



par Floris Lambrechts
<floris/at/linuxfocus.org>

L'auteur:

Je suis l'éditeur principal de LinuxFocus/Hollande depuis plusieurs années. J'étudie pour devenir 'ingénieur en électronique industrielle' à Leuven, en Belgique et je "perds" mon temps à jouer avec Linux, PHP, XML et LinuxFocus, tout en lisant des livres de Stephen Hawking et (en ce moment:) de Jef Raskin, 'L'Interface Humaine'.

Traduit en Français par:
Georges Tarbouriech
<georges.t/at/linuxfocus.org>

Découvrir XML



Résumé:

Voici une brève introduction à XML. Vous y rencontrerez Eddy le chat meta, la police de la syntaxe XML, et quelques DTD. Ne vous inquiétez pas, nous allons expliquer ;-)

Introduction

Durant l'été 2001, quelques uns des éditeurs de LF se sont retrouvés à Bordeaux pendant le LSM. De nombreuses discussions au sein du groupe documentation du LSM ont porté sur le même sujet : XML. De longues (et plaisantes) heures ont été passées à expliquer ce qu'est vraiment XML, à quoi il sert et comment l'utiliser. Au cas où vous seriez intéressés, c'est exactement ce que cet article va essayer d'aborder.

J'aimerais remercier Egon Willighagen et Jaime Villate de m'avoir amené à XML. Cet article repose en

grande partie sur le travail de Jaime, que vous trouverez dans la rubrique liens en bas de page.

Qu'est-ce que XML

Nous les "besogneux" en documentation, savions tous plus ou moins ce qu'était XML. Après tout, sa syntaxe est très semblable à celle du HTML et ce n'est rien qu'un autre langage à balises tel que le SGML et (encore) le HTML, d'accord ? D'accord. Mais c'est quand même un peu plus. XML possède certaines propriétés qui en font un "manipulateur" de données utilisable pour presque tout. Il semble être capable de gérer les choses les plus complexes tout en restant lisible à l'oeil humain et facile à analyser dans des programmes. Comment est-ce possible ? Penchons-nous sur ce langage étrange.

Eddy, le chat meta

Tout d'abord, XML est un *langage à balises*. Les documents écrits dans un langage à balises contiennent essentiellement deux choses : des *données*, et des *meta-données*. Si vous savez ce qu'est exactement une donnée, prévenez-moi, mais en attendant, parlons de la meta-donnée ;). Pour faire simple : la meta-donnée est une information supplémentaire qui ajoute un contexte ou une signification particulière à une donnée. Un exemple : prenons la phrase '*Mon chat s'appelle Eddy*'. Un humain tel que vous sait que '*chat*' est le nom d'une espèce animale, et '*Eddy*' le nom de l'animal. Toutefois, les programmes d'ordinateurs ne sont pas humains et ne savent pas tout ça. Nous utilisons donc des meta-données pour ajouter un sens à la donnée (grâce à une syntaxe XML, bien sûr !) :

```
<sentence>  
  Mon <animal>chat</animal> s'appelle <name>Eddy</name>.  
</sentence>
```

Maintenant, même un programme bête peut dire que "chat" est une espèce et que 'Eddy' est un nom. Si nous voulons un document dans lequel tous les noms sont écrits en bleu, et toutes les espèces en rouge, XML nous rend les choses très simples. Rien que pour le plaisir, voici ce que nous obtiendrions :

Mon chat s'appelle Eddy.

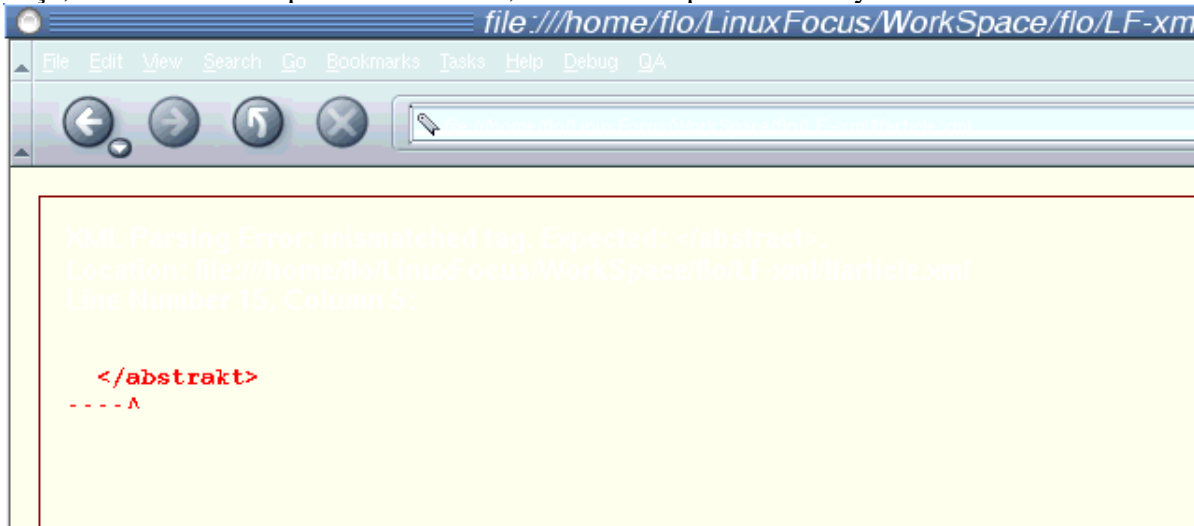
Maintenant, nous pouvons théoriquement écrire les informations de mise en page (les couleurs dans notre cas) dans un fichier séparé, nommé feuille de style. En faisant cela, nous séparons en réalité l'information de mise en page du contenu, chose considérée par certains comme le Saint Graal de la création de sites web. Jusque là, nous n'avons rien fait d'extraordinaire : les langages à balises ont été conçus pour ajouter des meta-données. Alors, qu'est-ce qui rend le XML si particulier ?

La police de la syntaxe

Tout d'abord, XML a une syntaxe très stricte. Par exemple, en XML toute <balise> doit avoir une </balise> fermante. [Note: il est un peu idiot d'écrire <tag></tag> lorsqu'il n'y a rien entre les deux, vous pouvez donc écrire <tag /> et ainsi économiser une ou deux minutes de votre vie]. Une autre règle stipule que vous ne pouvez pas mélanger les balises. Les balises doivent être fermées dans l'ordre inverse de leur ouverture. Quelque chose comme ce qui suit est impropre :

```
<B> texte en gras <I> texte en gras et en italique </B> texte en italique </I>
```

Les règles de syntaxe spécifient que vous devez fermer la balise `</I>` avant de fermer la balise ``. Sachez aussi que tous les éléments d'un document XML doivent être écrits entre des balises (sauf les deux balises externes, bien sûr !). C'est pourquoi, dans l'exemple ci-dessus, nous avons écrit les balises `<sentence>` autour de la phrase. Sans elles, certains mots de la phrase ne seraient pas inclus entre balises, et ça, comme beaucoup d'autres choses, rend folle la police de la syntaxe.



La police de la syntaxe de Mozilla au travail...

Mais une police forte a ses avantages : elle fait régner l'ordre. Puisque le XML applique des règles de syntaxe très stricte, les programmes peuvent le lire facilement. De même, les données de vos documents XML sont très structurées, ce qui les rend faciles à lire et à écrire.

De fait, la structure du XML rend même possible l'écriture de bases de données (essayez donc avec du HTML ! :p). C'est exactement ce qu'Egon Willighagen a réalisé pour l'équipe Hollandaise de LinuxFocus et son article sur le sujet est disponible à partir des liens en bas de page.

Si vous parvenez à devenir amis avec les vérificateurs de syntaxe, il y aura même des cas où vous pourrez laisser la police faire le travail à votre place. Mais pour cela, vous devrez être malins dans la manière d'utiliser une DTD...

La DTD

Dans notre exemple ci-dessus, 'Eddy, le chat meta', nous avons inventé nos propres balises XML. Bien évidemment, un acte aussi créatif est intolérable pour la police ! Les "hommes en bleu" veulent savoir ce que vous faites, comment, quand et (si possible) pourquoi. Pas de problème, vous pouvez tout expliquer grâce à la DTD...

Une DTD vous permet "d'inventer" de nouvelles balises. En réalité, elle vous permet d'inventer de véritables nouveaux langages, tant que vous respectez la syntaxe XML.

La DTD, ou **D**ocument **T**ype **D**efinition (Définition de Document Type), est un fichier qui contient la description d'un langage XML. C'est en fait, une liste de toutes les balises, de tous leurs attributs et de toutes leurs combinaisons envisageables. La DTD décrit ce qui est réalisable et ce qui ne l'est pas dans votre langage XML. Ainsi, lorsque nous parlons d'un 'langage XML', nous discutons en réalité d'une DTD spécifique.

La police au travail

Parfois, la DTD va vous *forcer* à faire quelque chose dans un endroit précis. Par exemple, la DTD peut vous contraindre à inclure une balise contenant le titre du document. Ce qui est très bien à ce sujet, c'est qu'il existe des logiciels (un module emacs, par exemple), capables d'écrire automatiquement les balises requises.

De cette manière, certaines parties de la structure de votre document sont remplies automatiquement. La syntaxe étant si stricte et bien définie, la DTD peut vous guider tout au long de l'écriture d'un document. Et lorsque vous faites des fautes, comme d'oublier de placer une balise de fin, la police est là pour vous en informer. Tout compte fait, les flics ne sont pas 'fous' du tout; si dans la vie réelle les flics disent 'Vous avez le droit de vous taire', la police du XML vous informe amicalement d'une 'erreur de syntaxe @ la ligne xx : '... :)

Et pendant que la police fait tout ce travail à votre place, vous, vous pouvez continuer et vous concentrer sur le contenu.

L'amalgame

Une des caractéristiques essentielles du XML est son aptitude à utiliser plusieurs DTD en même temps. Ca signifie que vous pouvez mêler différents types de données dans un document.

Cet 'amalgame' est effectué par les "namespaces" (zones de nommage) de xml. Par exemple, vous pouvez inclure la DTD Docbook dans votre document .xml (par le *préfixe* 'dbk' dans l'exemple qui suit).

Toutes les balises Docbook sont alors prêtes à être utilisées dans votre document sous cette forme (voici une balise docbook <une balise>):

```
<dbk:une balise> quelques mots </dbk:une balise>
```

Grâce au système de zone de nommage, vous pouvez utiliser toutes les balises et tous les attributs de n'importe quelle DTD xml. Ceci ouvre un monde de possibilités comme nous allons le voir dans le chapitre suivant...

Les DTD disponibles

Voici une petite collection de DTD déjà utilisées (au moins en partie).

- **Docbook-XML**

Docbook est un langage destiné à écrire des documents structurés, tels que des livres et des articles. Mais il est également utilisé pour beaucoup d'autres choses. Docbook est une DTD SGML, mais il en existe aussi une très populaire version XML. C'est l'une des plus répandues DTD XML.

- **MathML**

MathML est le Mathematical Markup Language (Langage Mathématique à Balises), qui est utilisé pour décrire des expressions et des formules mathématiques. C'est vraiment l'outil idéal pour le monde des maths. Les chimistes n'ont pas à être jaloux de leurs collègues matheux; pour eux, il existe CML, ou Chemical Markup Language. A noter : Mozilla 1.0 supporte MathML par défaut.

- **RDF**

RDF est le Resource Description Framework (Environnement de Description de Ressource). Il est destiné à encoder et à réutiliser des meta-données; en pratique, il est souvent utilisé sur des sites web pour informer d'autres sites des nouvelles qu'ils proposent. Par exemple, le site Hollandais linuxdot.nl.linux.org se sert du fichier RDF d'autres sites pour afficher ses propres nouvelles. Les sites de nouvelles les plus populaires (tels que Slashdot) proposent un fichier RDF vous permettant de copier leurs nouvelles dans un cadre de votre page personnelle.

- **SOAP**

SOAP signifie Simple Object Access Protocol. C'est un langage utilisé par des processus pour communiquer entre eux (échanger des données et exécuter des appels distants - RPC). Avec SOAP, les processus peuvent communiquer à distance, en utilisant par exemple le protocole http (internet). Atif, ici à LF, peut vous en dire beaucoup plus; voir les liens :-)

- **SVG**

Scalable Vector Graphics. Le trio PNG, JPEG2000 et SVG est susceptible de représenter le futur de l'imagerie sur le web. PNG remplacera GIF (moins de perte de qualité pour les bitmaps compressés avec transparence), et JPEG2000 pourrait remplacer le .jpg d'aujourd'hui (bitmaps avec un degré de perte de qualité paramétrable à la compression). SVG n'est pas un format d'image bitmap mais un format vectoriel, c'est-à-dire que les images ne sont plus représentées par des pixels mais par des formes géométriques (lignes, carrés,...). SVG possède également des fonctions de script et d'animation, pouvant ainsi être comparé d'une certaine manière au Flash de Macromedia. Vous pouvez utiliser du JavaScript dans les fichiers .svg, et grâce à lui vous pouvez écrire du code .svg. Plutôt souple, non ?

Mais svg est relativement nouveau; pour le moment il n'existe qu'un seul plugin SVG de qualité pour les navigateurs, qui nous vient d'Adobe et ne concerne que les plates-formes Windows et Mac. Mozilla travaille sur un "viewer" SVG intégré, mais il n'est pas encore terminé et vous devez télécharger une version du navigateur spécialement compilée pour la circonstance afin de pouvoir l'utiliser.

NOTE: les fichiers .svg peuvent devenir très gros et c'est pourquoi vous trouverez souvent des fichiers .svgz. Ce sont des versions compressées par l'algorithme gzip.

- **XHTML**

XHTML est la variante XML du HTML 4.01. En raison de la syntaxe stricte de XML, il y a quelques différences - certaines choses réalisables en HTML ne le sont plus en XHTML. Pourtant, une page écrite en XHTML est aussi une page HTML valide. Le programme HTML tidy est capable de convertir vos pages HTML existantes en XML.

- **Les autres**

De nombreux nouveaux formats de fichier utilisent XML, souvent en combinaison avec une compression .gz ou .zip. Un exemple : les formats de fichier de KOffice sont des DTD XML. C'est très pratique puisque ça permet à l'utilisateur de mêler les fonctionnalités de différentes applications dans un seul document. Par exemple vous pouvez écrire un document KWord en y insérant un graphe de KChart.

Liens

Le W3C, ou World Wide Web Consortium

Des informations sur XML, MathML, CML, RDF, SVG, SOAP, XHTML, namespaces...

www.w3.org

Quelques travaux de Jaime Villate (vous aurez peut-être besoin d'un logiciel de traduction en ligne pour lire les deux premiers :))

Introduction à XML(Espagnol)

Comment générer du HTML avec XML(Espagnol)

LSM-slides

HTML tidy, le programme :

www.w3.org/People/Raggett/tidy

Docbook

www.docbook.org

Projet SVG de Mozilla.org

www.mozilla.org/projects/svg

Articles de LinuxFocus en relation:

Utiliser XML et XSLT dans la construction de LinuxFocus.org(/Hollande)

Créer des documents PDF avec DocBook

<p>Site Web maintenu par l'équipe d'édition LinuxFocus © Floris Lambrechts "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Floris Lambrechts <floris/at/linuxfocus.org> en --> fr: Georges Tarbouriech <georges.t/at/linuxfocus.org></p>
--	--