XML based Legal Document Drafting Information System

Ovidiu Vasutiu(a,b), David Jouve(b), Youssef Amghar(a), Jean-Marie Pinon(a)

(a) Laboratoire d'InfoRmatique en Images et Systèmes d'information – INSA de Lyon
(b) Caisse Nationale des Allocations Familiales

(a){ovidiu.vasutiu, youssef.amghar, jean-marie.pinon}@insa-lyon.fr
(b){david.jouve}@cnedi69.cnafmail.fr

Introduction

Legal documents play an important role in all activities related to the legal domain. In particular they represent an efficient human communication mean to transmit legal knowledge. Legislations are often complex and prone to change. Organizations that base their daily work on a set of legal documents have to deal with a massive amount of legal documents (hundreds of thousands) and numerous legal updates (several legal updates every week). Each legal update must be incorporated into the organization's internal legal documentary corpus and in its Legal Information System.

Motivations

For any legal sub-domain, the set of relevant legal documents is partially structured [1], documents can be laid out in a pyramidal structure according to their legal importance, *Fig 1*. For organizations such the French Family Benefits Office, *Cnaf*[1], external received legal documents are modified, updated and enriched within the organization. In this way legal texts are evolving, legal content is enriched to produce highly operational degree documents at the lowest level of the pyramid: represented by technical instructions used for the design of the information systems components [2]. We witness a large legal information redundancy between pyramid levels; different documents at different levels express the same legal knowledge in different forms.

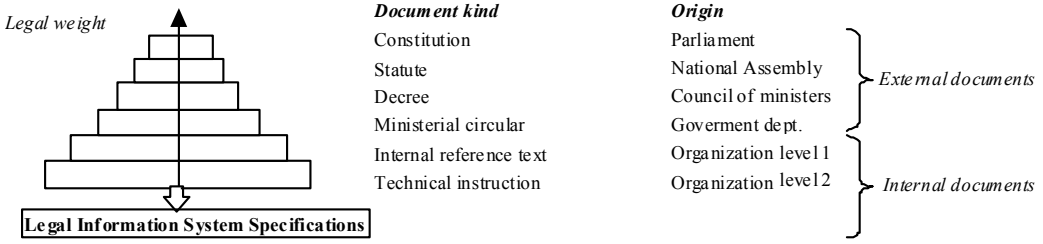

Fig. 1. Legal documents hierarchy

Moreover legal documents have normative semantics [3,4], they reflect a normative knowledge. Due to the legal context of each country, legislation is constantly added, deleted or modified, like an informational flow in perpetual evolution. One change at a given level of the legal pyramid will have repercussions to all beneath levels until the lowest level is reached. Assuring the coherence and the consistency of the documentary corpus becomes very difficult. By example, the *Cnaf,* has to deal with an average of two legal updates a week; large teams work to continually update the documentary repository. Therefore collaborative work and traceability for legal changes become very important.

While many existing document management systems are specialized in publication, indexation or administration of document repositories and in information retrieval, they do not respond to specific demand of a legal document drafting environment. In this position paper, we present the operational solution for the legal document drafting information system of our organization with integrated and automated mechanisms to guarantee consistency, conformity and coherence.

Legal Document Drafting and Referential Information System

Our solution to the issues presented here is an XML based solution for the drafting referential Document Management Information System (DMIS).

Meta-model of the DMIS

The legal document referential embeds a set of *Document collections.* Each collection is provided with a set of descriptors defining all the types of contained resource: document types (i.e. decree, ministerial circular, etc.), asset types, reference types… The collection meta-description also contains: *taxonomies, docflow definitions* and *references types.*

Basic features

- *Document format and structure :*

To avoid redundancy, often present in the legal domain, two structures are used: the document persistence structure (content stored at only one place) and the document presentation structure (user ended documents containing richer and more detailed information, an aggregation from different persistent documents). All documents respect a generic structure having two top elements: the *meta* element (all document meta-data: document type, ID, version, revision and edition identifiers, origin, state and docflow step history) and the *content* element (contains the legal information). The system uses a native XML database to store persistence structures. For backward or external interoperability raisons it can also wrap with any other content type.

- *Version and revision management*

The repository contains multiples versions and revisions of a same document. Each version is associated to a different legal drafting workspace answering to a legal update. At any time, editors can synchronize, using a merge tool, their working version of the document with the reference version of the document (the public version of the document).

[1] Cnaf - Caisse Nationale des Allocations Familiales

- *Resource identification*

Legal content is scattered all over the document corpus thus preserving link integrity of the document repository is a crucial problem. A solution to precisely addressing a resource is the use of Universal Resource Names or URNs [5,6]. A revision of a document is precisely identified by the unique document identifier, its version and revision number. It is referenced by a storage independent URN formed as follows:

```
collection-name : resource-kind : resource-type : resource-id[: version : revision]
```

URN resolution rules will identify the physical document, taking in consideration if the version and/or revision are specified or not. Documents with the current or reference version and those in the last revision are privileged.

- *Document reference model / Document Addressing Model*

A reference is encoded as an XML element with a set of attributes that is mapped to the URN of the target:

```
<ref col="sair-daf" kind="docobject" type="tache" res="00000003"
 [ver="REF" rev="0001" nodes="id1 id2" ref-type="originContent"]/>
```

The *col*, *kind*, *type*, *res* define the URN of the referenced resource. The optional *nodes* attribute can address a specific sub-fragment of the document. Referenced document's label, description, type are retrieved. Sub-fragments identified by the "nodes" attribute, content defined by the "ref-type" are included in the presentation of the referencing document. All the information concerning the target document is stored within so that any update is automatically available. We can identify two categories of reference types: (1) *simple reference*: a document points out to another document implying a simple link, (2) *specialized reference*: attribute *ref-type* is present defining the semantics of the link. The semantic meaning defines the transformations to be made during loading to build the presentation document.

- *Docflow*

Each document at a given moment is associated to a state in its lifecyle (specific to each document type). User contextual menu functions are defined according to user's habilitations and document state. An internal docflow engine listens to external (function activation, service call, etc...) or internal (document transitions) events that might (if all conditions are validated) launch a transition to another state. Actions can be executed on in/out transitions.

- *Information retrieval*

The retrieval mechanism is very flexible allowing users to specify a search expression (containing multiple keywords or phrases) and/or document expected metadata (identifier, document type, current state, date).

Legal document drafting functionalities

- *Unique referential*

The resource addressing feature and link resolution algorithm allow us to reduce redundancy without losing any valuable performance. A link can either point to the last revision of a document, or it can be strictly forced to point out to a specific revision number, even if obsolete. So documents in parallel editions workspaces (different versions) can be referenced without any interference. Using the "ref-type" attribute, we can define meaningful links between two documents (i.e. origin of the document, legal knowledge dependency or similarity) or how content from a document is included in the presentation structure of another document.

- *Legal drafting process management*

Responsibility in legal document drafting is often shared between many actors. Docflow mechanisms allow us to define the automation of the drafting process, the roles of each participant, the flow of documents and information, in whole or part, from one decisional instance or state to another for action.

- *Legal knowledge and information traceability*

Meaningful links allow us to trace the usage of the legal knowledge and its propagation while meta-data allow us to trace evolutions, contributors, contributions and updates of the documents.

- *Dependency and impact analysis*

Due to traceability in our system we can identify the documents that are in a way related to a legal text and that need to be updated with each legal evolution, giving an estimate of the impact. Furthermore as Information System design documents are related to the law, it allows us to link requirements back to corresponding design artifacts and code.

Discussions and conclusion

In the Motivations section we have presented several important issues that a legal document drafting system must address for better production, access and retrieving legal documents. The legal knowledge and document management system presented here and implemented at the *Cnaf* responds to theses demands.

For instance coherence is assured by the reference and traceability mechanisms. References can allow us to avoid redundancy, often present in legal documents. Knowledge equivalence reference allow us to identify documents, although different but expressing the same legal knowledge [7]. The complexity of the legal knowledge transforming and drafting process is managed with the docflow and version/revision control mechanism

All the feed-back received after a user validation phase shows that even if the new mechanisms implied more constraints in the work process, they help to reduce the risk of error. Traceability helped users to precisely identify the documents that need to be updated and version/revision control allowed then to constantly be able to control changes and evolutions.

Further more this kind of architectures must allow to establish a link between legal documents and information systems with their specification and design documents. Our next approach is to use UML models of the Legal Information System together with the specification and design documents to determine impacts[8] (documents or application components) and therefore to assure the coherence of documentary corpus and the Legal Information System.

References

1. Jouve, D., Chabbat, B., Amghar, Y., Pinon, J-M. : Modélisation sémantique de la réglementation. *Ingénierie des Systèmes d'Information*, Vol. 6, No. 2, pp. 95—119 (2001)
2. Jouve, D., Amghar, Y., Chabbat, B., Pinon, J-M. : Conceptual Framework for Document Semantic Modelling : an Application to Document and Knowledge Management in the Legal Domain. Data & Knowledge Engineering, Vol. 46, No. 3, pp. 345--375 (2003)..
3. von Wright, G.H.: Norm and Action: A Logical Enquiry. International Library of Philosophy and Scientific Method, Routledge and Kegan Paul, London (1963)
4. Kelsen, H.: General Theory of Norms. Translation of ''Allgemeine Theorie der Normen'' by Hartney. Clarendon Press, Oxford, UK, (1991)
5. Amghar, Y., Chbeir, R.: Active Server for the Management of Structured Documents Link Integrity. In : IFIP WG8.1 International Conference on Engineering Information Systems in the Internet Context, Kanazawa, Japan September 25-27 (2002)
6. Berners-Lee, T.: Universal Resources Identifiers in WWW : a unifying syntax for the expression of names and addresses of objects on the network as used in the WWW. RFC 1630 (1994)
7. Jouve, D. : Modélisation sémantique de la réglementation, PhD Thesis, INSA de Lyon, 258 p. (2003). Retrieved Octomber 25, 2007, from http://csidoc.insa-lyon.fr/these/2003/jouve/these.pdf
8. Vasutiu, O., Amghar, Y., Jouve, D. : Gestion des changements et étude d'impact dans un système d'information réglementaire. In : XXIVème Congrès INFormatique des Organisations et Systèmes d'Information et de Décision (INFORSID 2006), pp. 1007—1022 (2006)