xmLegesEditor: an OpenSource Visual XML Editor for supporting Legal National Standards

Tommaso Agnoloni, Enrico Francesconi, Pierluigi Spinosa
`{agnoloni,francesconi,spinosa}@ittig.cnr.it`
Institute of Legal Information Theory and Techniques (ITTIG-CNR)
Italian National Research Council http://www.ittig.cnr.it

Abstract. The NormeinRete (NIR) project aims at providing improved accessibility to normative documents. To this end XML and URN standards have been estabilished for norms representation and identification. In this paper xmLegesEditor, an Open Source visual XML editor providing a unified access to software tools for supporting the adoption of NIR standards, is presented. Thanks to its modularity and flexibility xmLegesEditor can be reused as a developing platform for supporting any XML standard and in particular other Legislative XML standards.

Keywords: Legislative Standard, XML Editor, OpenSource

1. Background

In recent years a standardization process of legislative documentation, stimulated by the migration of legislative data collections on the web, has been started both in the National and European environment. In Italy, the NormeinRete (NIR) project started in 1999, proposed by the Italian Ministry of Justice, with the aim of building a distributed cooperative system to access juridical documentation. The main goal was to improve accessibility to laws, by offering unified access to Italian and European Union legal material published on different institutional web sites through a specialised portal. The project has achieved the following results:

– a site for providing a unique access point for searching the Italian legislative corpus. The site (www.normeinrete.it) offers search and retrieval services operating on all Italian laws since 1904 and utilities for automated hyperlinking. The system is based on a federation of legislative data bases developed on different platforms and built on the basis of a co-operative technological architecture;

– an XML standard for norms representation. DTDs (Document Type Definition) for Italian legislation have been defined, able to represent metadata and all the significant information useful to automate legislative documents life-cycle management;

– a standard for norm persistent identification, based on an identifier derived from URN (Uniform Resource Name) and a resolution system able to resolve logical identifiers into physical addresses.

See (Lupo, 2005). The XML NormeinRete standard is today widespread and used by main normative document producers. Standardization allows to design cooperative systems involving different institutions achieving interoperability mainly through documents format compatibility.

In this context, a number of software tools for supporting the adoption of such standards, have been developed or are under development with the aim to cover the whole law life-cycle from drafting to Official Journal publication.

2. Introduction

xmLegesEditor is a specific integrated Legislative drafting environment developed at ITTIG/CNR for supporting the adoption of Italian Legislative National XML Standards (NIR). It is distributed with an Open Source license.

By "specific" it is meant that, besides integrating various modules for the treatment and processing of normative texts, it has been developed from scratch as a *native* XML editor oriented to Legislative Drafting and not as an adaptation of general purpose word-processors or generic XML Editor.

The peculiarity of this approach is a fundamental one in order to appreciate the features offered by xmLegesEditor.

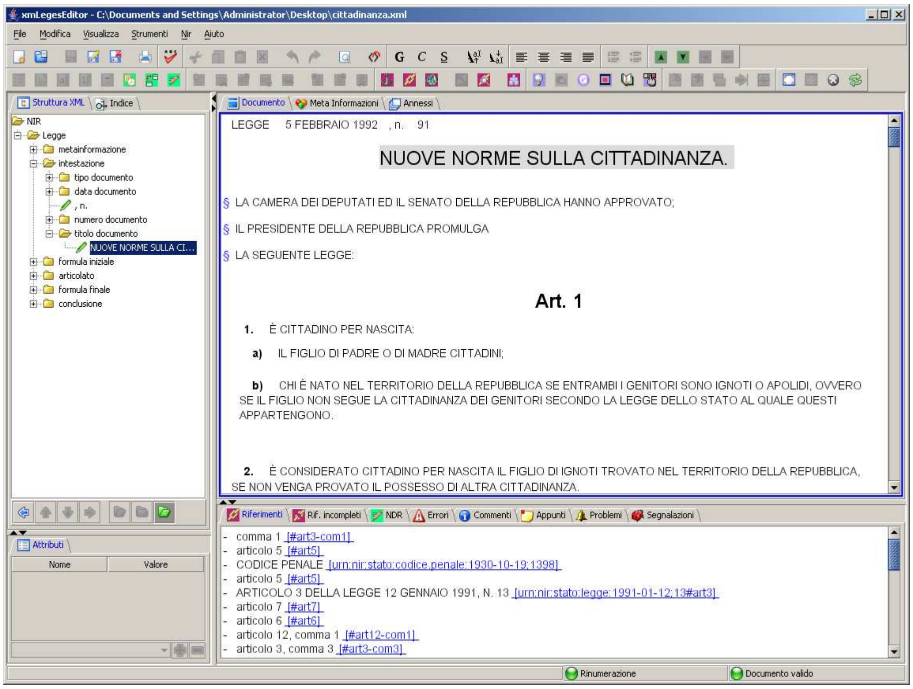
This approach has emerged from an analysis of existing widespread general purpose editors which pointed out that none of them offers the possibility of exploiting the XML underlying document structure in a user-friendly way. In fact, typical WYSIWYG word-processors, though with the strength of being widely spread and used, are mainly oriented to texts' *style markup* than to their *structural and semantic markup.* A mapping between the latter and the software tools offered for the former has appeared as a weak and restrictive approach with respect to the development effort needed for such adaptation. On the other hand, generic XML editors are almost exclusively oriented to technical users with a strong background and awareness of XML underlying theory, which turns out to be useless in an office productivity environment in which different skills are assumed. See (Saqib, 2005) for an overview.

The effort made with the development of xmLegesEditor has been to estabilish a trade-off between a user-friendly approach to text authoring hiding the underlying XML structure, and the maximum flexibility and extensibility in the exploitation of the high potentiality of content expression offered by XML documents and grasped by Legislative XML standards. In a way an effort has been made towards the overcoming of traditional distinction between WYSIWYG and not editors, which is underlying in the XML approach where content and presentation are kept well distincted.

Moreover, a crucial point in the adoption of an XML standard and therefore with the choice of an editor handling related documents, is the issue of document validity in an XML-sense. The definition of a strict Document Type Definition (or XML Schema) is a powerful tool for the standardization of the formal structure of documents and its mandatory respect is a key tool in order to guarantee interoperability among all the software agents involved in document lifecycle.

However its enforcement and the respect of such rules during documents production is a non-trivial task for typical editors. On the other hand, failure in validation or manual management of XML tags can, particularly for complex standards as the Legislative ones, cause a complete failure of the original objectives of standard compliant document production.

xmLegesEditor proposes an original approach to this problem in which, while guaranteed to be safe and successful for XML validation (and not only well-formedness) is completely "transparent" to the user and does not require any additional effort or XML awareness than typical word processing operation. Such approach can be indicated as an *a priori validation* approach and will be more deeply described in following sections. The basic idea 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.
Finally, the specific domain of Legal Drafting and Normative Documents Management calls for the integration of a variety of additional tools in order to setup a complete Integrated Normative Drafting Environment which are most easily integrated in a specifically developed platform.



LEGGE 5 FEBBRAIO 1992 , n. 91

NUOVE NORME SULLA CITTADINANZA.

§ LA CAMERA DEI DEPUTATI ED IL SENATO DELLA REPUBBLICA HANNO APPROVATO;

§ IL PRESIDENTE DELLA REPUBBLICA PROMULGA

§ LA SEGUENTE LEGGE:

Art. 1

1. È CITTADINO PER NASCITA:

a) IL FIGLIO DI PADRE O DI MADRE CITTADINI;

b) CHI È NATO NEL TERRITORIO DELLA REPUBBLICA SE ENTRAMBI I GENITORI SONO IGNOTI O APOLIDI, OVVERO SE IL FIGLIO NON SEGUE LA CITTADINANZA DEI GENITORI SECONDO LA LEGGE DELLO STATO AL QUALE QUESTI APPARTENGONO.

2. È CONSIDERATO CITTADINO PER NASCITA IL FIGLIO DI IGNOTI TROVATO NEL TERRITORIO DELLA REPUBBLICA, SE NON VENGA PROVATO IL POSSESSO DI ALTRA CITTADINANZA.

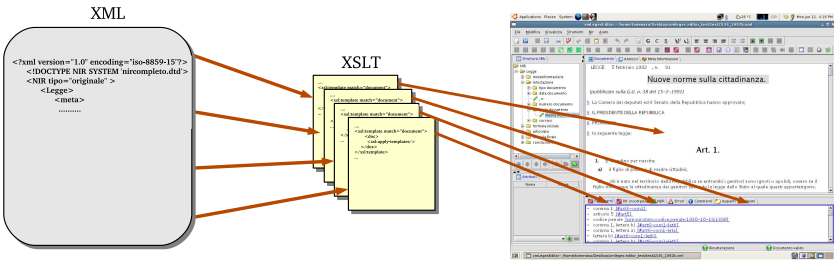
- comma 1 [#art3-com1]
- articolo 5 [#art5]
- CODICE PENALE [urn:nir:stato:codice.penale:1930-10-19;1398]
- articolo 5 [#art5]
- ARTICOLO 3 DELLA LEGGE 12 GENNAIO 1991, N. 13 [urn:nir:stato:legge:1991-01-12;13#art3]
- articolo 7 [#art7]
- articolo 6 [#art6]
- articolo 12, comma 1 [#art12-com1]
- articolo 3, comma 3 [#art3-com3]

Figure 1. A screenshot of xmLegesEditor

3. xmLegesEditor

3.1. Description

In Fig. 1 a screenshot of xmLegesEditor is depicted. The user interface is the typical one with common functions offered by traditional word processors. However notice how 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. Migrating XML approach to the editor, the idea is in fact to view and manage the same content, stored and accessed directly in XML, through different views or tabs. Each view is simply an XSLT stylesheet file applied to the content, see Fig. 2. At runtime each XML document can be associated with multiple views or tabs. 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 obtainable by applying a stylesheet can easily become an interactive panel (not read-only) of xmLegesEditor,



```
<?xml version="1.0" encoding="iso-8859-15"?>
    <!DOCTYPE NIR SYSTEM 'nircompleto.dtd'>
    <NIR tipo="originale" >
        <Legge>
            <meta>
            ..........
```

```
....
<xsl:template match="document">
        <doc>
            <xsl:apply-templates/>
        </dox>
</xsl:template>
...
```

Figure 2. xmLegesEditor panels obtained by applying XSLT stylesheets to the XML document.

synchronized with all the others, and all the data inserted from each view are directly stored in the unique shared XML file.
In xmLegesEditor, besides full text view the following additional views are integrated:

Tree Panel offers a structural view of the document reproducing its tree structure. It can be browsed as a document index to quickly access document's partitions or edit partition's properties.

Attributes Panel shows the attribute contents of XML elements selected from any panel; allows editing of attributes, association of specific editors to specific data types (dates, URNs, etc.) removing/adding non mandatory/allowed attributes querying the DTD.

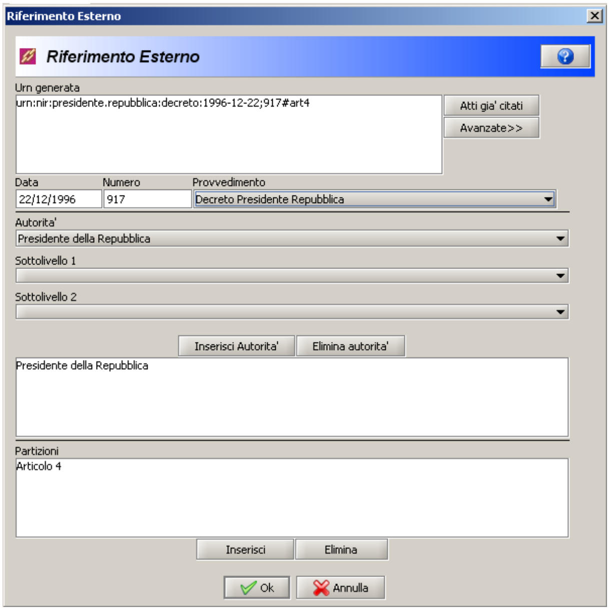
Reference Panel presents an index of all the normative references in the document and related URN. Clicking on the URN, web access to the resolutor and to the cited document is provided.

Notes Panel summarizing view of the notes to the document.

Annexes Panel a separated full-text view of the annexes to the main document.

General Metadata Panel summarizing view of the main document's geneal metadata.

each view can also serve as a quick access point to the corresponding content in the full text view. As all this panels are simply provided by XSLT stylesheet application to the XML document, they can be customized and parametrized in the most various stylistic and structural ways exploiting the features of XSLT standard.


Riferimento Esterno

Urn generata
urn:nir:presidente.repubblica:decreto:1996-12-22;917#art4

Data	Numero	Provvedimento
22/12/1996	917	Decreto Presidente Repubblica

Autorita': Presidente della Repubblica

Sottolivello 1:

Sottolivello 2:

Presidente della Repubblica

Partizioni: Articolo 4

Figure 3. Guided insertion of a reference in the URN format.

3.2. XML elements management

On the top of the screen in Fig. 1 , see the specific toolbars for the access to the function for the management of NIR Elements. Buttons are grouped in toolbars by function *i.e.* red toolbar containing functions for managing normative references, green toolbar for accessing functions for managing partition insertion etc.

Each button accesses specifically developed complex function for the management of specific XML Elements and related attributes, possibly through forms guiding the user in the insertion of related informations. See for example Fig. 3 depicting the form guiding the user in the insertion of a normative reference according to the URN standard. XML Elements management is approached in an *incremental* way, from a rough mode through contextual right-click operations to evolved functions accessed by toolbars. This allowed us to cover the management of all NIR-Elements from the beginning and improve it during software development.

A fundamental feature of xmLegesEditor is the management of buttons enabling. In fact, as introduced in previous sections, in order to ensure *a priori validation* of the document, only the buttons of functions that can be applied at the current cursor position are enabled. In the same way, insertion of XML Elements from right click is managed in such a way that only the tags whose insertion preserves document validity are proposed to the user. Moreover, such information is not cabled in the program but directly accessed from the DTD files, this meaning that a change in the DTD or adoption of a different DTD directly affects such feature without additional coding. More details on the modules responsible of such feature will be given in subsequent sections. What is important at this stage is that xmLegesEditor implicitly guides the user to the production of a valid XML document according to the related DTD without additional effort than usual typewriting.

3.3. External modules integration: xmLeges Suite

As pointed out in previous sections, xmLegesEditor aims to be a complete drafting environment providing facilities for handling both legacy content and drafting of new documents.

To this end, a number of independent software modules, developed for managing both formal profile (XML structure and reference recognition) and functional profile (semantic information extraction and annotation) (Biagioli et al., 2005) have been integrated into the editor in order to provide the user with a unique point of access to different tools. Notice that, external modules integration is made particularly easy by the software modules interoperability naturally offered by XML, along with the fact that xmLegesEditor natively supports standard XML format. It is therefore enough that each external module, provides an XML-NIR output for it to be connected to the editor.
For the moment, four different external modules related to the xmLeges project have been integrated:

xmLegesLinker extracts references in normative texts and describes them according to the corresponding URNs. The URN-NIR standard estabilished a grammar to identify documents within the NIR domain. This grammar has been defined according to (Moats, 1997) andis able to generate URNs using information on: the enacting authority; the type of measure, a number of details as: date of issue, different later versions of the document; the annexes (see (Spinosa, 1997), (Biagioli et al., 2003) for details). A normative text may contain a lot of cross references to other measures that have to be described using the related URN, so that references can be transformed in effective links when documents are published on the Web. Especially in the phase of legacy content conversion, the manual construction of a URN for each reference can be a time-consuming work. For this reason xmLegesLinker, a module able to automatically detect cross-references and assigning them the related URN has been developed. The parser is generated using LEX and YACC technologies , (Lesk, 1975), (Johnson, 1975) on the basis of the vocabulary of the citations and the URN grammar expressed in EBNF.

xmLegesMarker is a structural parser able to transform a legacy normative document in plain text, HTML or doc format into XML-NIR 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. For the header and the footer a different strategy was adopted, since their partitions are not usually identified by particular typographical symbols. 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 these hidden states. For this reason, to parse these two sections we adopted a strategy based on Hidden Markov Models (HMMs).

xmLegesClassifier as regards the automatic detection of the semantics in a normative document, xmLegesClassifier is designed to automatically classify paragraphs into provision types according to the provisions model defined in NIR standard. Two machine learning approaches to document classification have been tested: *Naïve Bayes* and *Multiclass Support Vector Machines* (Biagioli et al., 2005)

xmLegesExtractor designed to automatically detect the arguments of a provision. Knowing the provision type detected by *xmLegesClassifier*, this module uses the provision specific grammar to extract the provision arguments using NLP techniques. Basically the purpose is to select text fragments corresponding to specific semantic roles that are relevent to the different types of provisions. It is realized as a suite of Natural Language Processing tools for the automatic analysis of Italian texts, specialized to cope with the specific stylistic convention of legal parlance (Bartolini et al., 2005).

3.4. Software Architecture

xmLegesEditor has been developed on the experience of prototype NirEditor (Biagioli et al., 2005). In this new version a complete reingeneering

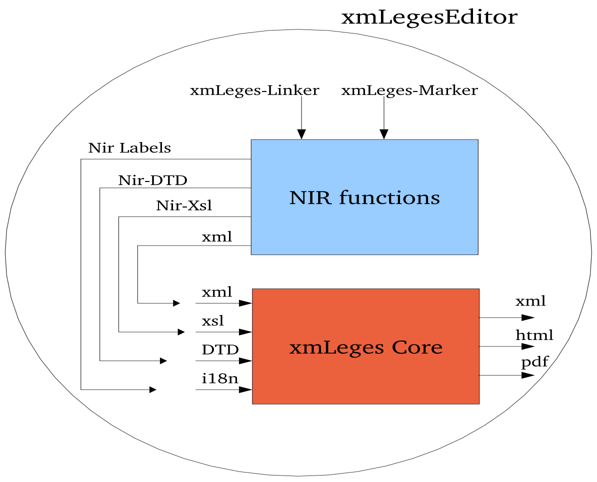


Figure 4. xmLegesEditor Architecture.

of the software has been made. The aim was to guarantee maximum extensibility, modularity and reusability of the different software modules. To this end xmLegesEditor has been structured on a component-based architecture where each component provides specific services. Moreover a strict separation between generic XML modules and specific NIR modules has been followed. This, as a by-product, gave the significant result that 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 simply composing the different services specified in an application composition 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. In Fig. 4 such architecture is schematized. xmLegesEditor can be seen as the overlapping of two layers:

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

- ***Validity Management:*** This is accomplished through the access to the *rulesManager* service. As reminded in previous sections, a central feature in xmLegesEditor is that only valid operations are allowed: this is obtained by contextually querying the rulesManager which provides methods to access the DTD 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 operate over the document. Changing the DTD automatically affects rulesManager answers. This was originally needed in order to follow the variations of a standard under evolution without continuosly changing the code, but is of course even more useful to provide such Validity Management strategy to different standards. It is planned to develop a similar component providing services for querying XML-Schema in order to extend xmLegesCore support to standards based on such format.

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

- ***Document Visualization:*** As reminded earlier the strategy in document visualization consists in the visualization in the various different panels of an HTML document obtained by applying an XSLT stylesheet to the XML Document. However, 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 taken from localization files accessed through a unique key which is what is actually written in the source code. 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.

each implemented in separated and standard independent components.

xmLegesNIR At the top level xmLegesNIR provides parameters to the Core level in order to specialize it to the NIR standards. Moreover, a number of NIR specific components have been developed to provide enhanced functionalities to the support of NIR-Elements. At this level external modules integration is also provided.

Notice that thanks to such architecture most of the functionalities of validation, visualization, basic elements management can then be easily shared for reuse in developing editors for other standards.

4. OpenSource Project

A fundamental feature of xmLegesEditor is that it is a free resource, distributed with an Open-Source license.

Besides a number of good motivations for being like that, especially for software in public administrations (Cospa, 2006), this is a natural choice for implementing tools supporting an *open standard* like the NIR one, shared by a number of subjects with different specific needings.

The NIR standard interests the Italian Parliament, Courts, Public Authorities, Regional Assemblies, at different levels and with different specific needings. The idea is to offer a shared highly customizable platform on which to develop specific functions and easily integrate existing or new designed tools as external modules.

Moreover xmLegesEditor is entirely written in *Java*, a platform independent language, thus leaving freedom in the choice of the platform on which to run it, included free and opensource operating systems like *GNU/Linux* which use is increasingly encouraged in Public Administrations. 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 are managed by open, *W3C* standards compliant, and commercial free, software tools. Being xmLegesEditor a *native* XML editor, this means that no conversion to or from proprietary formats is ever made, thus preventing any vendor lock-in possibility and providing the added value of complete transparency in data management which is nonnegligble especially in the context of *public data* management.

In this view the two layers architecture described in previous section should encourage contributions (Sun, 2006) 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 at least, but not only, among free resources. (Saqib, 2005)

xmLegesEditor is currently available for free download in its release candidate version as a binary distribution at *www.ittig.cnr.it/xmleges*, the source code repositories are currently online and a project homepage for accessing source code, documentation both for users and developers, collaborative resources like mailing lists, wiki, bug tracking system, is in preparation.

Finally, xmLegesEditor for its described features could be considered as an initial contribution for developing a shared platform over which specialize different National Legislative Editors and eventually for supporting the forthcoming European Standard of which National Standards should become specializations following the same modularity principle which inspired xmLegesEditor development.

5. Conclusions

In this paper xmLegesEditor, an Open-Source Visual XML Editor for supporting NIR standards has been proposed. xmLegesEditor can be seen as an integrated legislative drafting environment integrating a variety of software tools for supporting the adoption of Legislative standards. xmLegesEditor differs from usual general purpose word processors as well as from generic XML editors as it has been specifically designed for supporting descriptive XML documents in a user friendly way while exploting all the features offered by XML standards, with the aim of bringing non-expert users closer to XML.
Thanks to its architecture xmLegesEditor is highly customizable and extensible to support additional views and modules for improving and assisting legal drafting. Moreover the core level xmLegesCore, provides full functionalities for implementing a general purpose Visual XML Editor. As an Open-Source project the goal is to share this resource in the XML community as well as in the IT in Law community to improve it and put it at disposal of interested parties.

xmLegesEditor is currently under experimentation at Italian Senate with the aim to extend its functionalites to the support of bills, from preparatory works to amendaments management. It is also in use in Regional projects to provide mark-up of *consolidated* legislative texts (normative texts carrying all the modifications that have been introduced over time by other normative texts) to provide free access to the in-force version of laws in specific domains.

References

Bartolini, R., Lenci, A., Montemagni, S., Pirrelli, V., Soria, C. Automatic classification and analysis of provisions in italian legal texts: a case study. In *Proceedings of the Second International Workshop on Regulatory Ontologies.* 2004

Biagioli, C., Francesconi, E., Spinosa, P., Taddei, M. The nir project: Standards and tools for legislative drafting and legal document web publication. In *Proceedings of ICAIL Workshop on e-Government: Modelling Norms and Concepts as Key Issues*, 2003, pp. 69-78

Biagioli, C., Francesconi, E., Spinosa, P., Taddei, M. Legislative drafting support tool based on XML standards. In *Proceedings of DEXA 2005*, Copenhagen, Denmark, 22-26 August 2005

Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., Soria, C. Automatic semantics extraction in law documents. In *Proceedings of International Conference on Artificial Intelligence and Law*, 2005, pp. 133-139

COSPA - Consortium for Open Source in Public Administration. *http://www.cospa-project.org* retrieved on Dec. 18 2006

Johnson, S. *Yacc - yet another compiler compiler.* Technical Report CSTR 32, Bell Laboratories, Murray Hill, N.J., 1975

Lesk, M. *Lex - a lexical analyzer generator.* Technical Report CSTR 39, Bell Laboratories, Murray Hill, N.J., 1975

Lupo, C. Beyond NormeinRete. In *Proceedings of the 3rd Legislative XML Workshop*, 2005

Moats, R., K.R.S *Urn syntax.* Technical Report RFC 2141, Internet Engineering Task Force (IETF), 1997

Saqib, A. *XML: WYSIWYG to WYSIWYM A brief look at XML document authoring.* Free Software Magazine Issue 3, April 2005 *http://www.freesoftwaremagazine.com/articles/practical_ applications_ xml* retrieved on Dec. 18 2006

Spinosa, P. Identification of legal documents through urns (uniform resource names). In *Proceedings of the EuroWeb 2001, The Web in Public Administration*, 1997

Sun Microsystems Free and Open Source Licensing White Paper. *www.sun.com/software/opensource/whitepapers/free_ open_ licensing.pdf*, 2006 retrieved on Dec. 18 2006

Megale, F., Vitali F. I dtd dei documenti di norme in rete. In *Informatica e Diritto* ***1***, 1990, pp. 167-231