Such a state machine controls the determination of the **document** structure by interpreting the sequence of paragraph styles used in the document (actually, the state transitions correspond to the paragraph styles).
- eNorm determines the document structure and validates it against the underlying structure by inspecting the sequence of paragraph styles used, and by matching the paragraph styles with the leaf elements of the document structure. This is achieved using a state machine (encoded in XML) where the transitions are implied by the sequence of paragraph styles.

*Notes:*

- eNorm's *Strukturdarstellung* command checks and presents the document structure using the above-mentioned state machines. In contrast to any generic parser, these state machines provide for better user guidance in case of any structural errors. In addition, using precalculated state machines is more efficient in terms of performance.
- For the sake of simplicity, we mentioned above just one INI and one XML configuration file per document type. In fact, these INI and XML files are organised hierarchically, so as to allow using re-usable building blocks. Consequently, configuration for a single schema can be spread over several INI and XML files, which can be shared by different schemas.
- eNorm documents are saved in Microsoft Word's .doc or .docx format; conversion to XML format is performed before promulgation, only. (The resulting XML format can currently not be re-imported to Word/eNorm).

### ***Supporting another structure***

The kernel of LegisWrite/eNorm is independent of the underlying structure. In fact, LegisWrite may be considered as an implementation of a simple structure regarding EU legislation, whereas eNorm is an implementation of a complex structure regarding German Federal legislation. The eNorm team considers that a straightforward implementation of Akoma Ntoso would be feasible using Microsoft Word along with the LegisWrite/eNorm structuring approach, as well. Development of support for xNorm is on-going, eNorm supports xNorm with regard to the main document types. An expansion is planned.

A single document template can be used for implementing different schemas. For example: the LegisWrite COM template is used for all legislative documents (featuring different structures, but a common style sheet). The actual schema to be used is defined during document creation and stored as a document property.

Implementing a new document type with a new schema requires some expertise. One important step consists in specifying and implementing the necessary document template and its corresponding paragraph styles (or re-using an existing one). Transforming the schema into the

INI representation is rather straightforward. Provision of the XML-based state machine for a new schema is fully automated.

### 6.1.3 Benefits and trade-offs

#### 6.1.3.1 User-friendliness

##### **User guidance and WYSIWYG**

Among the benefits of eNorm are the user guidance features (described in section 2), available from the interface, and the “WYSIWYG” interface: the user edits the text with the styles – not simply text content – which provides a view on the final document.

The use of a familiar working environment (word processor) is also seen as a benefit: *“With the aim of achieving tangible practical benefits as quickly as possible, the first step was to make the first eNorm software particularly user-friendly and ensure it could be employed immediately. In order to attain this, standard document templates and additional functions were developed which would be compatible with Microsoft Word, the most widely used word processing application within the Federal administration”*.<sup>29</sup>

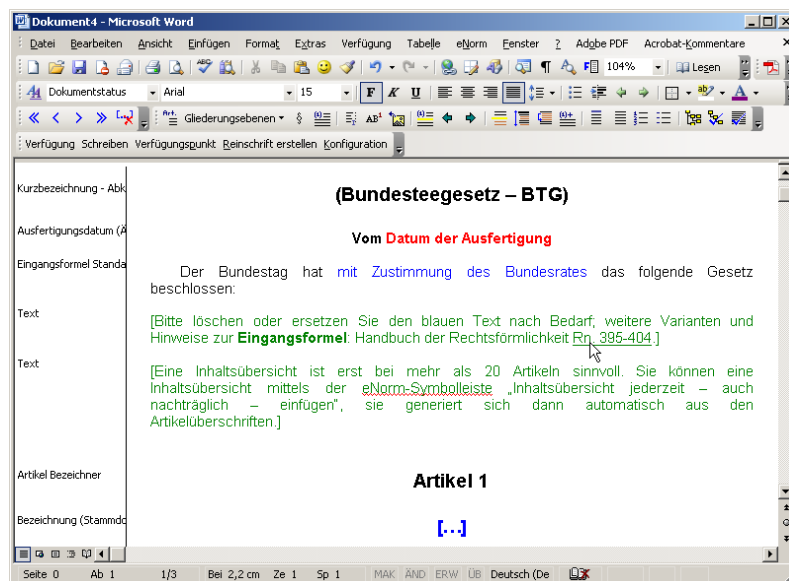


Figure 11 eNorm screenshot: editing in a word processor interface

##### **Freedom of drafting**

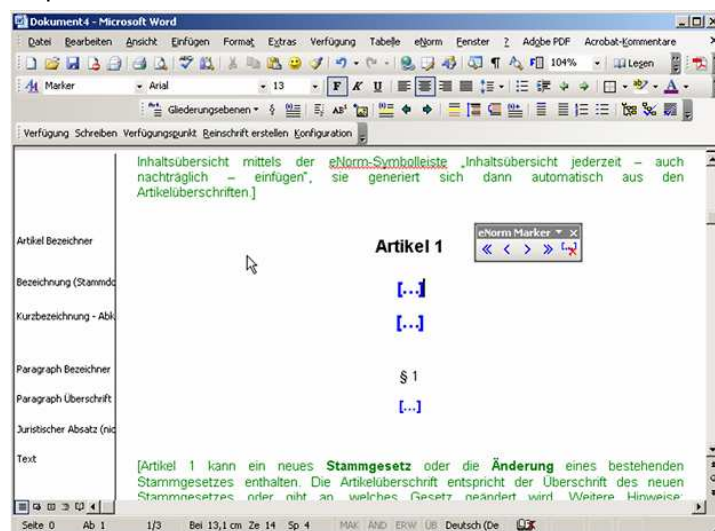
Although eNorm provides a structure with high granularity, the users mention that there is “a low level of constraint on the input” and precise that Microsoft Word users can use the tool virtually without restriction. This provides the users with a freedom of drafting – he may start writing text without needing to conform to a structure. Validation is not performed during drafting. Actually, eNorm documents are “pure” Microsoft Word documents, i.e. they can be revised in a non-eNorm environment with pure Word functionality. (Moreover, most of the eNorm commands can

<sup>29</sup> eNorm flyer [www.enorm.bund.de](http://www.enorm.bund.de)

be used for eNorm documents that do not conform to the underlying schema. There are, of course, some limitations regarding structure-related commands for such documents.)

This freedom of drafting fits the business needs in the legislative world – the users have to conform to tight deadlines, while drafting informally among internal teams before the document is uploaded in a system and/or transferred to the next stakeholder in the workflow. There is a need for flexibility of the tool; users are able to focus on content, and conform to the structure at a different point in time.

Freedom of drafting is provided by the a posteriori validation, but a few features of eNorm described in the user support provide additional freedom of drafting. One is the option of changing the space line for reading on the screen or reading print (by default, it is single line in the imposed structure). Another option is how to insert temporarily comments or information on “unfinished” sections in the text; eNorm support (FAQ section) explains that comments, suggestions or notes in a document can be added either by inserting a manual line break (the notes will then be on a dedicated line, not in an inappropriate place in the structure), or using the “blue anchor links” which can be inserted manually in the text – and signal areas where additional work is required.



**Figure 12 eNorm screenshot - anchor links**

The eNorm team is also investigating how to extend XML conversion for such documents so as to enable XML export also for structurally invalid documents (for drafting exceptions).

### ***Use of a word processor***

Building a legislative drafting tool using a word processor allows to benefit from existing functionality in word processors such as spell checking and grammar checking, and provide users with a familiar environment.

There is a trade-off for choosing to work with proprietary software: eNorm is dependent on Microsoft Word versions, and needs to be compatible with these different versions. The maintenance of the tool is therefore linked not only to the tool and users, but also to the evolution of the proprietary software – this can be referred to as “double maintainability”. Moreover, there is no control on the roadmap of feature development and maintenance of available features.

The choice of specific file formats may also be imposed (or changed) by policies and interoperability frameworks, and the same format is not necessarily used across the complete drafting process (for example: need to support Open Office).

### 6.1.3.2 Structure not imposed

According to the users' feedback, one of the drawbacks of this solution is that "Microsoft Word provides users with a high level of freedom for drafting, which sometimes affects the structuring of data". This does not mean that data is not structured, it emphasises the fact that the structure is not imposed. Although the structure used in eNorm is very granular, the user is not forced to conform to it, which leads to creating invalid documents. Some tools support editing in XML and enforce correct use of the structure.

eNorm maps styles to elements, which is an approach enabling the user to apply a structure, it does not enforce it. A benefit of eNorm is that the users do not feel there is a rigid imposed structure. However, the high degree of structuring in eNorm and its checks clearly indicate that a structuring is provided for eNorm documents (see Figure 13 eNorm screenshot: structure view and Figure 9 eNorm screenshot – structure elements and drafting phase).

The screenshot displays the eNorm interface with two main panes. The left pane, titled 'Strukturansicht', shows a hierarchical tree of document elements. The right pane, titled 'XML-Ansicht', shows the corresponding XML markup for the selected element.

**Left Pane (Strukturansicht):**

- Paragraph Bezeichner: § 1f
- Paragraph Überschrift: Bürgerliches-Gesetzbuchf
- Juristischer Absatz (Nr. 1): Das Bürgerliche-Gesetzbuch in der Fassung der Bekanntmachung vom 2. Januar 2002 (BGBl. I S. 42, 2909; 2003 I S. 738), zuletzt geändert durch (...), wird wie folgt geändert.
- Nummerierung (Stufe 1): 1. → § 1631b wird wie folgt geändert.
- Nummerierung (Stufe 2): a) → In Satz 1 werden die Wörter ist nur mit Genehmigung des Familiengerichts zulässig durch die Wörter bedarf der Genehmigung des Familiengerichts ersetzt.
- Nummerierung (Stufe 2): b) → Nach Satz 1 wird folgender Satz eingefügt.
- Nummerierung (Stufe 1): 2. → § 1666 wird wie folgt geändert.
- Nummerierung (Stufe 2): a) → Absatz 1 wird wie folgt gefasst.
- Nummerierung (Stufe 2): b) → Absatz 3 wird wie folgt gefasst.
- Nummerierung (Stufe 3): aa) Gebote, öffentliche Hilfen wie zum Beispiel Leistungen der Kinder- und Jugendhilfe und der Gesundheitsfürsorge in Anspruch zu nehmen.
- Nummerierung (Stufe 3): hh) Gebote, für die Einhaltung der Schulpflicht zu sorgen.
- Nummerierung (Stufe 3): cc) Verbote, vorübergehend oder auf unbestimmte Zeit die Familienwohnung oder eine andere Wohnung zu nutzen, sich in einem bestimmten Umkreis der Wohnung aufzuhalten oder zu bestimmende andere Orte aufzusuchen, an denen sich das Kind regelmäßig aufhält.
- Nummerierung (Stufe 3): dd) Verbote, Verbindung zum Kind aufzunehmen oder ein Zusammentreffen mit dem Kind herbeizuführen.
- Nummerierung (Stufe 3): ee) die Ersetzung von Erklärungen des Inhabers der elterlichen Sorge.
- Nummerierung (Stufe 3): ff) die teilweise oder vollständige Entziehung der elterlichen Sorge.
- Nummerierung (Stufe 1): 3. → Dem § 1696 Abs. 3 wird folgender Satz angefügt.
- Nummerierung (Stufe 1): 4. → In § 1712 Abs. 1 Nr. 2 werden die Wörter einschließlich der Ansprüche auf eine anteilige Unterhalts zu gewährenden Abfindung gestrichen.
- Paragraph Bezeichner: § 2f
- Paragraph Überschrift: Vorrang- und Beschleunigungsgebotf

**Right Pane (XML-Ansicht):**

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<document xmlns="http://www.isa.at/norm" >
  <h1 >Bürgerliches-Gesetzbuch</h1>
  <list type="ordered-decimal" >
    <list-item >
      <text >Das Bürgerliche-Gesetzbuch in der Fassung der Bekanntmachung vom 2. Januar 2002 (BGBl. I S. 42, 2909; 2003 I S. 738), zuletzt geändert durch (...), wird wie folgt geändert:</text>
      <list type="ordered-lower-alpha" >
        <list-item >
          <list-item >
            <text >1. → § 1631b wird wie folgt geändert:</text>
            <list type="ordered-lower-alpha" >
              <list-item >
                <text >a) → In Satz 1 werden die Wörter ist nur mit Genehmigung des Familiengerichts zulässig durch die Wörter bedarf der Genehmigung des Familiengerichts ersetzt.</text>
              </list-item>
              <list-item >
                <text >b) → Nach Satz 1 wird folgender Satz eingefügt.</text>
              </list-item>
            </list>
          </list-item>
          <list-item >
            <text >2. → § 1666 wird wie folgt geändert:</text>
            <list type="ordered-lower-alpha" >
              <list-item >
                <text >a) → Absatz 1 wird wie folgt gefasst:</text>
              </list-item>
              <list-item >
                <text >b) → Absatz 3 wird wie folgt gefasst:</text>
              </list-item>
            </list>
          </list-item>
          <list-item >
            <list-item >
              <list-item >
                <text >aa) Gebote, öffentliche Hilfen wie zum Beispiel Leistungen der Kinder- und Jugendhilfe und der Gesundheitsfürsorge in Anspruch zu nehmen.</text>
              </list-item>
              <list-item >
                <text >hh) Gebote, für die Einhaltung der Schulpflicht zu sorgen.</text>
              </list-item>
              <list-item >
                <text >cc) Verbote, vorübergehend oder auf unbestimmte Zeit die Familienwohnung oder eine andere Wohnung zu nutzen, sich in einem bestimmten Umkreis der Wohnung aufzuhalten oder zu bestimmende andere Orte aufzusuchen, an denen sich das Kind regelmäßig aufhält.</text>
              </list-item>
              <list-item >
                <text >dd) Verbote, Verbindung zum Kind aufzunehmen oder ein Zusammentreffen mit dem Kind herbeizuführen.</text>
              </list-item>
              <list-item >
                <text >ee) die Ersetzung von Erklärungen des Inhabers der elterlichen Sorge.</text>
              </list-item>
              <list-item >
                <text >ff) die teilweise oder vollständige Entziehung der elterlichen Sorge.</text>
              </list-item>
            </list-item>
          </list-item>
          <list-item >
            <text >3. → Dem § 1696 Abs. 3 wird folgender Satz angefügt.</text>
          </list-item>
          <list-item >
            <text >4. → In § 1712 Abs. 1 Nr. 2 werden die Wörter einschließlich der Ansprüche auf eine anteilige Unterhalts zu gewährenden Abfindung gestrichen.</text>
          </list-item>
        </list>
      </list-item>
    </list>
  </list>
  <h2 >Vorrang- und Beschleunigungsgebot</h2>
</document>
```

Figure 13 eNorm screenshot: structure view

### No a priori validation

"Legislative texts generated by eNorm can be converted into structured data. This is only possible if there is a consistent and correct use of eNorm".

A low level of constraint on the input affects the structuring of data. Two counter measures exist and are applied by eNorm: support /guide the users in applying the structure, and train the users to create valid documents.

The comprehensive features available in eNorm and the high granularity of the structure imply some complexity of use. *“Well trained users are not the problem for using eNorm. Users who have never used eNorm are more of a problem.”* There is extensive user support (tutorials, FAQ, hotline) and training is provided – a 4 hour training for a basic module covers the following topics: *Create documents and learn the first steps in the document - Using the tree representation - Edit - Number insertion and update and the revision mode, references from Juris [legal database], document quality check and error correction, presentation and export for printing, explanation of the manual.*<sup>30</sup>

There are structural errors, in some cases also semantic and formal errors and display errors detected. These can be partly corrected automatically, but errors can still be made, there is no automatic correction of all errors. This is for the following reason: some items detected are not always considered as being wrong, therefore some are labelled as “warning”, only. In addition, there are cases where it requires human intelligence in order to resolve the error, automatic correction would be too critical – e.g. the proper formulation of amending instructions.

This a posteriori validation of the document requires submitting it for validation, then applying the corrections – which adds a second step in the drafting process.

### ***Ensuring the use of the tool***

Benefiting from a structure includes ensuring a wide use of the tool supporting it. eNorm benefits from a large user base. eNorm was split off from LegisWrite in 2005 and is used since then by the Ministry of Justice. It was introduced in the German parliament in 2006. eNorm is also used by other administrations than the federal government and parliament: some German federal states are using eNorm already for similar and for different types of documents, and planning to extend the usage with additional features, or integration with different systems. Others are evaluating the possibility of using eNorm for their federal state government. The first federal state started using eNorm in 2009. In the German Parliament, eNorm is also used in the secretariats of parliamentary committees. At government level, eNorm is used in most ministries, among which are also ministries producing voluminous or a high number of bills (e.g. departments for justice, health, finances, environment, interior<sup>31</sup>). eNorm is used for many different types of texts e.g. for government proposals, laws and regulations, parliamentary communication, administrative directives, treaties, amendments and it is used to create consolidated texts.

However, the use of eNorm and its structure for drafting legislative proposals is currently mandatory inside the Ministry of Justice only. Since the Ministry of Justice is in charge of the adherence to the drafting rules and also of the promulgation, the usage of eNorm in the entire process is requested, but it cannot be enforced yet. Nevertheless, eNorm is used in most German Federal ministries and by most of the Parliament’s committees. Extending the use of eNorm to the entire legislation process could require the import of Open Office documents. The adaptation of eNorm to xNorm ensures the support of an open XML schema which is also a state standard, and therefore an enabler for a process-wide adoption of a common structure/tool.

<sup>30</sup> <http://www.enorm.bund.de/forum/viewforum.php?f=13&sid=24fece8f118ff4c58b26e9b83abfaf34>

<sup>31</sup> The ministry of interior affairs plans to distribute eNorm to about 3000 work stations.

### 6.1.3.3 Adapting the tool and the structure to users' new needs

eNorm is a tool which continuously evolves in order to map its features with users' needs.

#### ***Supporting legislative drafting needs***

Current users would like to see enhancements such as an improvement of the user-friendliness through a clearer user interface and context-sensitive help. Moreover, in the long-term, they would welcome a better automation of the creation of normative texts.

The eNorm team is working on a proof of concept which would allow *on the fly* document structure check (currently, this is available only if the user requests a structure check). There are currently no on-the-fly checks yet; to a large extent, the reason for this is that many different Word versions are to be supported at the same time, due to the heterogeneous environments in which eNorm is used. On-the-fly checks and interactive user guidance are discussed for the near future, e.g. context-sensitive guidance and hints, or a dynamically updated structure check.

#### ***Supporting exceptions***

The users would like an increase in the robustness of the application in terms of greater tolerance to user errors. The eNorm team is studying how to extend XML so as to enable XML export also for invalid documents in order to support drafting exceptions.

In the eNorm versions for federal government and parliament, there is a basic metadata handling. For one of the federal states using eNorm, advanced metadata integration was developed, in which various document metadata can be extracted and uploaded to a company providing legal services in Germany.

#### ***Support of a National standard - xNorm***

Among the adaptations to evolving needs, eNorm is working on importing and exporting XML files compliant with a new government standard for legislative texts (xNorm – an open structured data format), which will be used across the complete drafting process.

Users mentioned: *“the structured XML data can be incorporated into the printing / publishing programs and archiving, for further processing. In theory, 100% of the data can be exported, but in practice much less”*. There is an underlying DTD which is layout oriented, eNorm so far aimed at exporting data for publishing purposes, but little of the data is currently re-used by other applications.

The next phase (current objective since January 2011) is to check the introducing of the XML format (xNorm) throughout the *entire legislative process* (the tool will need to import XML files). The import and export of documents would cover base acts (not consolidated laws). The xNorm format can be exported and imported at any time, meaning it could replace storing eNorm documents (in .doc and .docx format) on the longer term. This will also allow exchanging documents from other word processors such as Open Office.

### ***“eNormisiering” of documents (legacy management)***

The team is also working on a feature which can be named eNormisierer, which will probably be ordered and developed in the course of this year. The main objective is to turn any text document into a proper Word/eNorm document by applying the proper paragraph styles. Most of the paragraphs of German legislative documents are numbered; consequently, eNorm-ising can be automatized up to a significant degree; by employing structural knowledge, also most of the unnumbered paragraphs can then be dealt with (semi-)automatically. In addition, the import of xNorm documents will facilitate the import of legal acts as soon as the federal database for the documentation of the law will be ready for xNorm support.

#### **6.1.4 eNorm: conclusion**

eNorm is a tool which has a comprehensive set of features and is user centred in its approach. While users sometimes understand that freedom of drafting generates some flaws in the structure, they also understand the benefits of a common structure to be used throughout the complete legislative process and are adapting the tool accordingly – with the implementation of xNorm. eNorm evolved to support importing and exporting of data in this National open standard for data structuring. The adaptation of eNorm to xNorm is therefore an enabler for a process-wide adoption of a common structure/tool.

The strategy for eNorm to support the users in applying the highly granular structure is based both on user guidance in the tool and support/training of the users. This approach – preserving the freedom of drafting – fits the legislative drafters’ needs, their first aim is to provide legislative content in a document and complying to a structure is yet another step in the drafting process. The evolving features of eNorm will try to provide better user support in the future by *on the fly* structure checks, which will suppress or minimize this extra step.

#### **6.1.5 Annex: eNorm DQC Criteria (Document Quality Check)**

The following table lists the DQC criteria available in eNorm and their individual checks; the lists of individual checks have been grouped together, where appropriate.

<b>Name</b>	<b>Type of check</b>	<b>Individual checks / errors detected</b>	<b>Repair available</b>
Structure check	Structure	<ul style="list-style-type: none"> <li>Adherence to the underlying schema; two types of errors: <ul style="list-style-type: none"> <li>Element is missing</li> <li>Element is not allowed in this context</li> </ul> </li> </ul>	(yes) <sup>32</sup>
Modified paragraph margins	Layout	<ul style="list-style-type: none"> <li>(Footnote) Margin is modified</li> <li>Margin in revision statement is modified</li> </ul>	yes

<sup>32</sup> Insertion of missing elements is feasible in LegisWrite, yet not in eNorm.



Superfluous whitespace	Layout	<ul style="list-style-type: none"> <li>• (Footnote contains) Multiple white space</li> <li>• (Footnote contains) Multiple white space at paragraph beginning</li> <li>• (Footnote contains) Empty paragraph</li> <li>• Empty footnote</li> </ul>	yes
Embedded objects	Technical / layout (graphics)	<ul style="list-style-type: none"> <li>• Embedded objects (graphics wrongly inserted) or other objects present</li> </ul>	
Modified numbering	Technical	<ul style="list-style-type: none"> <li>• Automatic numbering was removed or modified</li> </ul>	yes
Markers contained	Content	<ul style="list-style-type: none"> <li>• Placeholders that should be filled ([...]) are still contained in the document</li> </ul>	
Document outline and table of contents	Content & layout	<ul style="list-style-type: none"> <li>• Heading is missing</li> <li>• Headings are used inconsistently</li> <li>• Legal paragraph needs to be numbered (if there are several)</li> <li>• Legal paragraph mustn't be numbered (if there is only one)</li> <li>• More than five legal paragraphs</li> <li>• Element (table, graphics) not recommended at this position</li> <li>• Superfluous higher divisions</li> <li>• Table of contents missing (main act)</li> <li>• Table of contents missing (embedded act)</li> <li>• Division contains an addendum (e.g. "Kapitel 5a" instead of "Kapitel 5" – only valid in amending statements)</li> <li>• Legal paragraphs or sub items are not numbered consecutively (error in automatic numbering, or manual numbering misused)</li> <li>• Wrong numbering format used</li> <li>• More than one sentence (relevant for specific parts of the acts)</li> <li>• Article heading missing (for specific document types)</li> <li>• Wrong number of articles (some documents should feature only 2 to 4 articles)</li> <li>• Numbering should be restarted (misuse in numbering)</li> <li>• Wrong sequence of higher divisions</li> </ul>	



Character formatting changed	Layout	<ul style="list-style-type: none"> <li>• Different font used</li> <li>• Font size changed</li> </ul>	
Amending/revision statements	Content & layout	<ul style="list-style-type: none"> <li>• Revision statement not valid (there is only a certain set of allowed wordings)</li> <li>• Revision statement not well-formatted (style)</li> <li>• Revision of parts of a sentence is too long</li> <li>• Superfluous closing quotation mark behind revision statement</li> <li>• Superfluous leading quotation mark in front of revision statement</li> <li>• Levels of quotation marks wrongly used (in nested amendments)</li> <li>• Wrong punctuation in revision statement</li> </ul>	
Checks for enacting terms in different document types	Content	<ul style="list-style-type: none"> <li>• Name of laws is wrong ("Gesetz" must be contained)</li> <li>• Name of regulations is wrong ("Verordnung" must be contained)</li> <li>• Parentheses used in act title</li> <li>• Short act name or abbreviation required</li> <li>• Adoption formula not valid (there is a restricted set only)</li> <li>• Article title invalid</li> <li>• Order of articles on publication and entry into force invalid</li> <li>• Article of entry into force malpositioned</li> <li>• Time precisions incomplete</li> <li>• Closing sentence wrong ("Bundesrat" must be part)</li> <li>• Closing sentence wrong (different criteria)</li> <li>• Short name of act missing (if act title larger than 5 words)</li> <li>• Act title of embedded act invalid</li> <li>• Short name of act not in parentheses</li> <li>• More than one word in short name</li> <li>• Abbreviation for act missing (for certain document types)</li> <li>• Abbreviation contains special characters</li> <li>• Short name does not obey to naming rule</li> <li>• Abbreviation does not obey to naming rule</li> <li>• More than one separator in short name element</li> <li>• No short name or abbreviation</li> </ul>	

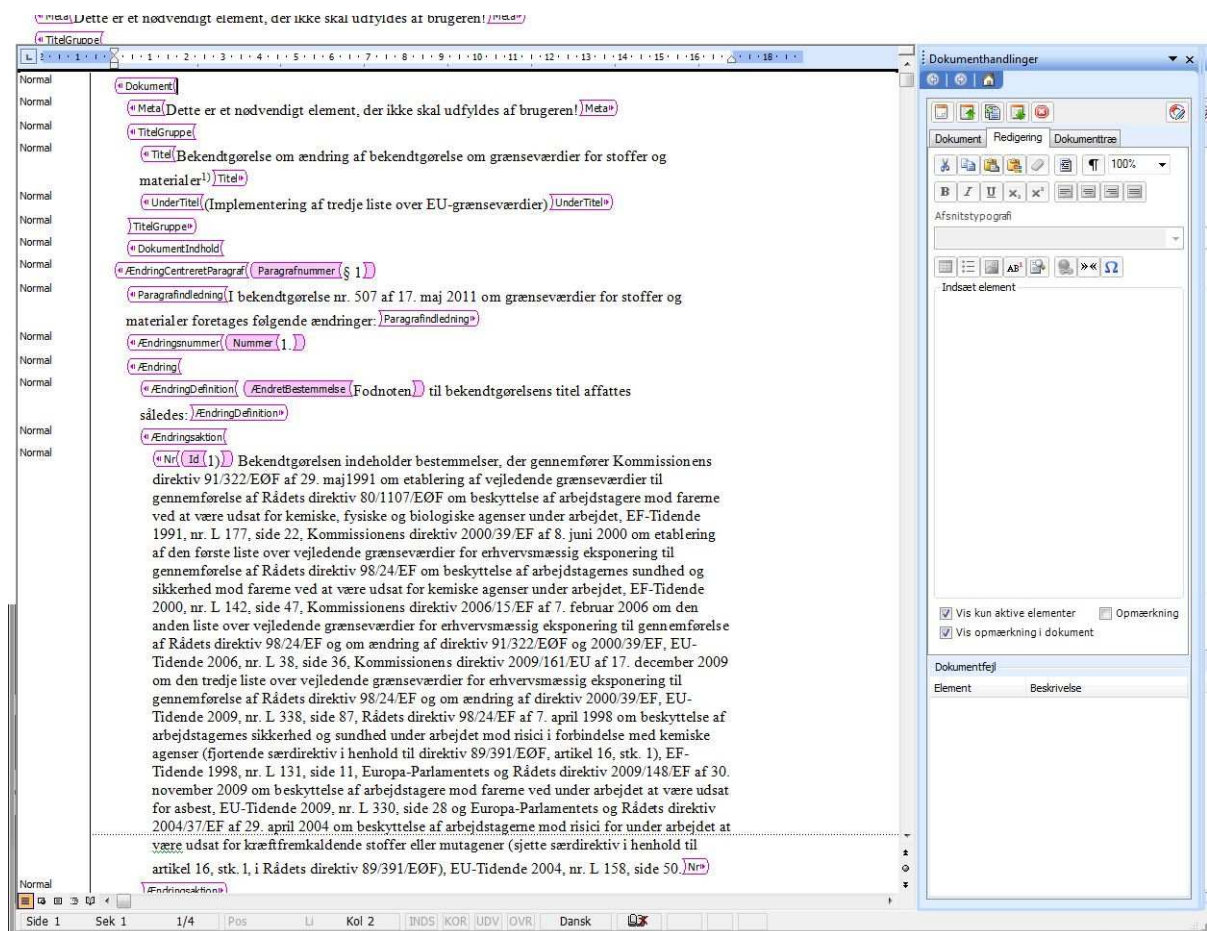
		<ul style="list-style-type: none"> <li>allowed (certain document types)</li> <li>Short name/abbreviation not formatted correctly</li> <li>Inconsistency between cover page and act title</li> <li>Inconsistency in short name between cover page and act title, or one of them missing</li> <li>Missing separator between short name and abbreviation</li> <li>Embedded base act may not contain entry into force</li> </ul>	
Wordings	Content	<ul style="list-style-type: none"> <li>Plural "Worte" (engl. words) shouldn't be used</li> <li>Abbreviation not allowed</li> <li>Date wrongly formatted</li> <li>"Euro" wrongly arranged/spelled</li> <li>Sentence too long (55 words)</li> <li>Word too long (30 characters)</li> <li>Superfluous decimals</li> </ul>	
Citations	Content	<ul style="list-style-type: none"> <li>Higher divisions must be mentioned in singular</li> <li>Abbreviations within citation</li> </ul>	
Legal links	Content	<ul style="list-style-type: none"> <li>Legal link to other act invalid (check via web service)</li> </ul>	yes
Meta data	Content	<ul style="list-style-type: none"> <li>Mandatory elements not filled</li> </ul>	
Inner cross references	Content	<ul style="list-style-type: none"> <li>Target does not exist any more</li> <li>Reference to sentence may be outdated</li> <li>Reference could not be updated automatically</li> </ul>	

#### Notes:

- The repair procedures can be run automatically or interactively.
- Repairing any change in the style sheet is performed automatically upon launching a document quality check.

*This case study is based on replies to the questionnaire and expert interview (involving the owners of the Lex Dania system at the Ministry of Justice), as well as on literature research.*

Lex Dania editor allows editing XML (Microsoft WordML) files using a word processor interface (Microsoft Word). The user is guided through the structure by inserting the content in between the user-friendly tags (Figure 14 Lex Dania editor interface screenshot). The structure is based on a complex XML schema (Lex Dania XML) and related user friendly schemas and templates implementing user needs, and hiding from the user the parts of the structure which he does not need to see. The editing is not WYSIWYG, in the sense that it does not provide the user with an immediate view of the content of his text with the final layout while he is drafting. The user is guided to focus on the content of his text.



**Figure 14 Lex Dania editor interface screenshot**

In order to understand the advantages of this solution in terms of user friendliness and user's freedom in creating text and templates, this case study analyses in a **first section**, the Lex Dania schema, the creation of templates and user friendly schemas, and the way the tool validates the document according to the schema.

A **second section** describes the benefits and corresponding trade-offs of the tool, considering aspects such as solution strategy (sustainability, maintainability ...) and user buy-in (training, governance model for "enforcing" the use of the structure/ tool, ...).

## 6.2.2 Lex Dania schema and user interaction

### 6.2.2.1 Lex Dania XML schema

#### *Introduction*<sup>33</sup>

**Lex Dania** XML<sup>34</sup> is an XML meta-schema used to compose other, more specific, XML schemas. In this manner, it provides a data-model for Danish Legislative Documentation. The purpose of Lex Dania is to provide a common syntax and addressing scheme for the multitude of Danish legislative document types. This data-model provides many advantages, such as being able to re-use components, which facilitates the writing of many schemas. It also makes the interchange of information easier between schemas as well as providing a basis for interoperability.

The meta-schema Lex Dania is used as a set of "building blocks" and, at present, consists of 5 XML schema modules: addressing, content, references, metadata, and structure. The data-model is (similar to MetaLex) a three tiered structure: a) the meta-schema (Lex Dania), b) Omnischemas (middleware) and c) specific legislative schemas.

#### *Analysis*<sup>35</sup>

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.

The most significant contribution of the Lex Dania 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 mechanism of elements and data types according to an object-oriented development philosophy. The inheritance mechanism allows incrementing the semantics and the details of the data types for each layer in Lex Dania schema system.

<sup>33</sup> From: <http://www.lri.jur.uva.nl/standards2003/LexDania-handout.pdf>

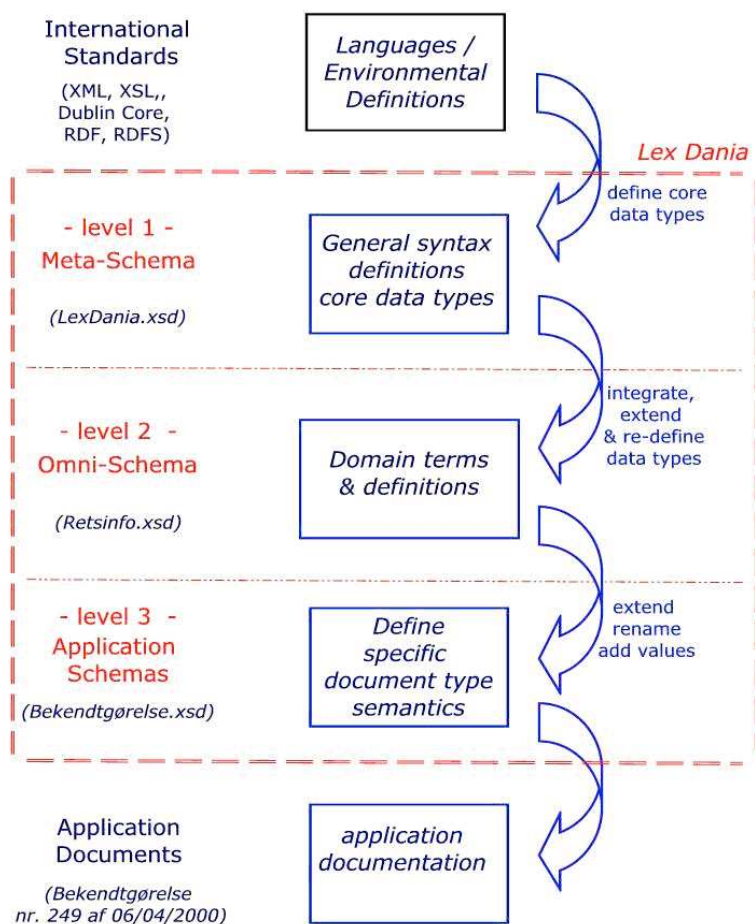
<sup>34</sup> Schemas available at <https://www.retsinformation.dk/offentlig/xml/schemas/>

<sup>35</sup> From: <http://www.estrellaproject.org> Deliverable 3.1 General XML format(s) for legal Sources - Caterina Lupo et al. University of Amsterdam - Estrella Project - European project for Standardised Transparent Representations in order to Extend Legal Accessibility

On the other hand, including restrictions as regards data types increases control over document contents but, as a drawback, it can represent a burden for the drafting.

The fundamental syntactic details are defined in a single schema Lex Dania.xsd - sometimes referred as the meta-schema. The intention is that the data types of the Lex Dania meta-schema will be used as the basis for defining specific legislative document schemas (using the XSL inheritance constructions). In other words, the schema Lex Dania.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.)

The Lex Dania 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 15 Lex Dania XML Schema structure).



**Figure 15 Lex Dania XML Schema structure**

## Granularity of Lex Dania structure

In terms of granularity of the structure, in Lex Dania editor, there are only low level and high level documents. Low level documents cover loose documents and annexes; all other documents have a high level of granularity.

An example of high level granularity is the amending act (see Figure 16). In the example, there is:

AmendmentNumber(8), containing an Amendment, that contains AmendmentDefinition and AmendmentAction.

The AmendmentAction can contain all types of structure, in this example it is an Article, that contains Sections

Normal	<code>&lt;Endringsnummer&gt;</code>
Normal	<code>&lt;Endringsnummer&gt; (Nummer (8.))</code>
Normal	<code>&lt;Endring&gt;</code>
Normal	<code>&lt;EndringDefinition&gt; (&lt;EndretBestemmelse § 8&gt;) affattes således: &lt;EndringDefinition&gt;</code>
Normal	<code>&lt;Endringsaktion&gt;</code>
Normal	<code>&lt;Paragraf&gt; (&lt;Paragrafnummer § 8&gt;)</code>
Normal	<code>&lt;Stk&gt; Pligten til fast bopæl ved erhvervelse af en landbrugsejendom, jf. § 12, §§ 14-16 og § 20, skal opfyldes af en erhverver i 10 år. Tilsvarende gælder ved erhvervelse af jord til supplering af en landbrugsejendom, jf. § 29. Pligten skal opfyldes ved, at erhververen eller landmanden i et selskab efter lovens § 20 tager fast bopæl på den pågældende landbrugsejendom eller på en anden landbrugsejendom, som den pågældende eller dennes ægtefælle eller samlever er ejer eller medejer af. Boligen skal være hovedbopæl i skattemæssig forstand. &lt;Stk&gt;</code>
Normal	<code>&lt;Stk&gt; (&lt;Stknummer&gt; (Stk. 2.)) Bopælspligten efter stk. 1 kan endvidere opfyldes ved, at en anden person end erhververen eller landmanden i et selskab efter lovens § 20 tager fast bopæl på ejendommen eller på en anden landbrugsejendom, som erhververen er ejer eller medejer af. Erhververen kan i så fald kun opholde sig på ejendommen, når dette sker i naturlig tilknytning til ejendommens drift. &lt;Stk&gt;</code>
Normal	<code>&lt;Stk&gt; (&lt;Stknummer&gt; (Stk. 3.)) Ministeren for fødevarer, landbrug og fiskeri kan fastsætte regler om kravet om fast bopæl og om, hvornår en person kan betragtes som samlever. &lt;Stk&gt;</code>
Normal	<code>&lt;Stk&gt; (&lt;Stknummer&gt; (Stk. 4.)) Ministeren kan dispensere fra stk. 1 og 2, når særlige forhold taler for det. &lt;Stk&gt;</code>
Normal	<code>&lt;Stk&gt; (&lt;Stknummer&gt; (Stk. 5.)) For erhvervelser, jf. § 12, §§ 14-16 og § 20, hvor bindende købsaftale er indgået inden 1. april 2010, gælder dog fortsat, at pligten til fast bopæl skal opfyldes af en erhverver i 8 år. &lt;Stk&gt;</code>
Normal	<code>&lt;Paragraf&gt;</code>
Normal	<code>&lt;Endringsaktion&gt;</code>
Normal	<code>&lt;Endring&gt;</code>
Normal	<code>&lt;Endringsnummer&gt;</code>
Normal	<code>&lt;Endringsnummer&gt; (Nummer (9.))</code>
Normal	<code>&lt;Endring&gt;</code>
Normal	<code>&lt;EndringDefinition&gt; (&lt;EndretBestemmelse § 8 a&gt;) ophæves. &lt;EndringDefinition&gt;</code>

Figure 16 Structure of an amending act in Lex Dania

An example of a loose document in terms of granularity of the structure, is an annex to a document (see Figure 17). There is an element FormattedText in which the user can add word styles according to a preselected list.



	trådte i kraft den 1. januar 2008. Den foreløbige elpatronordning for el udløber således ultimo 2011.
Tekst9	Elpatronordningen er nærmere beskrevet i forslaget til elpatronloven, Forslag til Lov om ændring af forskellige miljø- og energilovslove (Afgiftslempelser på fjernvarme m.v. som led i finanslovsaftalen for 2006, afgiftslempelser på kvælstof i lastbilers miljøfiltre samt mikrokraftvarme samt justeringer af forbrugsregistrering m.v.). Forslaget blev fremsat af skatteministeren den 16. november 2005.
Tekst9	Det fremgår af de almindelige bemærkninger, afsnit 2.1., til forslaget til elpatronloven fra 2005 (L 81 fremsat den 16. november 2005 af skatteministeren), at energi- og CO <sub>2</sub> -afgifterne på el og brændsler til fjernvarmeproduktion uden samtidig elproduktion ved forslaget reduceres ned mod niveauet for brændselsbaseret kraft-varme for fjernvarmeleverandører med kraftvarmekapacitet. Det fremgår videre, at med elpatronordningen rettes der op på tidligere uhensigtsmæssigheder, og at der opnås både en væsentlig samfundsøkonomisk gevinst og væsentlige miljømæssige fordele i form af lavere emission af CO <sub>2</sub> , SO <sub>2</sub> og NO <sub>x</sub> . Det skyldes reduktion af de tidligere afgiftsmæssige incitamenter til en for stor elproduktion på kraftvarmeværkerne samt oget incitament til fornuftig indenlandsk anvendelse af elproduktionen frem for uøkonomisk eksport af el. Der henvises til den nærmere gennemgang af elpatronordningen i nævnte forslag til elpatronloven.
Tekst9	EA-Energianalyse og Risø-DTU har den 9. juni 2009 færdiggjort ovennævnte uafhængige analyse i form af rapporten: Bedre integration af vind. Analyse af elpatronloven, treledstariffen for mindre kraftvarmeanlæg, afgifter og andre væsentlige rammebetingelser.
Tekst9	På baggrund af analysens anbefalinger og med henblik på bedre at integrere vind i energisystemerne har regeringen (Venstre og Det Konservative Folkeparti), Socialdemokraterne, Dansk Folkeparti, Socialistisk Folkeparti og Det Radikale Venstre indgået nævnte aftale, som bl.a. vedrører ordningen i elpatronloven. Denne aftale har følgende indhold:
BemTilV	1. Elpatronloven gøres permanent.
BemTilV	2. Krav om ejerskab lempes så ordningen også gælder:
Normal	« Opregning »
Pind9	« Pind (Id - ) Fjernvarmeværker uden egen kraftvarmekapacitet, der leverer til fjernvarmenet, hvor hovedparten af varmen kan forsynes fra kraftvarmeanheder. »
Pind9	« Pind (Id - ) Virksomheder med kraftvarmekapacitet, hvor varmen bruges i samme interne vamesystemer. »
BemTilV	» Opregning »
BemTilV	3. Skatteministeren fremlægger inden den 1. juni 2010 en redegørelse om muligheder for og virkninger af ændrede afgifter på elektricitet med særlig henblik på bedre integration af VE (dynamiske afgifter).
BemTilV	4. Klima- og energiministeren fremlægger inden den 1. juni 2010 en redegørelse om muligheder for og virkninger af ændrede afgifter på elektricitet med særlig henblik på bedre integration af VE (dynamiske

Figure 17 Structure of an annexe in Lex Dania

### 6.2.2.2 Lex Dania in the European context

#### **Lex Dania editor and other schemas - some re usability**

The concept of having user-friendly schemas on top of very advanced schemas below can be re-used, and the general architecture of the editor as the editor is customised by schemas. So everybody could more or less use the editor if schemas, and user friendly schemas and the conversion between the two different schemas are made.

Based on the understanding of the concept, the architecture used for Lex Dania could be used for example with the Akoma Ntoso schema, but this would need to be validated by a proof of concept.

#### **Compliance to a common standard for a structure across the EU – no use case**

Most of the Lex Dania XML schemas were finalised in 2004, and at that time Lex Dania experts were cooperating with both Leibniz Center for Law of the University of Amsterdam (initiators of

MetaLex) and Cirsfid of the University of Bologna (initiators of Akoma Ntoso), but Lex Dania was working on the field with a limited budget and a fixed dead line, so Lex Dania schemas had to be stable before both Metalex and Akoma Ntoso were stable.

Moreover, there has not been so far any use case or business case in the Lex Dania project proving the necessity of been compliant on a schema-level across different legal systems in the EU. If a law from another Member State needs to be accessed, it is more with an approach of a global framework of reference: including the experts of this law, the doctrine, and the complete final text of law (readable by humans, not by machines). Accessing the contents of the law in XML from outside Denmark is not needed.

Lex Dania project experts participated in a study organised by the European Forum of Official Gazettes: “ Use of XML for the production and distribution of the official gazettes” - Final Report of the Forum Working group on XML. The conclusions of this study mention:

*“A common vocabulary on structure would not be successful from two different reasons:*

- 1. Important differences in legislative culture causing different structures in legislative documents [for example, some “legislative cultures” do not require the publication of the “considering” (preamble) or basis of the act]*
- 2. The advanced status of XML based projects in the different countries leaving no possibility to introduce a new schema on structure.”*

Moreover, a common structure could be considered in the context of EU Directives; but even then, this may not be useful. Occasionally, the laws of a member state may already comply with the outcome of the implementation of the directive, and the state involved would be required only to keep its laws in place. More commonly, member states are required to make changes to their existing laws (commonly referred to as transposition) in order for the directive to be implemented correctly.

However, some Member States do not have a schema. MetaLex may be used as a basis for creating a more detailed and specific schema. One may also build upon an existing, more specific, MetaLex compliant schema , (such as Akoma Ntoso) and prune undesirable elements and add desired ones according to the legislative culture of the country.

### ***Towards a common standard for identifying European Legislation***

Experts across Europe are currently working at a EU level on a reference framework / a common system for the identification (URI) and metadata of law– the European Legislation Identifier (ELI).

This initiative is in line with trends following the implementation of the PSI Directive (Public Sector Information), which requires the publishing of data (including legal texts) generated by Public Sector. Data published in this context is very often available under open standards, such as Web standards and linked data. Creating a European Legislation Identifier follows this trend, and enables interoperability among different existing systems.



Currently, Lex Dania has defined all its metadata<sup>36</sup> in English, and will support the ELI standard. Lex Dania gives free access to all XML documents for re-use by a set of web services thus already fulfilling the PSI-directive.

### 6.2.2.3 User-friendliness

#### ***User-friendly schemas and document templates for specific user group***

Lex Dania solution allows having user groups specific document templates - the ministry can make these specific alternations within the limits of the more general XML Schemas. Lex Dania XML is a set of schemas, both very abstract and specific so not easy to understand for drafters and up-markers; therefore user-friendly schemas were created. The term user-friendly schemas means that in these schemas, a lot of the technical aspects in Lex Dania XML- necessary for the workflow/production/styling/presentation/reuse/indexing - is not visible for the end user. There are 36 user-friendly schemas and 36 document types. In order to impose the correct use of the schema, user-friendly schemas ensure that only allowed elements are active and mandatory elements are prompted in the user interface. User-friendly schemas are created by mapping the schema to the specific user vocabulary; this requires two types of expertise: the knowledge of the business and the understanding of the schema. Specific experts with this "double" knowledge manage these tasks.

Specific work templates are then created compliant to the user-friendly schemas; the templates are different representations of the user-friendly schema, based on the different users' needs. Templates (XML files) are saved in a designated read only folder on every desktop. These templates must be valid, and the validation against the user-friendly schema is made in a generic XML editor. Validation of the document according to the general schema is then made through the Lex Dania editor on upload or export. The new templates appear in the "New Document" choice list. There is a short way for creating a new document template by creating a specific Lex Dania editor document and removing specific parts (leaving template parts). Template creation is also managed by experts.

#### ***Freedom of drafting***

The user-friendly schemas ensure the legislative structure, so only allowed elements are active and mandatory elements are prompted. User-friendly schemas are generally speaking strict, thus not allowing the end user to be creative regarding the structure. However, the user can benefit from a freedom of drafting by saving his work on his desktop. The document is on-the-fly validated against the user-friendly schema and whenever the document or a valid section of the document is previewed either in HTML or PDF the document is validated against Lex Dania XML.

---

<sup>36</sup> [https://www.retsinformation.dk/offentlig/xml/schemas/2011/09/26/Meta.LexDania\\_2.1.xsd](https://www.retsinformation.dk/offentlig/xml/schemas/2011/09/26/Meta.LexDania_2.1.xsd)

With regards to the formatted appearance of the text, the tool focuses only on the content and on the structure. The text is not formatted in any style during the drafting. If the user needs to edit formatted text, there is an option for this in the FormattedText element, the user can add word style according to a preselected list.

## WYSIWYG

The aim of editing content in XML is to allow options for publishing the data according to any applied style. The user does not see any formatted text. However, he may at any point in his drafting see the content of his work without the tags (see Figure 18).

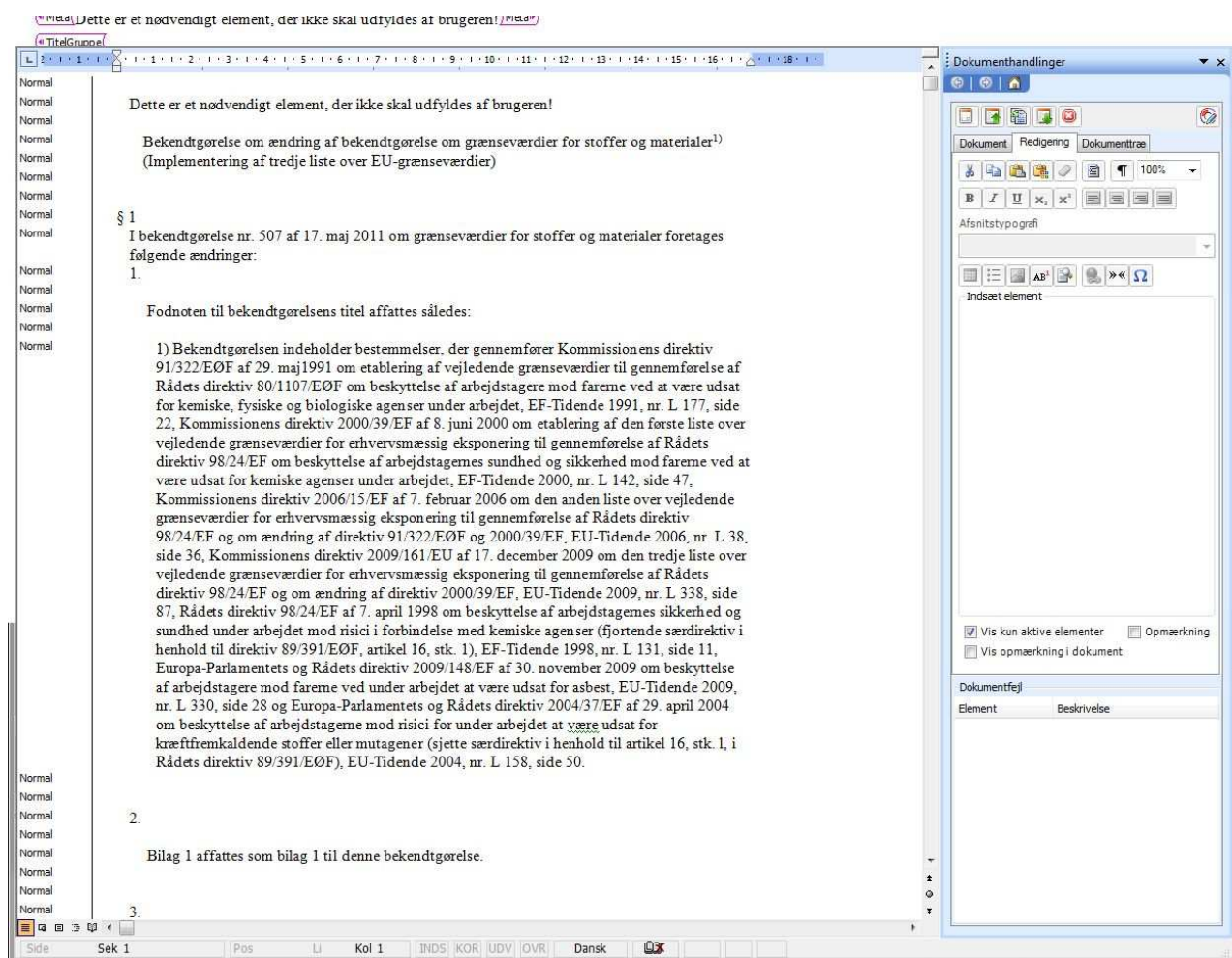


Figure 18 Lex Dania editor – view document without mark-up tags

The user can at any time have a preview either in HTML both off line and on line or in PDF (which then shows his text with the layout needed for publishing in the Official Journal) only on line. The content is checked for validation against the schema before the PDF and html files are generated.

## 6.2.3 Lex Dania editor: benefits and trade-offs

### 6.2.3.1 Reduced errors

With the use of this new version of the editor – ensuring correct use of the structure by editing the content in the structure in WordML, with user-guidance and user-friendly schemas, errors have been reduced by 90% compared to the former version of the editor.

#### ***Applying a structure - a priori validation***

The former version of the Lex Dania tool was based on another approach: the users were applying styles to the text in order to apply a structure: one formatting style was converted into one element. The helpdesk workload was extremely high, strong support was needed to ensure validity of the document. The old converter also only makes a posteriori validation (no guidance on the structure) hence the drafter/up-marker was in many cases left only with trial-and-error when making valid documents. Moreover, the concept of mapping an element to a formatting style is difficult to implement, as this is not completely the case in real life, so the converter was very complex and difficult to maintain. The old editor was legacy software from the old data capturing system, which didn't use XML and structure.

This new version of the editor ensures a priori validation, strong control of documents as well as high quality of documents. There is less support needed for drafters and up-markers. Moreover, this version ensures much better control and support of tables and pictures. It also ensures a good level of performance with large documents. The tool allows a faster up marking of existing documents, as it automatically recognises sections on selection e.g. by selecting an article it automatically marks up the article, number, section.

#### ***Learning to use a structure***

In the legal document drafting world, there is a culture of focusing on styles (Arial Bold 16) rather than on content (a title); this is legacy from the previous editor. Users have learned that presentation is everything i.e. it is important that the text is Arial, Bold, Centred, 16 pt. . The use of the new version of the editor requires a one-day training course, with the prerequisite that the user has experience with the previous version, and knowledge of legislative technique is required.

The new way of working in the structure creates a level of abstraction for the users; they do not have a WYSIWYG interface when drafting: they not see immediately their text in a publishing layout, but see it in the different elements of the structure. The user needs to overcome the “fear of these structure elements” in order to use the tool as fluently as the previous version.

#### ***Broad variety of users***

The efficiency of the legislative drafting process is linked to the fact that Lex Dania has a large user base. The Lex Dania editor is used by 400 drafters. The users are:

- Drafters using the editor, when they draft the document from scratch (civil servants in parliament and in the government).
- Up-markers using the editor to mark up all ready drafted documents (pre-press contractor - if the task is out sourced and office staff/students).

### ***Adapting the structure to users' needs***

A trade-off of applying a structure and having a large user base is the need to adapt it to all the users' needs. As mentioned in the case study previously, specific expertise is required to create user-friendly schemas and templates; this cannot be done by any user. As an example, a specific case for Lex Dania was that the editor and the templates/user-friendly schemas couldn't be used for a consolidated act for the Faeroe Islands. A new schema and template were generated, for a budget of around ten thousand euro. Sixty five hours of work were needed to cover the complete production process, from generating the schema and template to adapting the workflow to the new document type and testing the system.

### ***Ensuring the use of the tool and the structure***

There is no enforced policy for using the new version of Lex Dania editor; users can still use the old editor based on formatting styles but they are strongly encouraged to use the new Lex Dania editor both for quality and production time. However, only Lex Dania XML valid documents (either from the old editor or Lex Dania editor) can be published and promulgated, the workflow system ensures that, and all ministries are obliged to use the workflow system - this was decided by the Minister of Justice in a secondary legislation. Ministries who do not want to use the editor can use a contractor thus outsourcing the mark-up, but they need to pay for the work.

#### **6.2.3.2 Using word processors**

### ***Functionality from Word processors***

Building a legislative drafting tool using a word processor allows to benefit from internal XML formats – in the case of Lex Dania, the editor uses WordML, a Microsoft Office XML format/schema for Microsoft Word2003. Microsoft Office XML formats are stored as plain single monolithic XML files. Embedded items like pictures are stored as binary encoded blocks within the WordML.

The tool also benefits from existing functionality in word processors such as spell checking and grammar checking, but to a certain extent, as Lex Dania editor currently makes no use of the track changes – the amendment is a type of template and there is currently no consolidation supported<sup>37</sup>.

ODF and the use of Open Office Writer was considered, but the choice was made of using of Microsoft Word based on the solution meeting best the needed functionality (the amount of references to pictures in an Open Office document is limited to a number which does not meet

---

<sup>37</sup> In future developments, for which Lex Dania editor is prepared, Lex Dania will introduce with the Parliament semi-automatic consolidation of proposed amendments. This is done by the structure of amendments that contains "definition" and "action", where the definition states "where" and "how", and the action contains "what".

potential need of Lex Dania users). Moreover, if Lex Dania were to develop the functionality in Open Office, the needed expertise was not found to be easily available in Denmark.

### ***Formats and Vendor lock-in issues***

There is a trade-off for choosing to work with proprietary software: Lex Dania editor is dependent on Microsoft Word 2003, and will need to be updated to Microsoft Word 2010. The maintenance of the tool is therefore linked not only to the tool and users, but also to the evolution of the proprietary software – this can be referred to as “double maintainability”.

Moreover, there is no control on the roadmap of feature development and maintenance of available features. For example, the editor provides “pink tags” – a more user friendly version of XML tags. These user-friendly tags are - according to the MS roadmap of today - not supported by Microsoft Word anymore in 2018.

Using open source tools may provide more visibility and control over the roadmap and feature development – although it is not straightforward to find resources for this development, but the example of Open Office which “forked” into Libre Office when Oracle bought Sun Microsystems shows that some unpredictability also exists with non-proprietary solutions.

The choice of specific file formats may also be imposed (or changed) by policies and interoperability frameworks, and the same format is not necessarily used across the complete drafting process (interoperability frameworks may not apply to parliament for example). In the specific case of Lex Dania, the choice of using MSWord is based on a solution tendered for specific needs, and the choice of a format was based mostly on the eGovernment user need; this implies that even if all government users have a specific solution for their desktop environment/word processor which is not Word 2003, they will also use Word 2003 because it is part of the Lex Dania solution.

#### **6.2.4 Lex Dania conclusion**

This case allowed to understand the benefits and trade-offs of the Lex Dania tool . While meeting all the users’ needs in terms of document types, and ensuring a high efficiency of the drafting process, this approach needs to face other challenges such as adapting the structure to users’ needs (a common challenge when applying a standard to the “real world”) and maintaining the tools based on proprietary software.

The approach based on user-friendly schema and templates guide the user in a most efficient way through a complex XML schema structure. The user has some drafting freedom because he may save his work in a non-valid document, but once he uploads the document in the workflow system (mandatory for use for all legislation), the document must be valid according to the schema. The user is not obliged to use the Lex Dania editor, he has the option to outsource (and pay) the conversion of his document into a valid XML document.

While the Lex Dania editor benefits from existing functionality of word processors, there are still features which need to be developed in order to provide State-of-the-Art functionality supporting the legal drafting process (ex: support of consolidation, versioning).

## CASE STUDY 1 CONCLUSION

While both tools address user guidance through a structure in a similar way, one focuses more on extensive user guidance through functionality, and the other more on presenting the structure in a user friendly manner with user friendly schemas. Both structures applied are highly granular. While freedom of drafting is still more prevalent in one solution, the tool focuses on enhancing structure compliance during drafting by developing functionality to support the user. The other tool aims at driving the user directly into the structure, and in doing so enforces some “a priori” validation - its use reduced drastically the structural error rate. However, users are not imposed to use the tool and can outsource (and pay) the conversion to the XML schema. Both tools face common challenges such as vendor lock-in, maintenance and constant adaptation to users' needs.

---

## CASE STUDY 2: “XMLISATION”

This second case study focuses on describing the management of legacy documents (“XMLisation”) in various cases found during the LEOS study. These illustrate four different solutions and their approaches - and do not intend to compare them - as they are linked to the needs, context, legacy and aim of each project.

“Mark-up” or “XMLisation” is the activity of marking up a text with a set of defined tags assigning specific meanings to specific text partitions, enriching the text with semantic information (or metadata) that describes it. Some mark-up tools presented (Norma Editor, xmLegesEditor) help the end-user mark-up a legal text in a transparent way, that is, without requiring any knowledge of tag set (the DTD or the XML schema) or any knowledge of the technology involved (XML). Other tools are part of a complete process and are handled by dedicated teams.

A first case is the solution used by the Dutch Publication of the Official Gazette, for which all documents provided in various formats are XMLised through a four step process, which includes various schemas.

The second case is the solution provided by the European Parliament: the online web tool AT4AM supports the creation of amendments at the European Parliament on existing documents in all linguistic versions – in XML format, and the “XMLisation” is carried out beforehand by the official journal unit of DG ITEC (Directorate General Innovation and Technological Support) of the secretariat of the European Parliament.

The third case Norma Editor - a tool used for the mark-up (XMLisation) of legislative documents by the Italian Official Gazette. It is a Microsoft Word add-in for Microsoft Windows XP, which provides mark-up of the structures of documents with automatics tools, automatically recognise and mark-up normative references, and create an XML version of document, for different XML schemas.

The fourth case is an Open Source visual XML editor for the desktop – xmLegesEditor - which provides an “import” function of legal texts and marks up the document with the Italian National XML standard (NIR). This tool is in use in Regional projects to provide mark-up of consolidated legislative texts, and has been positively evaluated for use at the Italian Senate.



## 6.3 SDU<sup>38</sup> XMLISATION PROCESS

*The information provided for this case is collected through desk research, attending conferences such as the ODF Plugfest<sup>39</sup> and interviewing an expert from Lunatech Research working for Sdu<sup>40</sup>.*

### 6.3.1 Overview of the publication workflow of legislative documents

This section presents a basic picture of the overall publication workflow of legislative documents, of which the XMLisation (or kraken as this process is called within Sdu) of the original documents is an --important-- step.

The raw text of the documents is written by different public administrations, national and local alike, that are involved in drafting the legislative documents and the Registry of the parliament for parliamentary texts. Most documents are produced in some Microsoft Word format version, but there are some notable exceptions, like tagged-ascii or in some cases pure XML. Contractual obligations state that Sdu must be able to handle all these formats. From Sdu's perspective they are considered as the external producers of the raw documents to be published.

When producing entities need to publish a document, they send it to the central service point, which falls under the authority of the Dutch State and which is known as "Digitaal Loket". From there Sdu collects them as publication orders and processes them according to the requested services. No authoring is done by Sdu. On the other hand, Sdu may, if such a service is requested, perform syntactical textual correction (in Dutch "verbeteren kennelijke fouten" – correct obvious mistakes), but this is done after XMLisation (with some exceptions for syntactic corrections like correcting 'quotation marks').

When all processing is done, all publication formats for a document have been finalised, they are returned to the central service point. The documents are then officially published (and sanctioned) under control of this central service point, which implies that they are released on the website <http://www.officielebekendmakingen.nl> (see Figure 19, Figure 20, Figure 21).

---

<sup>38</sup> SDU is a publisher in the Netherlands. <http://www.sdu.nl/>

<sup>39</sup> <http://opendocsociety.org/news/gouda-odf-plugfest/>

<sup>40</sup> <http://www.sdu.nl/>



The guide to all government information and services

**Overheid.nl** [High contrast](#)

**Regulatory**

[Home](#) [Public](#)

**Official Notices**

Welcome to OfficiëleBekendmakingen.nl, the site where you can find official notices published today in the Government Gazette, the Gazette and the Treaty Series. For a search of official notices and parliamentary documents, please visit [zoek.officielebekendmakingen.nl](#). On this site contact is also possible to let you draw attention to current laws and regulations.

**Official Notices**

- > Tractatenblad (0)
- > Gazette (10)
- > In Organization (179)
- > In Category (179)
- > Search Official Notices
- > Explanation renumbered Gazette from July 1, 2009

**Wednesday, February 29, 2012**

**Gazette**

179 Publications

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 17 | 18 | [Next](#) >

[Court of Leeuwarden, Bankruptcies, receiverships and debt restructuring 02.29.2012 | Gazette, 2012, 4308 | Legal notices | Court of Leeuwarden](#)

[Court of Haarlem, Bankruptcies, receiverships and debt restructuring 29.02.2012 | Government Gazette 2012, 4307 | Legal notices | Court of Haarlem](#)

[Court of Breda, Bankruptcies, receiverships and debt restructuring 02.29.2012 | Gazette, 2012, 4306 | Legal notices | Court of Breda](#)

**Actions**

[Print](#)

Figure 19 Screenshot NL Official Publications of 29/02/2012

The guide to all government information and services

**Overheid.nl** [High contrast](#)

**Regulatory**

[Home](#) [Public](#)

**Official Notices**

[Back to publications](#)

**Staatscourant**

Volume 2012  
Nr. 4308  
Wednesday, February 29, 2012 9:00

[View full table of contents](#)

**Bankruptcies, receiverships and debt restructuring**

**Court of Leeuwarden**

**Zaailand 102**

**bankruptcy rulings**

- op February 21, 2012 on (F.12/67) ADVANCED SURFACE TECHNOLOGY BV vest.adr. Foeke Sjoerdswei 3, 8914 BH Leeuwarden, Netherlands Chamber of Commerce North, 27193084. Cur: E.J. Rotshuizen, PO Box 222, 8901 BA Leeuwarden, tel 058-2122444. RC: Mr. J. Smit.

**end bankruptcies**

**for lack of assets**

**Actions**

[Authentic version \(pdf\)](#)

Figure 20 Screenshot of one Official Publication – web version

 **STAATSCOURANT** **Nr. 4308**  
29 februari 2012

Officiële uitgave van het Koninkrijk der Nederlanden sinds 1814

**Faillissementen, surseances en schuldsaneringen**

**Rechtbank Leeuwarden**

**Zaailand 102**

**Uitspraken faillissement**

- op 21 februari 2012 inzake (F.12/67) ADVANCED SURFACE TECHNOLOGY B.V., vest.adr. Foeke Sjoerdswei 3, 8914 BH Leeuwarden, KvK Noord-Nederland, 27193084. Cur: mr. E.J. Rotshuizen, POSTBUS 222, 8901 BA Leeuwarden, tel. 058-2122444. RC: mr. J. Smit.

Figure 21 Screenshot of one Official Publication – PDF version (authentic version)

Regarding the process handling within Sdu, this is all managed by a central custom-made system, which is responsible of controlling the workflow, initiating the necessary conversions upon request and handling the communication with the central service point and other interested parties.

The various conversions are all independently developed, maintained and packaged, though they build upon each other. They support the full range of different publication types that are produced by the various public administrations and parliament. A complete overview of the publications is found on the site: <http://www.officiëlebe bekendmakingen.nl> (Official Publications).

Sdu uses a dedicated schema for each publication type, while the Dutch State uses only one generic schema (BWB) as the official standard for all publication types. The reason for this difference is that Sdu needs to keep the production processes as specific as possible for each document type. This is detailed below.

### 6.3.2 Converting to XML

The process of transforming Microsoft Word documents to XML equivalent ones is done in several steps:

#### 6.3.2.1 Stripping and adding predefined styles

The documents are processed by a Microsoft Word service that applies word templates to first strip most (or all) of the styling information. The service then adds predefined styles to the stripped document based on:

- a. heuristics that look for specific phrases and textual elements like headings in the content of the document, and
- b. heuristics that are based on the general document layout and ordering of subjects, also known as the surface structure of a document.

The end result is a well-understood styled document. In some cases manual correction needs to be applied to the newly added styles, because the heuristics are not defined enough to be fool proof. Sometimes there are embedded graphics or embedded tables that need to be hand crafted into the resulting Microsoft Word document.

#### 6.3.2.2 Convert into XML using the styles

The restyled document is then treated by software (<http://www.upcast.de/>), which is configured to use the applied styles to convert the document to an equivalent intermediate XML document. Besides the styles that are added in the previous step, Upcast also uses all kind of Microsoft Windows specific structures when converting the Microsoft Word document to an XML equivalent (for instance list, table, image, frame or footer structures).

This type of software transforms documents using XSLT, creates mark-up based on regular expressions and on style information, provides list detection heuristics, validates against XML DTD, XML Schema and Relax NG. A specific document processing language was developed -

UPL (upCast Processing Language<sup>41</sup>), enabling document conversion from graphically marked-up documents into rich logically marked-up documents. An example:

To turn any paragraph containing more than 85% of text

1.that is bold and

2.whose font-size is between 16pt and 18pt

into a heading of level 1, the following UPL code could be used:

```
[element(uci:par) and %(@font-weight="bold" and @font-size >= 16pt and @font-size <= 18pt) > 0.85] { set-heading-level(1); }
```

### 6.3.2.3 Convert the XML into Sdu internal XML with XSLT

In the final step, the XML document that is produced by Upcast is converted again using XSLT style sheets to the Sdu internally used XML format for all official publication. This step completes the kraak process.

In these three steps, manual intervention may be needed but is limited to a bare minimum. Intervention may happen to handle individual complicated cases, but may also occur as part of a larger service offering, like correcting textual or grammatical errors in the content.

### 6.3.2.4 Convert the Sdu internal XML into BWB XML (Official Publications)

A separate conversion package converts XML documents that comply to a publication specific Sdu internal XML schema into the generic official publication XML schema (BWB). From there, the end products to be published (PDF, XHTML and ODT) are produced. For each end product there exists additional separate conversion packages.

## 6.3.3 The need for XMLisation after drafting

The need for XMLisation partly results from the structure of the overall publication process as described above. Sdu has no control in the way the source documents are authored, but is obliged to fulfill its contractual duties, which includes producing XML documents for publication from source documents in every conceivable format. Drafting is outside scope from Sdu point of view. They may participate in projects that aim for direct XML drafting, but these are projects initiated by entities/departments within the Dutch State.

Given the large quantity of documents processed per week (hundreds), an automated solution is a definite requirement. Currently, virtually all contemporary legislative documents are produced with the described process.

<sup>41</sup> <http://www.upcast.de/loop/assets/content/products/upcast/764b1743/doc/manual/html/upl.html>

In the future this may change and enhancements to the central system of Sdu have been built to cater for such a new situation. However, even with source documents that are delivered in XML format, conversions remain necessary - for instance because of specific layout requirements which need to be captured for publishing reasons, but that are irrelevant from a semantic point of view.

#### **6.3.4 Developing the XMLisation heuristics**

Development of the heuristics has been complex as there is a large range of publications. Initial development was between four and six months by a four men team, which covered all legislative publications. A second implementation round, also taking several months by the same team, added support for the parliamentary part.

Furthermore, improvements and enhancements have been made over a time span of around four years to include insights gained during daily production. Whenever possible, new improvements are implemented as reducing manually assistance helps in bringing down the overall cost. A rough estimate of time spent over this period is somewhere around six months in total by a loosely knit team evening out to about one man per year.

During the whole life-time of the product there is about 6 man-year spent on automating the XMLisation of documents. It is important to note, that fine tuning of this process never stops, but that the required effort is of course declining over time.

The current set of heuristics is bound to the Dutch legislative texts and the formalized XML-schema that abstracts the various constructions found in those texts. The initial internal Sdu XML-schema captured these abstractions of which many have found their way back into the official publication XML-schema. The same approach may work on other legislative texts (from other countries) as well, because juridical systems --albeit verbose-- do follow logical structures. The art is in deducing the juridical abstractions and distilling the relevant phrasing or other textual constructs to recognize them, but these are language dependent.

#### **6.3.5 Efficiency of the XMLisation process**

Documents are reviewed as part of the quality assurance - which is required, as the final publicized PDF documents (that are based on the prior XMLised documents) have legal binding. The XMLisation process is - and has to be - efficient, as the publication process is done on commercial terms. More manual intervention immediately undercuts the margins.

## 6.4 THE XMLISATION PROCESS FOR EUROPEAN PARLIAMENT AMENDMENTS

*The information provided for this case is based interviewing of experts in charge of the XMLisation process and in charge of the AT4AM project.*

### 6.4.1 Overview of the amendment workflow

The online web tool AT4AM supports the creation of amendments at the European Parliament on existing documents – in XML format. It is used by 1500 users: MEPs, MEPs assistants for drafting the amendments, and Secretariats of parliamentary committees, of political groups for preparing lists of amendments. It was released in production in January 2010. In order to use the tool, attending one hour presentation plus knowledge of amendments' drafting inside EP is enough. As it edits XML, layout is not needed; nevertheless, the text appears to the screen with a specific layout to conform to the current way to format amendments in EP, so as to provide users with a WYSIWYG interface.

It provides three types of documents:

- Amendments on proposals made by European Commission (COM proposals)
- Amendments on European Council Position (COD procedure)
- Amendments on European Parliament non legislative reports

AT4AM uses the Akoma Ntoso schema and works on XML versions of the proposals, positions and other reports mentioned above.

This “XMLisation” is carried out beforehand by the official journal unit at DG ITEC<sup>42</sup>. Then, deputies amend in one of the 23 official languages, using AT4AM; this draft amendment is verified -also in AT4AM, and the amendment is sent in XML to the administration (DG TRAD Translation Directorate General) which then translates the amendments. At this stage, the documents are generated in the Microsoft Word format and transferred to the Council.

The official journal unit at DG ITEC provides the XMLisation services for the translated documents of the AT4AM project (European Commission documents (COM documents); Parliamentary commission documents (draft reports), Council documents) as well as for the Europarlament<sup>43</sup> website (translated versions of texts adopted in plenary session of the European Parliament, in their provisory or final version).

---

<sup>42</sup> The secretariat of the European Parliament is the administrative body of the European Parliament headed by a Secretary-General. It includes the Directorate General Innovation and Technological Support (DG ITEC), which consists of The Directorate for Information Technologies (DIT), and the Directorate for Publishing and Distribution which deals with printing and distributing (also electronically) of Parliament's working documents (committee meetings, plenary sittings, etc.), as well as publishing legislation and documents in the Official Journal (minutes, texts adopted, written questions, etc.) and producing other publications in cooperation with the Office for Official Publications.

<sup>43</sup> <http://www.europarl.europa.eu/>

## 6.4.2 Converting to XML

The conversion is currently made to SGML then to an internal XML schema at export, and in a next step the files are converted into the Akoma Ntoso XML schema. The process of this SGML Workshop – providing the conversion of several types of documents, through an iterative process of:

1. standardisation,
2. control,
3. quality control.

then distributes and archives the converted documents.

### 6.4.2.1 Standardisation

The standardisation process is a set of *transformation steps* where:

- the input file of the step 1 is the Word document file to be treated;
- the input file of the step i) is the SGML document file created in step i-1);
- the output file in the last step is a XML document file.

Each of the *transformation steps* of a document is a specific program in the software tool Omnimark<sup>44</sup>, used to parse/mark-up the document. This approach is based on breaking the conversion process up into multiple steps – which simplifies development and maintenance of the conversion, rather than writing conversion scripts to do the entire conversion in one pass through the data – which would create a very large and complex application that will be hard to develop and debug, and hard to maintain and adapt when new content sources or new business rules are introduced.

Each type of document treated has its own specificities, but there are common areas to all. The standardisation process therefore handles 2 types of transformation: standard and oriented.

### **Standard transformation**

It is accomplished through several transformation steps as described in Figure 22 Standard transformation – EC and EP documents.

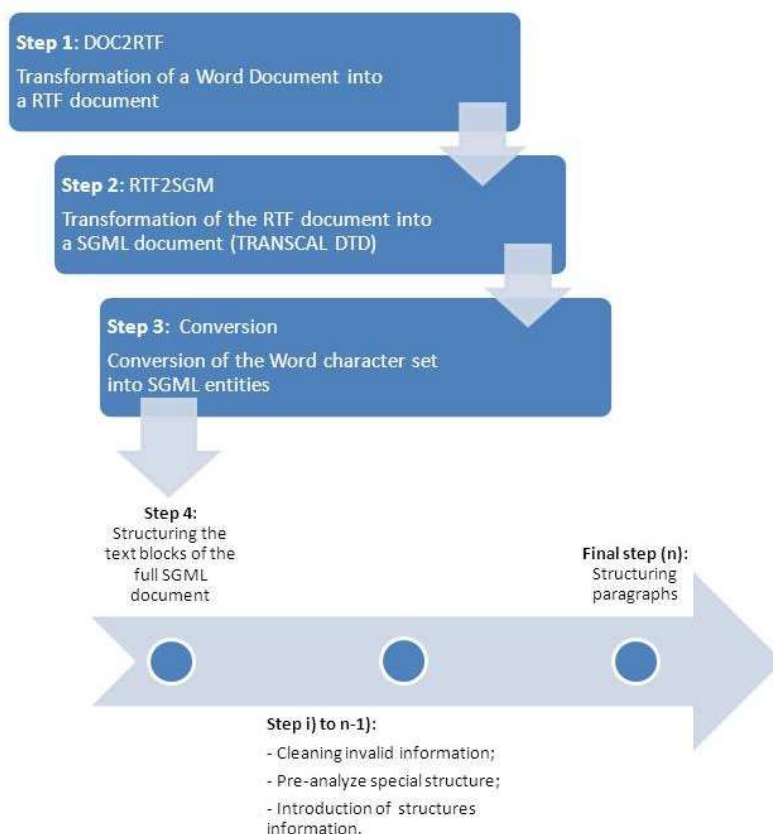
Those transformation steps are governed by the following rules:

1. a block of text is defined by textual data delimited by :
  - the beginning of the document;
  - carriage returns;
  - the end of the document;
2. a paragraph is a block of text which starts by a upper-case letter and can be identified due to digits, letters, special characters such as hyphens, dots, ... ;

---

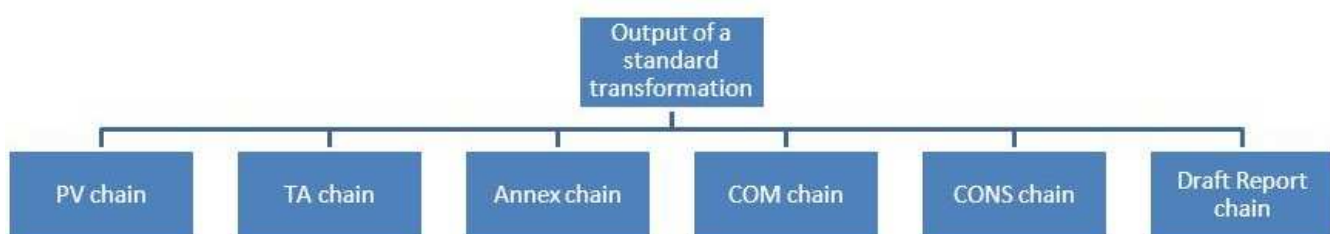
<sup>44</sup> <http://www.stilo.com/OmniMark/tabid/57/Default.aspx>

3. a title is a centred paragraph;
  4. a sub-title is a centred paragraph following a title;
  5. other blocks of text are considered as “normal blocks”;
  6. a citation is a set of structures 2), 3), 4) and 5) delimited by quotes;
- In all cases the operator can modify the way the system works, by applying specific Word styles on blocks of text, in order to work around these rules if needed.



**Figure 22 Standard transformation – EC and EP documents**

The outcome of the standard transformation is then adapted to each document type (PA, Annex, COM, ...).



## Oriented transformation

Each type of document is governed by a Document Type Definition (DTD) which is why each of them has a particular production chain; the result of an oriented transformation is in line with the specifications defined by that DTD. Given that a document has the same content and structure in its 22 linguistic versions, only one linguistic version is used throughout the oriented transformation.

The oriented transformation uses a master structure and a merging function.

### Master structure

The French document is used as the master document in order to create the “Master Structure”. The first step of this oriented transformation consists in the structuring of the document **lists**. This transformation steps are governed by the following rules:

1. a list can be introduced by a block of text (any type of block) with the character “.” at the end;
2. An item of a list is a normal block
3. all items of a list are preceded by the same type of symbol (digit, letter, ...);
4. a paragraph ends the list previously defined;

In all cases the operator can modify the way the system works, by applying specific Word styles on blocks of text, in order to work around these rules if needed.

The rest of the transformation steps consist in determining the semantic structure of the document defined by the DTD.

At the end of the oriented transformations, the “Master Structure” of the considered is defined.

To illustrate the “Master Structure”, let us consider the simple “SIGNATURE” structure which is defined by the place where the document has been signed (“PLACE” structure), the person (“NAME” structure) and his/her quality function (“QUAL” structure).

The XML structure for the French document is:

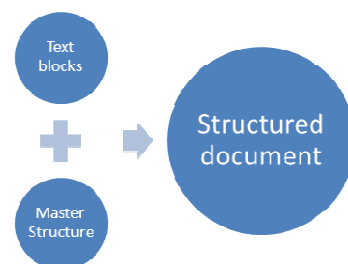
```
<SIGNATURE>
  <PLACE>Fait à Bruxelles, le ...</PLACE> “text block 1”
  <QUAL>Président</QUAL> “text block 2”
  <NAME>Jerzy Buzek</NAME> “text block 3”
</ SIGNATURE >
```

The “Master Structure” is:

```
<SIGNATURE>
  <PLACE><!-- place of text block 1 --></PLACE>
  <QUAL><!-- place of text block 2 --></QUAL>
  <NAME><!-- place of text block 3 --></NAME>
</ SIGNATURE >
```

### Merging function

The Merging function consists in taking the text blocks of a document other than the French and inserting them at their reserved place inside the Master Structure.





The approach used in the standardisation phase allows identifying easily where and why a problem occurs. Where this problem occurs, the corresponding step in production can easily be adapted. A new transformation chain can quickly be created for new types of documents.

#### 6.4.2.2 Control

##### Control of the structures

Two types of control are operated, the first one being during the standardization phase (merging phase) - this is the control of the structures. The merging shows where documents are missing translated sections.

##### Content control and synoptic control

Another type of control is the *content control*. For this, a list of data - named "Independent language data" – is defined by the fact that they follow writing rules such as:

- references,
- numbering,
- numeric dates,

or as data belonging to a countable list such as the months within a date (alpha numeric dates).

When all linguistic versions of a document are well structured, a control of all "independent language data" content is done –this is the content control.

Moreover, a "*synoptic*" control is defined as a content control comparing the "independent language data" belonging to a given context in *all* linguistic versions. This "synoptic" control generates the error report of the treated document for all linguistic versions. Add screenshot

#### 6.4.2.3 Quality control

The standardization phase guarantees that for a certain type of document, an XML version of the document is generated following a specific DTD, and that an error report of that document is created. However this phase is almost automatic and one cannot guarantee that all the created structures are well formed in the context where they appear. Besides, documents often contain mistakes because:

- drafting rules are not always respected by the writer;
- some structures are very complex or can introduce ambiguities which hinders defining the appropriate structures

A visual control of the structure is made of the structure generated during the standardisation phase. This visual control checks conformity with the master document, using the editor Epic. The controller is able to create, delete or modify a structure.

In the case where it is difficult to modify a structure, the controller can request a **new standardization phase** for the French Microsoft Word document in which he has assigned a special style to some blocks of text. This is done in order to force the system to create the needed structure. The controller, by confirming the end of the quality control operation, creates a new "Master Structure" and sends the request for finalizing the production. The Merging

function, using the new “Master Structure”, applied on all other linguistic versions of the treated document, finalizes the production. The result of this production can then be archived and distributed.

### 6.4.3 Benefits and efficiency

The main benefit is the use of the XMLised content of documents for amendments in the tool AT4AM. Previously the tool used to amend documents was Microsoft Word with templates, with a configuration script used to assign the needed committee to a type of document. The tool used over 1000 macros, and was difficult to maintain because of the high number of templates (130 templates, each in 23 languages). Moreover, the tool was layout oriented, and not fitting the needs of the drafters who wanted to express themselves – needed to pass a message, and in doing so, often did not conform to the structure, linked to the layout.

With AT4AM, the administration services - which checked not only the amendments, but also the format, - find that 50% of their time is saved. The tool provides currently 4000 individual logins, supports 50 users simultaneously, and provides many status messages to the users during their amending.

The time spent making an amendment used to take nine minutes, and now takes less than a minute (which generated an increase in the amendments made).

Some of these online services provides by AT4AM can be generalised and used by other administrations, for example the spell checker and the “diffing” (versioning view in two columns), as well as amendments basic actions.

Converting a document according to the process described above can take from an hour to several weeks, depending on the size and the initial quality of drafting. This multiple step approach makes the conversion process more maintainable, but it is also much slower to execute and uses more resources because of the need to serialize and parse the content between each step. Moreover, a lot of this time is spent in checking the coherence in between the different language versions of the converted documents. Incoherence appears for example when one section is not translated into one language – the tool signals if each translation version has the same number of paragraphs than the others. This could/should be done at the outcome of the translation process, so it actually burdens the transformation process, but provides a benefit in terms of quality of translation of documents.

The team includes one IT expert and six full time operators. The statistics regarding documents managed for the year 2011 are presented in the table below (for example, in January, 169 files, in 22/23 linguistic versions added up to 3746 documents counting 60 272 pages are converted). For a year, this adds up to almost 800 000 pages converted (+/- 400 pages per day per person).

	Nr Files	Nr Linguistic versions	Pages	MBytes
Jan-2011	169	3.746	60.272	609,23
Feb-2011	201	3.825	55.173	520,40
Mar-2011	314	6.440	91.941	824,42
Apr-2011	233	4.964	64.277	614,57
May-2011	275	5.561	82.930	888,12
Jun-2011	201	4.160	58.241	584,89
Jul-2011	246	4.873	73.561	742,79
Aug-2011	76	1.643	47.897	360,05
Sep-2011	221	4.376	65.832	593,95
Oct-2011	191	3.774	62.235	598,12
Nov-2011	272	5.036	75.992	867,48
Dec-2011	232	4.678	59.571	600,85
	2.631	53.076	797.922	7.804,87

## 6.5 THE MARK-UP OF A DOCUMENT WITH THE NORMA EDITOR

*The information provided for this case is collected through desk search and interview of the expert who developed the tool.*

### 6.5.1 Introduction

Norma Editor (part of Norma System) is a tool used for the mark-up (XMLisation) of legislative documents by the Italian Official Gazette, (100 000 documents in 5 years) and developed by Monica Palmirani<sup>45</sup>.

Norma-Editor is a Microsoft Word XP add-in which provides a comprehensive list of features, which include possibility to acquire Doc, ASCII, RTF, TXT and XML-format documents, mark-up the structures of documents with automatics tools, automatically recognise and mark-up normative references, and create an XML version of documents. This mark-up and conversion to XML is further detailed in the following sections.

### 6.5.2 Converting to XML

The process of transforming Microsoft Word documents to XML equivalent ones is done in several steps:

1. Clean-up: The text imported into the Norma-Editor environment is stripped of any pre-existing tagging and formatting. Some this pre-existing formatting and tagging may help the user detect information useful for mark-up, so the user has the option of enabling some text recognition before proceeding to the clean-up stage.
2. Detection of tables: Any tables included in the text are marked up. Tables have special formatting, and it is important to detect it before feeding the text to the parser for automatic recognition.
3. Detection of annexes: The main text is split off from any annexes, with the editor managing as well the hierarchy of the annexes (annexes appended to other annexes).
4. Detection of annotations: Any annotations are automatically detected.
5. Mark-up of the preamble: The heading and the preamble of the legislative text is automatically detected by the parser and manually corrected by the user.
6. Detection of quoted text: Any new text added by insertion or substitution is usually enclosed in quotation marks (e.g., Article 1 Decision 2000/185/EC is hereby amended as follows: 1. in the first subparagraph of Article 1 this “three years running from 1 January 2000 to 31 December 2002” shall be replaced by “four years running from 1 January 2000 to 31 December 2003”).
7. Mark-up of body text: The main partition (or running text) of the document is marked up in automatic or semiautomatic mode.
8. Multimedia links: The legislative document may include multimedia objects, and the editor will manage this kind of link.
9. Mark-up of closing partitions: Mark-up of the final parts of the legislative document (signatures, ending clause, date, place, etc.).

---

<sup>45</sup> <http://www.e-p-a-p.com/dlib/9788883980466/art15.pdf>

10. Mark-up of normative references: The parser helps the user to automatically detect normative references, by locating such references and detecting the text's structural partitions (document type, date, number) necessary to build a Uniform Resource Name(URN).

11. Qualification of normative references: A specific tool helps the user qualify normative references as being of a certain type. This applies to all normative provisions generally (any text partition carrying normative meaning for example), and in particular to "modificatory" provisions, which are detected and qualified as such in order to enable automatic construction of an updated text.

12. Mark-up of metadata: The foregoing information enables automatic inference of different kinds of metadata (document type, level in the hierarchy of legal sources, type of act,...); other metadata will have to be put into the system manually (examples being the date of entry into force and the modificatory provision's date of application).

13. Semantic check: A semantic check will detect any inconsistencies the text may contain from the legal point of view or any discrepancies from the rules of proper legal drafting.

A **semantic check** is based on drafting rules. For example, it checks, when there is a link, if the link is correct/complete. A structure check would verify only if there is a link. This controls the semantic correctness of the mark-up, e.g. it is possible to insert the title of the act inside of each comma - for the XML-Schema it is valid from a technical point of view – but the title of the act could be wrong. It coordinates the semantic of the structure (if a title mentions "Mr. Kafka", it checks if the URL mentions the same name).

The editor produces a report of any errors or inconsistencies found, thus enabling the user to go back and correct them.

14. Validation: Once the semantic check is carried out (and any inconsistencies set straight), the editor will convert the document from Microsoft Word format to XML and will validate it in keeping with the DTD or XML schema selected by the user. Several reports show any errors in the conversion and validation process, thus aiding the user in correction.

15. Sending package to the server: The final legislative document (complete with annexes and multimedia files) is sent to the server, which will then handle the package for processing and storage.



## Document structure

Decreto legislativo 30 luglio 1999, n. 282 (in Gazz. Uff., 16 agosto, n. 191). - Disposizioni per garantire la riservatezza dei dati personali in ambito sanitario.

Il Presidente della Repubblica:

Visti gli **articoli 76 e 87 della Costituzione**;

Vista la **legge 31 dicembre 1996, n. 675** e successive modificazioni ed integrazioni;

Sulla proposta del Presidente del Consiglio dei Ministri, di concerto con il Ministro di grazia e giustizia;

Emana il seguente decreto legislativo:

**Articolo 1.** Ambito di applicazione e definizioni.

1. Dopo il **comma 1 dell'art. 23 della legge** sono inseriti i seguenti:

"1-bis. Con decreto del Ministro della sanità adottato ai sensi **dell'art. 17, comma 3, della legge 23 agosto 1988, n. 400**, sentiti la Conferenza permanente per i rapporti tra lo Stato, le regioni e le province autonome di Trento e Bolzano e il Garante, sono individuate modalità semplificate per le informative di cui all'art. 10 e per la prestazione del consenso nei confronti di organismi sanitari pubblici, di organismi sanitari e di esercenti le professioni sanitarie convenzionati o accreditati dal Servizio sanitario nazionale, nonché per il trattamento dei dati da parte dei medesimi soggetti, sulla base dei seguenti criteri"

2. Il medesimo decreto disciplina anche, limitatamente a quanto specificamente previsto, i trattamenti di dati idonei a rivelare lo stato di salute in ambito sanitario da parte di soggetti diversi da quelli indicati nel comma 1.

3. Ai fini del presente decreto si applicano le definizioni elencate **nell'art. 1 della legge 31 dicembre 1996, n. 675**, di seguito denominata 'legge'.

Il presente decreto, munito del sigillo dello Stato, sarà inserito nella Raccolta ufficiale degli atti normativi della Repubblica italiana. E fatto obbligo a chiunque spetti di osservarlo e di farlo osservare.

Dato a Roma, addì 30 luglio 1999

CIAMPI

Amato, Presidente del Consiglio dei Ministri

Bianco, Ministro dell'Interno

Visto, il Guardasigilli: Fassino

Heading

Opening  
Formula

Division into  
Articles

Closing Formula

Conclusion

© CIRSIFID-Beni eni

**Figure 23 Screenshot Norma Editor document structure<sup>46</sup>**

Norma-Editor needs to coordinate with the server during four different phases of the mark-up process, while (a) marking up normative references, (b) updating the DTD or XML schema, (c) checking the URN for completeness, and (d) checking to make sure that the text version produced aligns with the rest of the normative system.

Automatic text parsing is done using structural rules and vocabularies *external* to the software programme<sup>47</sup>. Vocabularies contain:

- list of issuing authorities
- types of documents
- ordinal numbers
- latin numerals adverbs
- names of partitions
- mapping between word's elements and XML tags
- qualifications of provisions
- easily personalization of vocabularies

<sup>46</sup> <http://www.sciweavers.org/publications/xml-editor-legal-information-management>

<sup>47</sup> <http://www.parliaments.info/downloads/11%20Monica%20Palmirani%20-%20Norma-Africa.pdf>

### 6.5.3 Re-usability

Norma editor has been developed and enhanced over the last 10 years, based on Microsoft Word add-ins (which count 40 000 lines of code). It is compliant with Microsoft Windows XP – and Microsoft Windows XP only, as the libraries of the interface are shared with the operating system. It will not be updated to be used on other platforms, but the knowledge, concepts and experience are being used for developing a vendor independent tool based on Open Office (see Bungeni case study) and supporting Akoma Ntoso and MetaLex XML schemas.

## 6.6 THE MARKUP OF A DOCUMENT WITH XMLEGESEDITOR

*The information provided for this case is collected through desk search and interview of the expert who developed the tool.*

### 6.6.1 Introduction

xmLegesEditor is a tool is used for the mark-up (XMLisation) of various legislative documents in Italy:

- administrative acts of Tuscany for the PACTO<sup>48</sup> project, an observatory for monitoring the quality of administrative acts
- regional acts of Campania for the IRE-SUD project, which aims to improve and simplify the delivery of some eGovernment services, particularly in the area of Justice<sup>49</sup>
- bills for the Italian Senate within the TafWeb development<sup>50</sup>.

xmLegesEditor is a specific editor for legislative documents based on NIR (NormeInRete) Italian Legislative standard (XML DTD/Schema and URN). It is a native XML editor (directly manages XML documents compliant to NIR format) entirely written in Java and using open source standard Apache libraries. It is a visual XML Editor with “word-processor look-and-feel” (hidden XML syntax). It is open source (Licensed in GPL) and entirely customizable to support different DTDs or XML-Schema.

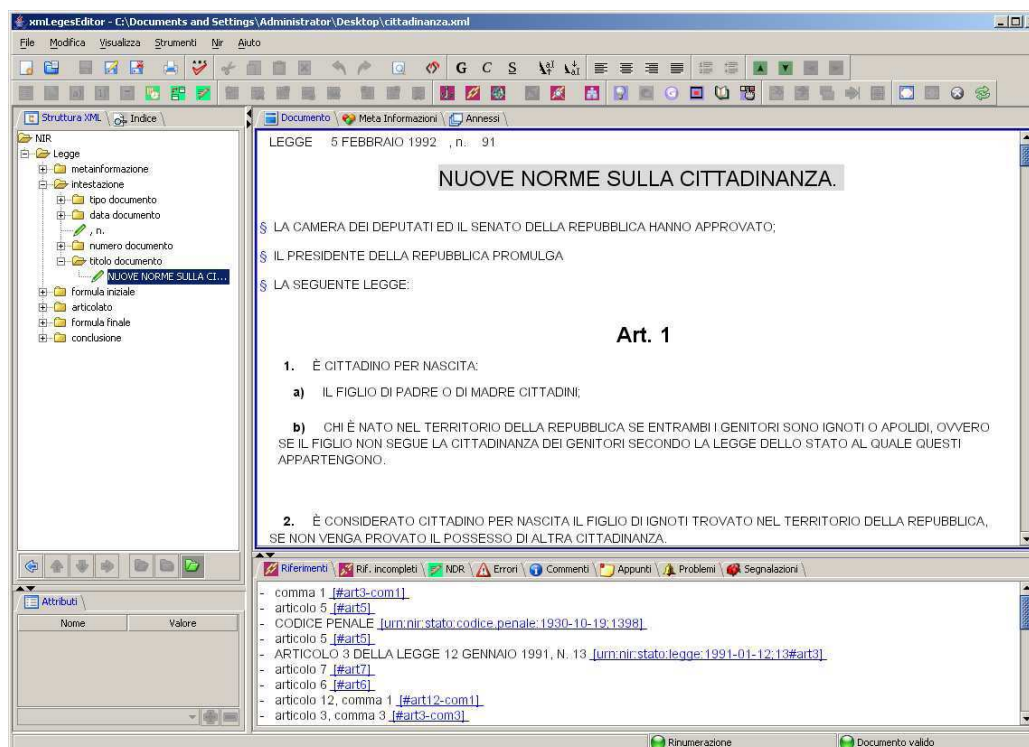
---

<sup>48</sup> <http://www.pacto.it>

<sup>49</sup> [http://archivio.cnipa.gov.it/site/it-IT/Attivit%C3%A00 - Archivio\\_storico/Efficienza\\_interna\\_della\\_PA/Progetto\\_IRE-Sud/](http://archivio.cnipa.gov.it/site/it-IT/Attivit%C3%A00 - Archivio_storico/Efficienza_interna_della_PA/Progetto_IRE-Sud/)

<sup>50</sup> Lorenzo Bacci, Pierluigi Spinosa, Carlo Marchetti, Roberto Battiston Automatic mark-up of legislative documents and its application to parallel text generation <http://ceur-ws.org/Vol-465/paper6.pdf> pages 8-10;





**Figure 24 xmLegesEditor screenshot**

The user interface is similar to those of word processors. However, the frame is divided into different panels each offering a different view of the document. The main one is the textual panel reporting the full text content of the document over which typical textual typewriting and related facilities are allowed. Moreover, a number of different summarizing and specific panels are provided. At runtime each XML document can be associated with multiple views. Each view can deal with just one aspect of a large XML document or may simply provide another way of looking at the same content. This means that every possible view of the document is obtainable by applying a style sheet can easily become an interactive panel (not read-only) of xmLegesEditor.

xmLegesEditor provides a comprehensive list of features for drafting new texts (specific document templates according to different type of acts, automatic numbering according to specific drafting rules, text editing (cut/copy/paste/move) within the main drafting panel with corresponding structural elements moving within the structure panel, implementation of external references and related links according to the URN-NIR standard, etc...).

Moreover, legacy content is managed with the xmLeges Marker and xmLeges Linker. The marker provides mark-up to a legal document. It provides XML conversion of different types of documents (laws, decrees, etc.). The tool allows the user to choose the type of text document, but the tool also recognises types of text based on words in the text ("law..."). It converts the following formats: doc, plain text, html, pdf. It also provides integration and correction of automatic tagging (sections merging/splitting – section rank promoting/reducing, tables tagging). The tool also automatically detects specific metadata (document URNs, publication date, etc.), and automatically recognises legal references and their conversion into hyperlinks. (The linker



detects references according to the citations, and transforms it into a URL according to the IETF standard URN:LEX).

This mark-up and conversion to XML is further detailed in the following sections.

**xmLegesMarker** is a parser for legacy content. It is a plugin in C++ for the XMLLeges editor and it is also available for documents tagging Web service. It provides automatic detection and tagging of the documents formal structure.. This set of functionalities is able to transform legacy contents (a legislative text) into NIR documents, but can be adapted to another XML schema such as Akoma Ntoso.

**XmLegesLinker** is used on several platforms, as Web and stand-alone service. XmLegesLinker results can be shown as a list of references and related URN or as text including references properly marked-up as hyperlinks. xmLegesLinker is able to manage different document formats (txt, html, XML specific format) and to produce reference tagging in different formats (txt, html, XML specific format).

## 6.6.2 Converting to XML

### 6.6.2.1 Importing

The user can suggest some hints to the tool by choosing the type of document he will import. In the example Figure 25, “Legge” (law) is chosen.

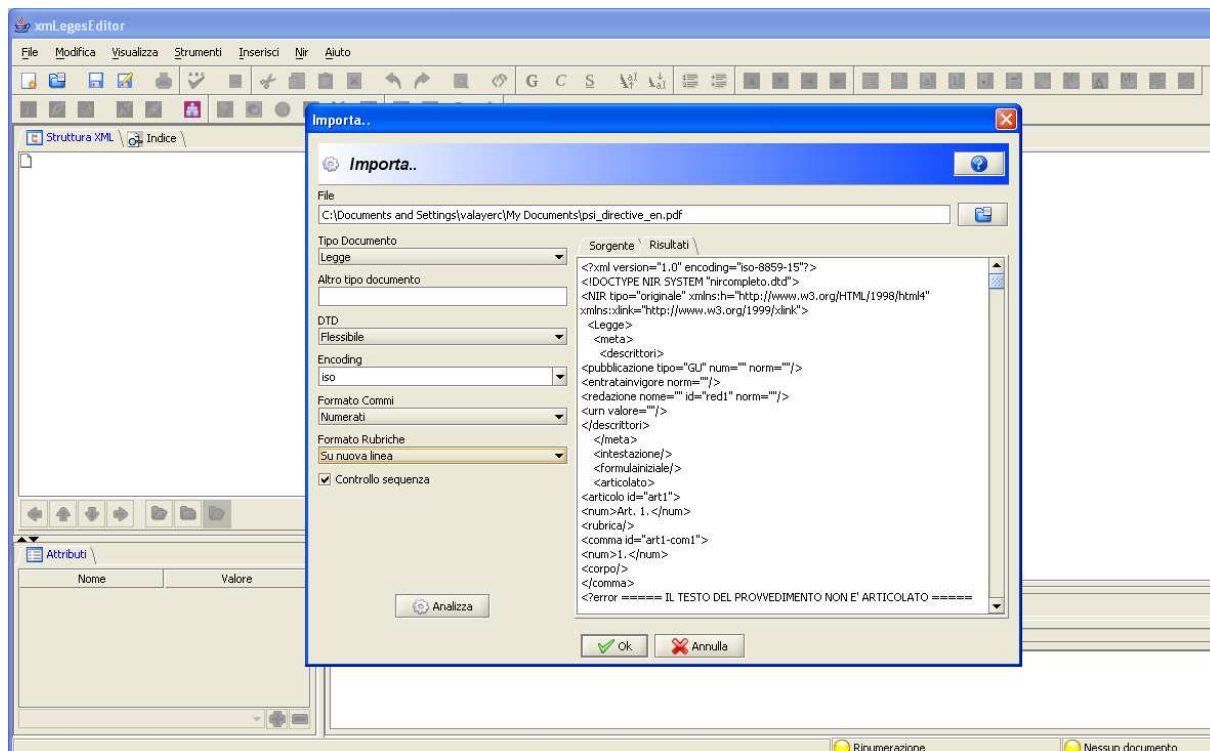


Figure 25 xmLeges screenshot: Importing a document

The user can chose:

- which DTD or schema to use (NIR, or a “light version”),
- which encoding (UTF-8, ...),
- which format for the paragraphs (numbering, nothing,...)
- how should paragraphs be separated (new line, ....).

If the user selects “OK”, the tool then imports the document in the main panel. The tool applies a stylesheet to present the document (all panels use XSLT technology to present the XML content in a user-friendly presentation).

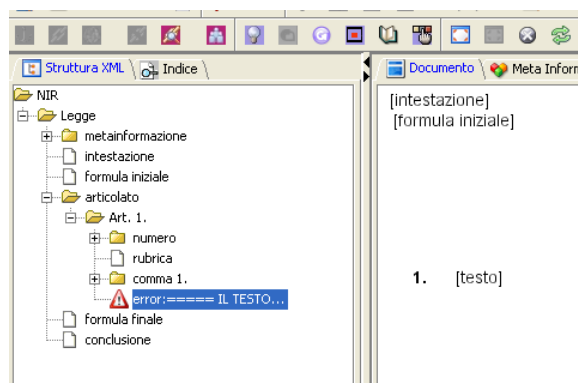
xmLEGESeditor then detects the references in the text. The user can manually edit the references detected having errors. By clicking on the URN, the tool queries the Italan database Normattiva.it resolver which resolves the URN of NIR references. It uses only the parameters of the citation.

The document can be exported in the browser (HTML) or in .PDF (the tool then uses the stylesheet of the Official Gazette).

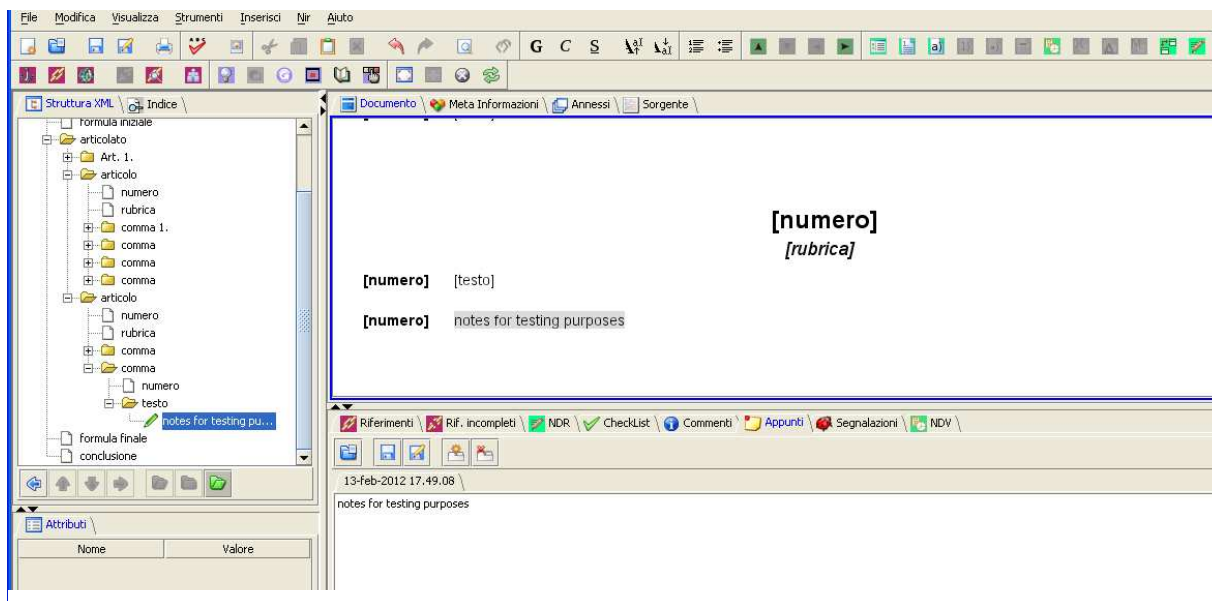
### 6.6.2.2 Structure check

The editor will provide messages when errors in the structure are detected. If the document imported does not have a structure, or if the user needs to deal with some parts of the document which do not fit in the structure, the content will be treated as a rejection by the tool (see Figure 27).

The tool then allows deletion of the part in the structure which has the error (but the content cannot be directly inserted in another structure – as in a drag and drop manner). The use of the note pad in the panel below is useful for storing the content which needs to be moved - before uploading it in the correct part of the structure (see Figure 26).



**Figure 27 xmLeges screenshot –**



**Figure 26 xmLeges screenshot – notes section**

The importing of very large documents takes a significant amount of time. The software would need some re-engineering for enhancing the speed of import.

### 6.6.2.3 Technology<sup>51</sup>

xmLegesMarker is a structural parser able to transform a legacy normative document in plain text, HTML or doc format into XMLNIR format. Two parsing strategies have been adopted for different portions of a document. For the body of a normative document, a non-deterministic finite-state automata (NFA) was implemented.

In legislative acts, the enacting terms section, in which articles and paragraphs lie, matches the data, while entities like title, number and type of document, subscribers and so on, are considered explicit meta-data. Other meta-data, defined in NIR schema, although not explicitly present in the input, are computed or added by xmLegesMarker, usually exploiting the values of the explicit meta-data (i.e.: automatic generation of URN [4]). The splitting of information in data and meta-data follows the physical structure of the document. While explicit meta-data are typically located in the header or in the footer, the body of the legislative act accommodates the enacting terms, namely the data. Besides the physical position, there is another important difference between the body and the header (or the footer) in a legislative act: the former is composed by partitions strictly organized and sorted in a hierarchical way, practically a tree of partitions, while the latter appears fuzzy and composed by expected and unexpected elements, often in a random order. This is the reason why xmLegesMarker adopts two different strategies in order to analyze the header, the footer and the body of a document.

The marker uses lexical analysers technologies. The engine is organised as follows:

The body of a legislative act coincides with the enacting terms section, which is typically well organized in known partitions, hierarchically arranged in a tree structure. On the other hand, the tree representing the enacting terms can be complex, long and nested. An automata approach is required in order to efficiently parse this kind of structure. The Flex<sup>52</sup> scanner generator allows the creation of very powerful text scanner based on a non deterministic finite state automata. The automata that handles the enacting terms strictly follows the constraints imposed by the NIR schema: the parsing process obeys to rules that depend on the automata states (start conditions), which match all the partitions defined in the NIR hierarchy. For example, an alphabetical list can be read only if the automata is in the paragraph state, because, according to NIR and to legislative drafting rules, a list should only stay inside paragraphs.

For the header and the footer a different strategy was adopted, since their partitions are not usually identified by particular typographical symbols. It uses a statistical and machine learning approach for meta-data extraction. The identification of such elements can only be based on the sequence of words appearing within them, with a probability that can be estimated and without knowing the states which produced such sequence. The aim of this approach is to uncover

---

<sup>51</sup> L. Bacci, P. Spinosa, C. Marchetti, R. Battistoni, "Automatic mark-up of legislative documents and its application to parallel text generation", in proc. of "LOAIT 2009: 3rd Workshop on Legal Ontologies and Artificial Intelligence Techniques", Barcellona, 8 Giugno 2009 <http://ceur-ws.org/Vol-465/paper6.pdf>

<sup>52</sup> Flex scanner generator (<http://dinosaur.compilertools.net/flex/>)

these hidden states. For this reason, to parse these two sections we adopted a strategy based on Hidden Markov Models – a probabilistic automata.

Legacy and manually edited documents often contain syntactical errors, like numbering errors, incorrect use of punctuation marks, errors in the layout of the document, unbalanced quotes. Some of these errors in the plain document have a limited effect in the XML output of xmLegesMarker, while others may cause totally disruptive behaviours of the automata used for the body parsing. For example, if quotes aren't balanced, the automaton jams in the amendment states, forcing all the remaining text into the amendments tags. The marker identifies these troublesome situations and embeds a self-explaining warning message in the XML output. Each issue is identified through a warning code that guides the user in the correction of the input (see Figure 28).

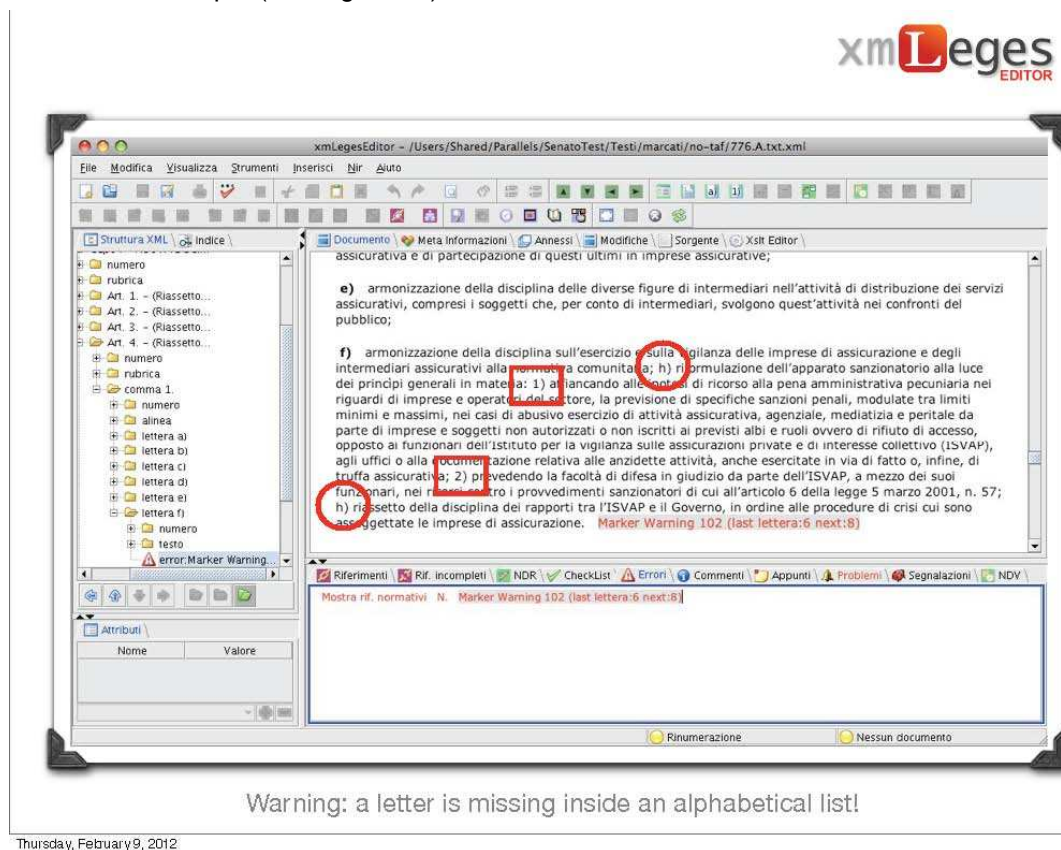


Figure 28 xmLeges screenshot identification of structure issues

### 6.6.3 Efficiency - accuracy evaluation<sup>53</sup>

A number of plain documents were provided by the Italian Senate, including both Senate and Chamber version of bills (for the training set – to train the tool: 60 documents, for the test set: 47 documents).

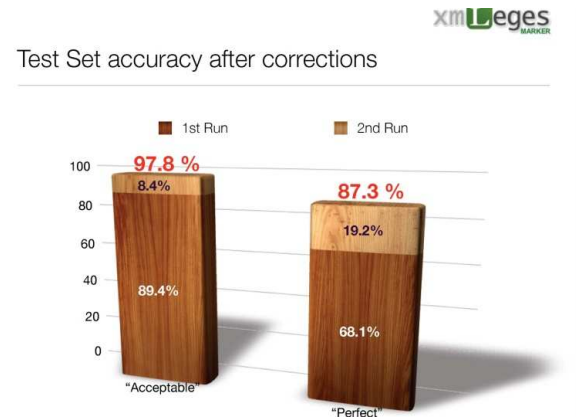
<sup>53</sup> L. Bacci, P. Spinosa, C. Marchetti, R. Battistoni, "Automatic mark-up of legislative documents and its application to parallel text generation", in proc. of "LOAIT 2009: 3rd Workshop on Legal Ontologies and Artificial Intelligence Techniques", Barcellona, 8 Giugno 2009 <http://ceur-ws.org/Vol-465/paper6.pdf>

Mark-up was defined as “Acceptable” for the user if there were at least 90% of correctly identified partitions, and “Perfect” if there were 100% of correctly identified partitions.

Two runs were made on the Test Set:

- 1st run: original plain documents
- 2nd run: plain documents with syntax error corrections following warnings triggered in the first run

The results of the second run were considered “acceptable” for 97,8% of the documents, and “perfect” for 87,3% of the documents.



## 6.6.4 Re-usability<sup>54</sup>

### 6.6.4.1 An architecture designed for adaptability and contribution

xmLegesEditor has been structured on a component-based architecture where each component provides specific services in order to guarantee maximum extensibility, modularity and reusability of the different software modules.

Moreover a strict separation between generic XML modules and specific NIR<sup>55</sup> modules has been followed.

Therefore the developed components can be reused in order to develop other specific visual editors for supporting other XML standards and in particular Legislative XML standards. The complete editor application is in fact obtained by composing the different services specified in an application composition XML file. Replacing NIR modules with other specific modules developed for supporting such different standards, a new editor can be obtained using the core XML editor infrastructure. The architecture is presented in Figure 29.

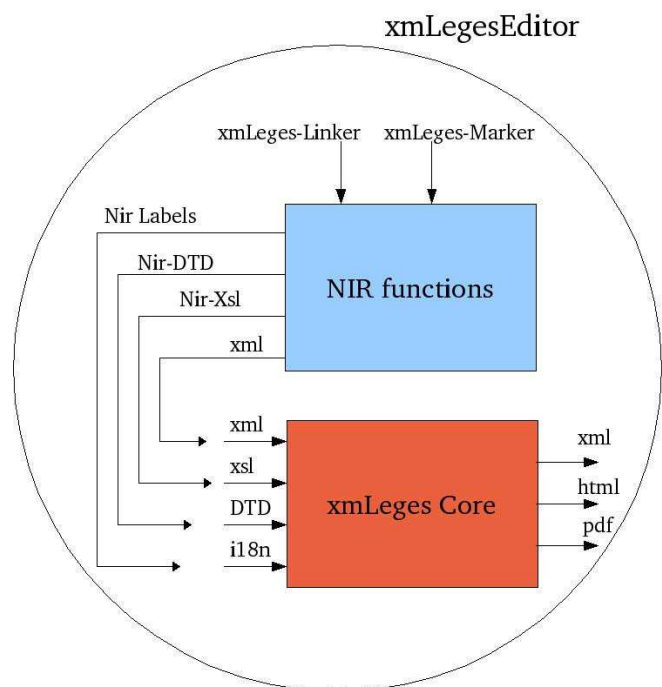


Figure 29 xmLegesEditor architecture

<sup>54</sup> Tommaso Agnoloni - ITTIG-CNR agnoloni@ittig.cnr.it

<sup>55</sup> NIR: Norma in Rete – the Italian National XML schema

#### 6.6.4.2 xmLegesCore: a generic and application independent Visual XML editor

##### **CORE Features**

xmLegesCore provides functionality for composing a basic generic visual XML editor. Its parameters are: the DTD or XSD files of the standard to be supported, the XSLT style sheet files for internal visualization inside the editor and for document export in output formats such as HTML or Pdf, one or more files containing textual labels and icons for the complete customization and internationalization of the user interface. All these aspects have been kept as application's parameters rather than cabled inside the code in order to guarantee maximum reusability.

Editing functions are

- Undo/redo
- Drag & drop (partially implemented)
- Cut Copy & Paste
- Find & Replace
- Comments/Processing Instructions management
- Document Tree View
- Attributes View
- Document Outline (as XSL visualization)
- Single Document support
- Multipanel (synchronized) view
- Open/close/Validate/Validation Errors log
- Save, Save As
- Image / Pictures insertion
- Table support (only html tables tags or depending on the format as noncore functionality)
- Export (HTML,PDF,rtf) xsl,xsl-fo,css support
- Spellcheck supports MySpell Openoffice OpenSource dictionaries - <http://lingucomponent.openoffice.org/>
- Editable Xsl filtered visualization of the document

##### **xmLegesCore functionalities**

xmLegesCore provides complete functionalities for:

**Validity Management:** This is accomplished through the access to the *rulesManager* service. Only valid operations are allowed in the XML document editing: this is obtained by contextually querying the *rulesManager* component which provides methods to access the DTD or XSD parsed in a Finite State Automata graph structure in order to a priori validate in such structure the effects of a certain operation without actually entering the content in the document. Only valid documents can be produced ("*a priori validation*"). Changing the grammar automatically affects *rulesManager* answers. This allows to provide such Validity Management strategy to any document standard expressed both in DTD or XSD.

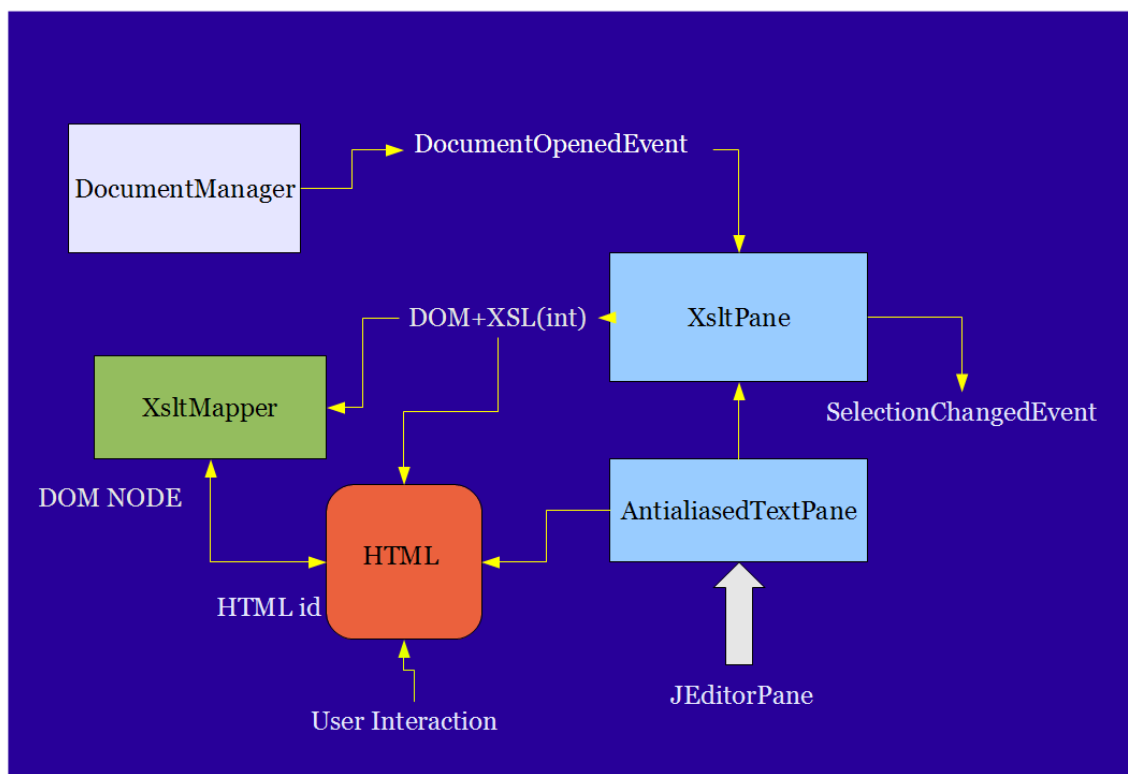


Document Management: A dedicated component (*DocumentManager*) provides services for accessing a generic XML Document. The document is initially parsed into a Document Object Model format and then made available in such format for access and modification to any other component. Functions for opening, saving, management of a multi-level undo/redo are made available by the *DocumentManager* component.

Document Visualization: *WYSIWYG – XSL filtered visualization*. This provides visualization in the various different panels of an HTML view obtained by applying an XSLT stylesheet to the XML Document. The number and the content of visualization panels is configurable. In order to provide editing functionalities from such views, a mapping between the original XML document elements and the visualized HTML text is provided in order to make this process bidirectional and store the input from the editor panels into the corresponding XML elements. This is accomplished by two different components *XsltMapper* and *XsltPane* which are at the heart of *xmLegesCore* visualization functionality.

Internazionalization: All the labels and the icons appearing in the editor User Interface are actually loaded from localization files accessed through a unique identifier. This means that by simply changing some locale file the whole UI can be translated and customized with different icons without actually changing the code. This service is provided by the *i18n* component.

#### 6.6.4.3 Extending xmLegesCore



Extending *xmLegesCore* to support different Legislative standards (XML schemas) consists in:

**Reusing** *xmLegesCore* functionalities “as is “. This allows basic valid document editing, contextual element and attribute insertion and deletion from right-click.

**Developing** the format specific layer with:

- Internal XSL visualization stylesheet to customize the WYSIWYG editing panels.
- External XSL export stylesheet for HTML and PDF publication.
- Developing the specific DOM components and related Forms, Actions, Menus and Toolbars according to the figure depicted above invoking the available core services previously described.
- Editing the i18n properties files for translation of the interface and customization of the icons in the chosen language.

**Declaring** the new configurations and developed components in a new application composition file replacing *xmLegesEditor.xml*

#### 6.6.4.4 Use of open standards

*xmLegesEditor* is entirely written in Java, a platform independent language, thus leaving freedom in the choice of the platform on which to run it. All the software libraries used in the development of *xmLegesEditor* are robust and highly reliable opensource libraries mainly from the Apache Software Foundation. At every level of processing, data is managed by open, W3C standards compliant, and commercial free, software tools.

*xmLegesEditor* a native XML editor, this means that no conversion to or from proprietary formats is ever made, this preventing any vendor lock-in possibility and providing the added value of complete transparency in data management – important aspect in the context of public data management.

This two layers' architecture should encourage contributions from wider developer communities very active in the field of XML tools development, at least to improve *xmLegesCore* in order to obtain a robust fully functional generic Visual XML Editor shared as a common resource, which is still missing. Moreover, the code is easily accessible, and detailed documentation is available.

## CASE STUDY 2 CONCLUSION

This case study provided various examples which allow an understanding of the need and benefits for XMLisation in the context of publishing, amending and legacy management.

The analysis also highlights various techniques used, and provides insight in the re-usability of the tools. Re-usability is based not only a sound architecture design and the use of open standards, but also strategic aspects of non-vendor lock-in to ensure adaptability of the tool to a large user base.

Re-usability also faces challenges linked to language dependency of the solutions, and some mark-up techniques are based on probabilistic heuristics for which the machines need to learn from correct documents.

This case study also allows understanding the expertise and time needed, the experience to be built with the specific documents to be XMLised and with the specific schemas to be used.



---

## CASE STUDY 3: LEGISLATIVE EDITOR USING OPEN OFFICE

### 6.7 BUNGENI, AN OPEN SOURCE DESKTOP EDITOR FOR LEGISLATIVE MARK-UP BASED ON OPEN OFFICE

This case study analyses the Bungeni editor, an open source desktop editor for legislative mark-up based on open office, licensed under GPLv3. The tool is a java application that extends the OpenOffice.org word processor. The Bungeni Editor can in theory work on any operating system that supports OpenOffice.org v 3.3.0, however it has been primarily tested on the Ubuntu Linux and Windows XP platforms. Production version will be ready second quarter 2012.

Several Parliaments have expressed interest in using and also institution concerned with publishing judgements and laws. Users need to know how a valid document must be structured, and what the semantic parts of a legal document are.

#### 6.7.1 Applying the structure

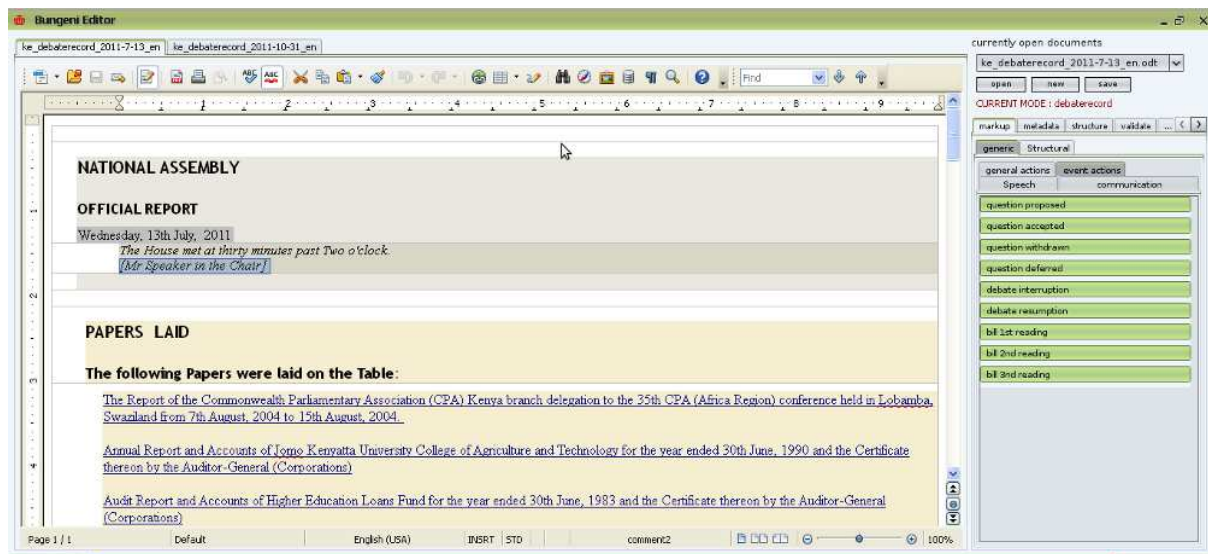
##### 6.7.1.1 Functional aspects

##### ***A mark-up tool***

The editor provides a hybrid user interface with mark-up tools around the main word processor window. Mark-up tools are provided by the extended UI to allow the user to identify parts of a legislative document and associate metadata to different parts of the document. The ultimate objective of going through the effort of marking up a document is to produce legislative XML. The Bungeni editor provides a facility to export marked up word processor documents in both Akoma Ntoso and MetaLex formats.

##### ***The editor user interface***

The Bungeni editor user interface embeds the OpenOffice.org writer user interface and provides additional functionality in the containing window. The left hand area is the word processor window where the user can edit one or more documents. The right hand area - a Control Panel - provides functionality to manage the workflow of legislative editing— and also provides actions to mark-up the document and get a structural view of the document being edited (see Figure 30).



**Figure 30 Bungeni editor screenshot user interface**

## Editing

This section describes the functionality available for editing (see Figure 31), according to the four tabs or steps:

### Step 1 Compose New / Edit Existing

The Editing process is managed primarily using the right hand Control Panel. The Control Panel uses a tabbed interface to group logically sequential steps of the editing process, such as opening an existing document, composing a new document, switching between documents, adding editor notes on the status of document corrections.

### Step 2 Metadata Panel

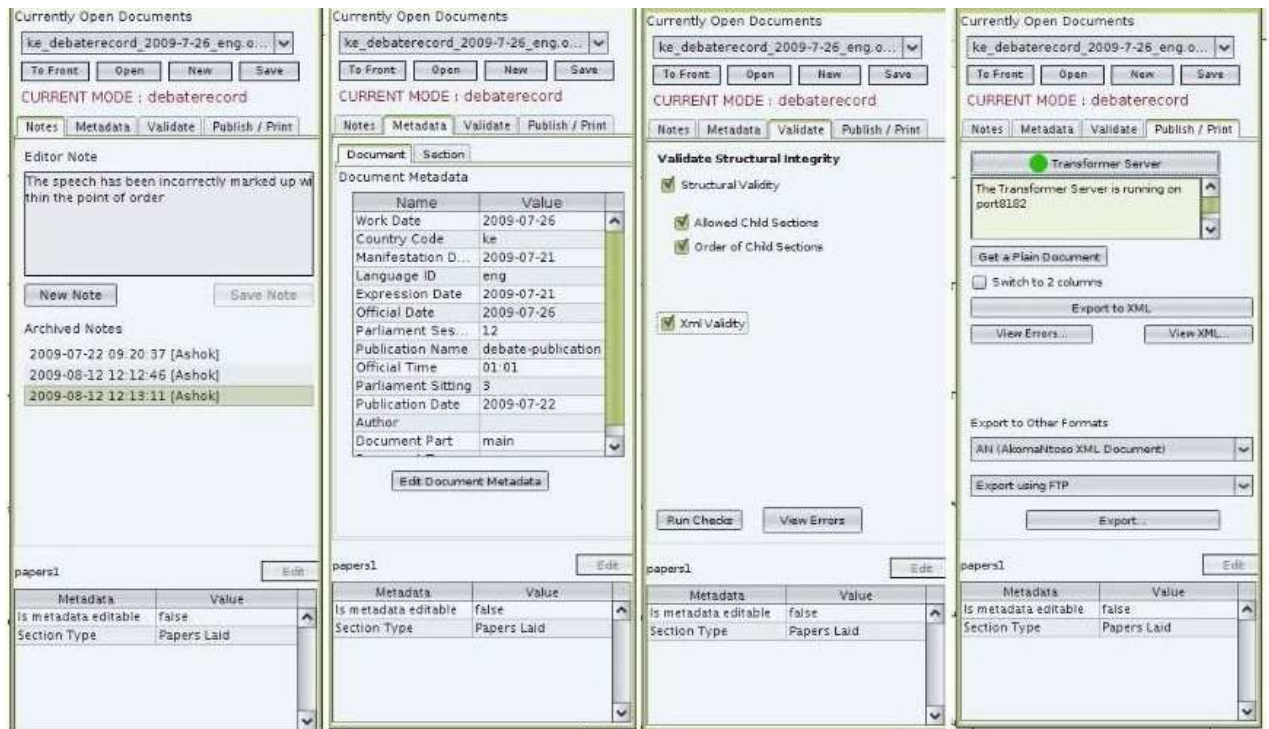
The Document Sub-tab within metadata displays the document level metadata. The Editor supports applying metadata either at the document level (i.e. The metadata is associated with the whole document) or with the main container element of the Editor the document section.

### Step 3 – Validate and Check document

The editor supports 2 levels of validation – the Structural Validation is a semantic rule checker which allows creation of custom rules. A semantic validation provides checks at the OpenOffice.org word processor document level. The XML validation layer converts the document to Akoma Ntoso and validates it against the Akoma Ntoso XML schema.

### Step 4 – Transform to different formats

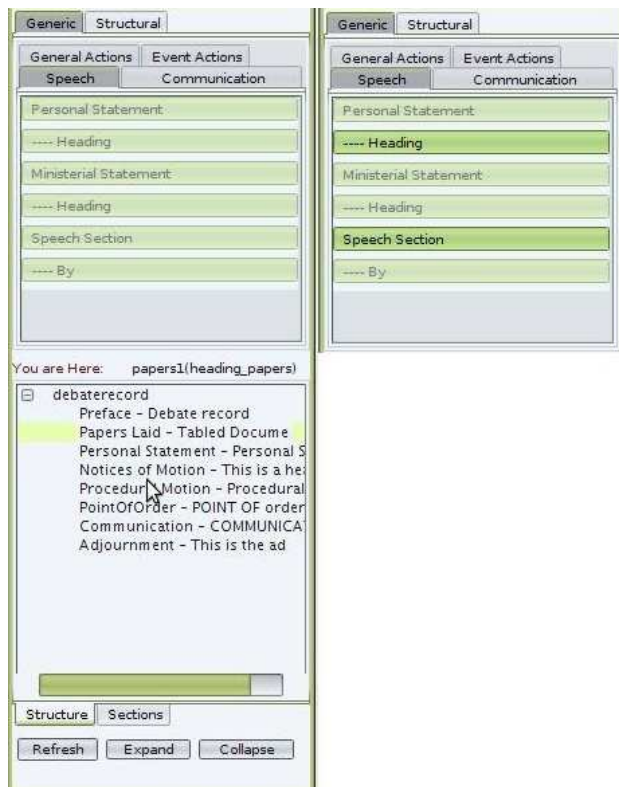
The editor supports transformation to various formats. It uses an XSLT based transformation engine. The tool can also convert the marked-up document to a plain document without any mark-up, indentation or styling.



**Figure 31 Bungeni screenshot editing functions**

## Marking-up

The Bungeni Editor also provides an assistive UI to help the user create proper mark-up. This section describes the mark-up functionality available in the panel (see Figure 32). Mark-up tools and icons are contextually highlighted and enabled to indicate valid mark-up actions to the user according to where the cursor is positioned in the text.



**Figure 32**Bungeni screenshot mark-up functions

## 6.7.2 User-guidance for applying the structure

### 6.7.2.1 Various document types

Depending on the selected document type, structural mark-up tools specific to a document type are provided in the user interface. These structural mark-up tools are context sensitive where the context is determined by a rule-based syntax. For example, if a bill document allows a clause only within a chapter - a rule can be specified using the rule syntax to enable the clause mark-up only when the cursor is within a chapter. These rules can be grouped per document type.

### 6.7.2.2 Creating a document

The default template in the editor is a standard word processor template with some special metadata to identify it as a Bungeni Editor document. Upon creation, the system prompts for essential metadata to name and identify the document.

Currently, the use-case managed by the editor is the mark-up of existing documents, no initial structure is imposed on the document, instead contextual mark-up tools are provided to the user for him/her to mark-up the text. The project team is planning to introduce specific support to guide users in the creation of new documents making use of document specific templates and styles.

### 6.7.2.3 Checking a document

Various levels of structural compliance checking are supported:

- Active checking via mark-up rules where actions are defined based on rules and these get activated based on the context of the cursor position in the document. For e.g. a clause heading mark-up action will only be enabled when the user highlights some text inside a container identified as a clause. (see 6.7.2.1)
- Structural rules – the Bungeni Editor supports a structural rule checker – which provides post mark-up validation checks. These are run by the user after mark-up and the rules are implemented using a custom XML syntax. It provides an extensible interface to allow a developer to implement their own structural rules.
- XML output validation – the XML output Akoma Ntoso can be validated for correctness.

## 6.7.3 Technical aspects

### 6.7.3.1 A structure with high granularity

The supported structure is Akoma Ntoso XML schema. The Bungeni editor supports identifying container hierarchies and marking up metadata within these hierarchies.

- If Questions and Answers are grouped in a debate, the Editor allows demarcating Questions and Answers within a container.
- Similarly typical nested structures like Chapter->Clause->Sub-Clause->Paragraph are also supported. There is no limit on depth of hierarchies. Structural nesting is controlled via mark-up rules.
- Metadata within a marked up container is also supported.
  - For e.g. if a speech is marked up in a debate, the name of the speaker can be marked up, the text of the spoken word can be marked up and if there was a person the speech was addressed to that can also be marked up.
  - In a bill, clauses, sub-clauses etc. may be numbered, this numeric metadata can also be marked up.

Note : Currently only mark-up of visible numbers on the document is supported. Some exceptional cases are also supported, for instance when a number a number is assigned to a group of paragraphs. This of course means that the output pipeline processor will need to implement a customized template to output such exceptions when converting to Akoma Ntoso.

### 6.7.3.2 Creation of the structure

The mark-up is done via standard word processor formatting like section breaks and formatting styles. Specific metadata is applied to section breaks and styles – and this information is used for correction and validation.

The Bungedi Editor is an Open Office Plugin, and it saves documents in .odt files. Open Office and Bungeni plugin exchange information according to the UNO<sup>56</sup> protocole. The .odt file is composed of a hierarchical tree structure with some specific files (see Figure 33 Screenshot .odt file content). Some of the files are used by Open Office for technical purposes; some other files are directly related to the document content and metadata. The ODT is compressed according to the zip format.

Folders	Name	Type
[fr_debaterecord_2012-2-3]	Configurations2	Folder
Configurations2	meta	Folder
meta	META-INF	Folder
META-INF	Pictures	Folder
Pictures	Thumbnails	Folder
Thumbnails	content.xml	XML Document
	manifest.rdf	RDF File
	meta.xml	XML Document
	mimetype	File
	settings.xml	XML Document
	styles.xml	XML Document

**Figure 33 Screenshot .odt file content**

The Bungeni Editor handles three types of information: metadata, content and notes. Metadata and notes are stored in meta.xml files, and content is stored in content.xml files.

Document level metadata can be found in ./meta.xml within the ODF package. Bungeni Editor writes document level metadata as ODF user-defined properties:

```
<meta:user-defined meta:name="BungeniCountryCode">ke</meta:user-defined>
<meta:user-defined meta:name="BungeniDocAuthor"/>
<meta:user-defined meta:name="BungeniDocPart">main</meta:user-
defined>
```

Akoma Ntoso defines different Element containers. These containers are of course not supported in ODF. Bungeni represents AN element containers in ODF using the text:section element. For example, the following is the representative mark-up in ODF of a speech in a debaterecord (content.xml):

```
<text:section text:style-name="Sect10" xml:id="id81842116"
text:name="speech1">
  <text:p text:style-name="Standard">
    <text:meta xml:id="id394891649">Mr. Francis Acheka: </text:meta>
  </text:p>
  <text:section text:style-name="Sect9" xml:id="id1636983406"
text:name="spbody1">
    <text:p text:style-name="Standard">Mr. Speaker, Sir, I beg to
reply.</text:p>
    <text:p text:style-name="Standard">
      <text:meta xml:id="id415166567">(a)</text:meta>
      <text:s/>
      <text:span text:style-name="an-list-item">The Arabia Airstrip is
of gravel surface, and [...] graded in June, 1988.</text:span>
    </text:p>
    <text:p text:style-name="Standard">
```

<sup>56</sup> [http://wiki.services.openoffice.org/wiki/Uno/Remote/Specifications/Uno\\_Remote\\_Protocol](http://wiki.services.openoffice.org/wiki/Uno/Remote/Specifications/Uno_Remote_Protocol)



```

<text:meta xml:id="id590467019">(b)</text:meta>
<text:s/>
<text:span text:style-name="an-list-item">According to the
Ministry's current Development Plan [...]Airstrip.</text:span>
</text:p>
</text:section>
</text:section>

```

This speech section has ODF metadata ; the RDF metadata for the section is recorded in a metadata RDF file ; the link to the metadata RDF file is made via the manifest.rdf file in the ODF package --

```

<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <ns1:anx xmlns:ns1="http://editor.bungeni.org/1.0/"
rdf:about="meta/meta.rdf">
    <rdf:type rdf:resource="http://docs.oasis-
open.org/ns/office/1.2/meta/pkg#MetadataFile"/>
  </ns1:anx>
  <ns2:Document xmlns:ns2="http://docs.oasis-open.org/ns/office/1.2/meta/pkg#"
rdf:about="">
    <ns2:hasPart rdf:resource="content.xml"/>
    <ns2:hasPart rdf:resource="meta/meta.rdf"/>
    <ns2:hasPart rdf:resource="styles.xml"/>
  </ns2:Document>
  <ns3:StylesFile xmlns:ns3="http://docs.oasis-
open.org/ns/office/1.2/meta/odf#" rdf:about="styles.xml"/>
  <ns4:ContentFile xmlns:ns4="http://docs.oasis-
open.org/ns/office/1.2/meta/odf#" rdf:about="content.xml"/>
</rdf:RDF>

```

Notes are considered as metadata by Bungeni. Each note is bundled into a unique section named "bungeniEditorNotes".

Additional information on mapping of Bungeni files to Akoma Ntoso is available in the Bungeni editor wiki at: [http://code.google.com/p/bungeni-editor/wiki/ODF\\_to\\_AN\\_XML\\_Mapping](http://code.google.com/p/bungeni-editor/wiki/ODF_to_AN_XML_Mapping)

As for how mark-up rules are applied, the Bungeni editor has a contextual toolbar driven editing interface that changes state based on the cursor position in the document. The behaviour of the toolbar is determined by a set of rule-based conditions defined in the settings, and activated via entries in toolbar.xml. The condition processor has been documented in the Bungeni wiki: <http://code.google.com/p/bungeni-editor/wiki/ConditionProcessorsInBungeniEditor>

## Managing document types

### Document types in Bungeni editor

The Bungeni Editor uses the Open Document Format (ODF) to represent all legislative documents. ODF allows setting of metadata at the document level and also for various artefacts like Images, Sections and Tables. The Bungeni Editor identifies an ODF document as a legislative document based on certain custom properties it sets on the document. The most relevant of these properties is the 'Document Type' property – this allows the Editor to identify a document either as a bill, a debaterecord, an act or a judgement.

The architecture of the Editor allows adding of new document types relatively easily – currently, the Editor supports the following document types :

- Debate Record
- Bill
- Judgement

The BungeniEditor supports multiple document types. Additional document types can be added by adding configuration entries into the editor's settings database.

Document types are mapped to individual action configuration files. The Editor allows associating a set of mark-up actions (grouped in a Toolbar action Xml config file) to a document type. This is done in the TOOLBAR\_XML\_CONFIG table in the Settings Database. Sample settings for the TOOLBAR\_XML\_CONFIG table are shown below :

DOC_TYPE	TOOLBAR_XML
bill	settings/toolbar_bill.xml
debaterrecord	settings/toolbar_debate_b.xml
judgement	settings/toolbar_judgement.xml

The complete documentation for writing action configuration files can be found on the Bungeni Editor wiki at: <http://code.google.com/p/bungeni-editor/wiki/ImplementingACompleteAction> .

#### **Creating a new document type**

The following steps are required to register a new document type in the editor:

- Create the template for the document type
- Create a Metadata Editor for the document type
- Register the new document type
- Registering Panels for the document type
- Enable available actions for the document type
- Registering an Action set for the document type
- Set the root section parameter for the document type

A complete documentation for generating new document types can be found on the Bungeni editor wiki: <http://code.google.com/p/bungeni-editor/wiki/RegisteringNewDocTypeForBungeniEditor> .

A new Document Type requires to be integrated via custom java code into the Bungeni java packages, and to build some xml config files and insert values in the internal Bungeni database. Adding a new document type cannot be done by an end-user. The process mentioned above involves java programming skills, and requires the following knowledge:



- An Open Office Template creation - .ott file (simple),
- Writing xml configuration files (for actions, metadata...)
- A Database configuration (into several tables)
- Writing java code in some cases (adding new metadata, tags or behaviour that are not available from the current templates)

### **Document templates**

The Bungeni Editor uses standard word processor templates to represent individual document types. This allows customization of styles and formatting per document type.

### ***Conversion to Akoma Ntoso***

The marked-up ODF document is converted to specific output formats (e.g Akoma Ntoso ) based on a transformation pipeline. Bungeni Editor implements a custom pipeline processor based on the Saxon XSLT transformer which processes marked up ODF to Akoma Ntoso XML. Structural mark-up and metadata is transformed to Akoma Ntoso elements and attributes. The pipeline is simply a XSLT template with template-matchers for specific elements, the template-matcher routes the template processing to a mini-XSLT file.

*Note:* The rule mark-up (see 6.7.3.2) is implemented independent of the document pipeline processor – which means they can be implemented to reflect similar structures in the mark-up and output document and/or completely different structures in mark-up and output document. The primary system components involved in the transformation process are as follows:

- ODFDOM - a free OpenDocument Format (ODF) library, which provides an easy common way to create, access and manipulate ODF files
- Saxon - an open source XSLT transformer
- Xerces Apache - an open source processor for parsing, validating, serializing and manipulating XMLthis is the DOM processor
- Restlet an open source tool used to provide a REST based API layer on the transformer. Client integration of the transformer is done over http using the Restlet http client API.

### ***Use of Open Document Format***

The Bungeni Editor uses ODF (Open Document Format) as the main document mark-up format. The main reasons are:

- ODF is an open standard and an XML format
- ODF supports extended metadata using RDF triples
- The availability of a choice of third party open source libraries for processing ODF
- Out of the box native rendering of ODF using OpenOffice / LibreOffice
- Easy conversion to various print and publishing formats
- Allows to use word processor environment familiar in the world of legislative mark-up

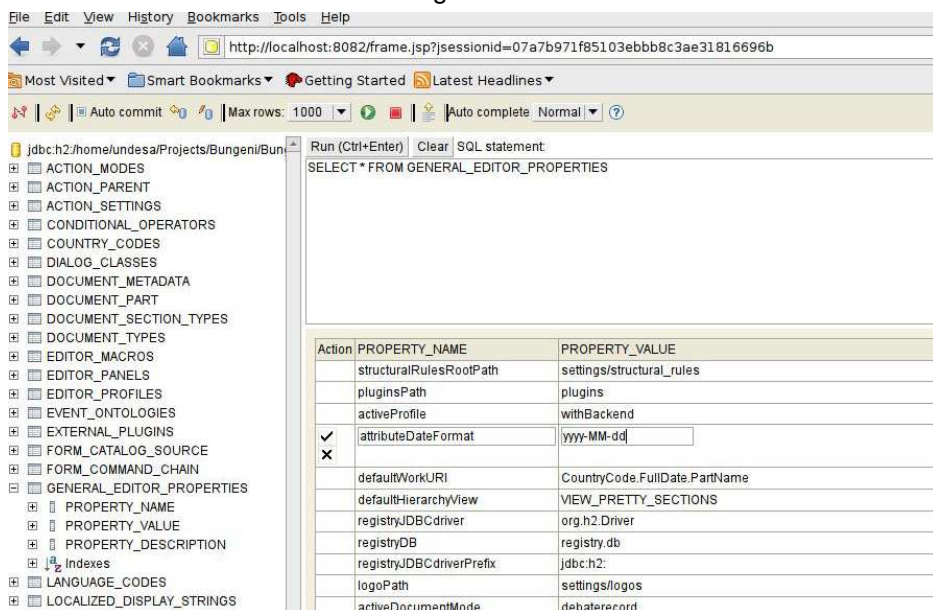
## Architecture

The design of the Bungeni editor architecture makes the editor highly customisable. As depicted in Figure 35 Bungeni editor architecture, and described in the sections 6.7.3.2, 0 and 0, various elements can be customised after configuration by the user: structure rules, document types and templates, as well as conversion to another XML schema. This ease of customisation of the tool is also due to the way the configuration of the registry is designed.

### Configuring the registry

The Bungeni Editor uses an embedded database for its configuration registry. This configuration registry allows for granular setup and configuration of the Editor – and allows the editor to adapt to different legislative scenarios without re-writing or modifying source code.

The Settings configurations can be edited using a browser based user interface (see Figure 34 Bungeni screenshot of the settings database configuration editor). The Settings editor runs as a webserver on the local editor installation and can be accessed using any web-browser. The settings editor provides a hierarchical view of various configuration options for the Editor. Further detail is available in the Bungeni wiki documentation.



**Figure 34 Bungeni screenshot of the settings database configuration editor**

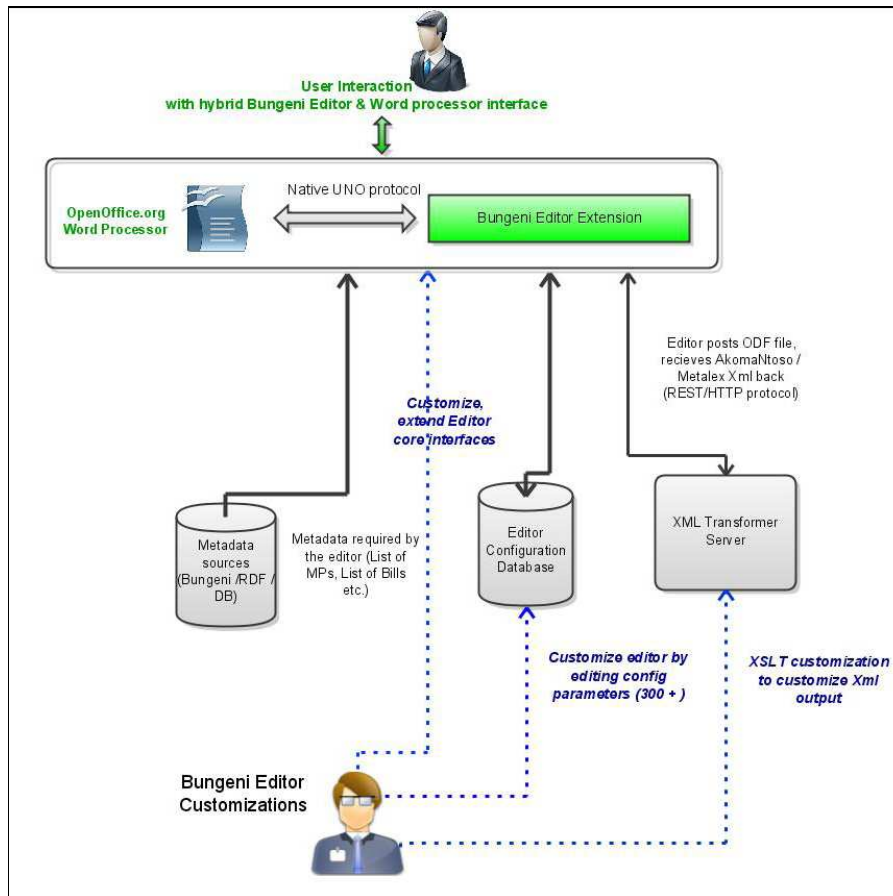


Figure 35 Bungeni editor architecture

### User interface<sup>57</sup>

A skin is a custom graphical appearance achieved by the use of a graphical user interface that can be applied to specific software and websites to suit the purpose, topic, or tastes of different users. Applying a skin changes a piece of software's look and feel. The user interface of the editor can be easily customised, the editor uses a skinning engine called "Substance" - various default skins are provided, and can be switched via a configuration file.

### Internationalization

In computing, internationalization is a means of adapting computer software to different languages; it is the process of designing a software application so that it can be adapted to various languages and regions without engineering changes. The Bungeni Editor an application supporting internationalisation.

String messages and user interface labels are all loaded from multilingual resource bundles. Furthermore, dynamically displayed user interface aspects (like displaying structural information about a document) can also be easily translated: metadata name and section type name aliases are all loaded from message bundles in the settings/bundles folder.

<sup>57</sup> <http://code.google.com/p/bungeni-editor/wiki/EditorTheming>

## 6.7.4 Benefits and trade-offs

### 6.7.4.1 Structure not imposed

There is a lower level of constraints in the tool. The user / drafter enters the text in a free manner, exactly like in a word processor.

The drafter can then mark-up the document via specialized tools. There is a user guidance for marking up the text, specific elements are highlighted according to where the cursor is placed, constraints can be enforced through rules on what styles can be used and where in the text.

Structure check and XML schema validity are done at request of the user.

#### ***Desktop application***

Bungeni is a desktop application; this has some advantages such as no need for an Internet connection, which adapts the tool to a potential digital divide among the users.

The downside of this is that it lacks the advantages of a browser application, which has a low barrier of entry. Bungeni editor needs to be installed, and other aspects such as maintenance/upgrade are also more complex than an application deployed using a Web browser.

However, the Bungeni team develops documentation and video tutorials to support the user, see <http://code.google.com/p/bungeni-editor/wiki/VideoTutorials> .

### 6.7.4.2 New project

#### ***Small user base***

The Bungeni editor is a new tool, soon in a stable version, with a user base under development. There is a wide interest though, as the tool is sponsored by the United Nations and implements Akoma Ntoso.

#### ***Expert contributors***

The project will benefit from the community of experts working with Akoma Ntoso. The standard is being fine-tuned with several parliaments across the world, and Bungeni editor is sometimes part of these projects. It receives funding for enhancement and adaptation to specific needs of the parliaments. The expert working on these projects has developed other legislation editing tools (the Norma Editor -see case study on XMLisation).

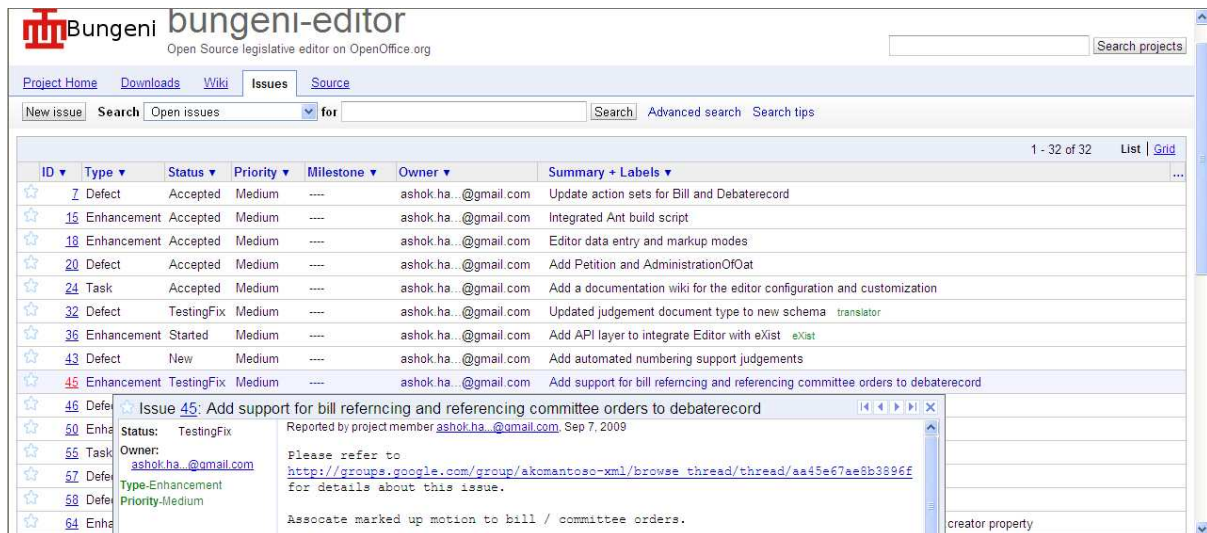
#### ***Enhancement of functionality and architecture***

The tool is in its first phase of development. Currently, it addresses the use case of document mark-up. New developments include for example to import Akoma Ntoso documents into marked-up ODF.

The following specific adjustments are identified by the project team to enhance Bungeni editor:

- Better support the creation of new document, e.g. Bill drafting using templates and supporting numbering and renumbering,

- Improve stability of Bungeni editor with more aggressive (and automated) user interface testing,
- Simplify the configuration architecture by unifying some of the configuration parameters which are presently spread across xml configuration documents and an embedded settings database,
- Implement a transformation error “back-reference” – which allows pointing validation errors identified in the output Akoma Ntoso document to the specific point in the original ODF marked-up document.
- The development can be followed on the bug tracking system of the project at <http://code.google.com/p/bungeni-editor/issues/list> (see Figure 36 Bungeni editor project issue tracker). It shows that the project has been under development for 3 years and that there are 4 developers listing bugs and enhancements, and mostly one active.
- The project development is transparent. Potential users / contributors can easily “enter” the project to develop listed enhancements, or develop their own suggested enhancements.



ID	Type	Status	Priority	Milestone	Owner	Summary + Labels
7	Defect	Accepted	Medium	---	ashok.ha...@gmail.com	Update action sets for Bill and Debatererecord
15	Enhancement	Accepted	Medium	---	ashok.ha...@gmail.com	Integrated Ant build script
18	Enhancement	Accepted	Medium	---	ashok.ha...@gmail.com	Editor data entry and markup modes
20	Defect	Accepted	Medium	---	ashok.ha...@gmail.com	Add Petition and AdministrationOfOat
24	Task	Accepted	Medium	---	ashok.ha...@gmail.com	Add a documentation wiki for the editor configuration and customization
32	Defect	TestingFix	Medium	---	ashok.ha...@gmail.com	Updated judgement document type to new schema <a href="#">translator</a>
36	Enhancement	Started	Medium	---	ashok.ha...@gmail.com	Add API layer to integrate Editor with eXist <a href="#">eXist</a>
43	Defect	New	Medium	---	ashok.ha...@gmail.com	Add automated numbering support judgements
45	Enhancement	TestingFix	Medium	---	ashok.ha...@gmail.com	Add support for bill referencing and referencing committee orders to debatererecord
46	Defect	TestingFix	Medium	---	ashok.ha...@gmail.com	Reported by project member <a href="#">ashok.ha...@gmail.com</a> , Sep 7, 2009
50	Enhancement	TestingFix	Medium	---	ashok.ha...@gmail.com	Please refer to <a href="http://groups.google.com/group/akomantoso-xml/browse_thread/thread/aa45e67ae8b3896f">http://groups.google.com/group/akomantoso-xml/browse_thread/thread/aa45e67ae8b3896f</a> for details about this issue.
55	Task	TestingFix	Medium	---	ashok.ha...@gmail.com	Associate marked up motion to bill / committee orders.
57	Defect	TestingFix	Medium	---	ashok.ha...@gmail.com	
58	Defect	TestingFix	Medium	---	ashok.ha...@gmail.com	
64	Enhancement	TestingFix	Medium	---	ashok.ha...@gmail.com	

Figure 36 Bungeni editor project issue tracker

#### 6.7.4.3 Customizable and re-useable

##### Customisation

The design of the Bungeni editor architecture makes the editor highly customisable (see 0) and therefore completely reusable. The mark-up (structure) and transformation (export to an XML schema) are fully configurable for different formats, the user interface is customisable, and the software supports internationalisation.

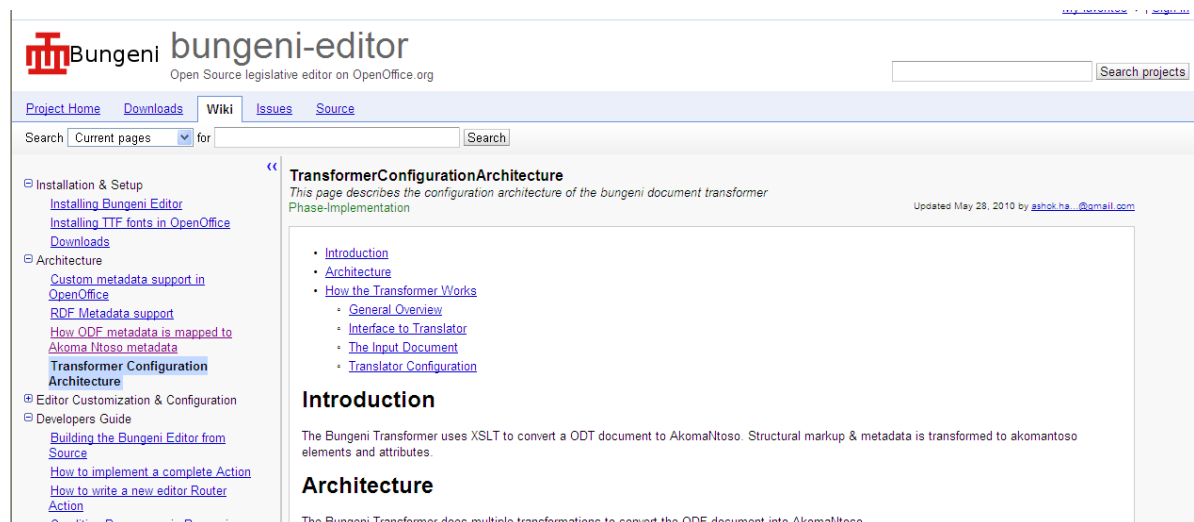
The editor is designed for high customisation, but this has some drawbacks and can be adjusted:

- Currently, there are many configuration parameters - because the tool aims to support a wide variety of document types – but this makes configuration complicated. The Bungeni team will simplify configuration architecture.
- Mark-up rules and transformation rules are implemented independently (this could also be an advantage), and in the future, the tool should support automatic generation of transformation rules from mark-up rules.

### ***Open source project***

The Bungeni editor is open source, which implies it can be re-used, the code can be accessed, modified and re-distributed. The project collaborative development environment is in place and public (<http://code.google.com/p/bungeni-editor/>) see Figure 37 Bungeni collaborative development environment screenshot.

This provides the framework for other users to join the project, understand the developments underway, the work needed to be done. Developers can request to contribute to the source code. The project provides a lot of documentation for developers about the customisation. Bungeni editor project is open source (in terms of the license), but it also is managed following an open source approach, enabling collaboration and promoting transparency.



**Figure 37 Bungeni collaborative development environment screenshot**

#### **6.7.4.4 Multilingualism**

The user interface is multilingual and can be translated to different languages by providing appropriate resource bundles for different languages (see section Internationalisation under 0). The main word processor interface supports all the languages supported by Open Office. The Bungeni Editor is multilingual and the document itself identifies metadata as a language, thus an English document is identified differently from a French document even though they may be translations of each other.

#### 6.7.4.5 Use of open standards

Bungeni uses only open standards, from the point of mark-up to the point of output. These standards are: ODF, Java, OpenOffice UNO, ODFDOM, XSLT, XML, OpenOffice RDF metadata. No conversion to or from proprietary formats is ever made, this preventing any vendor lock-in possibility and providing the added value of complete transparency in the processing of data.

#### 6.7.4.6 Use of a word processor

As legislative users are familiar with a word processor interface, the developers of the Bungeni editor decided to provide a familiar interface by building around the Open Office word processor window.

Using Open Office brings the benefits of using ODF format. The Bungeni editor also benefits from existing functionality in word processors such as spell checking, grammar checking, and multilingualism and track changes.

Using an open source application such as Open Office may provide more visibility and control over the roadmap and feature development than using a proprietary word processor, although “forking” (Libre Office was created as an independent branch of Open Office when Oracle bought Sun Microsystems) shows that some unpredictability also exists with non-proprietary solutions. Having access to the source code does provide freedom to build the application according to the needs.

The choice of specific file formats for word processors may also be imposed (or changed) by policies and interoperability frameworks, and the same format is not necessarily used across the complete drafting process (interoperability frameworks may not apply to parliament for example), but the choice of a common open standard ensures interoperability of the different tools across a process.

Maintenance of the tool is linked also to the word processing software and its evolution. The current release of Bungeni editor requires Open Office version 3.3.0.

### CASE STUDY 3 CONCLUSION

Bungeni editor is a young project, the production version is not yet available (as of March 2012). The tool is currently focusing on the mark up of legislative documents, but is planning additional functionality and addresses legislative drafters’ needs with a familiar word processor interface, and user guidance for applying a structure.

Bungeni editor is conceived as a re-usable and customisable tool by its architecture design and the use of open standards. A strategic aspect is creating Bungeni editor as an open source tool, with access not only to the source code but also to detailed documentation for users and developers and to the complete collaborative development environment. This provides transparency in the project management and enhances options for contributors to join the

community. The strategy avoids vendor lock-in because of the open source license, but also enables re-use and adaptability of the tool to a large user base thanks to the transparency of the management of the project.

It uses an emerging standard Akoma Ntoso as export XML schema format, and benefits from an expert contributor community active in various projects worldwide linked to the Akoma Ntoso standard.



## **CASE STUDY 4: LEGISLATIVE PROCESS SUPPORTED BY OPEN SOURCE DOCUMENT MANAGEMENT AND OPEN SOURCE WORD PROCESSOR**

### **6.8 S.O.L.O.N II, OPEN SOURCE WORD PROCESSOR AND DOCUMENT MANAGEMENT SYSTEM SUPPORT THE LEGISLATIVE PROCESS.**

*This case study is based on an interview of the project manager at the French Secretariat General of the Government as well as research in support documentation of the project.*

This case study analyses a solution currently being deployed (March 2012) and using an open source word processor (Open Office). The project is managed by the French General Secretariat of the Government, and is used by all stakeholders involved in the legislative process, including government and parliament. Data regarding the project was not available during the initial analysis of the As Is situation of LEOS project; developing this short case study on SOLON II describes the solution and also provides insight on various additional aspects such as change management and benefits of the solution compared to its previous version.

The SOLON I "Système d'Organisation en ligne des Opérations Normatives" (or the "online system for regulatory operations") solution was deployed in 2007. The system dematerialises the path of legislation published in the Journal Officiel de la République française (the "Official Gazette of the French Republic") in its "Laws and Decrees" edition via the involvement of ministries, the Council of State and the Secretariat General of the Government.

The newly deployed project SOLON II, much more ambitious, aims at being centred on the multiple stakeholders' needs, including the Assemblies and consultative bodies; it is a large project of document management, focused on adapting workflows and business process re-engineering, and providing drafters with guidance.

#### **6.8.1 Description of the solution**

##### **6.8.1.1 Components**

SOLON II is based on macros and usage of over thirty templates in Open Office. These templates created by the Secretariat General and are validated by DILA (Direction de l'Information Légale et Administrative) an institution also in charge of the publication of the Official Gazette.

The open source document management system Nuxeo handles the upload of documents including the pre-defined templates for types of law texts. Nuxeo is customised to meet the specific workflow needs of the French regulatory operations process.

Synchronized interfacing is done through web services (REST) in between SOLON's government processes and the ministries, and in between SOLON's parliamentary processes and the parliament.

The format used for the draft legislation is .ODT. The application also supports .doc, .docx which are formats used in the Assemblies, and XML for the metadata. For the annexe, the tool supports .ODS (spreadsheet) and .PDF. For the publication, the metadata is exported in XML, and the files in the folder are sent to the publications application. HTML is used for publishing on the website LegiFrance. For archiving, SOLON uses ODF and PDF/A for the annexes, these are recommended formats in the French Interoperability framework.

### 6.8.1.2 Functional aspects

The writer needs to conform to strict rules, but has some freedom during drafting, as the document is edited on the user's computer. When the document is uploaded into the system, the tool validates/checks the correct use of the styles according to the template. The tool also checks if the correct template is used.

### Creating a document

SOLON manages, for each act, a file which contains all the information regarding the act, which is found in each of the tabs at the centre of the screen. The system creates a new folder where all the information, annexes and other documents linked to the law text are found (see Figure 38).



Figure 38 SOLON file management screenshot

Tab	Description
Parapheur	(Signature book) Contains elements of the file to be published.
Fond de dossier	(File documents) Presents elements not to be published, in a folder defined according to the type of act.
Bordereau	(Slip) Presents information about the file.
Feuille de route	( Roadmap) Presents the roadmap (workflow) of the file.

**Journal** Presents the history of all actions on the document

**Traitement papier** Manages acts which are not in a digital format.



**Figure 39 SOLON screenshot: creating a document**

The user is guided in choosing a template when creating a document. He has to choose from a drop-down list among a series of types of acts, mandatory fields are proposed when creating a document.



**Figure 40 SOLON screenshot: transposition of an EU directive**

If for example, the user choses to create a law, the tool asks if this law is linked to a European Directive (see screenshot above). If this is the case, he will then need to fill in information regarding the Directive before finalising the creation of the new file.

## Updating a document

The workflow of the system provides the documents that need to be worked on by the user in his workspace under the tab “traitement” (see Figure 41)



**Figure 41 SOLON screenshot user interface – updating a document**


The folder needs to be “locked” by the user when he makes changes, in order to prevent other users to work simultaneously. (🔒)

## Changes and versioning management

The system provides a workflow for managing the documents from creation to publication. The act is created in a folder, which contains all documentation linked to the act.

The users who have worked on a document do not have access “downstream” to the changes in the document.

Changes in a text are tracked by generating a second file with the same NOR (unique identifier) in the file name, and storing it in the folder in the document management system, under a different version (see Figure 42).

Gestion des versions			
Version numéro 3 :	 Acte.doc	23/11/2011	Bdc
Version numéro 2 :	 Acte.doc	23/11/2011	Admin SGG
Version numéro 1 :	 Acte.doc	23/11/2011	Admin SGG
 Quitter			

**Figure 42 SOLON screenshot version management**

There is currently no management of consolidation.

## Document check

The use of templates allows for a consistent look by using the same style definition, a strict positioning of elements in the document, and enables the systematic change in the

configuration of all future texts by simply changing the style sheet. Only the administrator can edit, delete or create templates; this ensures that each model is numbered and updated.

The creator of the folder chooses the templates to be used in the folder. Experts in legal drafting need to be present when initiating laws in order to provide expertise on the choice of the right structure/ templates. The tool accepts only documents which content has complied with the styles in the templates.

The system checks compliance of usage of Open Office and the correct templates. These checks address:

- the use of open office writer format (except for laws);
- the use of the template linked to the type of act;
- the adequacy of the NOR number (identifier) in the metadata and in the corresponding field in the document.



**Figure 43 SOLON screenshot upload of a document**

Adding a document with a recognised template is allowed.

The Figure 44 shows the system mentioning that the file does not comply to the types of templates authorised in this type of folder. The user will then need to click on “annuler” (cancel).



**Figure 44 SOLON screenshot invalid file**

The administrators can add **new templates in a file folder**; the only mandatory fields in any template are: NOR number, Date of signature, Title of the Act. These fields are checked by the system when uploading the document.

## Metadata

At the creation of a folder, a series of metadata are automatically generated (see Figure 45 SOLON screenshot folder metadata). The user can then edit the remaining fields to complete the description of the act.

Données principales	
Information sur l'acte	
Statut	Initié
Type d'acte	Avis
NOR	PRMX1100124V
Titre de l'acte	
Catégorie acte	Réglementaire
Responsable de l'acte	
Ministère resp.	Premier ministre
Direction resp.	Secrétariat général du Gouvernement
Ministère rattach.	Premier ministre
Direction rattach.	Secrétariat général du Gouvernement
Nom du responsable du dossier	
Prénom resp. dossier	
Qualité resp. dossier	
Tél. resp. dossier	
Mél resp. dossier	
Créé par	Madame Thérèse MICHAUX 01 42 75 81 41 therese.michaux@pm.gouv.fr
Publication	
Date de publication souhaitée	
Signature	
Chargé de mission	
Conseiller PM	
Date signature	
Parution	
Pour fourniture d'épreuve	
Vecteur de publication	Journal Officiel
Publication intégrale ou par extrait	In extenso
Publication conjointe	
Mode parution	
Délai publication	
Publication à date précisée	
Parution JORF	
Date JO	
Page JO	
Numéro d'ordre dans le JO	
Parution BO	
SGG-DILA	
Zone commentaire SGG-DILA	
Base légale	
Base légale	

Figure 45 SOLON screenshot folder metadata

## 6.8.2 Users and change management

### 6.8.2.1 Users

SOLON is used for over 30 different types of legal texts, 30 000 acts per year. The types of texts include: laws, orders, notices, circulars, decisions, decrees, amnesties, lists, memos, ordinance, amendments, response from Ministries, and presentation reports.

Users are the Presidency of the Republic, the General Secretariat of the Government, Parliamentary Assemblies, Ministries, the Council of State and the consultative bodies.

The Ministers and directors of the various institutions use the tool, as well as secretaries. Experts in legal drafting need to be present when initiating laws in order to provide expertise on the choice of the right structure. The users need prior knowledge of word processing (Open Office).

### 6.8.2.2 Change management

The SOLON II solution is very different from SOLON I. User involvement during the SOLON II project elaboration is high. Five ministries volunteered for the pilots at first, then more ministries joined in.

They participate in various workshops aiming at supporting an iterative process for fine-tuning user requirements.

Animating the pilots is based on availability of the users, both from the business side and from the technical side. They all have access to the prototype, available on the intranet.

40 workshops of half a day were organised. These aimed at collecting user improvements on the prototype. Thematic workshops allowed focusing on specific aspects, and managing various expectations and agendas.

During the change management phase, training of trainers allows close coaching of current beta-users, and the creation of a network of experts in each institution, close to the users.

A “Grand Mess” is organised communicating on each important milestone achievement of the project. Manuals are created, and specific reference cards are published regularly informing the future users about the status of the project.

A hotline is also provided by the Secretariat General. This type of support will be organised with collaborators in each ministry in order not to flood the Secretariat General with too many basic questions.

## 6.8.3 Benefits and needed improvements of the new solution

Compared to the previous solution (based on an off-the-shelf tool, not specifically developed for the specific needs of the legislative drafting/publishing process, yet alone for the public administration needs), SOLON II is focused on the user.

The previous system was a less ambitious project, whereby users adapted their needs to the functionality of an off-the-shelf proprietary software. The budget of SOLON II phase 1 is fifteen times the budget of SOLON I. It includes the change management and the maintenance of the system over three years.



The fine-tuning of the functionalities and the business process re-engineering allowed an optimisation of the workflow an enhanced efficiency of the system.

The efficiency of drafting is enhanced thanks to the “forced” use of templates in the system. Enforcing valid structure at the beginning of the drafting process allows fewer resources needed for publishing. The resource used to check all the texts before publication is now working for the SOLON project.

The current phase 1 of SOLON II treats only metadata in XML. Moreover, there is no automatic consolidation of acts, the ministries are in charge of consolidating the pieces of law text. In SOLON Phase 2, all the acts will be generated in XML; digital signature will be supported (including of archived documents); management of consolidation will be addressed.

For the exchange of revisable and semi-structured documents, the French Interoperability Framework<sup>58</sup> recommends the use of document formats based on XML, with specifications standardised by ISO. Open Office is chosen as word processor as it supports ODF. The choice of Open Office as word processor was made before the project split into Libre Office. Although the call for tender mentioned Open Office, the SOLON team is analysing the feasibility of developing the system for Libre Office in the scope of the reply to the call for tender.

## CASE STUDY 4 CONCLUSION

SOLON II brings the users the benefits of a solution designed according to the business needs, based on the reengineering of the work processes and specific templates for each type of document. The use of a word processor and the validation on upload of the document provides the drafter with a degree of freedom. User guidance is mostly focused on choosing the right templates and ensuring metadata is encoded, but the drafter does not benefit from any additional support when drafting. There needs to be a legal expert for choosing the templates to be included in a file. The system focuses on an administrative workflow; drafters do not see what changes are made to texts down the line.

The project ensured user buy-in through extensive change management, with a high level of involvement in fine-tuning the requirements and regular communication on the achievements at the main milestones. The need for this involvement is mostly because of the re-engineering of the processes across multiple institutions.

Basing a solution on a word processor creates risks of unpredictability of roadmaps and support. SOLON II is investigating in adapting to Libre Office. Moreover, interoperability frameworks recommend specific document formats, and these need to be implemented by the word processors.

---

<sup>58</sup> [http://www.references.modernisation.gouv.fr/sites/default/files/RGI\\_Version1%200.pdf](http://www.references.modernisation.gouv.fr/sites/default/files/RGI_Version1%200.pdf)



## CASE STUDY 5: DESKTOP XML EDITOR CUSTOM DEVELOPED FOR LEGAL TEXTS

### 6.9 VEXPRO, A DESKTOP XML EDITOR FOR EDITING LEGISLATIVE TEXTS

*This case study is based on the replies to the questionnaires, desk research, user manuals and testing of the tool.*

Vexpro is an open source Java XML editor for legislative texts, developed by the State Chancellery in Estonia and licensed under EUPL v1.1. Legislative acts are published electronically on the Official Gazette website<sup>59</sup>. Vexpro is used mainly for publishing legislative acts and for creating consolidated texts (although it is technically possible to use the tool also in legislative drafting). Currently most Estonian regulative acts have been converted to XML-format with a converter specially developed for this purpose.

It is not platform dependent, can be installed easily on user's computer and it works without a web connection. The availability of a web connection is only necessary for automatic version updates.

user friendliness of interface and features, how the tool guides the user through the structure. This case study provides an overview of a desktop XML editor specifically developed for the Estonian legislative process, and an overview of the re-usability of such a tool in other legislative contexts.

#### 6.9.1 Rational for an XML editor<sup>60</sup>

In order to achieve interoperability, the Estonian public sector has established a strategic objective to make records management XML-language based. The State Chancellery is developing an XML tool for compiling and processing legislation, which can be easily used in a convenient user interface to create documents with complicated XML structure. The introduction of the tool will change user habits, but will not require knowledge of the XML language. The tool has been developed in accordance with the technical rules and complexity of legislative drafting.

The introduction of XML documents is a step forward towards ensuring the integrity of documents, as metadata is saved in the composition of documents and it is possible to monitor the process of legislative proceeding (the lifecycle of documents). The XML format is also a prerequisite for long-term storage of documents and complies with archiving requirements, which ensures the authenticity, integrity and reliability of documents. All documents that are submitted for legislative proceeding must have these features, which agencies can ensure through their internal records management. The project involves all agencies that participate in legislative proceeding: ministries, constitutional institutions and gradually all local government authorities as well.

<sup>59</sup> <https://www.riigiteataja.ee/>

<sup>60</sup> Information Society Yearbook 2009 – RISO (State Information Systems) <http://www.riso.ee/en/pub/2009it/#p=2-1-7>

## 6.9.2 Users

At present, the tool is not used in process of legislative drafting, it is used mainly for publishing legislative acts and for creating consolidated texts. It is technically possible to use the tool also in legislative drafting.

The tool is used to prepare consolidated texts of the following legislative acts are published in eRT, the website of the State Gazette: Laws, Decrees of the President of the Republic, Regulations and Orders of the Government of the Republic, Regulations of ministers, Regulations of the President of the Bank of Estonia, Regulations of the National Electoral Committee, Resolutions of the Parliament.

Users are about 60 people from the National Electoral Committee, the Bank of Estonia and all the Ministries. They are public officials taking part in normative or law formulation, specialised users in the preparation of consolidated texts.

Current plans include adopting the tool for use by local municipalities (over 200 in Estonia) in 2013 for publishing the complete text of regulatory acts in State Gazette from 2013 onwards. This means that trainings will be provided to all local municipalities and the obligation to use the tool will be levied with a minister's act. There will be a pilot phase with selected municipalities in Estonia.

The tool enables the user to create XML documents without knowing XML – special knowledge not needed. Training is needed to get accustomed to the functionality and to the use of the structure. Trainings for using VexPro usually last around 3-4 hours.

## 6.9.3 User friendly editing in a complex structure

Legislative Drafting Rules are established by a Government Regulation and are followed most of the time. The rules are established based on former practice, but there are situations where the legislative act does not meet the set rules and structure. Specific features may occur when consolidated texts are made on the basis of earlier documents.

The tool focuses on providing user-friendly editing in an XML file and providing a document valid according to an XML schema. The tool includes options that are commonly used for compiling documents with text editors. Information can be submitted by entering data in predetermined fields or by choosing values; the tool also automatically provides default values.

The format used for the draft legislation (currently not managed by Vexpro) is .doc or .odt, and XML for consolidated texts (managed by Vexpro). The format used for the annex is XML and can include PDF (forms, complex tables) or HTML – for non-structural text, tables and forms, and pictures. The final text is available in .XML and .akt – a compressed archive file.

### 6.9.3.1 User-friendly schemas

The XML schemas are available (in Estonian) online<sup>61</sup>.

Legislative drafting is based on predefined schemas (document templates) that determine the rules and restrictions for the use of elements in documents.

<sup>61</sup> [http://xmlr.eesti.ee/xml/schemas/oigusakt/tyviseadus\\_1\\_10.02.2010.xsd](http://xmlr.eesti.ee/xml/schemas/oigusakt/tyviseadus_1_10.02.2010.xsd) (root)

[http://xmlr.eesti.ee/xml/schemas/oigusakt/maarus\\_1\\_10.02.2010.xsd](http://xmlr.eesti.ee/xml/schemas/oigusakt/maarus_1_10.02.2010.xsd)

[http://xmlr.eesti.ee/xml/schemas/oigusakt/yksikakt\\_1\\_10.02.2010.xsd](http://xmlr.eesti.ee/xml/schemas/oigusakt/yksikakt_1_10.02.2010.xsd)

## Sub-schemas

There is one basic schema in which all elements are included. There is one sub-schema per template and one overall root schema, which is used in all templates except for one. Sub-schemas have fewer elements. The biggest difference are in metadata stages and statuses; there are nine templates: act of law; act of law to modify an act of law already in force; a regulation; a regulation to modify a regulation in force; individual act [legislation of specific application]; individual act to modify one already in force.

The content elements are: title, preamble, part, chapter, division, subdivision, sub-subdivision, section, subsection and clause.

Documents are displayed using the WYSIWYG (What You See Is What You Get) editor together with predefined style files. XSL style file is used, which is assigned to each element and its font size. The font used is Arial, the font size depends on the element: Title 38px, Part 15p, Chapter 13px, text 11px, footnote 9px.

## Creating a new template

Templates are created outside VexPro editor based on the root XML-Schemas for different types of legislative acts. A new template can be totally independent one or based on one of the existing templates. The tool has a versioning system, so it can access also older templates from a central repository. Creation of a new template does require understanding and some expertise in the structure of the schema.

### 6.9.3.2 Clear interface

The interface of the tool is very clear (see Figure 46), and the most commonly used functions are grouped in the menu bar. They include: creating a new document from template, opening an XML document, generating a print preview, printing, generating a PDF file, cut, copy, find, and online help (however, the link to Vexpro.eu does not work). Track changes and spell check are under development.



Figure 46 Vexpro screenshot menu bar interface

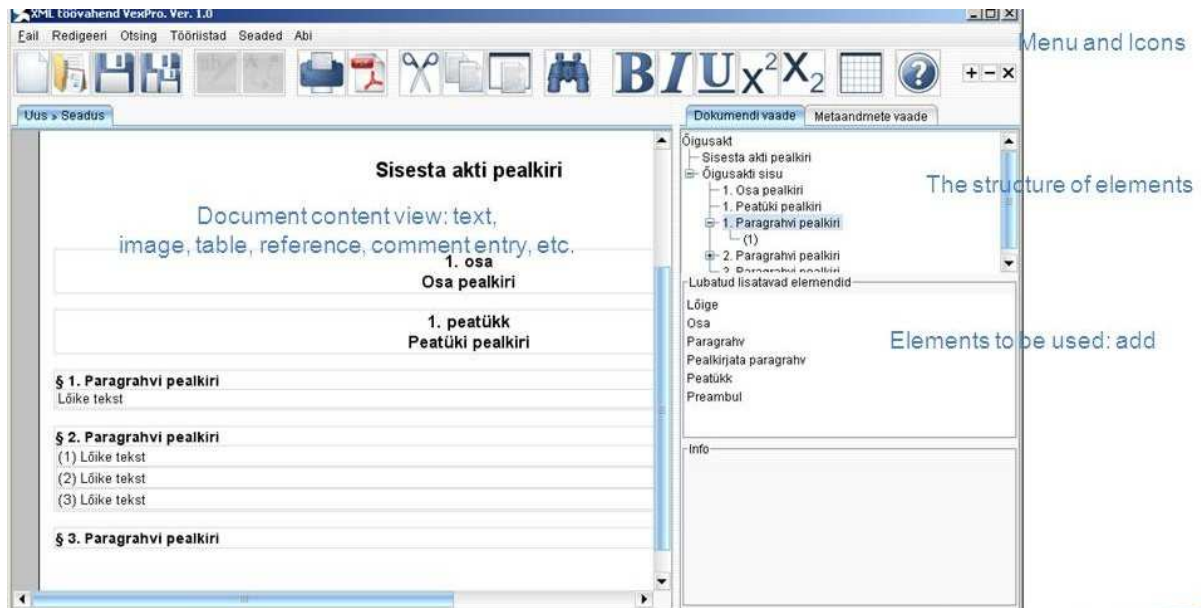
### 6.9.3.3 Editing a document

A document is created as follows:

1. An appropriate template is chosen – an empty template opens, with only a title template text. Drafting rules based on legislative schemes are integrated into the tool and the template helps the user to select elements of composition (e.g. chapter, title, subtitle, section, paragraph, etc.) Schemas are structured and contain only elements in the proper hierarchy.
2. From the right panel, the user can choose various types of elements to be used in the document. An element is chosen by double clicking on the name and it is added into the

template with element number and default text. The user then either types or copies the appropriate text of the document content. User input is similar to conventional word processors.

3. In another panel the whole structure of the document is available. When placing the cursor in a specific section of the text, the corresponding schema element in hierarchical structures can be seen status bar (see Figure 47).



**Figure 47 Vexpro screenshot - creating a document**

Elements can be moved to the right place in the document by dragging them in the structure panel, but only to where the elements are allowed to be. This allows relocating a single element or a group of elements together with their sub-elements in a structure tree.

Elements can also be deleted, including all the constituents, e.g. a section with all paragraphs, subsections and items, etc. While adding a new element in the text all the elements are automatically renumbered.

The user can also insert a reference, a note on the technical rules of legislative drafting, an explanatory note, an appendix, error correction or a signature (text field).

The user can also:

- insert tables, images, PDF files and HTML fragments to a document,
- use special characters, superscripts and subscripts and footnotes,
- emphasising elements (bold, italic, underline),
- add a legal act appendix (PDF), HTML container, references to other legislation and metadata, links.

The tool automatically adds a global ID to a legal document at its generation. An XML document passes different stages of the processing with the same ID, and new document versions are created and stored in various information and records management systems of agencies. The unique ID allows to find different versions of a document and to make them available for users.

The XML tool also adds global element-based IDs, which allows adding and storing references and links.

At present, amendments in a consolidated text are supplied with a note of updating, which reflects the date of publication of the amending document (a direct link is also added) and the date of coming into force of the amendment.

#### 6.9.3.4 Metadata

The tool supports three different types of metadata.

1. Default metadata (the user cannot change them - originate from different schemas depending on the type of the legislative act or are generated automatically by the tool), e.g. type of the document, type of text, date of the version, name of the schema, global ID.
2. Optional metadata (values that are described in the schema). A list is provided for the user.
3. User-defined metadata (the field type: number, text, date). The user can add the value.

The metadata view (see Figure 48) displays fields that are used during the various document processing stages. The user can choose the stage and status of processing, which generate metadata fields for which values can be chosen or entered by the user. The changes made are stored as metadata versions in the content of an XML document, and a version number and date is given to the document.

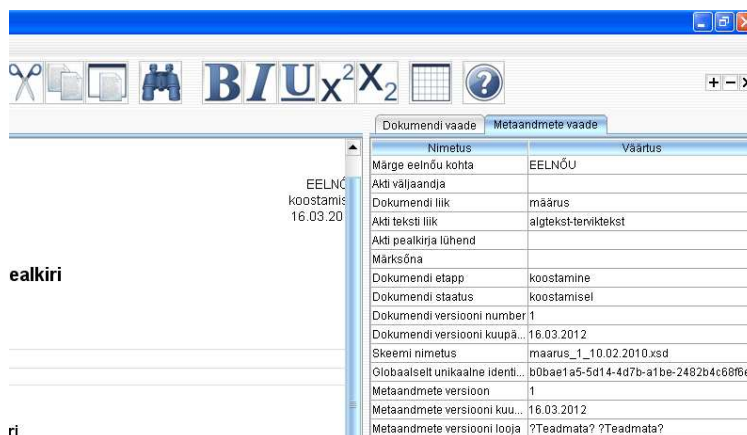


Figure 48 Vexpro screenshot metadata view

#### 6.9.3.5 The document check and user support

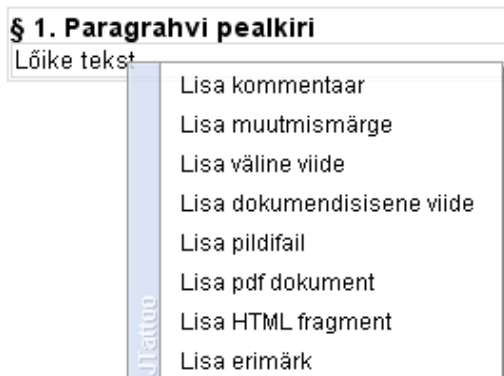
The structure of the document is checked against the schema and the user will be notified if the document is not valid by also indicating the incorrect element. The automatic editor also indicates inaccurate elements.

The correctness of the reference links are also checked (by accessing the Internet and the corresponding website) (see Figure 49).



**Figure 49 Vexpro screenshot reference link validation**

Contextual guidance is proposed, on right click on a piece of text, a menu proposes the valid elements which can be added in this section of the document (see Figure 50).



**Figure 50 Vexpro screenshot contextual guidance element proposal**

In the application menu “help”, there are links to online documentation, wiki and forum (but these links are broken). There is also a tab which opens a 20 page document on complete instructions for preparing and submitting texts according to the legislation and using the tool.

#### 6.9.4 Advantages and drawbacks

The tool is easy to use, and there is some assistance for creating the structure of the document. Advantages are linked to the fact that editing is made in the structure. Additionally, the XML editor provides support provided for adding PDF, HTML, images in the composition of the XML document. Special forms provided for creating a consolidated text.

Drawbacks exist however, because document creation is slightly different from the logic used in the usual routine of using a word-processor. The document is created by element-basis, i.e. only one element can be added at a time, which makes the work slower.

If a document that has been prepared in a different format than XML and contains elements that does not comply with the standard and technically required structure of the schema, it cannot be used as a text-based element. The tool allows to add these contents in an HTML container to preserve the initial structure of the text [a table; a formula; a picture, etc].

The tool allows working with only one document at a time.

There is also an issue of the XML tool performance when dealing with large documents. The issue is linked first to managing large files with image files and PDF files. Performance is achieved using a caching system. The issue is also linked to displaying a very large number of

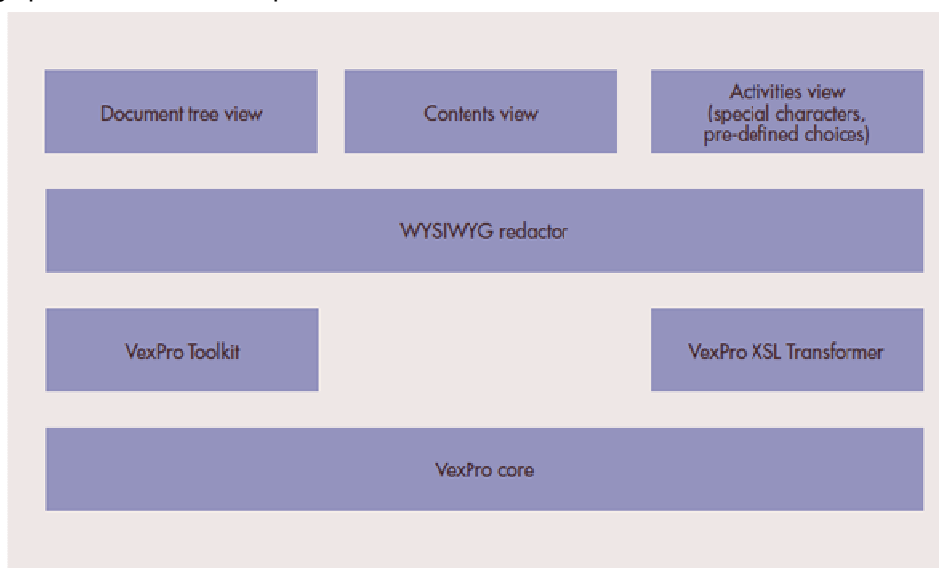
XML elements in the document view. The tool allows the user to display only parts of the elements of his document.

The tool does not provide XMLisation (or import of other formats of documents) – which is handled by another tool in the Estonian publishing process.

### 6.9.5 Re-usability

The user interface of the tool is in Estonian. As regards the architecture, Vexpro XML is a tool built on the following components: a text editor, and various user views. Views of the user interface are designed allowing the option to develop new views, without causing major changes to the existing application code.

The views address activities, content and document tree, and include displaying and editing metadata, displaying the document structure tree and removing the document, displaying files, creating special characters, export to PDF view, or validation view.



**Figure 51 Vexpro: General logic and architecture**

VexPro is certainly re-usable in other legislative solutions, as the business logic mentioned is less „Estonian-specific“ and more „legislative-specific“ in nature. XML Schemas are not embedded in the code of the tool. New schemas can be added, but not via the user interface of the tool (but by adding a schema document, its XSD and XSL to the schemas' folder of the VexPro tool, particular folders for schemas are in VexProConfig folder). The editor is not designed for editing the schema itself. Currently the editor is set up this way that it controls the schemas from the central server.

It is not clear or straightforward to estimate from existing documentation how easy it is to use another schema in Vexpro such as Akoma Ntoso. The Vexpro project is open source – the licence is EUPL v1.1, but internationalisation of the tool is not straightforward (there is no locale file with the user interface elements). There is no available collaborative project development environment, with access to feature roadmap, bug fixes, open issues, code contribution ...

Available documentation (in Estonian) addresses user manuals, and some description of the architecture.

## **CASE STUDY 5 CONCLUSION**

The aim of Vexpro is to allow users to edit XML in a user-friendly manner, and this goal is achieved. Vexpro is an XML editor used currently only for consolidation. Its use is also planned for drafting the initial legislation text, but this has not yet been implemented.

Re-usability of this open source tool is not very high because of the architecture design, availability of information regarding the development of the tool and the project management.



---

## CASE STUDY 6: A WEB XML EDITOR BASED ON AN OFF-THE-SHELF SOLUTION

### 6.10 WETSEEDITOR, A WEB XML EDITOR FOR AMENDMENTS

*This case study is based on replies to the questionnaire, interview of the principal architect and of the project manager at the Ministry of Justice and at ICTU (a body providing innovative ICT solutions for governments), as well as on findings following access to a demo version of the tool.*

Wetseditor is developed in the scope of the LEGIS program at the Ministry of Justice in the Netherlands. The tool is used in the Official Publications for quality review of the XMLised versions of legal texts; Basiswettenbestand is a large legal database containing almost all Dutch laws and decisions in XML format – using the BWB XML schema (see case study on Sdu XMLisation). Wetseditor is currently being introduced in the Parliament for producing amendments on consolidated texts, allowing the users to work directly in the XML files of the legal texts.

This case study provides insight in the implementation of a web based XML editor in terms of user buy-in: user benefits, user-friendly features for editing a structure, change management and governance of the use of the tool. It analyses user benefits, user friendliness of interface and features, how the tool guides the user through the structure. It also provides feedback on the pilot phase in Parliament.

#### 6.10.1 User benefits

##### 6.10.1.1 Direct view of effects of amendments on consolidated texts

The main objective and benefit of the Wetseditor is to always have a real time insight into the effects of the amendments on the consolidated texts (see Figure 52 Wetseditor screenshot: Consolidated text showing amendments). The user can load the existing law in the application, amend the text and then generate a new consolidated text. This greatly reduces the effort and chances for mistakes. The application also allows seeing where each sentence of legislation came from (such as Government and Parliament or date).

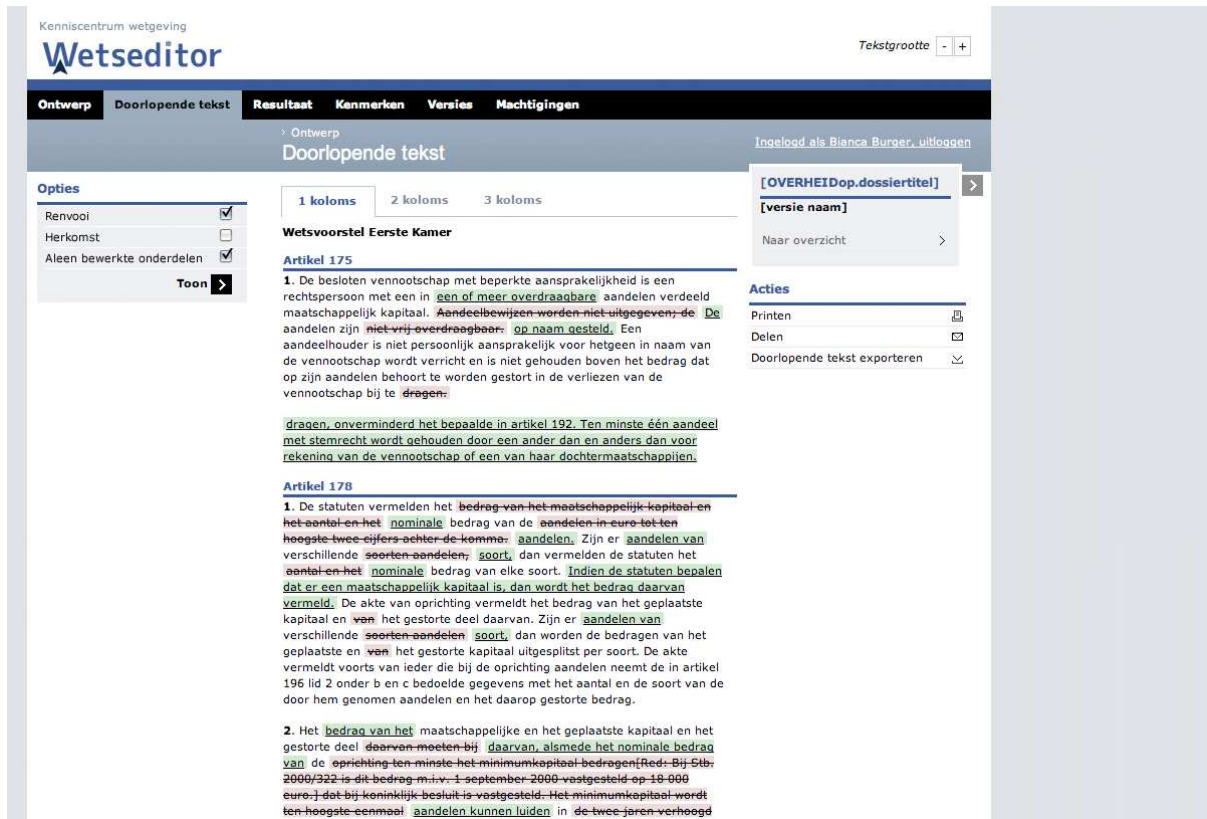


Figure 52 Wetseditor screenshot: Consolidated text showing amendments

### 6.10.1.2 Editing errors reduced

The other benefits of Wetseditor are:

- All authors always work with the latest version of the law and see the continuous text,
- It is easier to detect when there are collisions in between laws, because the final version of the law is immediately available and clear,
- The exchange of the source files (XML) between the parties implies that no more formatting process is necessary – this is done automatically, which leads to fewer steps, lowers the chance for errors, and lowers the work of proof reading, checking and correcting. The users do not need to apply the rendering rules.

### 6.10.1.3 Transparency of the drafting process

Kenniscentrum wetgeving  
**Wetseditor**

Tekstgrootte - +

Ontwerp Doorlopende tekst Resultaat Kenmerken Versies Machtigingen

Ontwerp Doorlopende tekst Ingelogd als Bianca Burger, uitloggen

**Opties**

- Renvooi ☐
- Herkomst ☐
- Aleen bewerkte onderdelen ☒

Toon

1 koloms 2 koloms 3 koloms

**Basis Wetten Bestand**

**Artikel 175**

1. De besloten vennootschap met beperkte aansprakelijkheid is een rechtspersoon met een in aandelen verdeeld maatschappelijk kapitaal. Aandelen worden niet uitgegeven; de aandelen zijn niet vrij overdraagbaar. Een aandeelhouder is niet persoonlijk aansprakelijk voor hetgeen in naam van de vennootschap wordt verricht en is niet gehouden boven het bedrag dat op zijn aandelen behoort te worden gestort in de verliezen van de vennootschap bij te dragen.

**Wetsvoorstel Tweede Kamer**

1. De besloten vennootschap met beperkte aansprakelijkheid is een rechtspersoon met een in een of meer overdraagbare aandelen verdeeld kapitaal. De aandelen zijn op naam gesteld. Een aandeelhouder is niet persoonlijk aansprakelijk voor hetgeen in naam van de vennootschap wordt verricht en is niet gehouden boven het bedrag dat op zijn aandelen behoort te worden gestort in de verliezen van de vennootschap bij te dragen, onverminderd het bepaalde in artikel 192. Ten minste één aandeel met stemrecht wordt gehouden door een ander dan en anders dan voor rekening van de vennootschap of een van haar dochtermaatschappijen.

**Wetsvoorstel Eerste Kamer**

1. De besloten vennootschap met beperkte aansprakelijkheid is een rechtspersoon met een in een of meer overdraagbare aandelen verdeeld maatschappelijk kapitaal. De aandelen zijn op naam gesteld. Een aandeelhouder is niet persoonlijk aansprakelijk voor hetgeen in naam van de vennootschap wordt verricht en is niet gehouden boven het bedrag dat op zijn aandelen behoort te worden gestort in de verliezen van de vennootschap bij te dragen, onverminderd het bepaalde in artikel 192. Ten minste één aandeel met stemrecht wordt gehouden door een ander dan en anders dan voor rekening van de vennootschap of een van haar dochtermaatschappijen.

**Artikel 178**

1. De statuten vermelden het bedrag van het maatschappelijk kapitaal en het aantal en het bedrag van de aandelen in euro tot ten hoogste twee cijfers achter de komma. Zijn er verschillende soorten aandelen, dan vermelden de statuten het aantal en het bedrag van elke soort. De akte van oprichting vermeldt het bedrag van het geplaatste kapitaal en van het gestorte deel daarvan. Zijn er verschillende soorten aandelen dan worden de bedragen van het geplaatste en van het gestorte kapitaal uitgesplitst per soort. De akte vermeldt voorts van ieder die bij de oprichting aandelen neemt de in

**Acties**

- Printen
- Delen
- Doorlopende tekst exporteren

**Figure 53 Wetseditor screenshot: view of a bill in various stages of the drafting process**

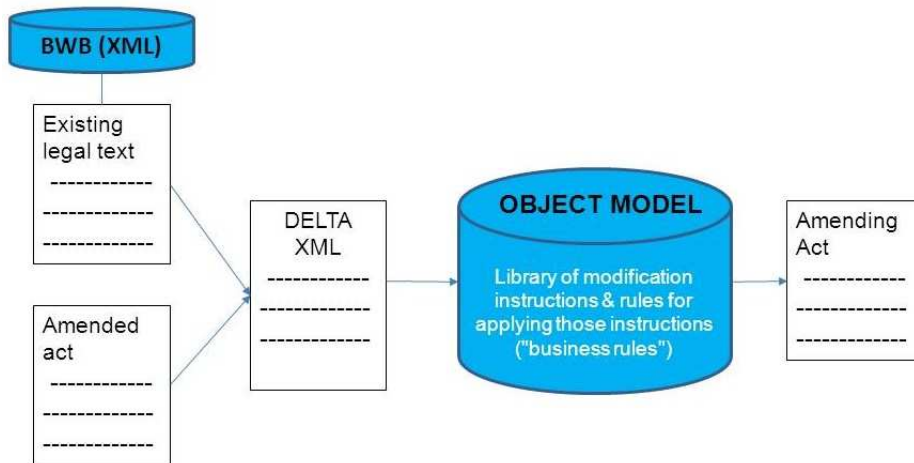
The drafters can follow the legislation drafting on a selected law; the tool provides a view of the continuous text of the bill according to its position in the drafting process. The screenshot in Figure 53 shows a presentation of the current valid bill in the first column, in the second column is the text which was proposed in Parliament and in the third column is the text sent to the Senate, after the Parliament voted on amending it. Being able to view where each sentence of a law text came from enhances the democratic process.

#### 6.10.1.4 Automatic formulation of amending instructions

Additionally, automatic formulation of amending instructions is handled by Wetseditor.

For example, a text imported from wetten.nl mentions “*Wanneer de reden van uitstroom een ongeschiktheid betreft geldt artikel 2 niet.*”. The text is changed using the Wetseditor to “*Wanneer de reden van het uitstromen een ongeschiktheid betreft geldt artikel 2 niet.*”

Wetseditor generates an amendment act: “*In onderdeel b wordt << de reden van uitstroom >> vervangen door: de reden van het uitstromen.*” (in section b is “de reden van uitstroom” replaced by “de reden van het uitstromen”).



**Figure 54 Wetseditor: technical model for generating automatic formulation of amending instructions**

The system exports in XML the existing law from the BWB database, available according to the BWB XML schema, and compares it with the amended act edited in Wetseditor. The difference generated by the comparison of the XML files is mapped with the existing library of modification instructions and rules for applying them, in order to generate the amending act.

There are currently 500 rules to generate amending acts. These are under constant revision by the “business” in order to reflect correctly the changes made to a legal text.

Previously to the development of the Wetseditor for amendments, all the work described above was generated by hand.

#### 6.10.1.5 Better working tools

For the user, the Wetseditor tool provides a series of features which were not available at all previously. These features are:

- Permanent availability of the continuous act, including consolidation and amendments,
- Automatic formulation of amending instructions,
- Online collaboration environment,
- One same source file edited throughout the whole process,
- Secured by authorization (access control) per file. There is one file per law. Authorisation is the same one as the users currently have. In the Parliament, all have access. In the government, access is provided to specific contacts who can then share the access with their department staff.

### 6.10.2 User-friendly editing in a complex structure

#### 6.10.2.1 User friendly schemas

The BWB schema provides a complex and detailed structure. Wetseditor provides an interface which hides the XML structure. Additionally, user-friendly schemas ensure that only allowed

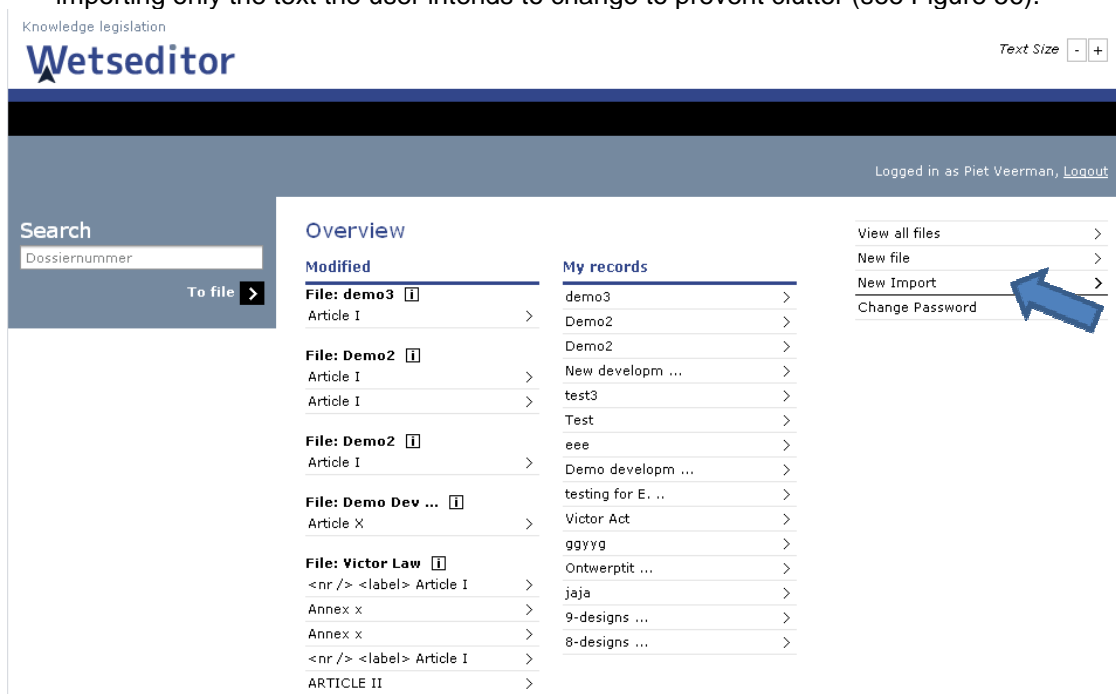
elements are active and mandatory elements are prompted in the user interface. These user-friendly schemas also depend on the type of document to be edited.

### 6.10.2.2 Editing interface and functionality

A fifteen minutes training is need for using the tool for the simple use cases. The interface is clear, not overloaded with command buttons.

The section below describes the editing of amendments in Wetseditor, through a series of screenshots illustrating the various steps.

1. In order to import a text of law from [www.Wetten.NL](http://www.Wetten.NL) the user can search (for example by title), using the Wetseditor interface (see Figure 55 ). The tool allows importing only the text the user intends to change to prevent clutter (see Figure 56).



Knowledge legislation

**Wetseditor**

Text Size - +

Logged in as Piet Veerman, [Logout](#)

**Search**

Dossiernummer

To file >

**Overview**

**Modified**

**File: demo3** ⓘ

Article I >

**File: Demo2** ⓘ

Article I >

Article I >

**File: Demo2** ⓘ

Article I >

**File: Demo Dev ...** ⓘ

Article X >

**File: Victor Law** ⓘ

<nr /> <label> Article I >

Annex x >

Annex x >

<nr /> <label> Article I >

ARTICLE II >

**My records**

demo3 >

Demo2 >

Demo2 >

New developm ... >

test3 >

Test >

eee >

Demo developm ... >

testing for E. .. >

Victor Act >

ggyyg >

Ontwerptit ... >

jaja >

9-designs ... >

8-designs ... >

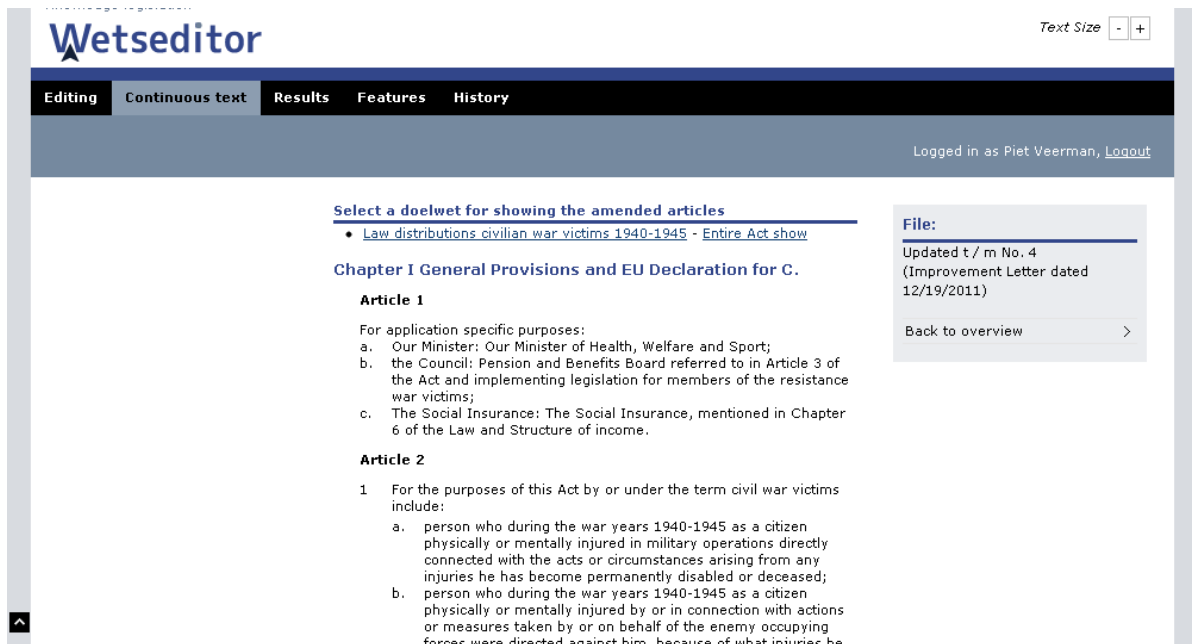
View all files >

New file >

New Import >

Change Password >

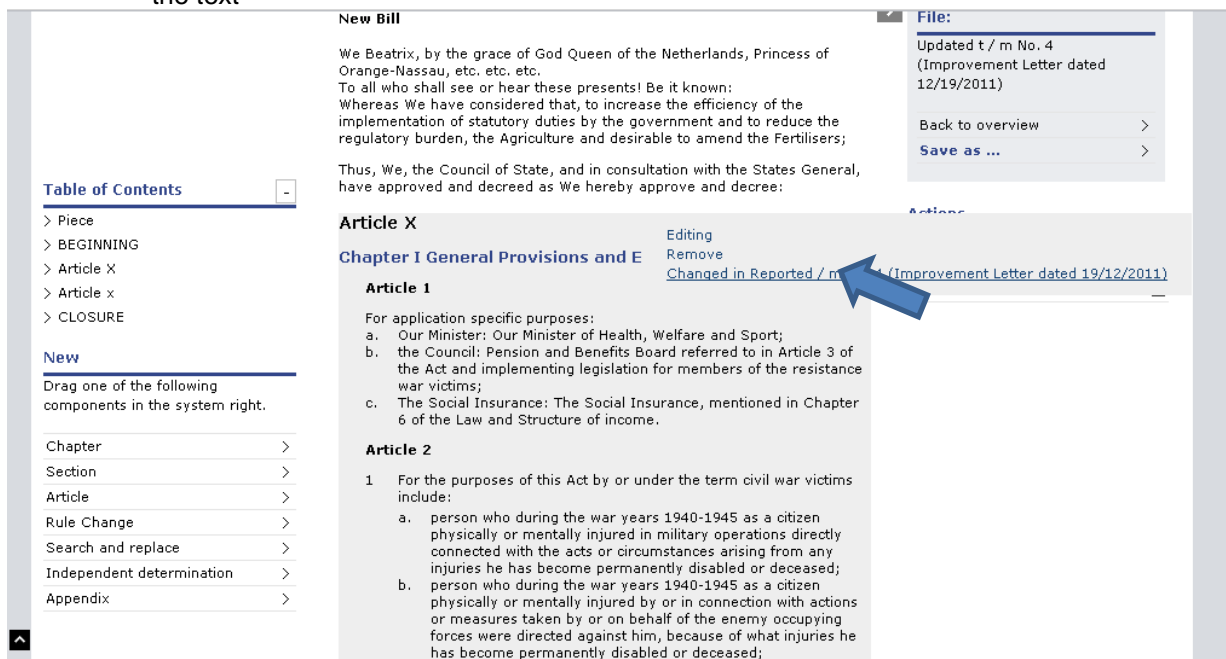
**Figure 55 Wetseditor screenshot: homepage**



The screenshot shows the Wetseditor interface. At the top, there's a navigation bar with tabs: Editing, Continuous text, Results, Features, and History. The 'Editing' tab is active. Below the navigation bar, there's a header area with the text 'Logged in as Piet Veerman, Logout'. The main content area displays a document titled 'Law distributions civilian war victims 1940-1945 - Entire Act show'. The document is structured into chapters and articles. The current view shows 'Chapter I General Provisions and EU Declaration for C.' and 'Article 1'. The text of Article 1 is displayed, detailing the purposes of the Act. On the right side, there's a sidebar with a 'File:' section showing 'Updated t / m No. 4 (Improvement Letter dated 12/19/2011)' and a 'Back to overview' button.

Figure 56 Wetseditor screenshot: view of text to modify

2. The user then selects a chapter to edit (see Figure 57) by passing the mouse over the text



The screenshot shows the Wetseditor interface with a focus on editing a chapter. On the left, there's a 'Table of Contents' sidebar with a list of components: Piece, BEGINNING, Article X, Article x, and CLOSURE. Below this, there's a 'New' section with a list of components: Chapter, Section, Article, Rule Change, Search and replace, Independent determination, and Appendix. The main content area displays the text of 'Article X' and 'Chapter I General Provisions and E'. The text of Article X is shown, detailing the purposes of the Act. A blue arrow points to the text 'Changed in Reported / m' in the 'Chapter I General Provisions and E' section, indicating the user's selection. On the right side, there's a sidebar with a 'File:' section showing 'Updated t / m No. 4 (Improvement Letter dated 12/19/2011)' and buttons for 'Back to overview' and 'Save as ...'.

Figure 57 Wetseditor screenshot: editing a chapter

3. An editing window opens (Xopus editor) – 2 changes are made (replace a word, and delete section b) then saved (see Figure 58).



Wij Beatrix, bij de gratie Gods, Koningin der Nederlanden, Prinses van Oranje-Nassau, enz. enz. enz.  
Allen, die deze zullen zien of horen lezen, saluut! doen te weten:  
Alzo Wij in overweging genomen hebben, dat het, ter vergroting van de doelmatigheid van de uitvoering van wettelijke taken door de overheid alsmede ter vermindering van de regeldruk, wenselijk is de Landbouwwet en de Meststoffenwet te wijzigen;

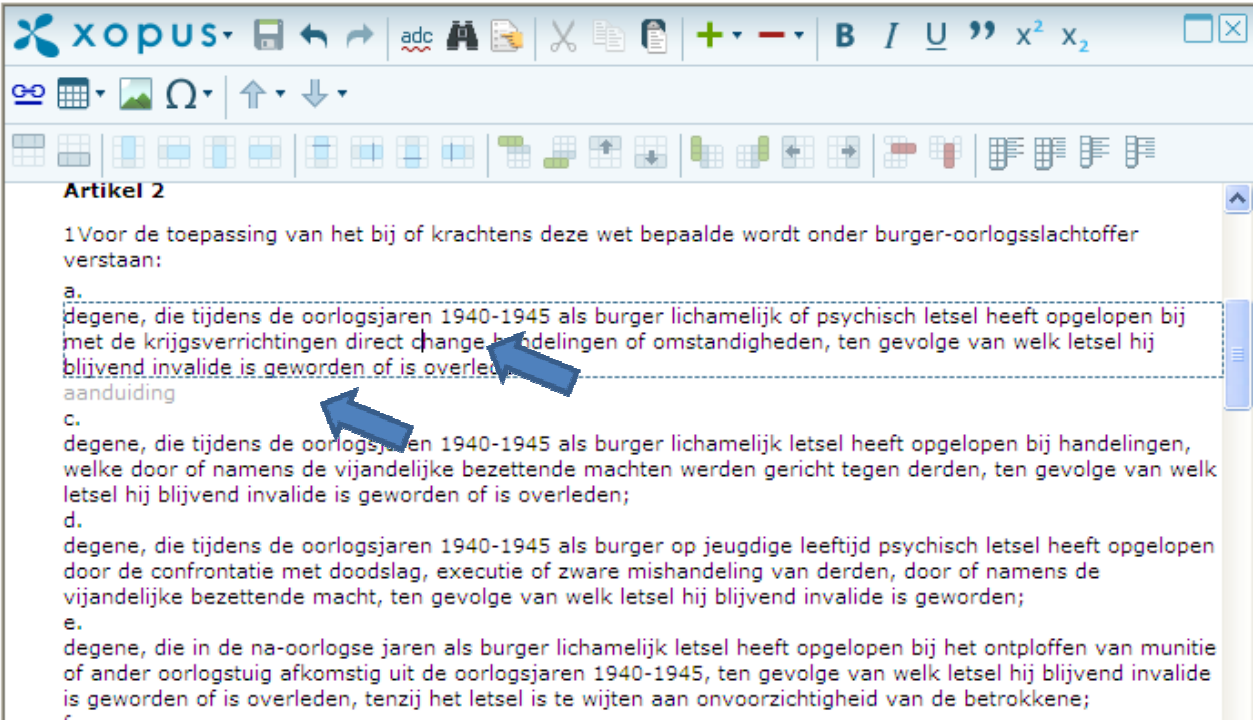
Zo is het, dat Wij, de Raad van State gehoord, en met gemeen overleg der Staten-Generaal, hebben goedgevonden en verstaan, gelijk Wij goedvinden en verstaan bij deze:

Verbetering d.d. 19-12-2011 )

Terug naar overzicht >

Opslaan als ... >

Acties



**Artikel 2**

1 Voor de toepassing van het bij of krachtens deze wet bepaalde wordt onder burger-oorlogsslachtoffer verstaan:

a. degene, die tijdens de oorlogsjaren 1940-1945 als burger lichamelijk of psychisch letsel heeft opgelopen bij met de krijgsverrichtingen direct ~~change~~ ~~handelingen~~ of omstandigheden, ten gevolge van welk letsel hij ~~blijvend invalide is geworden of is overleden~~ ~~aanduiding~~

b. degene, die tijdens de oorlogsjaren 1940-1945 als burger lichamelijk letsel heeft opgelopen bij handelingen, welke door of namens de vijandelijke bezettende machten werden gericht tegen derden, ten gevolge van welk letsel hij blijvend invalide is geworden of is overleden;

c. degene, die tijdens de oorlogsjaren 1940-1945 als burger op jeugdige leeftijd psychisch letsel heeft opgelopen door de confrontatie met doodslag, executie of zware mishandeling van derden, door of namens de vijandelijke bezettende macht, ten gevolge van welk letsel hij blijvend invalide is geworden;

d. degene, die in de na-oorlogse jaren als burger lichamelijk letsel heeft opgelopen bij het ontploffen van munitie of ander oorlogstuig afkomstig uit de oorlogsjaren 1940-1945, ten gevolge van welk letsel hij blijvend invalide is geworden of is overleden, tenzij het letsel is te wijten aan onvoorzichtigheid van de betrokkene;

e. degene, die in de na-oorlogse jaren als burger lichamelijk letsel heeft opgelopen bij het ontploffen van munitie of ander oorlogstuig afkomstig uit de oorlogsjaren 1940-1945, ten gevolge van welk letsel hij blijvend invalide is geworden of is overleden, tenzij het letsel is te wijten aan onvoorzichtigheid van de betrokkene;

f. degene, die in de na-oorlogse jaren als burger lichamelijk letsel heeft opgelopen bij het ontploffen van munitie of ander oorlogstuig afkomstig uit de oorlogsjaren 1940-1945, ten gevolge van welk letsel hij blijvend invalide is geworden of is overleden, tenzij het letsel is te wijten aan onvoorzichtigheid van de betrokkene;

**Figure 58 Wetseditor screenshot: amending a text in the editor**

Note: there is no support of the web browser Chrome with the Xopus editor

4. The user can view the changes made, under the history tab (Figure 59)

The new data input, fed to XSLT, generates sentences in a WYSIWYG manner. The resulting law describes changes on a word level - not sentence or paragraph - to prevent conflicts between different laws (generated from one same law, by different people at different times). For this, the structure's granularity is very high: down to parts of sentences.

The structure used in Wetseditor is an enhanced version of the BWB, the publisher's schema. The enhanced structure allows to define down to parts of sentences. The texts are automatically structured in this enhanced version of BWB on import in the tool.

Knowledge legislation

**Wetseditor** Text Size - +

Editing Continuous text Results Features History

Logged in as Piet Veerman, [Logout](#)

1-column 2-post

Updated t / m No. 4 (Improvement Letter dated 12/19/2011) [Current version]

423,787  
New Bill

We Beatrix, by the grace of God Queen of the Netherlands, Princess of Orange-Nassau, etc. etc. etc.  
To all who shall see or hear these presents! Be it known:  
Whereas We have considered that, to increase the efficiency of the implementation of statutory duties by the government and to reduce the regulatory burden, the Agriculture and desirable to amend the Fertilisers;  
Thus, We, the Council of State, and in consultation with the States General, have approved and decreed as We hereby approve and decree:

**Article X**  
**Chapter I General Provisions and EU Declaration for C.**

**Article 1**  
For application specific purposes:  
a. Our Ministers: Our Minister of Health, Welfare and Sport;  
b. the Council: Pension and Benefits Board referred to in Article 3 of the Act and implementing legislation for members of the resistance war victims;  
c. The Social Insurance: The Social Insurance, mentioned in Chapter 6 of the Law and Structure of income.

**Article 2**  
1 For the purposes of this Act by or under the term civil war victims include:  
a. person who during the war years 1940-1945 as a citizen physically or mentally injured in the military operations directly ~~connected~~ **Change** acts or conditions resulting from injuries which he has become permanently disabled or deceased;  
b. ~~person who during the war years 1940-1945 as a citizen physically or mentally injured by or in connection with actions or measures taken by or on behalf of the enemy occupying forces were directed against him, because of what injuries he has become permanently disabled or deceased;~~  
c. person who during the war years 1940-1945 as a civilian injured in physical acts by or on behalf of the occupying enemy forces were directed against any third party due to any injuries he has become permanently disabled or deceased;  
d. person who during the war years 1940-1945 as a civilian at a young age mentally injured by the confrontation with murder, execution, or grievous bodily harm to others, by or on behalf of the enemy occupying power, because of what injuries cause permanent disability has become;  
e. one who in the postwar years as a civilian physical harm suffered by the explosion of ammunition or other ordnance from the war years 1940-1945, because of what injuries he has become permanently disabled or deceased, unless the injury is due to carelessness of the person;  
f. the one in the post-war years in the former Dutch East Indies as a citizen physically or mentally injured in rioting that followed the war closely to December 27, 1949 there have occurred and that the nature and effects similar to the conditions referred to a , b , c or d , because of what injuries he has become permanently disabled or deceased.  
2 If the injury referred to in the first paragraph, no obvious reasons other than the war under the first paragraph is created, this is deemed to have been caused in the war, or in circumstances related. Taking into account the views and experiences of medical science on the relationship between war and mental and physical

**Figure 59 Wetseditor screenshot: view of the text of law with the changes made.**

- Wetseditor also allows to view only the changes made The tool generates an **automatic formulation** of the amendments, based on specific rules defined in collaboration with the user groups. (see Figure 60 - note: demo functionality for another law example than the one used previously)



Editing Continuous text **Results** Features History

Logged in as Piet Veerman, Logout

### Results

PROPOSAL OF LAW

We Beatrix, by the grace of God Queen of the Netherlands, Princess of Orange-Nassau, etc. etc. etc.

To all who shall see or hear these presents! Be it known:

Whereas We have considered that, to increase the efficiency of the implementation of statutory duties by the government and to reduce the regulatory burden, the Agriculture and desirable to amend the Fertilisers;

Thus, We, the Council of State, and in consultation with the States General, have approved and decreed as We hereby approve and decree:

#### ARTICLE I

The Act benefits civilian war victims 1940-1945 is amended as follows:

A

Chapter I is entitled "replaced by EU Declaration and General Provisions for C..

B

In Article 1, after

File:

Updated t / m No. 4  
(Improvement Letter dated 12/19/2011)

Back to overview >

Save as ... >

#### Actions

View Delta Xml >

See Proposal Xml >

Options

Options

Options

Figure 60 Wetseditor screenshot: automatic formulation of the amendment

6. The users can also view the new law, based on a number of amending acts, each including a series of amendments. The screenshot (Figure 61 Wetseditor screenshot: view of the initial text, the complete new text, and the editors) shows the initial text, the final text, the changes, and the users who have worked on these amendments.

Kenniscentrum wetgeving

## Wetseditor

Ontwerp Doorlopende tekst **Resultaat** Kenmerken Versies Machtigingen

Ontwerp Doorlopende tekst Resultaat Kenmerken Versies Machtigingen

Ingeleid als Bianca Burger, uitloggen

### Doorlopende tekst

1 kolom 2 kolom 3 kolom

#### Basis Wetten Bestand

##### Artikel 175

1. De besloten vennootschap met beperkte aansprakelijkheid is een rechtspersoon met een in een of meer overdraagbare aandelen verdeeld maatschappelijk kapitaal. De aandelen zijn op naam gesteld. Een aandeelhouder is niet persoonlijk aansprakelijk voor hetgeen in naam van de vennootschap wordt verricht en is niet gehouden boven het bedrag dat op zijn aandelen behoort te worden gestort in de verliezen van de vennootschap bij te dragen, onverminderd het bepaalde in artikel 192. Ten minste één aandeel met stemrecht wordt gehouden door een ander dan en anders dan voor rekening van de vennootschap of een van haar dochtermaatschappijen.

#### Wetsvoorstel Eerste Kamer

1. De besloten vennootschap met beperkte aansprakelijkheid is een rechtspersoon met een in een of meer overdraagbare aandelen verdeeld maatschappelijk kapitaal. Aandeelbewijzen worden niet uitgegeven; de aandelen zijn niet vrij overdraagbaar, op naam gesteld. Een aandeelhouder is niet persoonlijk aansprakelijk voor hetgeen in naam van de vennootschap wordt verricht en is niet gehouden boven het bedrag dat op zijn aandelen behoort te worden gestort in de verliezen van de vennootschap bij te dragen.

dragen, onverminderd het bepaalde in artikel 192. Ten minste één aandeel met stemrecht wordt gehouden door een ander dan en anders dan voor rekening van de vennootschap of een van haar dochtermaatschappijen.

#### Artikel 178

1. De statuten vermelden het nominale bedrag van de aandelen. Zijn er aandelen van verschillende soort, dan vermelden de statuten het nominale bedrag van elke soort. Indien de statuten bepalen dat er een maatschappelijk kapitaal is, dan wordt het bedrag daarvan vermeld. De akte van oprichting vermeldt het bedrag van het geplaatste kapitaal en het gestorte deel daarvan. Zijn er aandelen van verschillende soort, dan worden de bedragen van het geplaatste en het gestorte kapitaal vermeld. De akte van oprichting vermeldt het bedrag van het geplaatste kapitaal en het gestorte deel daarvan.

1. De statuten vermelden het bedrag van het maatschappelijk kapitaal en het aantal en het nominale bedrag van de aandelen in euro tot ten hoogste twee cijfers achter de komma. aandelen. Zijn er aandelen van verschillende soorten aandelen, soort, dan vermelden de statuten het aantal en het nominale bedrag van elke soort. Indien de statuten bepalen dat er een maatschappelijk kapitaal is, dan wordt het bedrag daarvan vermeld. De akte van oprichting vermeldt het bedrag van het geplaatste kapitaal en het gestorte deel daarvan.

[OVERHEIDop.dossier] [versie naam]

Naar overzicht >

#### Acties

Printen

Delen

Doorlopende tekst exporteren

Wetsvoorstel TK

2e NvW d.d. 7 okt 2010

3e NvW d.d. 10 nov 2011

Wetsvoorstel TK

A. Dijkstra

A. van Vliet en Koppejan

A. Jacobi

A. van Vliet en Koppejan

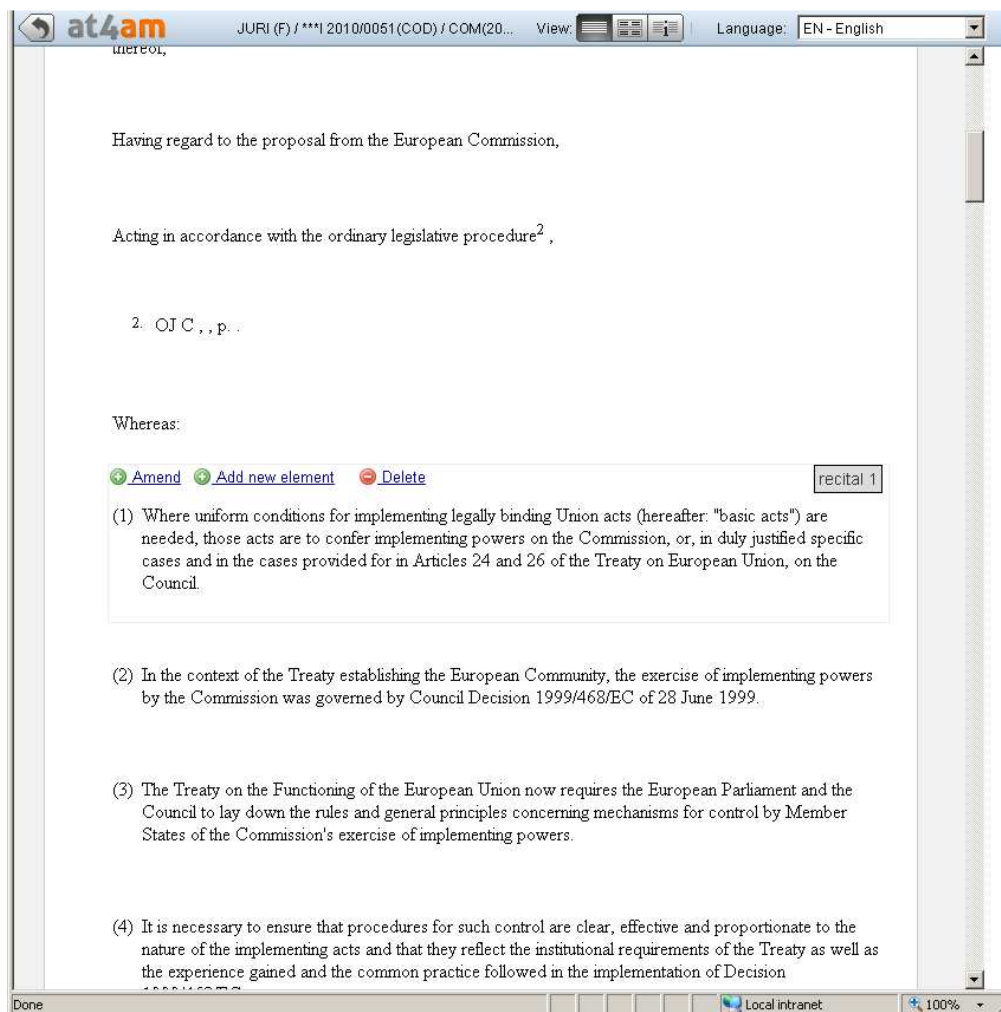
Figure 61 Wetseditor screenshot: view of the initial text, the complete new text, and the editors

### 6.10.2.3 A similar tool: AT4AM

In order to grasp the user-friendliness features of such tools, and compare them to a certain extent, this section presents the user interface of features of a similar tool, AT4AM, the web tool for authoring amendments at the European Parliament.

Comparison is feasible to a certain extent, as each system complies with specific amending rules. The main benefit for using AT4AM is to spare Members with the burden of applying the rendering rules for drafting amendments; these are applied automatically as for Wetseditor.

1. The Member opens her/his own language version of a Draft legislative Act in the Web based editor and hovers the mouse on the text
2. AT4AM tells whether the current paragraph is editable or not, and provides information (recital 1 in this case) that will be used as metadata for a possible amendment



**Figure 62 AT4AM screenshot: editable paragraphs**

- Add new amendment: recital 4

Original	Authors	Additional info
<p>It is necessary to ensure that procedures for such control are clear, <del>effective</del> and proportionate to the nature of the implementing acts and that they reflect the institutional requirements of the Treaty as well as the experience gained and the common practice followed in the implementation of Decision 1999/468/EC.</p>		<p>It is necessary to ensure that procedures for such control are clear, <u>adequate</u> and proportionate to the nature of the implementing acts and that they reflect the institutional requirements of the Treaty as well as the experience gained and the common practice followed in the implementation of Decision 1999/468/EC.</p>

ABC

< >

≡ ≡ ≡

|

☰ ☲ ☱

|

↶ ↷

|

✕ ✖

|

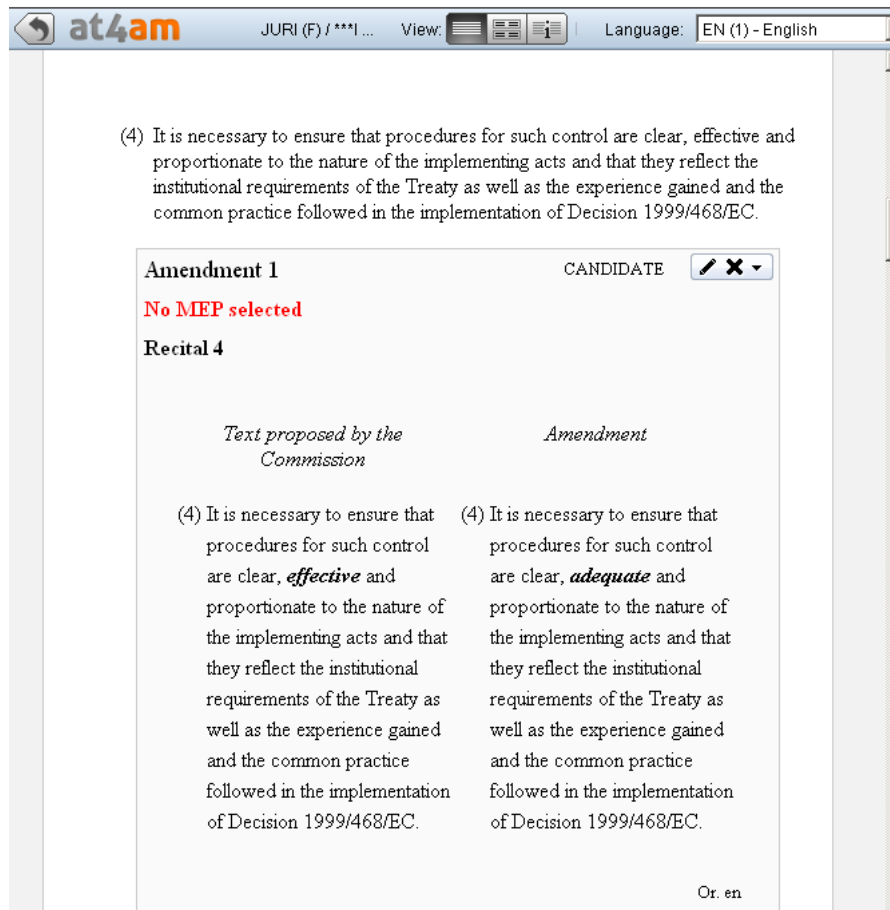
Ω ™

☐ This is a multilingual "panaché" amendment Or.  

Add as new amendmentCancel

**Figure 63 AT4AM screenshot: double column amending**

As a result, the amendment is generated. All the drafting rules are automatically applied when the floating editor is closed. In particular, the track changes are rendered in bold/italics. Metadata are also added automatically.



**Figure 64 AT4AM screenshot: amendment generated**

The individual Amendments, as content, rendering codes and metadata, are saved to the AT4AM Data base. They can be exported as Microsoft Word or XML files (according to the Akoma Ntoso schema).

## 6.10.3 User involvement and change management

### 6.10.3.1 Working with a core user group

The project involves a key user group of twelve to fifteen people, and a core user group of three people participating in the week to week decision process. The users are all involved in drafting laws, from Parliament, Government and Ministries.

These user groups are part of the project from the beginning. Although the software is now in use, there are still many adjustments to make and decisions to make from the business side. The core group is still strongly involved.

This early involvement and fine-tuning phase allows adoption and adaptation of the tool.

### 6.10.3.2 Planning the pilot with the right features

The first pilot – in Autumn 2011 - needed to be rescheduled; this first step failed, because the tool did not provide enough functionality for the users to feel an actual benefit in exchange of changing their work habits. The prototype proposed only allowed to handle the amendments generated by the parliament. Once more features had been developed in the tool, not only generating the amendment, but also including the entire amending act and consolidated version, the pilot was launched. It started in January 2012, and is successful so far because of good participation of the users.

### 6.10.3.3 Meeting users' needs

The strategy of Wetseditor usage in Parliament is to involve the users from the beginning, by having them test the various prototypes, and iteratively enhance the features.

#### ***Defining user-friendly schemas***

The BWB schema provides a complex and detailed structure. Wetseditor provides an interface which hides the XML structure. However, it is also important to adapt the structure to the type of user in order to impose the correct use of the schema. User-friendly schemas ensure that only allowed elements are active and mandatory elements are prompted in the user interface. User-friendly schemas are created by mapping the schema to the specific user vocabulary; this requires two types of expertise: the knowledge of the business and the understanding of the schema.

The editor adapts the terms used from the XML schema in the interface buttons to comply to vocabulary commonly used by different types of user groups. For example, a “change article” in the schema is named either an “article” or a “header” depending on the type of user/document generated. This adaptation takes about a week per type of document, in order to allow consensus on the choice of words by key users.

### ***Fine-tuning the formulation of amending instructions***

The formulation needs to reflect exactly the intention of the amendment. This is difficult to do in an automatic manner, with pre-defined rules. The users are still providing negative feedback on some of the wording generated when applying the rules, and are working with the project team on refining the rules.

#### **6.10.3.4 Governance for the usage of the tool**

For a general usage, 15 minutes training is enough for treating simple use cases. Although the use of the Wetseditor is straightforward, it is not imposed to the various drafters in the process of law drafting in the Netherlands. The current model is an XMLisation to BWB schema of all the texts (in various formats) at publishing.

The drafters can at any moment export their work from the Wetseditor in several formats, and email it to the next drafter, hence creating a parallel workflow. At the end of the process, the files are XMLised (see case study on Sdu XMLisation), and each department pays for its amount of files which need processing.

#### **6.10.4 Benefits and trade-offs**

Besides the user benefits mentioned in section 6.10.1, this section will analyse other benefits, as well as trade-offs of such a solution.

##### **6.10.4.1 A solution for all the challenges**

Wetseditor provides a technical and functional solution based on a series of challenges identified at the beginning of the project:

Challenge 1: The amendments affect more than one “law text” (such as the version through Parliament and through Senate). *Solution:* The tool edits directly in the structure, and the different law texts are generated automatically.

Challenge 2: There are often repeating changes made during an amendment. *Solution:* Wetseditor offers a “search and replace” function.

Challenge 3: The management of tables is not straightforward. *Solution:* the choice of a good XML Editor which supports tables.

Challenge 4: The management of pictures is not straightforward. *Solution:* The tool links to the BWB database where the pictures are stored.

Challenge 5: Install the solution for every user. *Solution:* Wetseditor runs in the web browser, no installation needed.

Challenge 6: Make sure there are not multiple versions of a same document being edited. *Solution:* Everyone works online

##### **6.10.4.2 Online**

Wetseditor is an online web solution, which makes it easy to deploy and maintain.

However, the users need to be always online to use the tool. The project architect mentioned at the beginning of the project that they would consider developing an off line version; it was decided not to, for various reasons including the lack of need from the users.

#### 6.10.4.3 Web standards

The choice of a web solution implies the use of web technology and open standards. Wetseditor uses REST, a style of software architecture for distributed hypermedia systems like the World Wide Web.

An important concept in REST<sup>62</sup> is the existence of resources (sources of specific information), each of which is referenced with a global identifier (e.g., a URI in HTTP). In order to manipulate these resources, components of the network (user agents and origin servers) communicate via a standardized interface (e.g., HTTP) and exchange representations of these resources (the actual documents conveying the information). When a client holds a representation of a resource, including any metadata attached, it has enough information to modify or delete the resource on the server, provided it has permission to do so.

The metadata in Wetseditor uses microformat. A microformat<sup>63</sup> is a web-based approach to semantic mark-up which seeks to re-use existing HTML/XHTML tags to convey metadata and other attributes in web pages. This approach allows software to process information intended for end-users automatically.

#### 6.10.4.4 A proprietary XML editor

Besides the web tool, Wetseditor uses as XML editor a proprietary tool (Xopus editor), identified as state-of-the-art in terms of functionality. Being able to integrate an existing editor in the solution is very interesting in terms of functionality support. The trade-off however of using such a tool is vendor lock-in when the tool is not open source. The potential issues are a lack of control on the feature roadmap in the supported different versions of the tool. Another issue could be the need to adapt the Wetseditor if the Xopus editor changes technology strategy.

### CASE STUDY 6 CONCLUSION

The Wetseditor is a powerful tool for managing amendments with high added value functionality. The user-friendliness of the interface and functionality of the web tool is striking. The user benefits are also very high and users can immediately grasp these benefits because it is easy to get familiar with using the tool. However, there is a strong need for change management, and including a core and key user group throughout the complete project is inevitable. These key users are part of the project, and define the user-friendly schemas which adapt the complex structure to the type of user/document. They also fine-tune the automatic formulation rules. A lesson learned in this project is that the prototypes used during the pilot phase needs to provide enough functionality for the pilot phase to be a success. The tool is not

<sup>62</sup> [http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)

<sup>63</sup> <http://en.wikipedia.org/wiki/Microformat>

imposed, the XMLisation of a document can be done at the end of the drafting process too, and each department pays its share.

In order to achieve this high level of functionality, Wetseditor is based on a proprietary web XML editor, and on a web application which supports the exchange of data with the database, features for interacting with the structure, automatic amendment formulation, etc.... The use of a web application has many advantages, including the openness of standards and the ease of maintenance. The need for a proprietary editor creates potential vendor lock-in issues.



## 7 OVERALL CONCLUSIONS AND RECOMMENDATIONS

The case studies covers various topics, from functional to technical, and include benefits and trade-offs of the tools as well as re-usability aspects.

### 7.1 CONCLUSIONS

This As Is study provides an overview of the existing situation and planned projects in Member States, national parliaments, publication offices and EU institutions. It identifies needs, trends, strengths, best practices and opportunities. The study takes into account a wide variety of stakeholders and their challenges. Stakeholders include legislative text drafters and solution providers, developers and project managers, as well as academic experts and standardisation bodies. Consulted documentation includes questionnaire replies, published research, experts' presentations, policies and available studies, frameworks, specifications and existing standards.

The study shows that there are existing and emerging standards. Recommendations regarding the use of these standards are presented in the section "7.2.1". The initial analysis of tools shows that there is a wide diversity in usage, coverage of needs and technology used. The case studies identify benefits and trade-offs of various types of tools, either based on word processors or developed to provide user-friendly XML editing.

#### 7.1.1 Solutions based on word processors

More than half of the solutions analysed are based on off-the-shelf word processors. Almost half of these are based only on templates and add-ons and do not include the use of XML, and they all intend to change the system and/or enhance the tool so to include the use of XML for structuring the documents and exchanging data.

Generally speaking, the main disadvantage of tools based on word processors is to miss validation during the drafting. On the other side, this approach – preserving the freedom of drafting – fits the legislative drafters' needs whose first aim is to provide legislative content in a document. Complying with a structure is yet another step in the drafting process.

Their main advantages are applying a structure to the document using user-friendly templates and providing semantic check of the document. The document can also be converted and validated according to the chosen XML schema thanks to the use of an internal data format (for instance .ODT in Bungeni editor), which allows users to easily switch between different XML standards at the conversion stage. The tools can import and export XML files.

The case studies on structure management and user guidance in tools based on word processors show that different levels of guidance and constraint can be applied, from the simple use and validation of templates (as in SOLONII), to extensive user guidance (as in eNorm) and even imposition of a structure (as in Lex Dania). eNorm is carrying out research to provide on

the fly structure checks, which will suppress or minimize the extra step of validation of the document. Lex Dania is a unique example as it imposes editing directly in the structure using a word processor, but the user does not benefit from a completely familiar word processor interface.

The case studies provided more insight in benefits and trade-offs of using off-the-shelf word processors.

This allows using existing functionality and benefiting from a user accustomed interface. But this also links the system to the evolution of software which is not controlled by the users (vendor lock-in), and can be referred to as “double maintainability”: maintaining not only the tool to the needs of the users but also maintaining compatibility with the different versions of the word processor. There is no control on the roadmap of feature development and maintenance of available features. The choice of open source word processors provides more transparency on the roadmap of the tool. It also provides the users with the option of joining the project to participate in the development and in the feature request.

Using a word processing tool implies the choice of a document format and the issue of interoperability of various document formats. The choice of these specific file formats may be imposed (or changed) by policies and interoperability frameworks, and the same format is not necessarily used across the complete drafting process. These tools are mainly used for drafting legislation, but some solutions are used also for providing a mark-up to existing documents (XMLisation).

### 7.1.2 Solutions for user-friendly XML editing

The other solutions are specific developments based on checking for compliance with a pre-defined structure (and only one) at all times, while providing some user-friendly interface (e.g. WYSIWYG). These tools can be web solutions or desktop solutions. Some can be based on an existing off-the-shelf web XML editor. The main benefit of this type of tool is that the user is transparently constrained by the editor to perform only valid operations on the document in such a way that, starting from a valid document, only valid documents can be produced.

The (known) use of these tools is not wide-spread and implementations are currently limited to some features (ex: amendments management), or to some parts of the organisation (publications, parliament) and other pilots. The case study analyses four of these tools: features of Vepro, Wetseditor and AT4AM, specific features such as XMLisation for XMLeges, and change management for Wetseditor.

In most cases, the usability of the interface of these tools is very high. The challenge for the user is to learn to work in the structure. The tool shows only the authorised options to the user. This is done through the creation of user-friendly schemas (or sub schemas), based on the complex complete XML schema, presenting only information useful for the user, and based on the type of text he needs to work on. Drafting functionality is not as developed as in tools based on word processors, for instance regarding semantic check. With a desktop XML Editor, one

cannot fragment texts among different working groups. The schema needs to be centrally validated only – unless sub schemas are created specifically for these working groups, which is a complex process to be done for each large document edited collaboratively. There is also a problem with the sharing of metadata.

However, there are high added value features possible when editing the structured content, such as automatic generation of amendments. These tools are used also for generating consolidated texts. Generally speaking, the use of an XML editor is very interesting when dealing with already XMLised documents, and working on the existing content, such as amending, or even translating (working with multiple language versions of a document in XML format). Additionally, these tools edit only XML, so there is no conversion to or from proprietary formats. This avoids vendor lock-in.

With regards re-usability of these tools, designing the architecture of the tool with re-usability of some core “generic” components in mind is important, as this type of tool is based on a specific XML schema. Re-usability is also linked to the legal requirements of legislation drafting and publishing. For example, the types of amendment produced (diffing/synopses, spell checker/red line approach, or amendments basic actions/mention what has been changed) can be different, and legally binding amendments presentation vary from one country to another.

## 7.2 RECOMMENDATIONS

The recommendations build on the results of the As Is analysis and lessons learned regarding legislative drafting methods across the EU. The study provides a better understanding of the way public administrations address the issue of drafting their legislations. The recommendations take into account the best practices identified, and propose improvements and areas where common efforts and developments are possible. They provide guidelines for a next phase in the LEOS project to support Member States and European Institutions in developing efficient and effective solutions.

These guidelines focus on two aspects:

- the identification of on-going standardisation efforts to define a common framework for the structure of legal texts,
- the identification of best practices and reusable solutions while evaluating the potential benefits offered by open source software.

These aspects are to be treated independently: a tool can support different structures, and a structure can be supported by different types of tools.

The recommendations were discussed during a workshop with experts, addressing both the adequacy of existing standards for structuring legislative documents and the choice of tools.

## 7.2.1 The need for a structure

### 7.2.1.1 Use of XML

The initial analysis results show that there is a move towards XML. This move is motivated both by the need for applying a common structure throughout the legislative process, from drafting to publishing, and the need for an open standard for exchanging data with other systems and therefore enhancing interoperability.

According to the IPEX survey results, almost half of EU Parliaments/Chambers uses XML for some legislative documents, and almost half plan to introduce XML mark-up for parliamentary documents or for new documents. The case studies show that there is a need for XMLisation of documents at - and before - publishing phase.

The case studies highlighted strong user benefits when the solutions allowed editing existing content in XML, such as automatically generating amendments.

Solutions exist to provide XML mark-up on legislative texts.

There are many initiatives – National and International – regarding standards or specifications for structuring legislative documents, and they address the definition of XML Schemas.

The use of XML, XML schemas (and user-friendly sub schemas) is identified in the As Is study as a good practice and as a main and common trend.

### 7.2.1.2 Standards and use of a common schema

#### ***Limited use cases for a common standard***

Whereas many Member States define National XML schemas, there is no convergence towards the use of an existing standard or specification for an XML schema.

Experts working on solutions in Member States and interviewed have not yet identified the need for being compliant on a schema-level across different legal systems in the EU. A common structure could be considered in the context of EU Directives. Even then, this may not be useful. Occasionally, the laws of a member state may already comply with the outcome of the implementation of the directive, and the state involved would be required only to keep its laws in place. Usually, member states are required to make changes to their existing laws (transposition) in order for the directive to be implemented correctly.

There is currently a lack of good practices for re-use of legal information, whether legislation or case law.

A study from the Forum of National Gazettes highlights that there are important differences in legislative culture causing different structures in legislative documents and hindering the possible creation/use of a common standard.

### ***Emerging initiatives for common standards***

Experts across Europe are currently working at a EU level on a reference framework / a common system for the identification (URI) and metadata of law– the European Legislation Identifier (ELI)? It is in line with the existing European Case Law Identifier (ECLI) initiative<sup>64</sup>.

According to the IPEX survey, almost half of the EU Parliaments/Chambers are expressly available or interested in making their XML documents compatible with XML documents produced by other Parliaments or institutions. Most of them require that a common standard is established to this end.

Some experts highlight the potential benefit for using a meta-schema which would enable a comparison of similar sections in legal texts which each have their own schema. This would allow for example a comparison of a transposition of a directive across Member State, but this has not been done yet. Such a meta-schema would facilitate the interconnection of legal databases and the performance of search engines. A meta-schema exists: CEN MetaLex is a jurisdiction-independent XML format for representing, publishing, and interchanging legal texts. It was developed to allow traceability of legal knowledge representations to their original source. As the need has not yet been identified concretely, the feasibility of conversion of the documents from existing National Schemas to MetaLex has not been addressed in the Member States. An exception is the MetaLex Document Server and website “doc.metalex.eu” which publishes all Dutch regulations (initially in publisher’s BWB XML schema) as CEN MetaLex and RDF Linked Data. The legislation.gov.uk portal publishes legislation form across the UK, and uses CEN MetaLex for the identifiers.

The rational for interoperability across the EU is strong. It is also pushed also by the paradigm of Open Data. Reuse of legislative data is a challenge. In the scope of the Public Sector Information Directive, although there are no clear rules on the obligations of government agencies producing legal information or on the level of service of re-use, institutions can publish regulations as linked data.

Some Member States do not have a schema. CEN MetaLex may be used as a basis for creating a more detailed and specific schema. Another option is to build upon an existing, more specific, CEN MetaLex compliant schema such as Akoma Ntoso and prune undesirable elements and add desired ones according to the legislative culture of the country.

The recently created OASIS LegalDocumentML (LegalDocML) Technical Committee has among its objectives to promote a common legal document standard based on Akoma Ntoso. It aims to collect requirements from the community of the stakeholders who create, manage and use legislative and legal documents in order to extend and refine the standard. It will create a common mechanism for naming and linking resources (URI) and achieve compatibility with Akoma Ntoso 1.0., CEN MetaLex and applicable LegalXML vocabularies. It is used by several parliaments, including the European Parliament.

---

<sup>64</sup> [http://europa.eu/legislation\\_summaries/justice\\_freedom\\_security/judicial\\_cooperation\\_in\\_civil\\_matters/l0056\\_en.htm](http://europa.eu/legislation_summaries/justice_freedom_security/judicial_cooperation_in_civil_matters/l0056_en.htm)

Using the Akoma Ntoso (LegalDocML) schema is identified as a good practice, for several reasons including the fact that the standard is emerging and contributors are welcome. An interoperability expert from the Belgian Federal ICT Public Service has requested to join the Committee.

The emerging standard is conceived to accommodate multiple legislative cultures. It is descriptive and can be used in civil law and in common law. It handles various types of representations of amendments, according to legally binding requirements of each country, whether they have to be narrative, presented in two columns or written in the text highlighting the previous text. Additionally, structured documents according to the Akoma Ntoso structure are easy to convert to FORMEX documents for example – an XML schema which is less detailed, and oriented towards publishing (presentation / typography oriented).

### 7.2.1.3 LEOS Phase 2 and the use of a structure

#### ***Akoma Ntoso***

As the use of Akoma Ntoso is identified as a good practice, in LEOS Phase 2 Member States and the European Commission may consider various ways of benefiting from this emerging standard:

- Evaluating its feasibility of use in Member States who do not have a schema and who intend to implement one

In the initial analysis of LEOS study, many Member States mentioned their intent to create a new system, or move to open standards, and/or to structure data. According to the IPEX study, eleven Parliaments/Chambers plan to introduce XML mark-up for parliamentary documents or for new and additional series of documents.

There is therefore a need for a structure. LEOS could organise a workshop to present the Akoma Ntoso schema and its design. During the workshop, attendees would also be assisted in evaluating its use by analysing a sample of some legislative texts and the adequacy of the schema. The workshop would be organised with some of the respondents to the LEOS questionnaire, and some IPEX members who intend to introduce a structure in their legislative process.

- Evaluating the feasibility of Parliaments which have XML documents to become compatible with Akoma Ntoso, in order to exchange documents among Parliaments (as identified in the IPEX study)

The IPEX study mentions that twelve Parliaments/Chambers are expressly available or interested in making their XML documents compatible with XML documents produced by other Parliaments or institutions. Most of them require that a common standard is established to this end.

LEOS project phase 2 could organise a workshop with IPEX members, present the Akoma Ntoso schema and support them in adopting Akoma Ntoso for their documents, or making their schema Akoma Ntoso compatible.

- Evaluating its scope of use and feasibility of use at the European Commission

The European Parliament is using the Akoma Ntoso schema. The European Commission produces a much larger scope of types of documents, and could evaluate together with some OASIS technical committee members which types of documents should use the schema. Some documents have a defined structure usually maintained throughout the legislative process, and freedom of drafting is required for other types of documents, which could then follow a separate workflow.

- Taking part in the specifications of the standard by joining the technical committee in order to ensure it is adapted to the needs

The LEOS project could support the workshop participants – including Member states and the European commission - in joining the OASIS technical committee. For example, some Member States have similar structure needs, due to a common legislative culture, and could designate a member to contribute to the standard on their behalf.

- LEOS structure task force

Following the organisation of the dissemination and supporting workshops mentioned above, the LEOS project could then create a task force with Member States strongly interested in the standard and participating in the OASIS technical committee, who could then support other interested parties by organising local workshops. The Members would also follow other standardisation activities, such as ELI - the European Legislation Identifier, a reference framework / a common system for the identification (URI) and metadata of law.

### ***MetaLex***

Although the use case for a common EU standard has not been identified so far, there could be a benefit from mapping content of legal texts from various Member States, such as comparing the different implementations by Member States of a directive.

LEOS Phase 2 could create synergies with IPEX project and some volunteer Parliaments in Member States using XML for mark-up of parliamentary documents, and test the feasibility and the outcome of such a comparison, using the MetaLex standard. The Akoma Ntoso standard could be evaluated as inadequate for exchanging XML documents from different parliament, due to incompatibility with existing schemas in the parliaments. The evaluation of MetaLex could then also be done during the workshops mentioned above.

#### **7.2.1.4 Conclusion**

The LEOS project is carried out at a phase where there is a move towards XML, at a moment when a significant number of Member States express the need for a structure, and for some, even a common structure.

The phase 2 of LEOS should take advantage of this timing to provide dissemination of existing knowledge and expertise, and promote the use and the contribution to an emerging standard aimed at complying with the different legislative cultures.

## 7.2.2 The need for divers tools

### 7.2.2.1 Support and freedom of drafting as important as structuring

The study shows that there is diversity in the types of tools used.

- Some specialise in user guidance for drafting, some in amendment generating, others support and XMLisation (mark-up according to a schema) of existing texts in various formats.
- Some tools are used from drafting to publications, and are based on word processors and XML.
- Drafting solutions exporting to XML are researching how to manage exceptions in the structure.
- Some tools are designed to edit an XML structure in a user-friendly manner, and are used for consolidating texts or amending or checking XML documents before publication.
- Some of the identified tools editing XML structures can be used for drafting, but have not (yet).
- One system using a tool providing drafting in a structure provides an option to the users to outsource the structuring of their texts.
- Editing tools are often based on word processors because of the variety of editing functionality available in the software.
- Editing tools based on word processors are researching how to provide on the fly structure checks.
- Editing tools focusing on XML structure editing provide less functionality for supporting users in drafting legislation.

From this, one can conclude that

- The need of freedom of drafting is as important as the need to comply to a structure,
- (Currently identified) tools editing in a structure are not meeting the needs of drafters.

### 7.2.2.2 One process, two cultures

Two main user groups can be identified: the initiators of content (the legislation drafters) and those working on existing texts (translators, publishers, legislation drafters creating amendments).

#### ***Initial drafting of documents***

The initiators of content of documents are often from a conservative user base. This user base is often large, and deals with a wide series of documents. Legislative rules are complex.

In some cases, as experts interviewed insisted: *“There is a need for XMLisation, but one does not want legal drafting rules to block political or legislative work.... A tool cannot simplify these*



*rules, formalness and validity should be only a final goal. XML is an export".* The drafting tool should allow the user to focus on the text, and keep on working even if the structure is not (yet) valid.

This user culture requires maintaining some freedom of drafting. Tools for drafters handling a strict structure need to accommodate exceptions. This can be done for example by creating a template for documents with barely any structure, and a corresponding sub schema generated from the common schema. Another option is to ensure XMLisation after drafting.

Drafters need to comply with a structure at the end of their drafting, because exporting to XML provides strong benefits including interoperability with other systems and enhanced efficiency throughout the complete legislative process. However, they should have an option to manage exceptions, by having a system allowing another actor to XMLise.

There are three scenarios for exporting drafted content to XML:

*Drafting => Validation in complete structure => complete content export in XML*

*Drafting => Non validation => XMLisation (with mark-up tools) => complete content export in XML*

*Drafting => Validation in exception template => partial content exported in XML => XMLisation (with mark-up tools) => complete content export in XML*

This shows the benefits of mark-up tools, such as Bungeni editor or xmLeges, which allow the mark-up of existing legal texts according to a schema. Additionally, Bungeni allows to first write content in the document before the structure is marked-up.

### ***Working on existing documents***

Generally speaking, the use of an XML editor is very interesting when dealing with already XMLised documents, and working on the existing content, such as amending, or even translating (working with multiple language versions of a document in XML format is efficient).

There is a user benefit for this type of tool. The case studies mention the use of the XML editor tools at publishing phase and at amendments or consolidation phases. The case studies also mention the intention of using this type of tool at the drafting phase, but this has not yet been done.

#### **7.2.2.3 A drafting and mark-up tool for LEOS Phase 2**

The legislative drafting community would benefit from a tool allowing mark-up and export to XML according to the XML schema mentioned above as best practice (Akoma Ntoso).

Following the best practices identified in the case studies, the mark-up tool should:

- i. be re-usable and customisable (export also to other schemas),
- ii. provide editing functionality
- iii. not be based on a word processor because:
  - Dependency on other solutions (maintenance, vendor lock-in)
  - not be incompatible with users' desktop policy and habits (formats in Interoperability frameworks).
  - Leos tool would need to be used across a very wide user base. but if there is a need to use a word processor, and choose among several document formats, it is recommended
    - to follow the Digital Agenda Guidelines for standard based ICT procurement<sup>65</sup>
    - take into account all aspects when evaluating a standard (as described in the ISA CAMSS – Common Assessment Method for Standards and Specifications<sup>66</sup>)
    - take into account the benefits of using ODF as described in the case study
- iv. use open standards.

The Bungeni editor complies with almost all of the recommendations. The LEOS Phase 2 project could therefore consider using the Bungeni editor as Proof of Concept for a mark-up tool. There is a user community involving experts who will contribute to further develop and enhance code and functionality of the editor, and the tool is highly re-usable. An ISA Member State expert in interoperability is considering analysing the tool.

Ideally, the mark-up tool could be built without using a word processor.

Considering xmLeges for LEOS Phase 2 is also an option, as it complies with all of the recommendations stated above – except for the high re-usability of the mark-up process technique which is based on probabilistic heuristics for which the machines need to learn from a very large number of correct documents. The tool also still has issues with managing large documents, an issue addressed in the development of the Vexpro tool and linked to the use of Java.

If LEOS were to consider the ideal tool, it would be a web application which could work online and offline, which does not provide a priori validation – respects the freedom of drafting – with on the fly structure check, and provides mark-up and semantic checks. The experts working around Bungeni and Akoma Ntoso are considering developing such a tool, with an open source

---

<sup>65</sup> EU Digital Agenda Action 23 [http://ec.europa.eu/information\\_society/newsroom/cf/fiche-dae.cfm?action\\_id=181](http://ec.europa.eu/information_society/newsroom/cf/fiche-dae.cfm?action_id=181)  
Draft guidelines available at: <http://cordis.europa.eu/fp7/ict/ssai/docs/ictprocurementworkshop-dec2011/draftguidelines-action23-21dec2011.pdf>

<sup>66</sup> [http://ec.europa.eu/isa/actions/02-interoperability-architecture/2-2action\\_en.htm](http://ec.europa.eu/isa/actions/02-interoperability-architecture/2-2action_en.htm)  
<https://webgate.ec.europa.eu/fpfis/mwikis/idabc-camss/>

Method available at:

license. It is interesting for the LEOS project to keep in touch with the Bungeni community and its future activities.

LEOS can consider providing a tool with the highest chances of re-usability, a basic tool which can be enhanced according to the various needs of Member States' legislative rules. This would be a web editor for texts, including spell checking and export to XML. It can consider re-using web services provided by the AT4AM project of the European Parliament (for instance, the spell checker). It is not clear yet which tools and/or can be re-used from AT4AM, but it is recommended to investigate further, as the European Parliament will develop soon a similar tool as AT4AM for drafting legislation.

#### 7.2.2.4 A tool for amendments and consolidation for LEOS phase 2

The legislative drafting community would benefit also from a WYSIWYG XML editor providing high-added value functionality such as AT4AM or Wetseditor – automatic consolidation or amendment generation.

This tool would be complementary to the mark-up tool, and help convince the users of the advantages of XMLisation because of its benefits. It allows users to make changes on valid documents which will produce only valid documents.

This tool should be highly customisable, because the legal validity of amendments depends on the requirements of different legislative cultures and is linked to how the amendment is presented (synopses as in AT4AM, or formulation of the amendment as in Wetseditor, ...).

As for using an existing tool for this proof of concept, the choice is small. Vexpro has low re-usability and adaptability. Wetseditor impose the use of proprietary software and license costs. xmLeges can be an option, but would need to be adapted to present legally binding amendments.

Another option would be to build on the best practices identified with Wetseditor and AT4AM, including the high usability of the interface, and develop a proof of concept as an online web application with WYSIWYG XML editing capabilities, re-using some of the existing functionality of AT4AM and adapt it to other requirements for amendments.

Developing such a tool can be complex when providing the option to edit any XML schema. LEOS could consider developing the tool supporting the Akoma Ntoso schema, thus addressing users who do not use an XML schema in Member States.

#### 7.2.2.5 LEOS Proof of Concept

As debated above, the LEOS Proof of concept could produce two types of tools, one providing mark-up and editing, the other providing consolidation and amendment features for existing XML documents.

If the proof of concept doesn't re-use existing tools, then one single web tool could be developed. Web solutions are to be considered because of the use of open standards and the advantages of web services such as REST, identified as a good practice with the Wetseditor tool.

Using the proof of concept should be targeted towards one specific type of document. The pilot phase would then use the proof of concept in the main work process. The XML file can easily be inserted in the current process, and the pilot is not disruptive for the current work of the user.

#### 7.2.2.6 LEOS tool task force

The LEOS project could create a task force with Member States in order to discuss the options for the proof of concepts, and prioritise them.

The members of this task force would include the developers of the tools and the future users. LEOS would benefit from the participation of the European Parliament AT4AM project manager, also currently developing a legislative drafting tool online similar to AT4AM. ISA experts or expert contacts in charge of the new legislative systems in Member States should also participate. Members of the OSASIS technical Committee would be interested in participating, as the TC aims also at creating an "interdisciplinary virtuous circle for legislative drafting guidelines that take advantage of XML standards and from tools that help apply XML to legal content in day-by-day uses".

The task force could debate on technologies to be used in the proof-of-concept, taking into account various emerging initiatives, interoperability of document formats, as well as advantages and drawbacks of the different types of tools.

The development of a new LEOS open source tool should ensure re-usability and customisation by other parties. In order to do this, the management and the development of the project should be done in a transparent manner – allowing other developers to join the project, understand the roadmap and contribute with their needs. Therefore, the use of an open source collaborative development platform is recommended - the Join Up platform of the European Commission is designed to support this type of initiative. The task force can also create a LEOS community on Join Up, as a meeting point and debate area.

#### 7.2.2.7 Conclusion

The recommendations cover considering both re-using existing open source tools and developing new ones. The next phase of LEOS should continue creating synergies among stakeholders, and benefit from cross collaboration among existing open source projects.

## 8 ANNEX 1: OTHER STUDIES ON LEGISLATIVE EDITING SOLUTIONS AND TOOLS

### 8.1 FEATURES OF SYSTEMS FOR MANAGING BILLS IN PARLIAMENTS

World eParliament Report 2010<sup>67</sup>.

Systems for managing bills must have a number of characteristics to be responsive to the needs of the members and staff of parliaments. The survey focused on several of these, including:

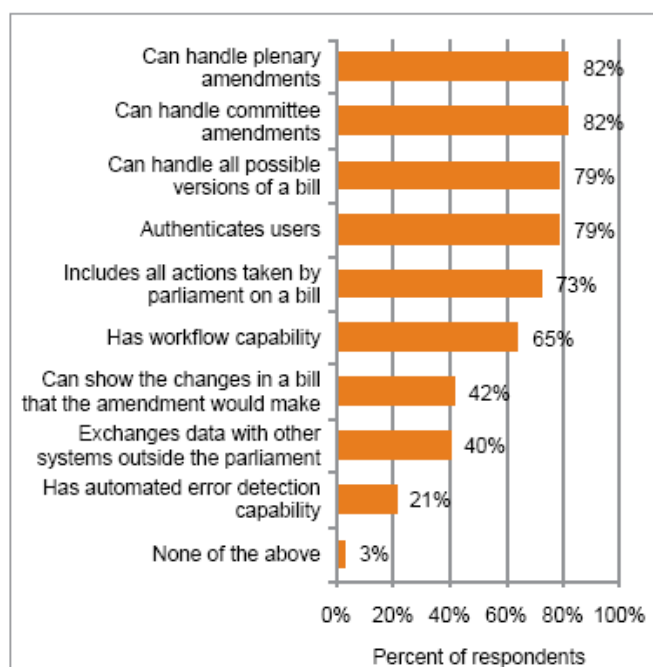
- Workflow. This allows bills to be moved automatically and smoothly among the members, officers and organizational units responsible for preparing and distributing them. Workflow also includes the ability to control versions so that authorized changes by one person or office are not overwritten by another.
- Accommodations of all versions of bills. It is important that all versions of proposed bills be introduced in the system as soon as possible. These include preliminary versions that are under active consideration for presentation to the body; versions that are considered and reported by committees, along with committee amendments if they are part of the process; versions considered and voted upon in plenary sessions, along with amendments considered in plenary; and versions sent from the legislature to the executive.
- Exchange and integration of documents and information. To have the complete legislative history of an act, it is essential that a bill system be able to integrate relevant documents and information related to a specific measure, such as amendments, plenary votes, status steps, and committee reports and activities, along with documents from other chambers, the government, or the judiciary.
- Accommodation of bills with special formats. Some types of bills, such as those dealing with the budget, may have particular requirements that affect their presentation online and in paper. A bill system must accommodate these requirements.
- Authentication of users. This is a crucial security procedure for ensuring the accuracy and authoritativeness of the text of the bill. There are various ways to implement authentication and the most secure systems may require both a fixed password and a constantly changing password or a physical token.

For those parliaments that have deployed a DMS for bills, the figure below shows their capabilities.

Well over 75% of parliaments have four of the nine features listed in the figure and two thirds have six of the nine. The features listed by less than half of all parliaments are among the most difficult to implement. One conclusion is that many of the DMS in place have many of the important functions that enhance their usefulness.

---

<sup>67</sup> <http://www.ictparliament.org/wepr2010>



## 8.2 FINAL REPORT OF THE WORKING GROUP “DATA CAPTURING AT THE SOURCE” - EUROPEAN FORUM OF OFFICIAL GAZETTES<sup>68</sup>

The following requirements which the Working Group has identified as being important in any system which are desirable for the successful implementation of **systems covering the entire process from drafting through to publication**:

- Techniques that are easy to understand and learn

Any system has to be capable of being used by a wide range of people who have different objectives to achieve and will have different levels of knowledge and experience and who may be infrequent users of the system. It is essential therefore that it is easy for everybody to understand and learn and must not be devised solely for experts who will use the system on a regular basis.

It is also essential that in specifying the system that all potential groups of users are involved in the process involved as the system is developed. This will ease the task of implementation and increase the chances of widespread acceptance of the system across the entire user community.

- Text manipulation software that is easy to use

Just as it is important for any system to be easily understood any text manipulation software should be easy to use. Many successful systems which are currently in use make full use of standard software packages (e.g. Microsoft Word) which most users use in their day to day

<sup>68</sup>

2007

<http://circa.europa.eu/irc/opoce/ojf/info/data/prod/data/pdf/Helsinki2007-dataCapturingAtTheSource-final.pdf>

work. If they become involved in the legislative process it is therefore easy for them to understand systems which are based closely on what they use already. From a cost and systems support point of view it also means that generally there are fewer software packages which have to be installed and maintained and the cost of implementation will be reduced.

- Editors who can produce simple and difficult texts reliably in content and layout and which can be validated automatically.

Ideally any system should enable users to produce both simple and difficult texts which can be produced reliably each time and which the user can see to be correct. This can be achieved by using standard templates which have been designed for handling the particular text elements and which also allow for wysiwyg presentation of the draft as it will appear when published in the Official Gazettes.

- It must allow both simple and complicated text with the same instruments to be worked upon (text, tables, graphics, pictures etc)

Although most legislation will be based on the use of text it is not unusual for complex formulae, tables, graphics and pictures to be incorporated into the drafts. This again necessitates the use of systems which are based on use of software which incorporate particular tools (e.g. Equation Editor within Microsoft Word) or can readily incorporate images and pictures (e.g. in tif and other formats), and spreadsheets produced in compatible systems (e.g. Microsoft Excel).

Again, where other data elements are being incorporated into the texts it is helpful for the user to be able to view a wysiwyg presentation of the draft as it will appear when published in the Official Gazettes.

- A general, well known data format (e.g. Microsoft Word), that should be used in text processing systems by Parliament, Government officials and anybody who is to be consulted on the draft legislation.

As for ensuring that any text manipulation software is easy to use it is also sensible to have systems based which use well known data formats which allows for the exchange of drafts with those who may not be directly involved in the legislative process but may nevertheless need to see and comment on drafts. Use of formats which may be in everyday use thus aids this process.

- The method of operation and instruments to be used should be widely available ensuring that all people involved in the process can work with the same working technique and the same instruments.

- It must bring the same results both online and offline

- It must support structural elements within which the working documents can be turned into publications automatically (e.g. automatic conversion into XML)

The benefits of holding legislative texts in XML formats has already been well documented in the report from the Working Group on the Use of XML for the production and distribution of the official gazettes which delivered its final report at the Ljubljana meeting. Any system which is implemented should enable legislative texts to be converted automatically to XML. This will mean that the data which is captured at the source should be formatted taking account of the XML Schema.

It must ensure the flow of work in two directions, without the loss of content and metadata. Particularly for graphics and tables (e.g. documents in XML back into doc)

- It may need to support more than one language so that the integrity of multilingual text issues can be guaranteed (content, structure and meaning)
- It must be capable of delivering outputs which end users can be secure in knowledge are authentic reproductions of the laws which have been passed by Parliament or adopted by Ministers.
- It should enable full version control of the drafts

### 8.3 FINAL REPORT OF THE WORKING GROUP ON XML - EUROPEAN FORUM OF OFFICIAL GAZETTES<sup>69</sup>

#### Summary:

The discussion in the working group showed that there are at least three solutions to the editor question:

- a specific editor (such as the NREditor),
- a generic XML editor such as XMetal which was used in the Swedish project and which can be configured in a way that all tags are hidden for the user, or
- a generic word processor (e. g. Microsoft Word) with additional processing into XML such as LegisWrite from the EU.

The discussion showed that there were various arguments for and against the different approaches, and therefore there could not be drawn any specific conclusions. Some of the arguments are:

“a specific editor will only allow valid XML with no use for an *a posteriori* validation to the annoyance of the drafter”

“it is easier to do additional processing with the legislative texts than introducing a new editor to the users”

“the cost to develop and support a specific editor is too high”

“cost of software license for a generic XML editor is too high, if the editor should be used by all drafters”

---

<sup>69</sup> 2006 - Use of XML for the production and distribution of the official gazettes.  
<http://circa.europa.eu/irc/opoce/ojf/info/data/prod/data/pdf/Ljubljana2006-useOfXml.pdf>



“a specific editor gives the possibilities to aid the drafter with legal techniques at a much higher level”

## 9 ANNEXE 2: THE AIM OF OPEN LEGISLATIVE DOCUMENT STANDARDS<sup>70</sup>

The aim of an open document standard is to define a set of simple, technology-neutral representations of Parliamentary documents for e-Parliament services and provide an enabling framework for the effective exchange of “machine readable” Parliamentary documents such as legislation, debate record, minutes, etc.

One of the main objectives of a standard for legislative documents is to be able to capture and describe similarities so as to unify and streamline, whenever possible and as far as possible, the processes and formats and tools related to Parliamentary documentation. This lends itself to reducing investments in tools and systems, helping open access, and enhancing cooperation and integration of governmental bodies both within the individual countries and between them. A standard would define a model for open access focused on the following issues:

- generation of documents: it should be possible to use the same tools for creating the documents, regardless of the type, country, language, and generation process of the document.
- presentation of documents: it should be possible to use the same tools to show on screen and print on paper all documents, regardless of their type, country, language and generation process.
- accessibility of documents: it should be possible to reference and access documents across types, languages, countries, etc., implementing the network of explicit references among texts into a web of hypertext links that allow the reader to navigate easily and immediately across them.
- description of documents: it should be possible to describe all documents, regardless of their types, languages, countries, etc., so as to make it possible to create repositories, search engines, analysis tools, comparison tools, etc.

Although each Parliament has its unique characteristics, all Parliamentary democracies have a number of characteristics in common: Actors, Structures, Procedures, Acts and Information. A good standard defines common building blocks in a single model that can be applied to each (or at least most) Parliamentary documents.

A good standard defines a set of recommendations and guidelines for e-Parliament services in an international context. The framework will be an essential prerequisite for interlinking and web-enabling Parliaments. It will address information content and recommend technical policies and specifications for connecting Parliament information systems across countries. It needs to be based on open standards.

---

<sup>70</sup> From: Global Centre for ICT in Parliament Working Paper No. 2 - January 2008 - Legal Informatics and Management of Legislative Documents - Chapter 4 Fabio Vitali (Department of Computing, University of Bologna) <http://www.ictparliament.org/>

Country Parliaments should use the guidance provided to supplement their national e-Government Interoperability Frameworks with an international dimension and thus enable international interoperability of Parliaments. So a good standard is meant to supplement, rather than replace, national interoperability guidelines that may exist by adding the interoperability dimension. Any such initiative will enable open access by focusing on both semantic and technical 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 applications, systems and interfaces for e-Parliamentary services are based on a shared core of technologies, languages and technical assumptions easing data interchange, data access and reuse of acquired competencies and tools. A good standard ensures technical interoperability by enforcing the use of open standards and open document formats, such as those based on the XML language, whose specifications are a worldwide standard and for which numerous tools and applications have been developed and are widely available.

There are three aspects to any Parliamentary document:

- Presentation - how the information looks e.g. the colour of the text used in the document, the headings and other such formatting issues;
- Structure - how the information is organized;
- Semantics - what the information represents or means.

The development of descriptive mark-up meta-languages such as XML allows adding information to any document that would make both the structure and the semantic of a document “readable” by a computer.

- Semantic mark-up – semantically identifies parts of the document (e.g., headings, names, references, provisions. In this way the “meaning” of the different parts can then be “understood” by machines as well, in the sense that a machine will be able to distinguish such parts and consequently to process them accordingly.
- Structural mark-up – this refers to the categorization of different parts of a document based on their functionality e.g. In a Parliamentary document you may want to indicate that a certain section of the document is the Preamble, Question, Motions etc...

## 10 ANNEXE 3: ADVANTAGES TO THE USE OF OPEN STANDARDS IN PARLIAMENTS

World eParliament Report 2010<sup>71</sup>.

Standards for documents – especially open standards for tagging the elements of records so that they can be interpreted properly by computers for editing, rendering, searching, exchanging, and preserving – are vital.

As outlined in the 2008 edition of this Report, there are a number of important advantages to the use of open standards in parliaments:

- Exchange of documents. Open standards make it easier to share documents between individuals and organizations, even if they use different software for editing and managing documents. They can facilitate such exchanges between departments within a parliament, with another chamber, between the parliament and the government, with citizens and civil society organizations, and with legislative bodies and organizations in other countries.
- Search. Search engines can provide more accurate results and users can formulate more precise queries if data is tagged for its specific content. Open standards permit documents to be indexed with a variety of search engines, thereby giving legislatures choices in the selection of a technology.
- Linking among documents. Legislative documents are highly interrelated. Open standards allow links among documents to be created automatically and even have the potential, depending on the depth of tagging, to support linking between elements within documents. For example, a section of a proposed bill could be automatically linked to the portion of an existing law that it would amend.
- Multiple forms of output. A source document tagged with an open standard could be rendered into different appearances such as for an online website, a paper copy, or a version modified to be incorporated into another document. XML can also be used to produce versions which could be easier for persons with disabilities to access by supporting, for example, large type fonts or audio output.
- Consistency in formatting. Tagging standards can be used to encourage or even enforce proper formatting so that members and others who prepare the texts do not have to know the exact conventions used when they draft bills or amendments.
- Ease of preparation. Open standards can be demanding to use but once understood they can ease the effort required to prepare a bill or amendment by guiding the drafter through the required formatting steps.
- Preservation. One of the most important uses of open standards is to ensure the long-term preservation of documents. Proprietary systems change constantly in response to market pressures for new capabilities. As these systems are enhanced, they often reach a point where they cannot be used to access documents prepared using older versions of the same

---

<sup>71</sup> <http://www.ictparliament.org/wepr2010>

software. Over time this has the potential for making it difficult, if not impossible, to access digital documents.

- Access for citizens. The problem of long-term preservation becomes most acute in the context of ensuring permanent access for citizens to legislative documents. Documents in digital formats that are accessible today may become inaccessible over time because previous media, software, and proprietary formats are no longer supported. And this could prevent public institutions from guaranteeing that archived public records in digital formats will remain accessible in the future.