XML TP1

E. Bruno

25 novembre 2010

Table des matières

Chapitre 1

TP1 - Edition de documents XML bien-formés et valides

1.1 Objectifs

- Utilisation des outils de base pour la création et l'édition de documents XML[1].
- Vérification des documents bien formés et validation d'un document par rapport à une DTD[2].
- Exemple de transformation avec XSLT[3] vers HTML[4], XSL[5] puis vers pdf.

Les documents exemples sont fournis à l'adresse http ://bruno.univ-tln.fr. Les outils sont installés sur la machine `sis.univ-tln.fr` (Vous commencerez par mettre en place une connexion avec votre clé publique vers cette machine.

1.2 Les outils pour l'édition et validation de documents

Les outils suivant sont utilisables pour l'édition :

- L'éditeur Emacs http ://www.gnu.org/software/emacs/emacs.html
  - Le mode psgml http ://sourceforge.net/projects/psgml/ - (pour activer esc-x xml-mode)
  - Le mode nxml (par défaut sur lsis, ne supporte pas les dtd) (pour activer esc-x nxml-mode)
  - Le mode tdtd http ://www.menteith.com/tdtd/
  - Un exemple de fichier de configuration de emacs est proposé dans emacs.dotfile (à renommer en .emacs).
- Le logiciel commercial xmlspy (http ://www.xmlspy.com) disponible sous windows ou sous linux avec wine.
- Le logiciel commercial oxygen (http ://www.oxygenxml.com) sur sis `/usr/local/Oxygen`
- Un plugin pour eclipse xmlbuddy (http ://xmlbuddy.com/)

Pour la validation en ligne de commande :

- Parser XML[1] : Xerces (http ://xml.apache.org/xerces2-j/index.html)
- Opensp (depuis emacs), parser SGML[6] adapté à XML[1] (http ://openjade.sourceforge.net/).

Pour la transformation de document XML[1] :

- Processeur XSLT[3] : Xalan (http ://xml.apache.org/xalan-j/index.html)

1.3 Validation d'un document associé à une DTD

Le fichier books.dtd.zip décrit un modèle de document qui permet de représenter une collection de livres. Le fichier books-errors.xml est un document XML[1] exemple qui est une instance de la dtd précédente, il contient des erreurs. Deux types d'erreurs peuvent apparaître. Des erreurs de non-conformité avec le langage XML[1] (balises

[1] Extensible Markup Language
[2] Document Type Definition
[3] Extensible Stylesheet Language Transformations
[4] HyperText Markup Language
[5] Extensible Stylesheet Language
[6] Standard General Markup Language

non fermés, ...), on dira alors que le document n'est pas bien formé ; et des erreurs de non conformité par rapport à une DTD[2], on dira alors que le document n'est pas valide par rapport à cette DTD[2].

Pour vérifier la conformité d'un document à la norme XML[1], on utilise un parser. La vérification peut aussi être faite depuis Emacs avec le parser onsgmls (ctrl-c ctrl-v) ou bien depuis Oxygen.

Corriger les erreurs syntaxiques du document books-errors.xml pour le rendre bien formé.

Pour valider un documents XML[1] par rapport à une DTD[2], on ajoute un DOCTYPE dans le document pour le lier à cette DTD[2], le parser l'utilisera pour vérifier la conformité du document. Rendre le document books-errors.xml valide par rapport à la dtd books.dtd. Vous pouvez soit réutiliser onsgmls, soit utiliser Xerces, soit oxygen :

```
/usr/local/bin/xerces-validate <doc.xml> (regarder le script !)
```

1.4 Edition d'un document XML

En utilisant l'éditeur Emacs et le mode psgml (pour insérer des éléments et créer des attributs) ajouter des livres et des catégories au document exemple (cf. http ://www.lavoisier.fr/fr/livres/index.asp ?texte=xml\&select=motcle\&exact=on\&tog

La documentation du mode psgml se trouve dans le fichier http ://www.snee.com/bob/sgmlfree/psgmqref.html, et le manuel ici http ://www.lysator.liu.se/\~ lenst/about\textunderscorepsgml/psgml.html. Vous utiliserez en particulier :

- C-c C-v (sgml-validate) validates a document. It must be done from the main document file, the one with the DTD[2] declaration at

the top. (For our documents that's usually doc/o/*/main.sgml.)

- C-c C-r (sgml-tag-region) tags a region of marked text. The system prompts for a tag name and will not accept an invalid one. You

can use the Tab key for auto-fill for the name.

- C-c C-e (sgml-insert-element) prompts for a tag name, and then puts an empty element (start and end tags) at the point at the

cursor location. The system checks for valid entries, and offers auto-fill.

- C-c = (sgml-change-element-name) only works if the cursor is inside the element's start or end tag. It prompts for a new tag name

and changes the element to what you enter. The system checks for valid entries, and offers auto-fill.

- C-c - (sgml-untag-element) removes the tags of an element. The cursor must be inside the element's start or end tag.
- ESC C-k (sgml-kill-element) removes the entire element with all its children, text, and tags intact. This is useful for moving whole

chunks of tagged text within and between documents. The cursor must be positioned before the start tag of the element.

- C-c RET (sgml-split-element) inserts an end tag for the current element followed by a beginning tag for the same kind of element,

effectively splitting the element at that point. This is an easy way to make two paragraphs. The cursor must be between (and not inside) the start and end tags of the element.

Vous mettrez éventuellement la DTD[2] à jour, et vous validerez le document modifié.

1.5 Espaces de noms

Modifier le document précédent en utilisant les espaces de noms. Vous structurerez de façon plus fine les auteurs pour leur ajouter un `title` (Dr., Pr., Ms., Mrs., Miss). Pour distinguer les éléments ayant le même nom, vous définirez deux espaces de noms, un pour les personnes et un pour les auteurs. Attention, les DTD[2] ne supportent pas les espaces de noms (cf. schemas) par contre vous pouvez utiliser les noms qualifiés (ie le même préfixe).

1.6 Création d'une définition de type de documents : DTD

1.6.1 Baliser des documents

Proposer une structuration XML[1] pour les classes et interfaces de la javadoc http ://java.sun.com/j2se/1.5.0/docs/api/index.html .Vous détaillerez une exemple avec les classes `String` et `StringBuilder`. Attention, le site Web est orienté présentation mais votre document XML[1] devra décrire la structure logique. Vous utiliserez en particulier la possibilité de décrire des listes et des tableaux et les références internes.

1.6.2 Construire une DTD

A partir des balisages que vous venez de définir, proposer une DTD[2] `javadoc.dtd` et valider les deux documents.

1.7 Une DTD industrielle, Docbook

La DTD[2] docbook (ainsi qu'une version simplifié) sont présentés sur les sites http ://www.docbook.org et http ://www.oasis-open.org/docbook/. Un exemple simple de document se trouve dans le fichier sample-docbook.xml.

La documentation complète de docbook écrite en XML[1] se trouve ici : http ://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/docbook/de une version HTML[4] (générée automatiquement) est disponible ici : http ://www.docbook.org/tdg/en/html/docbook.html.

1.7.1 Créer un document conforme à la DTD docbook

Écrire le manuel d'utilisation d'un projet que vous avez réalisé en utilisant la DTD[2] docbook.

1.7.2 Transformer un document XML

Un document XML[1] peut être transformé en un autre document XML[1] (ou en un fichier texte) en utilisant des feuilles de style XSLT[3] (celles-ci seront étudiées ultérieurement en cours, nous allons ici simplement les utiliser).

Trois feuilles de style sont proposées pour transformer un document XML[1] valide pour la DTD[2] docbook en

- HTML[4] : /usr/share/xml/docbook/stylesheet/nwalsh/html/docbook.xsl
- XHTML[7] : /usr/share/xml/docbook/stylesheet/nwalsh/xhtml/docbook.xsl
- XSL[5]-FO : /usr/share/xml/docbook/stylesheet/nwalsh/fo/docbook.xsl

La commande pour appliquer une feuille de style est :

```
/usr/local/bin/xalan -IN <source.xml> -XSL <stylesheet.xsl> [-OUT result.???] (Regarder le script
et les autres paramètres !)
```

Transformer le document de l'exercice précédent en un document HTML[4], puis en un document XHTML[7].

La troisième feuille de style produit un document XML[1] dans un format appelé FO (Formating Object) qui décrit un document formaté. Ce document peut ensuite être converti vers un format propriétaire (dvi, ps, pdf, . . .). Pour réaliser cette transformation, deux outils (en cours de développement) sont disponibles FOP (http ://xml.apache.org/fop/) et PassiveTeX (http ://www.tei-c.org.uk/Software/p

Deux commandes possibles fop et pdfxmltex :

- fop <fichier.fo> -pdf <fichier.pdf>
- pdfxmltex <fichier.fo>

Les commandes docbook2pdf, html, rtf, . . . utilisent un autre langage de feuilles de style.

[7] Extensible HyperText Markup Language