
Parsing the French « Journal Officiel » to Show the Evolution of Law

Vincent Rasneur - vrasneur@free.fr

Parsing the JORF...

Goals of the project (name: `juridiff`)

- get a very detailed view of how French law texts change over time
- example: what are the changes in French law between the 1st of January and the 1st of February?
- leverage existing software tools to do that

Parsing the JORF...

Are there any existing tools?

Légifrance website only has consolidated views. No way to see the differences over time.

No *diff view* in commercial databases (LexisNexis, Dalloz, ...) either.

I want something that looks like `rfcdiff`, but for French laws and with a timeline.

`rfcdiff` URL: <http://tools.ietf.org/rfcdiff>

Parsing the JORF...

Screenshot of rfcdiff

... skipping to change at **page 6, line 27** ...

```
namespace atom = "http://www.w3.org/2005/atom";
start = atomFeed | atomEntry
```

Both kinds of Atom Documents are specified in terms of the XML Information Set, serialised as XML 1.0 [X3C_REC-xml-20040204] and identified with the "application/atom+xml" media type. Atom Documents MUST be well-formed XML. This specification does not define a DTD for Atom Documents, and hence does not require them to be valid (in the sense used by XML).

Atom allows the use of IRIs [RFC3987], as well as URIs [RFC3986]. Every URI is an IRI, so any URI can be used where an IRI is needed. While IRIs must, for many protocols, be mapped to URIs prior to dereferencing, they MUST NOT be so mapped for comparison when used in atom:id. Section 3.1 of [RFC3987] describes how to map an IRI to a URI when necessary.

Any element defined by this specification MAY have an xml:base attribute [X3C_REC-xmlbase-20010527]. When xml:base is used in an Atom Document, it serves the function described in section 5.1.1 of [RFC3986], establishing the base URI (or IRIs) for resolving any relative references found within the effective scope of the xml:base attribute.

Any element defined by this specification MAY have an xml:lang attribute, whose content indicates the natural language for the

... skipping to change at **page 8, line 15** ...

3. Common Atom Constructs

Many of Atom's elements share a few common structures. This section defines those structures and their requirements for convenient reference by the appropriate element definitions.

When an element is identified as being a particular kind of construct, it inherits the corresponding requirements from that construct's definition in this section.

3.1 Text Constructs

A Text construct contains human-readable text, usually in small quantities. The content of Text constructs is Language-Sensitive.

... skipping to change at **page 6, line 27** ...

```
namespace atom = "http://www.w3.org/2005/atom";
start = atomFeed | atomEntry
```

Both kinds of Atom Documents are specified in terms of the XML Information Set, serialised as XML 1.0 [X3C_REC-xml-20040204] and identified with the "application/atom+xml" media type. Atom Documents MUST be well-formed XML. This specification does not define a DTD for Atom Documents, and hence does not require them to be valid (in the sense used by XML).

Atom allows the use of IRIs [RFC3987]. Every URI [RFC3986] is also an IRI, so a URI may be used wherever below an IRI is named. There are two special considerations: when an IRI which is not also a URI is given for dereferencing, it MUST be mapped to a URI using the steps in Section 3.1 of [RFC3987]; when an IRI is serving as an atom:id value, it MUST NOT be so mapped, so that the comparison works as described in Section 4.2.6.1.

Any element defined by this specification MAY have an xml:base attribute [X3C_REC-xmlbase-20010527]. When xml:base is used in an Atom Document, it serves the function described in section 5.1.1 of [RFC3986], establishing the base URI (or IRIs) for resolving any relative references found within the effective scope of the xml:base attribute.

Any element defined by this specification MAY have an xml:lang attribute, whose content indicates the natural language for the

3. Common Atom Constructs

Many of Atom's elements share a few common structures. This section defines those structures and their requirements for convenient reference by the appropriate element definitions.

When an element is identified as being a particular kind of construct, it inherits the corresponding requirements from that construct's definition in this section.

Note that there MUST NOT be any whitespace in a Date construct or in any URI. Some XML-writing implementations erroneously insert whitespace around values by default, and such implementations will emit invalid Atom Documents.

3.1 Text Constructs

A Text construct contains human-readable text, usually in small quantities. The content of Text constructs is Language-Sensitive.

Diff view between 2 RFC* drafts

RFC: internet specification (may be a standard, not always)

Parsing the JORF...

Where are the changes?

- New laws and decrees are published in the JORF (« Journal Officiel »).
- The changes affecting existing texts are described like that in a new law or decree:

Le code civil est ainsi modifié :

A la première phrase du premier alinéa de l'article 778, le mot : « divertis » est remplacé par le mot : « détournés ».

English translation:

The Civil Code is amended as follows:

In the first sentence of the first paragraph of Article 778, the word « divertis » is replaced by the word « détournés ».

Parsing the JORF...

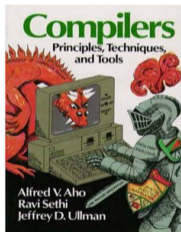
Properties of the changes

- They are written in a natural language (French)
⇒ a natural language is very difficult to parse...
- But... the grammar describing the changes is *relatively* simple
⇒ the set of actions is limited (*replace, append, remove, ...*)
⇒ words have a specific meaning (*article, sentence, paragraph, ...*)

Parsing the JORF...

Technical solution

- Just do as if the changes are described in a kind of programming language



Standard textbook for learning the internals of programming languages

- Add some techniques from information retrieval (*i.e.* used by search engines) to ease parsing

Parsing the JORF...

How to apply the differences?

Same steps as in existing programming language interpreters.

- *tokenization*: split sentence into tokens (*i.e.* verbs, dates, adjectives, ...)
- *parsing*: merge tokens into useful groups, tricky part!
 - filter useless words
 - stem the tokens, so there are no plurals nor conjugated verbs
 - apply a simplified French grammar
- *evaluation*: locate where the change happens, and apply it

Parsing the JORF...

Parsing example

Parsed representation of the Civil Code modification from above:

```
Action<word=remplacer, ...,
  before=[
    Reference<word=article, ..., name=[u'778']> ->
      Reference<word=alinea, ..., locators=[Ordinal<word=premier, nb=1>]> ->
        Reference<word=phrase, ..., locators=[Ordinal<word=premier, nb=1>]> ->
          Reference<word=mot, ..., name=[Quoted<length=8, ...>]>],
        after=[Reference<word=mot, ..., name=[Quoted<length=9, ...>]>]>
```

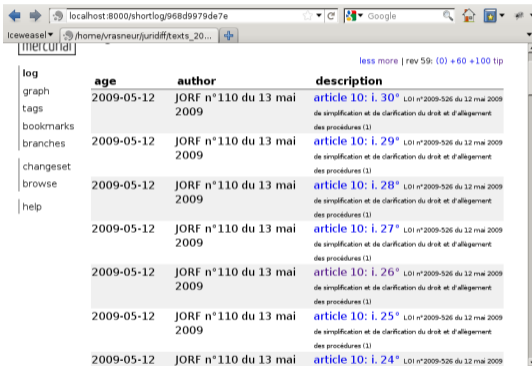
Parsing the JORF...

Versioning the changes

- use an existing system that works well: `mercurial`
- each law (and the texts changed by the law) is stored in a separate place, called a *branch* in `mercurial`.
- Changes are merged with the existing texts when they come into force

Parsing the JORF...

Screenshot



localhost:8000/shortlog/968d9979de7e

mercurial

less more | rev 59: (0) +60 +100 tip

	age	author	description
log	2009-05-12	JORF n° 110 du 13 mai 2009	article 10: i. 30 <small>LOI n°2009-526 du 12 mai 2009 de simplification et de clarification du droit et d'allègement des procédures (1)</small>
graph	2009-05-12	JORF n° 110 du 13 mai 2009	article 10: i. 29 <small>LOI n°2009-526 du 12 mai 2009 de simplification et de clarification du droit et d'allègement des procédures (1)</small>
tags	2009-05-12	JORF n° 110 du 13 mai 2009	article 10: i. 28 <small>LOI n°2009-526 du 12 mai 2009 de simplification et de clarification du droit et d'allègement des procédures (1)</small>
bookmarks	2009-05-12	JORF n° 110 du 13 mai 2009	article 10: i. 27 <small>LOI n°2009-526 du 12 mai 2009 de simplification et de clarification du droit et d'allègement des procédures (1)</small>
branches	2009-05-12	JORF n° 110 du 13 mai 2009	article 10: i. 26 <small>LOI n°2009-526 du 12 mai 2009 de simplification et de clarification du droit et d'allègement des procédures (1)</small>
changeset	2009-05-12	JORF n° 110 du 13 mai 2009	article 10: i. 25 <small>LOI n°2009-526 du 12 mai 2009 de simplification et de clarification du droit et d'allègement des procédures (1)</small>
browse	2009-05-12	JORF n° 110 du 13 mai 2009	article 10: i. 24 <small>LOI n°2009-526 du 12 mai 2009 de simplification et de clarification du droit et d'allègement des procédures (1)</small>
help	2009-05-12	JORF n° 110 du 13 mai 2009	

A law has been parsed and its changes are stored in the versioning system

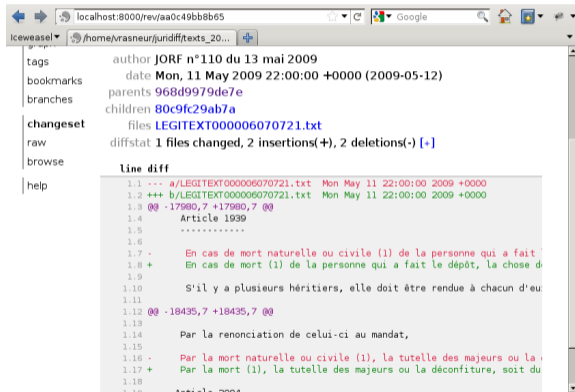
Parsing the JORF...

How to view the differences?

- use an existing system to view the differences:
mercurial itself or diffing tools (*i.e.* kdiff3)

Parsing the JORF...

Screenshot



```
tags          author JORF n°110 du 13 mai 2009
bookmarks     date Mon, 11 May 2009 22:00:00 +0000 (2009-05-12)
branches      parents 968d9979de7e
              children 80c9fc29ab7a
changeset     files LEGITEXT000006070721.txt
raw           diffstat 1 files changed, 2 insertions(+), 2 deletions(-) [+]
```

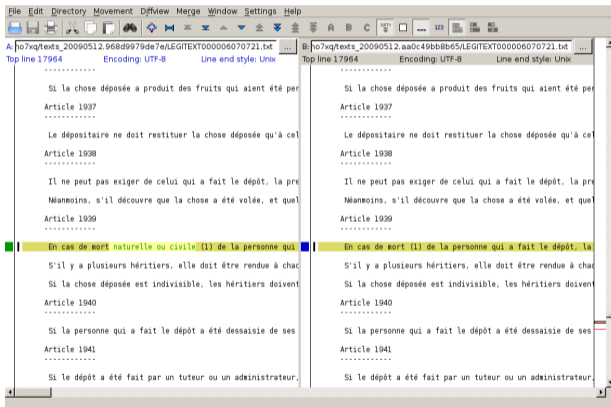
Line diff

```
1.1 --- a/LEGITEXT000006070721.txt Mon May 11 22:00:00 2009 +0000
1.2 +++ b/LEGITEXT000006070721.txt Mon May 11 22:00:00 2009 +0000
1.3 @@ -17980,7 +17980,7 @@
1.4     Article 1939
1.5     .....
1.6
1.7 -     En cas de mort naturelle ou civile (1) de la personne qui a fait
1.8 +     En cas de mort (1) de la personne qui a fait le dépôt, la chose d
1.9
1.10     S'il y a plusieurs héritiers, elle doit être rendue à chacun d'eu
1.11
1.12 @@ -18435,7 +18435,7 @@
1.13
1.14     Par la renonciation de celui-ci au mandat,
1.15
1.16 -     Par la mort naturelle ou civile (1), la tutelle des majeurs ou la
1.17 +     Par la mort (1), la tutelle des majeurs ou la déconfiture, soit d
1.18
1.19     Article 2004
```

Viewing the change in mercurial

Parsing the JORF...

Screenshot



Viewing the change in `kdif3`

Parsing the JORF...

Difficulties

- automation can never be perfect: may need to manually apply the changes
- grammar mistakes/misspellings in the JORF and in the laws and codes themselves! (some are corrected by the parser...)
- specific rules (coming into force, changes that affect all the existing texts without naming them, ...)

Parsing the JORF...

Technical details

The project (about 3000 lines of Python code) uses external libraries (open source)

- `python-stemmer`¹ for French stemming
- code (and ideas) borrowed from `NLTK`² for French language parsing
- `mercurial`³ VCS API for text versioning
- `kdifff3`⁴ for text comparison

1: <http://snowball.tartarus.org/>

2: <http://nltk.org/>

3: <http://mercurial.selenic.com/>

4: <http://kdifff3.sourceforge.net/>

Parsing the JORF...

Future work

- improve parsing
- test with more laws to know the parser accuracy
- extend the version control system to handle law amendments
- compute statistics about the changes
- reduce memory usage

Parsing the JORF...

Background (about me)

- have studied law (at Panthéon-Sorbonne university) and cryptography (Limoges)
- work as an R&D engineer in information security
- write detection engines for (web) security attacks
- lots of parsing (manual or grammar based)
- use `mercurial` with products composed of millions of lines of code

⇒ I use the same technologies at work, wanted to know if they could be useful to have a better understanding of French law.

Parsing the JORF...

Thank you

Questions?