

# MetaVex: Regulation Drafting meets the Semantic Web

Saskia van de Ven, Rinke Hoekstra and Radboud Winkels  
Leibniz Center for Law, Faculty of Law, University of Amsterdam  
PO Box 1030, 1000 BA, Amsterdam  
{s.vandeven,hoekstra,winkels}@uva.nl

## ABSTRACT

Currently almost all legislative bodies throughout Europe use general purpose word-processing software for the drafting of legal documents. These regular word processors do not provide specific support for legislative drafters and parliamentarians to facilitate the legislative process. Furthermore, they do not natively support metadata on regulations. This paper describes how the MetaLex regulation-drafting environment (MetaVex) aims to meet such requirements.

## Keywords

XML, RDF(S), OWL, metadata, regulation, drafting, semantic web, tool

## 1. INTRODUCTION

Legislative drafting, and designing amendments to existing or new legislation are important parts of the work done by national parliaments, regional assemblies, city councils and ministries in Europe. Currently almost all of these legislative bodies use general purpose word-processing software to create legal documents.

However, these regular word processors generally do not provide users with targeted support to facilitate the legislative process. They are often badly integrated with legacy systems that support storage, search and publishing facilities, and provide no streamlined environment for drafting and discussing legislation and other kinds of regulations. Such an environment would integrate workflow, search facilities, tools to support group dynamics (including versioning and distribution) and features that facilitate publication.

Legislative drafting is a complex process that takes place in a political and dynamic environment, which involves many stake-holders. Since a new or adapted regulation is often connected to existing laws, the drafters and other stake-holders should be aware of relationships between the law under construction and those existing legal sources. Legal drafting practice has learned that legal quality can benefit from the use of specific legal drafting patterns.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SW4Law June 8, Palo Alto, CA USA  
Copyright 2007 ACM ...\$5.00.

The SEAL project (Smart Environment for Assisting the drafting and debating of Legislation)<sup>1</sup> develops a supportive environment that enables easy construction of legal drafts using drafting patterns and creation of connections from and to existing legal sources. The infrastructure will provide access to a repository with existing laws, draft versions and amendments and will offer easy to use access methods. Collaboration between stake-holders will be supported by groupware facilities such as automated signalling functions and routing of drafts and amendments. This environment will be developed for three European parliaments. An initial working environment is foreseen in 2007. This will be tested, refined and implemented in co-operation with the parliaments and legislation drafters during the project.

The MetaLex regulation-drafting environment (MetaVex) is developed at the Leibniz Center for Law and is one of the three environments being evaluated in SEAL. For information about the other two environments, see [1] and [8]. In the following sections we identify the requirements, introduce the XML document standard underlying the system, and describe its current status.

## 2. REQUIREMENTS

MetaVex aims to streamline the legislative process by addressing the problems discussed in the introduction. In this section we describe the requirements against which the environment is evaluated. These criteria can be summarized as follows:

**Look and Feel** The editing environment should provide a look and feel similar to normal word processors. Document editing should be done in a WYSIWYG<sup>2</sup> interface; this way legal drafters can create document structure and content without knowledge of specific commands or technical notations.

**Drafting Patterns** Legislative drafters should be supported by the editor in complying to prescribed legal drafting patterns. Offering users suggestions and predefined phrases in the form of templates improves and speeds up the process of generating document structure and content.

**Referencing** The use of references to other legal sources is an important way by which drafters add structure and

<sup>1</sup>SEAL is a project in the e-Participation initiative of the European Commission. See <http://www.eu-participation.eu/seal>

<sup>2</sup>What You See Is What You Get

meaning to a document. The editor should facilitate the frequent use of these references and offer ways to validate the legal sources they cite. References should be *detailed*, i.e. point to the smallest relevant element of a regulation.

**Metadata** A way to store extra information about a document e.g. author, version, modification, should be provided. Possibilities to add information concerning document structure as well as content is regarded as an advantage.

**Version Management** The environment should offer support to manage document versions, starting from the first draft until and beyond the time at which the document is published. This allows users to always be able to identify the latest version of a document.

**Groupware** By using groupware facilities, drafters can collaborate on the same project. These facilities will not only consist of sharing comments or amending existing legislation, but will allow for elaborate authorisation and accountability management.

**Workflow** Workflow support should be an integral part of the environment to be able to divide tasks into sub-tasks, assigning them to people and keeping track of progress.

**Storage** Users should be able to store documents in a *local data repository*, providing them with advanced search mechanisms. It should also be possible to connect to a server with various types of clients over the *internet*, e.g. by using a browser.

**Publishing** The environment should allow straightforward publishing of texts in legacy formats. This allows publishing of legal drafts in an early stage, which makes it possible to interact with the public (businesses, citizens and interest groups) during the drafting process.

### 3. SYNTAX & SEMANTICS: METALEX

MetaVex is a regulation-drafting environment for MetaLex documents: texts are saved as XML documents that comply with the MetaLex [3] format for legal sources. This standard provides a generic and easily extensible framework for the XML encoding of the structure and contents of legal documents. It addresses many of the requirements introduced in the previous section, as is described in e.g. [11].

MetaLex is currently undergoing a CEN standardisation process. It is input to the CEN workshop on an Open XML interchange format for legal and legislative resources<sup>3</sup>. The MetaLex/CEN schema is based on best practices from amongst others the previous versions of the MetaLex schema, the Akoma Ntoso schema [10], and the Norme in Rete<sup>4</sup> schema. A first version of this schema was adopted as part of a CEN workshop agreement on 6 December 2006<sup>5</sup>.

A standard interchange format enables *public administrations* to link legal information from various levels of authority and different countries and languages. Moreover, the

<sup>3</sup>[http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/activity/ws\\_metalex.asp](http://www.cenorm.be/cenorm/businessdomains/businessdomains/iss/activity/ws_metalex.asp)

<sup>4</sup><http://www.nir.it>

<sup>5</sup>See <http://www.metalex.eu/wiki/>

standard will enable *companies* that are active in the field of legal knowledge systems to connect to and use legal content in their applications, which allows them to support a much larger market. An open interchange format will also protect customers of such companies from vendor lock in. Finally, the standard will help to improve transparency and accessibility of legal content for both citizens and businesses.

As the standard is primarily intended as an interchange format, a single MetaLex document can contain several versions of a text. The standard not only includes an event-based model for managing multiple versions of legal documents *through time* [4], multiple *language* versions of the same text can also be included in just one MetaLex document.

MetaLex provides extensive mechanisms to add metadata both to specific parts of a document and to the document as a whole. Every element of a legal text can be uniquely identified through a URI, and annotated with information regarding e.g. its version, publication date, validity interval, efficacy, language, jurisdiction, and authority. Furthermore, the standard introduces the possibility for marking references, both to elements of (other) regulations and to individual entities not part of a regulation, such as institutions or concepts defined by the regulation.

All metadata statements in MetaLex conform to the triple model of RDF<sup>6</sup>. This means that any MetaLex metadata can be used to generate an RDF triple: statements about entities are interpreted as subject, predicate, object triples. And conversely, because every MetaLex element has a unique identifier, it is possible to make external statements in RDF referring to any element of a legal text. The MetaLex CEN workshop furthermore adopted the RDFa<sup>7</sup> standard for embedded metadata. RDFa does not have its own namespace: the significance of XML elements and attributes to RDFa processors is determined entirely by names. An RDFa element is defined as any XML element that contains one or more RDFa attributes: **about**, **property**, **rel**, **href** or **content**. An RDFa processor generates RDF triples from RDFa elements.

This feature of MetaLex provides a strong connection to other semantic web standards as well, such as RDF Schema and OWL<sup>8</sup> as both have RDF/XML syntax. A MetaLex document can be translated into OWL by means of XSLT transformations. OWL provides additional expressive power, which can be used to describe not only the *content* or *domain* of a regulation, but also the *authority* through which a regulation is enforced, and the history and background of *modifications* of the regulation. The MetaLex CEN schema defines a general framework for describing events and actions in OWL. More information about this framework can be found at the MetaLex CEN Wiki<sup>9</sup>.

More importantly, MetaLex allows formal representations of legislation to refer to and be grounded in the documents containing the official texts. An example is LKIF, the Legal Knowledge Interchange Format [2], currently being developed in the ESTRELLA project<sup>10</sup>, a vendor neutral rep-

<sup>6</sup>The Resource Description Framework. See <http://www.w3.org/RDF/>

<sup>7</sup><http://www.w3.org/2006/07/SWD/RDFa/>

<sup>8</sup>The Web Ontology Language. See <http://www.w3.org/2004/OWL>

<sup>9</sup>See <http://www.metalex.eu/wiki/>

<sup>10</sup>ESTRELLA: European project for Standardized Transpar-

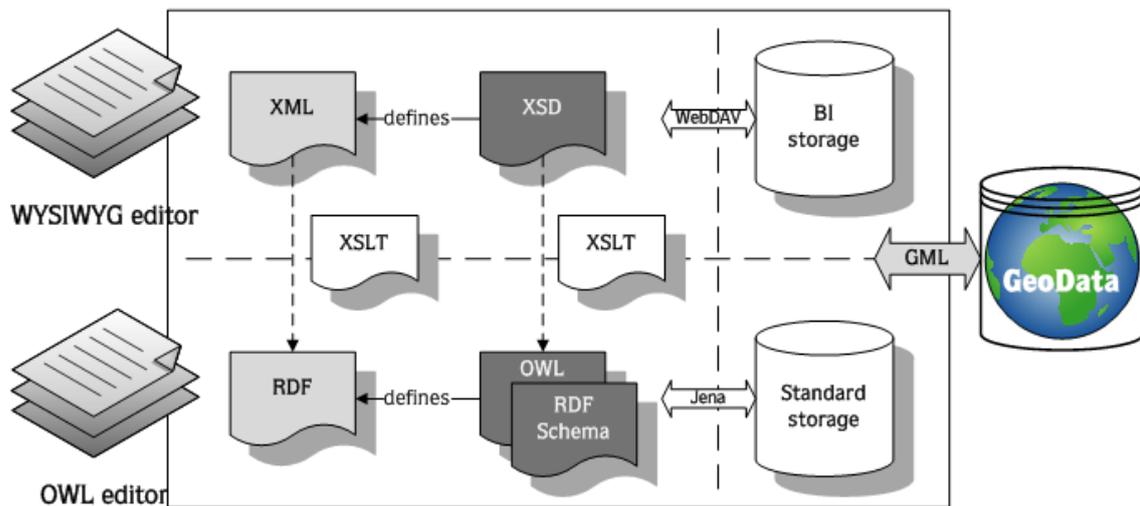


Figure 1: MetaVex Architecture

resentation format for legal knowledge. Existing Semantic Web initiatives are aimed at modelling concepts (OWL “ontologies”) and rules (RuleML and SWRL). The LKIF builds on but goes beyond this generic work to allow further kinds of legal knowledge to be modelled, including: meta-level rules for reasoning about rule priorities and exceptions, legal arguments, cases and case factors, values and principles, and legal procedures. It is based on a layered approach, providing a method of using OWL and SWRL, and contains two sublanguages, for defeasible rules and for subjunctive betterness. Furthermore, the LKIF is grounded in a core ontology of basic legal concepts: LKIF Core [6]. The ontology covers a base level of components required for explaining epistemological, situational, and mereological patterns as they occur in legal reasoning.<sup>11</sup>

These formal representations of the semantics of legal texts can be used to perform elaborate legal reasoning, such as consistency checking, legal assessment etc. and for building knowledge-based applications which can be used by citizens to gain advice on complex legal issues.

An example of the flexibility of the MetaLex schema is the combination of regulations, maps and spatial planning adopted in the Legal Atlas tool [12]<sup>12</sup>. The MetaLex region attribute can be used to refer to the geographical region to which rules in a regulation can be applied: i.e. it specifies geographical jurisdiction. In Legal Atlas, this is used in combination with RDF and GML<sup>13</sup> to show spatial planning regulations both as maps and as texts.

#### 4. METAVEX

MetaVex is a platform independent open source editor, and shares a large part of its codebase with the Visual Editor for XML (Vex)<sup>14</sup>. It is developed within the Java

ent Representations in order to Extend Legal Accessibility, IST-2004-027655, <http://www.estrellaproject.org>

<sup>11</sup><http://www.estrellaproject.org/lkif-core>

<sup>12</sup><http://www.leibnizcenter.org/projects/current/legal-atlas>

<sup>13</sup>Geography Markup Language

<sup>14</sup>Vex is currently no longer under active development. See

Eclipse<sup>15</sup> development platform, which allows future development of plug-ins and add-on functionality. Figure 1 shows the MetaVex architecture and the way in which the various components are integrated. In short, an XML document is constructed against an XML schema, and can be translated to RDF/OWL using XSLT transformations. MetaLex XML documents are stored in an XML storage facility (to be developed by one of the partners in the SEAL consortium) through a WebDAV interface. The RDF/OWL representation of the MetaLex document uses the vocabulary specified in the MetaLex OWL schema. It is stored using a Jena database backend. Currently, not all of the components shown in the picture have been developed. Future development issues are described in the discussion.

MetaVex is specifically intended to support the creation of documents complying with the MetaLex standard for legal sources, but flexible enough to be easily adapted to different XML schemas.

Target users of MetaVex will be drafters and members of parliament. These users cannot be expected to be familiar with editing an XML-structure directly. For this reason, the editor offers a WYSIWYG interface, which does not require any knowledge of or experience in creating XML-code. In fact, the editor shows close resemblance to a conventional word processor, and at the same time allows a user to alter or create content while keeping the integrity of the underlying structure intact.

Since XML documents themselves do not carry information about how to display the document, MetaVex uses CSS<sup>16</sup> to determine formatting. Users are able to choose from different types of predefined formatting, but cannot change the formatting in line.

The use of XML as an underlying document structure makes it easy to validate the structure of documents created with MetaVex against the rules defined in the MetaLex schema file. This schema file restricts element and attribute names and allowed combinations. MetaVex uses this schema

<http://vex.sourceforge.net/>

<sup>15</sup><http://www.eclipse.org>

<sup>16</sup>Cascading Style Sheets, see <http://www.w3.org/Style/CSS>

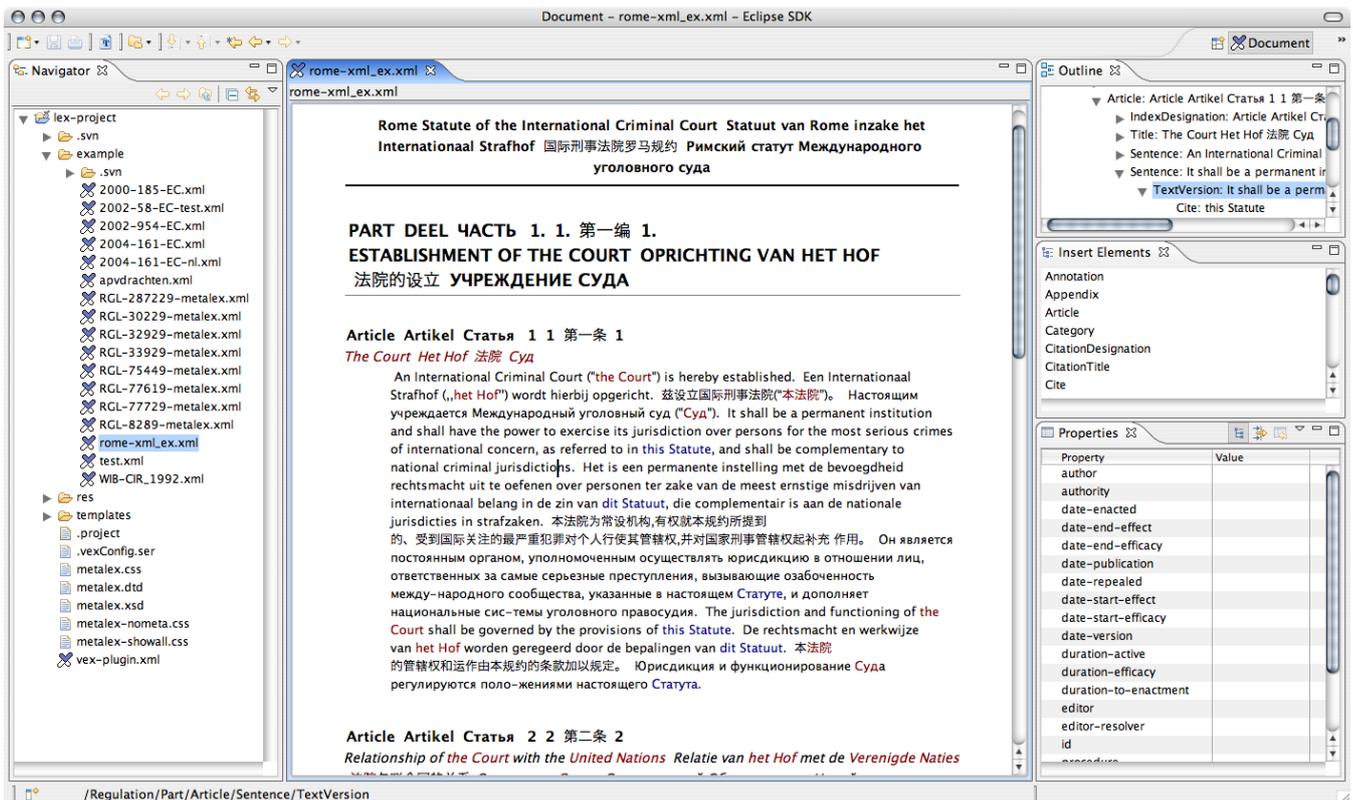


Figure 2: Editing the Rome Statute in MetaVex

file to check which elements can be inserted at a certain position in a document, while sustaining a well-formed and valid XML structure.

To make sure such a structure is obtained during composing or editing a document, the editor provides the user a menu of elements that are allowed at a particular point in the document. By enforcing users to choose from these elements, users can only add valid elements to the document.

This functionality is one of the major differences between MetaVex and normal word processors. Using a normal word processor, a user can just start typing and does not have to bother about adding specific text elements: The use of templates strongly reduces this difference by offering users the ability to add new elements or whole blocks of elements at once: creating e.g. an article is similar to form-filling.

In the Netherlands, legislative documents are required to be composed according to what is prescribed in the Dutch Guidelines for Legal Drafting [5]. These guidelines do not only apply to technical aspects of writing legislation, but emphasise content too. MetaVex provides a set of templates that are structured according to these guidelines. These templates can provide not only structure, but standard content as well.

Currently, the templates included in MetaVex follow Dutch guidelines and cannot be used to support drafting in other countries. However, other templates can be easily imported and used in MetaVex as well. This extensive use of templates not only offers guidance, but can also save users a lot of work.

The MetaVex user interface offers an “Outline” pane that shows the overall document structure as a hierarchy (see

Figure 2). This structure can be collapsed or expanded and allows a user to easily navigate through parts of the text. Conversely, the cursor position within the XML structure is reflected both in the selected element in this outline pane and in an XPath expression in the status bar. The “Insert Elements” panel shows a list of all elements that can be validly inserted at the current cursor position. The same list is accessible through a context sensitive right-mouse menu. Furthermore, as mentioned in the previous section, MetaLex supports an extensive set of meta-data attributes which allow users to link many different kinds of extra information to a document. The user interface of MetaVex allows the user to edit the values of these attributes through the “Properties” panel. This panel displays a table of all attributes and their values available on the currently selected XML element. Finally, the “Navigator” panel shows a list of the files available in the current project. Each of these panels can be moved, closed or enlarged to suit a users’ preference.

Most prominent in Figure 2 is the editing panel. Multiple versions of the same text can be simultaneously edited in a single MetaLex document. The screenshot illustrates this, showing the Statute of Rome (which introduces the International Criminal Court) in both Dutch, English, Chinese and Russian language versions. As this can be confusing to users, MetaVex can hide irrelevant information: users can select a desired time interval or language version and hide other versions available in the document (see Figure 3). The selection of text versions is also used by the export functions of MetaVex. Any valid document can be exported to various common formats such as PDF and HTML using export

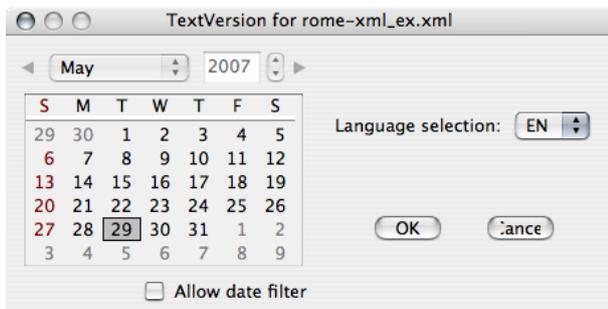


Figure 3: Selecting a text version

wizards, which apply XSLT<sup>17</sup> transformations to the XML source to produce the desired output format.

Although MetaVex is primarily intended for editing and producing legislative documents, this does not mean that MetaVex is limited to “legal” documents only. In general, for any document bearing rules for a certain domain, a structured way of creating and filing those documents is certainly useful. MetaVex can be used to suit the needs of many kinds of domains by providing structuring and formatting. For example, companies can develop their own general representation of data in the form of an XML schema file, according to which the intended documents will be composed and validated.

## 5. DISCUSSION AND FUTURE WORK

Besides the functionality MetaVex already offers, there are still a few items on the requirement list that are currently not supported.

Firstly, MetaVex should enable intuitive construction and maintenance of references in legal documents, by functions for adding, editing and validating. [7] describes a parser that can automatically detect references in legislation with high accuracy. Embedding this parser into MetaVex certainly would be a useful extension

The use of MetaLex means a strong focus on semantic web technologies such as RDF(S) and OWL. For now it suffices to support these formats and edit OWL documents in a separate, already existing OWL-editor, such as Protégé<sup>18</sup> or TopBraid<sup>19</sup>. In the future, MetaVex will offer means to view and edit OWL-documents using an OWL-editor embedded inside MetaVex. The idea is to develop a separate view showing the RDF triples corresponding to a selected MetaLex element. The view should also enable users to edit the RDF triples directly. Admittedly, concurrent editing of the RDF and XML version requires relatively complex synchronisation.

The MetaVex architecture (Figure 1) shows a connection to geodata. At this moment, MetaVex does not provide this connection yet, but it will provide one in the future. The connection will be similar to the approach of Legal Atlas.

So far we have not discussed MetaVex’ storage mechanism and how certain features like versioning, security, groupware facilities etc. will be integrated. Although an implicit way of

maintaining version information using MetaLex’ attributes already exists, the ability to cope with several versions of a document in an explicit matter should also be taken into account. A content management system satisfying the requirements of section 2, will be developed within SEAL. However, MetaVex will commit to standards-based interfaces to open source RDF repositories such as Sesame<sup>20</sup> and Jena/Joseki<sup>21</sup>.

MetaVex is still under construction and there is a lot of work that needs to be done. Nevertheless a solid, easy extendable and highly adaptable solution for editing XML-structured documents in a user-friendly environment already exists. As soon as all proposed features are fully present, MetaVex will lift the editing of legal documents to a whole new level by its unique combination of syntax and semantics.

## 6. REFERENCES

- [1] T. Agnoloni, E. Francesconi, P. Spinosa. xmLegesEditor, an OpenSource visual XML editor for supporting Legal National Standards. In C. Biagioli, E. Francesconi and G. Sartor, editors, *Proc. of V Legislative XML Workshop*, pages 239-252, European Press Academic Publishing, 2007.
- [2] A. Boer, T. Gordon, K. van den Berg, M. Di Bello, A. Föhrhécz, and R. Vas. *Specification of the Legal Knowledge Interchange Format (LKIF)*. Deliverable 1.1, Estrella, 2007.
- [3] A. Boer, R. Hoekstra, R. Winkels, T. van Engers, and F. Willaert. METALex: Legislation in XML. In T. Bench-Capon, A. Daskalopulu, and R. Winkels, editors, *Legal Knowledge and Information Systems (Jurix 2002)*, pages 1-10, Amsterdam, 2002. IOS Press.
- [4] A. Boer, R. Winkels, T. van Engers and E. de Maat. A Content Management System based on an Event-based Model of Version Management Information in Legislation. In T. Gordon (ed.), *Legal Knowledge and Information Systems. Jurix 2004*, pages. 19-28. Amsterdam, 2004. IOS Press.
- [5] *Aanwijzingen voor de regelgeving (Directives for regulations)*, regulations for legislative drafting issued by the Prime Minister, November 26, *Stcrt.* 1992.
- [6] J. Breuker, R. Hoekstra, A. Boer, K. van den Berg, R. Rubino, G. Sartor, M. Palmirani, A. Wyner, and T. Bench-Capon. *OWL ontology of basic legal concepts (LKIF-Core)*. Deliverable 1.4, Estrella, 2007.
- [7] E. de Maat, R. Winkels, and T. van Engers. Automated detection of reference structures in law. In Tom M. van Engers, editor, *Legal Knowledge and Information Systems. Jurix 2006: The Nineteenth Annual Conference*, volume 152 of *Frontiers in Artificial Intelligence and Applications*, pages 41-50, 2006. IOS Press.
- [8] M. Palmirani and R. Brighi. An XML Editor for Legal Information Management, *Proceedings of DEXA 2003, the 2nd International Conference on Electronic Government - EGOV 2003*, pages 421-429. Prague, 2003. Springer-Verlag.
- [9] N. Shadbolt, T. Berners-Lee, and W. Hall. The

<sup>17</sup>eXtensible Stylesheet Language Transformations, <http://www.w3.org/Style/XSL/>

<sup>18</sup><http://protege.stanford.edu>

<sup>19</sup><http://www.topbraidcomposer.com>

<sup>20</sup><http://www.openrdf.org>

<sup>21</sup><http://jena.sourceforge.net>

Semantic Web Revisited. *IEEE Intelligent Systems* 21(3), pages 96-101, 2006.

- [10] F. Vitali, F. Zeni. Towards a country-independent data format: the Akoma Ntoso experience, In C. Biagioli, E. Francesconi and G. Sartor, editors, *Proc. of V Legislative XML Workshop*, pages 239-252. European Press Academic Publishing, 2007.
- [11] Radboud Winkels, Alexander Boer, Emile de Maat, Tom van Engers, Matthijs Breebaart, and Henri Melger. Constructing a semantic network for legal content. In Anne Gardner, editor, *Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL)*, pages 125-140, Bologna, Italy, June 2005. IAAIL, ACM Press.
- [12] R. Winkels, A. Boer, and E. Hupkes. Legal Atlas: Access to Legal Sources through Maps. In Radboud Winkels, editor, *Proceedings of the 11th International Conference on Artificial Intelligence and Law (ICAIL)*, Stanford, 2007. ACM Press.