Legislative drafting support tool based on XML standards

C. Biagioli, E. Francesconi, P. Spinosa, M. Taddei

ITTIG - Institute of Legal Information Theory and Technologies
Via Panciatichi 56/16- 50127 Florence - Italy
{biagioli, francesconi, spinosa, m.taddei}@ittig.cnr.it
WWW home page: http://www.ittig.cnr.it

Abstract. The NIR project aims at making the retrieval and navigation among normative documents in a distributed environment easier. XML and URN standards have been established, as well as tools have been developed for the adoption of the established standards. In this paper a specific law drafting environment, NIREditor, is presented. It is able to produce normative documents according to the NIR standards. Particularly, it is able to handle the formal structure and the semantics of a law according to legislative drafting rules. Specific tools able to automatically transform legacy contents according to NIR standards and to detect the semantics have been developed. Moreover, a module able to plan a new bill from a conceptual point of view is proposed.

1 Introduction

In the last few years a standardization process of legislative documentation, promoted by the Internet migration of legislative data collections, has been started both in a national and in the European environment. This fact encourages the development of legal information systems with characteristics of interoperability and effective of use, as well as the definition of standards to build up legal documents access facilities for both citizens and legal experts. In Italy the "Norme in Rete" (NIR) project initiative[1] aims at creating a unique access point on the Web with search and retrieval services of normative documents, as well as a mechanism of stable cross-references able to guide users towards relevant sites of public authorities participating in the project. To achieve these purposes, the NIR project adopted XML and URN standards to represent and identify normative documents [1]. In order to make the adoption of such standards easier, a number of tools have been developed. The main one is a legislative drafting tool (*NIREditor* [2]) and modules which aim at managing new or legacy documents according to the established standards.

This paper is organized as follows: in Section 2 the standards established by the NIR project are introduced; in Section 3 the semantics of normative

[1] supported by CNIPA (Italian Nationa Center for Information Technology in the Public Administration) and Italian Ministry of Justice

documents within NIR is discussed; in Section 4 the main features of NIREditor are presented. As future development, in Section 5 the project of a module able to plan a new bill from a conceptual point of view is proposed. Finally, in Section 6 some conclusions are discussed.

2 The NIR standards

Two specific national NIR work groups produced two main official standards for normative documents:

– a standard for identifying documents defined by the uniform name (URN) technique: an unambiguous identifier, that allows the references to be expressed in a stable way, independently of document physical location [1];

– an XML standard to describe documents, defined by three DTDs (NIR-DTDs) of increasing degree of complexity in text hierarchy description [3].

Basically NIR-DTDs allow normative documents to be described using two main kinds of elements: Structural elements and Metadata. Structural elements can be divided into: *generic document elements* (references to other laws, formatted text-embedded relevant entities (tables, lists, etc.)) and *specific law text elements* (heading, sections, articles, paragraphs, etc.). Structural elements describe the form of a law text (*formal profile*).

Similarly NIR-XML standards consider two kinds of metadata: *General metadata* (subject classification, publication date; relationships among acts) and *Analytical metadata* [4] (*Provision types* (Amendments (Insertion, Abrogation, Substitution) and Rules (Obligation, Definition, Penalties, etc.)) and *Arguments* (for example the *addressee* of an Obligation)). General metadata provide general information on the act, as well as analytical metadata describe the semantics of provisions (*functional profile*).

The detection of the functional profile of a law text consists in describing the provisions it contains through a model [4] made of provision types (*regulative profile*) and their arguments (*thematic profile*). The first one reflects the lawmaker directions, the second one the peculiarities of the regulated field. Using the analytical metadata, a fragment of a law text can be qualified according to the model of provisions. For example, the following fragment of the Italian privacy law:

> *"A controller intending to process personal data falling within the scope of application of this Act shall have to notify the "Garante" . . . "*

besides being considered as a part of the physical structure of a law text (a *paragraph*), can also be viewed as a component of the logical structure of it (a *type of provision*). In particular, it can be qualified as a *provision* of type *obligation*, whose arguments are:

Addressee: "Controller";
Action: "Notification";
Third-party: "Garante".

The formal profile represents the traditional habit of organizing law texts in chapters, articles, paragraphs, etc.; on the other hand the functional profile is related to how the semantics of the text is organized. The functional profile is traditionally described by the legislator assigning titles to formal partitions: partition titles are nothing but *ante-litteram* metadata, therefore analytical provisions basically are a formalized version of these titles and their NIR-XML version is the way how they are adopted within the NIR project.

3 The semantics of provisions

The provision model describes the *functional profile* of a normative document; the model [4] has been adopted as NIR standards. As introduced in Section 2, provision types are divided into two main groups: Amendments and Rules.

Amendments can be *content amendments* which modify literally the provision content or, regarding the meaning, without literal changes; *temporal amendments* which modify the times of a provision (come-into-force and efficacy time); *extension amendments* which extend or reduce the cases on which the provision has effects.

Rules are provisions which aim at regulating the reality considered by the including act. Adopting a typical law theory distinction, they consist in *constitutive rules*, which are mainly rules on entities of the regulated reality (they consist basically in those ones introducing entities and those ones which assign a juridical profile to the entities ("empowering norms"), and *regulative rules* which are mainly rules on actions. They consist in those ones disciplining actions and those ones which discipline the substantial and procedural defaults ("remedies").

4 The NIREditor

The NIR-DTDs identify a wide and complex subset of documents: basically law texts and regulative acts. The production of new documents, as well as the transformation of legacy contents according to the NIR standards, can be a hard problem to face without an editing system guiding and supporting the user.

Even though programs for XML drafting already exist, the have limits whether used for a specific class of documents, especially as concerns the generality and inadequacy of their editing functions. They should be adapted to cope with a specific XML standard (es: NIR-DTDs). Other possible solutions, as adapting Microsoft Word or Open Office to adhere to a specific XML standard, suffer from the same limitations. Users are more familiar with such tools, however when such editors are used to produce documents according to a specific XML standard, they have to be personalized as well. Moreover, these solutions suffer from another limitation: they describe a document using a proprietary format, therefore the syntactic rules contained in the DTDs of the standard under consideration have to be mapped to the proprietary format. For these reasons we have decided to develop a specific environment, based on previous studies on legislative drafting [5], to handle XML-NIR documents in their native format.

As to produce HTML documents according to the HTML-DTD, specialized editors exist, similarly to help law texts drafting according to NIR-DTDs standards, a specialized visual editor (*NIREditor*) has been developed [2]: it consists of a law drafting environment supporting specific Italian legislative technique rules. Similar initiatives exist at European level, as for example MetaLex [6]. Metalex is a knowledge management system for legislative drafting: it aims at supporting users providing both content management and decision support components. With respect to MetaLex, *NIREditor* is more focused at providing facilities for legal drafting with the aim of giving users a tool able to make the adoption of NIR standards easier. It operates within the NIR-URN and DTD framework in two working situations: it is designed to process legacy documents, as well as to assist the drafting of new texts. In both these two working situations *NIREditor* is designed to handle the formal as well as the functional profile of a law text, using both manual and automatic facilities.

In Section 4.1 facilities dealing with legacy contents are discussed; in Section 4.2 the main functions dealing with the composition and the organization of new acts are described respectively. In Figure 1 the *NIREditor* drafting environment is shown.

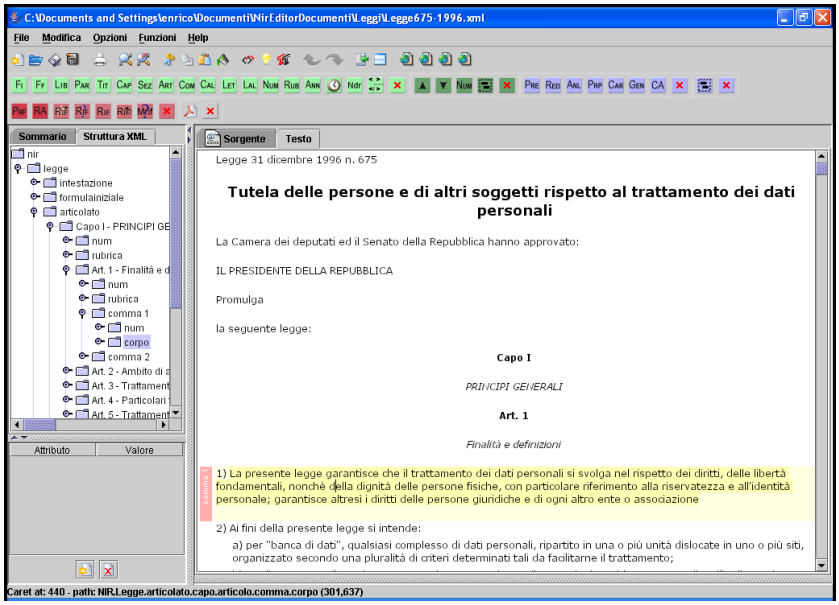

Fig. 1. The NIREditor environment

4.1 Legacy content handling

A particular attention has been addressed to design automatisms for legacy content handling, since they represent key-factors for promoting the adoption of the standards. Four modules have been implemented, to be used within *NIREditor* or as indipendent tools:
1. the *Cross-Reference Parser*, designed to automatically detect cross-references and constructing the related URNs;
2. the *Structure Parser*, designed to automate the XML-NIR conversion of legacy contents;
3. the *Provision Automatic Classifier*, which automatically classifies paragraphs into provisions according to the NIR provision model [7];
4. the *Provision Argument Extractor*, which automatically identifies the arguments of the provisions [7].

The first two modules are able to detect the formal profile of a document, producing its XML-NIR description. The last two modules are able to detect the functional profile of a document, producing its consequent XML-NIR semantic annotation. Here below a brief description of these modules is reported.

The Cross-Reference Parser. The URN-NIR standard established a grammar to identify documents within the NIR domain. This grammar has been defined according to [8] specifications and it is 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 (for a detailed discussion on URN-NIR syntax see [1], [2]). A normative text may contain lots 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. Information to build URNs are usually contained in the citation (for example the citation: "Act 24 November 1999, No. 468" generates the following URN-NIR "`urn:nir:stato:legge:1999-11-24;468`"). 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 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 [9], [10], on the basis of the vocabulary of the citations and the URN grammar expressed in EBNF (Fig. 2).

```
<NSS-nir> ::= <document> ["@" <version>]
<document> ::= <authority> ":" <measure> ":" <details> [":"<annex>]
...
```

Fig. 2. The top part of the URN grammar espressed in EBNF

Using the LEX technology a lexical analyzer is generated able to subdivide the text into *tokens* (words, numbers and punctation marks) belonging to the citation vocabulary. Then using the YACC technology a syntactical analyzer is

generated, able to detect a sequence of tokens, generated by LEX, as representing a reference and to construct the related URN.

The Structure Parser. The structure parser is able to transform a legacy normative document into an XML-NIR format. So far the expected document native formats are HTML and plain text; other proprietary formats will be considered in the next versions of this module. 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), basically probabilistc automata where states are inaccessible.

The Automatic Provision Classifier. As regards the automatic detection of the semantics in a normative document the *Provision Automatic Classifier* is designed to automatically classify paragraphs into provision types. Two machine learning approach of document classification was tested: *Naïve Bayes* and *Multiclass Support Vector Machines* [7]. Currently within *NIREditor* the two approaches can be alternatively used, but the approach based on *Multiclass Support Vector Machines* is recommended since it reported better results.

The Provision Argument Extractor. The *Provision Argument Extractor* is designed to automatically detect the arguments of a provision. Knowing the provision type detected by the *Provision Automatic Classifier*, this module uses the provision specific grammar to extract the provision arguments using NLP techniques. Basically the purpose of this module is to select text fragments corresponding to specific semantic roles that are relevant for 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 conventions of the legal parlance [11].

4.2 Composition and organization of new acts

NIREditor is conceived as a visual editor, supporting the user in producing valid documents according to the NIR-DTDs. No XML validation function is necessary within the editing environment, since NIREditor allows the user to perform only valid operations. Specific facilities are: the insertion of partitions according to the insertion point context; the automatic numbering of the divisions; the updating of internal references in the event of text movements or variations; the external and internal cross-references construction by hand or using the *Cross-Reference Parser*; the analytical metadata insertions by hand or using the *Provision Automatic Classifier* and the *Provision Argument Extractor* as a support.

It is possible to construct a new text by determining *a priori* the structure and insert the content of the various parts afterwards (*top-down composition strategy*), or else text fragments can be inserted in no particular order, then organized and inserted into a suitable structure at a later time (*bottom-up composition strategy*). Two alternative text organization strategies can be followed: the *formal organization strategy* and *the functional organization strategy* [4], [12]. The *formal organization strategy* considers the text according to the formal profile: the partitions of similar rank, to be grouped in a new partition of higher rank are chosen explicitly by the draftsman. The *functional organization strategy* considers the text according to the functional profile, where the elementary component, explicitly qualified, is a *provision*. The partitions to be grouped in a new one are chosen according to their content, affinities, etc., making queries on the analytical metadata (provision types, arguments and argument contents), as well as it is decided where they should be placed in the text, according to the preferences of the drafter and the customary procedure of presentation.

5 Planning a new act

Facilities to produce an organic and well-structured normative text are desirable. A well-structured normative text can be considered as the one where the semantic organization of the text (functional profile) follows its formal organization (formal profile) [13]. A module has been designed able to guide the drafter at planning a new organic bill. This module allows the drafter to plan a new bill from a conceptual point of view, then constructing the best structural organization of the text able to effectively communicate its semantics. The classical process of drafting (from structure to semantics) is inverted (from semantics to structure).

The planning module is conceived as a visual editor of provisions: first of all the user is required to collect terms (manually or from an ontology (ex. JurWordNet [14])) representing entities of the domain to be regulated, then in a visual panel the drafter will insert objects representing the provision types of the new bill and collected terms will be used as values for the provision arguments. In this context the use of an ontology is of primary importance: it allows to obtain a normalized form of the terms with which entities are expressed, so that they can be indexed and used in the analytical metadata querying process of normative document search and retrieval. The collection of the necessary terms represents a specific glossary of the bill under construction. At this stage the functional profile of the bill is defined and users will be provided with visual facilities, as well as tools, to express criteria (queries) to group sematically correlated provisions into formal partitions. So the formal profile is obtained and the XML skeleton of the new bill can be generated. Proposals of partitions wording can be generated on the basis of the defined functional profile [15].

6 Conclusions

In this paper a specific law drafting environment, *NIREditor*, dealing with the formal structure and the semantics of a normative document has been presented. It is developed within the NIR project and it works within the URN and XML standards established by NIR. Manual and automatic facilities dealing with legacy and new documents have been shown. Particularly four modules able to automatically recognize the formal structure and the semantics of documents, according to formal and semantics models adopted by NIR, have been presented. To provide facilities allowing the construction of new documents, a module able to help the drafter in planning a new bill starting from a semantic point of view has been proposed.

References

1. Spinosa, P.: Identification of legal documents through urns (uniform resource names). In: Proceedings of the EuroWeb 2001, The Web in Public Administration. (1997)
2. 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) 69–78
3. Megale, F., Vitali, F.: I dtd dei documenti di norme in rete. Informatica e Diritto **1** (2001) 167–231
4. Biagioli, C.: Towards a legal rules functional micro-ontology. In: Proceedings of workshop LEGONT '97. (1997)
5. Biagioli, C.: Law making environment. In: Proceedings of Workshop on Legal Knowledge and Legal Reasoning Systems, Tokyo. (1992)
6. Boer, A., Winkels, R., Hoekstra, R., van Engers, T.: Knowledge management for legislative drafting in an international setting. In: Proceedings of JURIX 2003: Legal Knowledge and Information System. (2003) 91–100
7. 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) 133–139
8. R. Moats, K.R.S.: Urn syntax. Technical Report RFC 2141, Internet Engineering Task Force (IETF) (1997)
9. Lesk, M.: Lex - a lexical analyzer generator. Technical Report CSTR 39, Bell Laboritories, Murray Hill, N.J. (1975)
10. Johnson, S.: Yacc - yet another compiler compiler. Technical Report CSTR 32, Bell Laboritories, Murray Hill, N.J. (1975)
11. 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)
12. van Kralingen, R.: Frame-based Conceptual Models of Statute Law. Kluwer Law International (1997)
13. Branting, L.K., Lester, J.: Justification structures for document reuse. In: Proceedings of the Third European Workshop on Case-Based Reasoning. (1996) 76–90

14. Gangemi, A., Sagri, M.T., Tiscornia, D.: Jur-wordnet, a source of metadata for content description in legal information. In: Proceedings of the ICAIL Workshop on Legal Ontologies & Web based legal information management. (2003)
15. Burges, C.: Towards a normalised language to clarify the structure of legal discourse. In: In Martino (ed.) Deontic Logic, Computational Linguistics and Legal Information Systems. A.A. Martino eds., Amsterdam: North Holland (1982)