Dorine MAZEYRAT

XML, DOM & XSL
Synthèse de lecture

Novembre 2008

NFE107 - Urbanisation des SI

SOMMAIRE

1. LE LANGAGE XML

1.1. Définition

XML (*Extensible Markup Language*, « langage de balisage extensible ») est un langage informatique de balisage générique. C’est un ensemble de conventions pour la conception de formats texte permettant de structurer des données. On entend par "données structurées" des éléments tels que des feuilles de calcul, des carnets d'adresses, des paramètres de configuration, des transactions financières, etc.

Le World Wide Web Consortium (W3C), promoteur de standards favorisant l'échange d'informations sur Internet, recommande la syntaxe XML pour exprimer des langages de balisages spécifiques.

1.2. Historique

Il existe deux versions de XML :

- ✓ La version 1.0, publiée le 10 février 1998,
- ✓ La version 1.1, publiée le 4 février 2004 qui apporte des améliorations dans le support des différentes versions d'Unicode.

Le W3C recommande aux processeurs XML de reconnaitre les deux versions, bien que la première version soit beaucoup plus répandue que la seconde.

1.3. Objectif

Avant l'apparition de XML, existaient :

- **SGML** (Standard Generalized Markup Language, ou langage normalisé de balisage généralisé) : un langage de balisage normalisé, riche en sémantique mais relativement lourd à mettre en œuvre et inadapté au Web.

- **HTML** (Hypertext Markup Language, ou langage de balisage hypertexte) : un langage parfaitement adapté au Web (puisque développé uniquement pour cette application) mais dont les applications sont limitées par une bibliothèque de balises figée et réduite.

Il convenait donc de définir un langage qui ait la facilité de mise en œuvre de HTML tout en offrant la richesse sémantique de SGML. C'est la raison d'être de XML.

C’est un sous-ensemble au sens strict de SGML, dont il ne retient pas les aspects trop ciblés sur certains besoins. En cela, il représente un profil d'application de la norme SGML. XML est en effet une simplification de SGML dont il retient les principes essentiels comme :

- ✓ La structure d'un document XML est définissable et validable par un schéma,
- ✓ Un document XML est entièrement transformable dans un autre document XML.

Son objectif initial est donc de faciliter l'échange automatisé de contenus entre systèmes d'informations hétérogènes (interopérabilité).

1.4. Présentation générale

```
<?xml version="1.0" encoding="UTF-8"?>
<book>
	<chapter>
		<title>Introduction</title>
	</chapter>
	<chapter>
		<title>Récit</title>
		<subChapter>
			<title>Partie 1</title>
		</subChapter>
		<subChapter>
			<title>Partie 2</title>
		</subChapter>
	</chapter>
	<chapter>
		<title>Index</title>
	</chapter>
</book>
```

1.5. Avantages

- ✓ XML bénéficie de la base installée de HTML, de HTTP et des navigateurs Internet,
- ✓ XML est simple et portable : C'est un fichier texte, donc il sera toujours lisible dans des décennies. Les commentaires sont des éléments prévus par la spécification. On peut en rajouter dans le fichier sans casser la structure. On garantit ainsi une meilleure pérennité de l'information.
- ✓ XML est plus qu'un simple langage de balise, c'est une vaste famille ! ("XML family" : technologies associées au XML),
- ✓ XML est standard : Cela signifie qu'il existe de nombreux outils informatiques qui permettent de lire ou d'écrire du XML. On trouve des librairies C, C++, Java, PHP, ... De plus en plus d'outils sont capables de lire des fichiers XML (Internet Explorer, Excel, ...).
- ✓ La mise en forme des données est totalement séparée des données elles-mêmes. Cela permet de séparer complètement l'information (le contenu) de son apparence (le contenant), et donc de fournir plusieurs types de sortie pour un même fichier de données, en fonction de l'utilisateur ou de l'application demandeuse (tableau, graphique, image, animation multimédia, fichier HTML, fichier PDF...).

1.6. Inconvénients

- ✓ Le XML est verbeux : les fichiers XML sont plus gros que des fichiers binaires ou tabulaires. Mais on peut facilement les compresser pour le stockage (avec des outils OpenSource par exemple).

2. L'INTERFACE DOM

2.1. Définition

Le Modèle Objet de Document (DOM) est une interface de programmation d'applications (API) pour documents XML (et HTML), permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour dynamiquement le contenu, la structure ou le style de documents XML.

2.2. Historique

Il y a plusieurs niveaux dans la spécification DOM :

- ✓ DOM Level 1 : elle comprend des interfaces fondamentales de bas niveau et des interfaces additionnelles pour donner une vue plus pratique d'un document HTML.
- ✓ DOM Level 2 : elle a été conçu pour HTML 4.01, XML 1.0 et les espaces de noms XML.
- ✓ DOM Level 3 : elle dispose d'extensions pour XML 1.1 et la gestion des services Web.

2.3. Présentation générale

DOM permet de construire une arborescence de la structure d'un document et de ses éléments. Il parcourt et mémorise l'intégralité du document avant de pouvoir effectuer les traitements voulus. Pour cette raison, les programmes utilisant DOM ont souvent une empreinte mémoire volumineuse en cours de traitement.

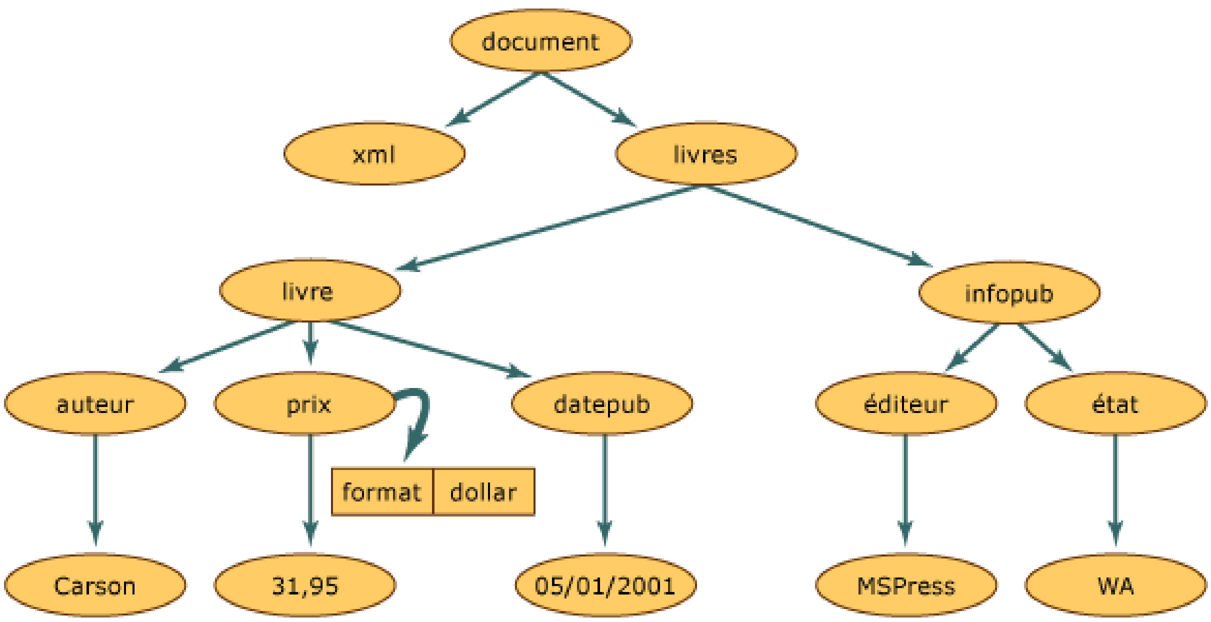
Le programmeur dispose d'objets, qui ont des propriétés, des, méthodes et des évènements qui interfacent le document XML (ou HTML).
En résumé:

- un ensemble d'objets,
- un modèle pour la façon dont ces objets peuvent être combinés,
- une interface pour y accéder les manipuler.

✓ Exemple de code XML :

```
<?xml version="1.0"?>

  <LIVRES>

    <LIVRE>
        <AUTEUR>Carson</AUTEUR>
        <PRIX format="dollar">31.95</PRIX>
        <DATEPUB>05/01/2001</DATEPUB>
    </LIVRE>

    <INFOPUB>
        <EDITEUR>MSPress</EDITEUR>
        <ETAT>WA</ETAT>
    </INFOPUB>

  </LIVRES>
```

✓ Arbre DOM associé :



Les principales interfaces fournit par DOM sur les documents XML sont les suivantes :

✓ **Interface Document**
L'interface Document représente tout le document HTML ou XML. Conceptuellement, il s'agit de la racine de l'arbre du document, et fournit l'accès principal aux données du document.

✓ **Interface Node (ici chaque cercle de cette illustration représente un nœud)**
L'interface Node est le type de donnée principal de tout le Modèle Objet de Document. Il représente un nœud unique de l'arbre du document. Bien que tous les objets

implémentant l'interface Node fournissent des méthodes pour traiter leurs enfants, tous les objets implémentant l'interface Node n'ont pas forcement d'enfants.

- ✓ **Interface Attr (ici : format)**

L'interface Attr représente un attribut d'un objet de type Element. Typiquement, les valeurs autorisées de l'attribut sont spécifiées dans une DTD (Définition de Type de Document).
Les objets Attr héritent de l'interface NODE, mais comme ils ne sont pas vraiment des noeuds enfants de l'élément qu'ils décrivent, le Modèle Objet de Document DOM ne les considère pas comme partie intégrante de l'arbre.

- ✓ **Interface Elément (ici : livres, livre, auteur, infopub, …)**

Les objets les plus couramment rencontrés par les utilisateurs parcourant un document (mis à part le texte lui-même) sont de loin les nœuds d'éléments (Element).

A chacune de ces interfaces est associé un ensemble de méthodes/propriétés permettant de modifier, lire, traiter toutes les données ainsi que les nœuds.

2.4. DOM et SAX

DOM et SAX sont deux moyens de parser (= analyser la syntaxe) un document XML et en utiliser le contenu.

DOM est le plus simple, le plus intuitif. Il charge le document en mémoire sous forme d'arborescence et permet au programmeur d'appliquer des fonctions sur les éléments de l'arbre.

Sax est plus rapide et consomme moins de mémoire. Il est orienté évènements. Il associe des méthodes aux balises, elles sont activées quand les balises sont atteintes lors de la lecture. Les éléments sont lus en séquence, une seule fois. Il faut fournir son propre modèle de document, alors qu'il en est fourni un avec DOM.
Dans les cas ne nécessitant pas de manipuler les documents XML, mais juste de les lire, la méthode SAX peut également être choisie car elle traite les éléments de façon successive sans charger le document en mémoire. Elle s'impose donc quand la taille du document excède la capacité de la mémoire.

3. LE LANGAGE XSL

3.1. Définition

XSL (eXtensible Stylesheet Language) est le langage de description de feuilles de style du W3C associé à XML pour mettre en forme des données au même titre que les CSS (Cascading StyleSheets) pour le langage HTML.

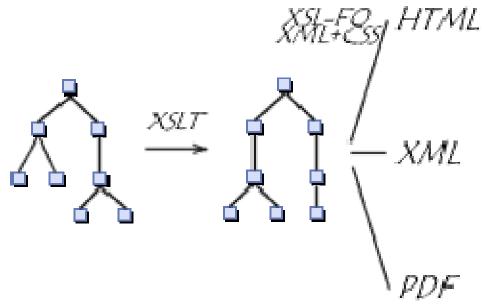
3.2. Présentation générale

Une feuille de style XSL est donc un fichier qui décrit comment doivent être présentés (c'est-à-dire affichés) les documents XML basés sur une même DTD (Définition de Type de Document : document permettant de décrire un modèle de document XML) ou un même schéma.

Toutefois, contrairement aux CSS, XSL permet aussi de retraiter un document XML afin d'en modifier totalement sa structure, ce qui permet à partir d'un document XML d'être capable de générer d'autres types de documents (PDF, HTML, ...) ou bien un fichier XML de structure différente.

Ainsi la structuration des données (définie par XML) et leur représentation (définie par un langage tel que XSL) sont séparées. Cela signifie qu'il est possible à partir d'un document XML de créer des documents utilisant différentes représentations (HTML pour créer des pages web, WML pour les mobiles WAP, ...).

XSL possède deux composantes :



- ✓ **Le langage de transformation des données** (XSLT) permettant de transformer la structure des éléments XML.
- ✓ **Le vocabulaire de mise en forme des données** (XSL/FO), c'est-à-dire un langage permettant de définir la mise en page (affichage de texte ou de graphiques) de ce qui a été créé par XSLT.

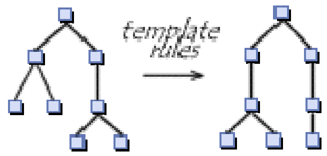
3.3. XSLT

3.3.1. Définition

XSLT (*eXtensible Stylesheet Language Transformations*), défini au sein de la recommandation XSL du W3C, est un langage de transformation XML de type fonctionnel.

3.3.2. Présentation générale

Le processeur XSLT (composant logiciel chargé de la transformation) crée un structure logique arborescente (on parle d'arbre source comme vue avec DOM) à partir du document XML et lui fait subir des transformations selon les « template rules » contenues dans la feuille XSL pour produire un arbre résultat représentant, par exemple, la structure d'un document HTML ou XSL-FO. Les composants de l'arbre résultat sont appelés objets de flux. Chaque « template rule » définit des traitements à effectuer sur un élément (nœud ou élément) de l'arbre source.



L'arbre source peut être entièrement remodelé et filtré, du contenu peut également être ajouté à l'arbre résultat, si bien que l'arbre résultat peut être radicalement différent de l'arbre source.

Cependant, le langage XSLT permet aussi les transformations vers tout autre type de document, au format texte ou dans un format binaire.

XSLT s'appuie sur XPath pour désigner une partie d'un arbre XML. XSLT est lui-même un dialecte XML. Un programme XSLT est donc, avant tout, un document XML :

```
<?xml version="1.0" ?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL" version="2.0">
<xsl:output method="xml" indent="yes"/>

<xsl:template match="person">
    <name username="{@username}">
       <xsl:value-of select="name" />
    </name>
</xsl:template>

</xsl:stylesheet>
```

L'une de ses principales particularités est d'être centrée sur les données. Un XSLT doit s'appuyer sur un XML, c'est un couple obligatoire, et on ne peut créer en XSLT que des boucles parcourant des données sélectionnées par XPath.

3.3.3. XPath

XPath est une syntaxe (non XML) pour désigner une portion d'un document XML. Initialement créé pour fournir une syntaxe et une sémantique aux fonctions communes à

XPointer et XSL, XPath a rapidement été adopté par les développeurs comme un petit langage d'interrogation.

```
<?xml version="1.0"?>

 <racine>
  <encyclopedie nom="Wikipedia" site="http://fr.wikipedia.org/">
   <article nom="XPath"></article>
  </encyclopedie>
 </racine>
```

Voici les expressions XPath suivantes qui peuvent être utilisées avec le document XML précédent :

Expression XPath	**Résultat**
/	sélectionne un nœud "fictif", dit *root element*, qui englobe tout le document sauf <?xml version="1.0"?>
/root	sélectionne le nœud vide, puisqu'il n'y a pas d'élément "root" (mais "racine")
//article	sélectionne tous les éléments "article" du document où qu'ils soient
/racine/encyclopedie	sélectionne l'unique élément "encyclopedie" puisqu'il est ici le seul fils de "racine" portant ce nom
//article[@nom='XPath']	sélectionne tous les éléments "article" du document où qu'ils soient, ayant un attribut "nom" dont la valeur est "XPath"

XPath est le langage de description des nœuds dans XSLT. Il est aussi utilisé comme langage de requêtes dans les bases de données XML.

Son principal concurrent est XQuery.

3.4. XSL-FO

3.4.1. Définition

XSL-FO est un langage de description de mise en page de documents destiné principalement à l’impression (ex : type PDF).

XSL-FO est un langage d'une haute technicité qui s'adresse principalement aux typographes afin de fournir avec les outils de gestion de documents, un outil typographique du niveau attendu par les publications imprimées.

3.4.2. Présentation générale

XSL-FO est aussi un dialecte XML. Ce jeu de balises XML permet de faire une mise en page par zones (Mise en page, corps du document, ...).

- ✓ Exemple de code XSL-FO :

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" >

<!-- Déclaration de la mise en page -->
<fo:layout-master-set>
   <fo:simple-page-master master-name="ma-page" margin="2cm">
      <fo:region-body />
   </fo:simple-page-master>
</fo:layout-master-set>

<!-- Corps du document -->
<fo:page-sequence master-reference="ma-page">
   <fo:flow flow-name="xsl-region-body">
      <fo:block>XSL-FO, c'est simple</fo:block>
   </fo:flow>
</fo:page-sequence>

</fo:root>
```

XSL-FO est un format très adéquat pour générer du contenu destiné à l'impression. Mais on se rend vite compte que d'écrire dans ce format directement est assez lourd. Ainsi ce format est souvent utiliser comme format de sortie généré à partir de documents écrits selon la syntaxe XML via les transformations XSL (XSLT).

L'outil principal pour traiter des fichiers XSL-FO (.fo) est FOP, un logiciel en Java qui permet de générer des fichiers PDF mais également PS, TXT...

4. LES APPLICATIONS DES TECHNOLOGIES XML

A titre d'exemples, nous pouvons citer les champs d'application suivants :

- ✓ Messageries : XML en tant que format standard pour l'échange de données,
- ✓ Traitement de données : déplacement du serveur vers le client (ex : Commerce/Echange électronique),
- ✓ Gestion documentaire : XML permet d'exprimer toute l'intelligence du document dans le document, les applications documentaires peuvent se développer sans contrainte (ex : base de données XML),

- ✓ Collaboratif : les Intranets bénéficieront largement des documents XML. Avec HTML, la même information était préparée pour tout le monde, maintenant le choix de la restitution peut revenir au client. L'auteur décide du contenu, le lecteur de la présentation,

- ✓ Publication multi-support de documents : présentation variable de l'information ; publication dans différents formats avec XSL ; publication automatisée (par exemple à partir des bases de données),

- ✓ Intégration de système : EAI, etc.

Aujourd'hui, un document XML permet donc d'exploiter des textes au kilomètre, mais structurés par un balisage puissant et riche comme celui d'une infrastructure relationnelle. Mais l'univers XML s'est aussi enrichi de mécanismes assurant la confidentialité et l'intégrité des données, ainsi que l'authentification des services et des utilisateurs.

XML s’est donc imposé sur le plan fonctionnel, mais aussi technique. A ce niveau, il joue le rôle essentiel de format pivot entre des systèmes hétérogènes, assurant l'interopérabilité là où elle n'existait pas.

SOURCES BIBLIOGRAPHIQUES

- ✓ Comment ça marche :
  - http://www.commentcamarche.net/contents/xml/xmlxsl.php3

- ✓ W3C :
  - http://www.w3.org/XML/1999/XML-in-10-points.fr.html

- ✓ Wikipédia :
  - http://fr.wikipedia.org/wiki/Xml
  - http://fr.wikipedia.org/wiki/XSL
  - http://fr.wikipedia.org/wiki/XSLT
  - http://fr.wikipedia.org/wiki/Document_Object_Model

- ✓ XMLFR.org :
  - http://xmlfr.org/documentations/articles/000321-0001
  - http://xmlfr.org/w3c/TR/REC-DOM-Level-1/introduction.html

- ✓ 01net :
  - http://www.01net.com/article/242457_a.html