Transformer un document XML, le cas de mes liens Wikipédia

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 février 2010. Dernière mise à jour le 4 mars 2010

`https://www.bortzmeyer.org/transformation-texte.html`

Comme le savent mes fidèles lecteurs, ce blog comporte pas mal de liens vers Wikipédia (plusieurs milliers différents `</auto/summary.html>`) pour expliquer un sigle, une technique ou un concept. J'essaie de les vérifier mais, évidemment, certaines tâches sont mieux faites par un logiciel qu'à la main. C'est le cas de la désambiguation de sigles (mettre un lien Wikipédia vers "*Domain Name System*" plutôt que vers DNS qui est ambigu). Ce blog étant réalisé en XML `<https://www.bortzmeyer.org/blog-implementation.html>`, est-il facile d'utiliser les techniques XML pour remplacer toutes les occurrences de `<wikipedia name="DNS">` par `<wikipedia name="Domain Name System">` ? Non, et c'est pour cela que j'ai dû récemment adopter une nouvelle méthode.

L'ancienne méthode `<https://www.bortzmeyer.org/transformation-xslt.html>` reposait sur un principe simple : la force de XML est la disponibilité d'un très grand nombre d'outils facilitant les tâches du programmeur (par exemple XSLT). Il semblait donc simple d'écrire un programme XSLT (en ligne sur `https://www.bortzmeyer.org/files/update-wp.xsl`) qui assurait ce remplacement des acronymes par des extensions, s'assurant ainsi que le lien vers « FAI » allait bien aboutir sur la page « Fournisseur d'accès à Internet » et pas sur la « Fédération anarchiste ibérique » ou bien sur la page d'homonymie.

Mais ce programme très simple n'est pas, je trouve, adapté à mes méthodes d'écriture. En effet, s'il respecte l'"*infoset*" (le document XML abstrait, avec sa structure et son contenu), il ne respecte pas du tout la syntaxe. Par exemple, il remplace les **entités** XML alors que je voudrais les garder pour faciliter l'édition (regardez le source XML de cet article `<https://www.bortzmeyer.org/rfc-editor-at-isi.html>`, par exemple, avec l'entité `&rfced;`). C'est une limite fondamentale de tous les outils XML.

Il peut donc être préférable (oui, je sais, c'est un affreux bricolage) de travailler au niveau du texte, par exemple avec des expressions rationnelles. Ce n'est en général pas conseillé du tout (XML a une syntaxe contextuelle et ne peut en général pas s'analyser uniquement avec des expressions rationnelles, une petite recherche sur Stack Overflow `<https://www.bortzmeyer.org/stack-overflow.html>` montre en effet plein d'articles sur la question `<http://stackoverflow.com/search?q=xml+regexp>`).

Mais, ici, c'est la seule solution que j'ai trouvé (c'est aussi parce que je voulais travailler sur le fichier avec les acronymes expansés, autrement j'aurais pu juste rajouter une étape au traitement qui va du source XML à la page HTML).

Donc, comment est-ce que je fais ? (Le programme responsable est `scripts/canon-wp.py` si vous récupérez l'archive complète des programmes de ce blog (en ligne sur `https://www.bortzmeyer.org/files/blog-support-files.tar.gz`).) Le programme est en Python et utilise le module d'expressions rationnelles <`http://docs.python.org/library/re.html`>. Il lit un fichier texte (`schemas/wikipe` dans l'archive du blog), cherche un lien `<wikipedia>` et remplace éventuellement le nom de l'article de Wikipédia. Un exemple d'une partie de la table est :

```
...
CMS Système de gestion de contenu
CPU Processeur
CSS Feuilles de style en cascade
CVS Concurrent versions system
...
```

L'expression rationnelle utilisée (et qui est **loin** d'être parfaite, voir les commentaires dans le programme) est :

```
(.*?)<wikipedia *((( |^ *)name *= *\"([^\"]+)\" *>([^<]+))|(*>([^<]+)))</wikipedia>
```

Une alternative possible, suggérée par Emmanuel Saint-James, est de travailler avec SAX, qui respecte les entités. Je testerai ça un jour, avec la bibliothèque Sax de Python <`http://docs.python.org/library/xml.sax.html`>.