Le langage XML – Introduction

(voir également les transparents de cours)

A. Bouras

Le Web a engendré un échange de documents électroniques à une échelle sans précédent et a atteint tous les publics, tant professionnels que particuliers.

XML (*Extensible Markup Language*, ou Langage Extensible de Balisage) est le langage destiné à succéder à HTML sur le World Wide Web. Comme HTML, c'est un langage de balisage (*markup*), c'est-à-dire un langage qui présente de l'information encadrée par des *balises*. Mais contrairement à HTML, qui présente un jeu limité de balises orientées présentation (titre, paragraphe, image, lien hypertexte, etc.), XML est un *métalangage*, qui va permettre d'inventer à volonté de nouvelles balises pour isoler toutes les informations élémentaires (titre d'ouvrage, prix d'article, numéro de sécurité sociale, référence de pièce…), ou agrégats d'informations élémentaires, que peut contenir une page Web.

XML est un langage universel d'échange de données particulièrement performant : il est simple et peut être véhiculé grâce à des protocoles standards de transport Web comme HTTP (HyperText Transfer Protocol). Comme tous les composants d'un document XML sont délimités et explicités grâce à des balises, chacun d'entre eux vient avec sa propre description, qui peut être structurée sous forme de métadonnées (c'est-à-dire de données sur les composants). N'importe quelle application peut interpréter un document XML en utilisant un interpréteur (ou parseur) et retirer l'information qu'il contient. En donnant une sémantique au contenu Web, XML remédie à de nombreuses insuffisances d'HTML, tout en conservant la simplicité qui a fait son succès.

Historique de XML :

- **La décennie 80 : SGML et les aides en ligne**

SGML (*Standard Generalized Markup Language*, ou langage normalisé de balisage généralisé), adopté comme standard en 1986 (ISO 8879), a été la première tentative systématique de créer de réels documents électroniques, c'est-à-dire des documents qui n'étaient plus des documents papier sous forme électronique. La principale idée sous-jacente étant de séparer le *contenu* (logique) d'un document de sa *forme* (matérielle/imprimée). Mais l'intention finale était toujours *principalement* de produire des documents imprimés, quoique plus économiquement — un unique document (logique) étant transformé automatiquement en différents formats imprimés. SGML a été une percée, mais il était si complexe que sa manipulation s'est trouvée restreinte aux spécialistes.

Presque parallèle a été le développement de la documentation en ligne, interactive, la première forme de documentation à être purement électronique. Et avec elle est arrivée la popularisation des liens hypertextuels. Mais cette forme de documentation restait une "aide", accessoire à la documentation papier.

- **Les années 90 : Le WWW et HTML**

A partir de 1992, le WWW et HTML (langage de balisage hypertexte, conçu vers 1990) sont devenus une réalité, et ont popularisé les documents électroniques hypertextuels sur une immense échelle. Depuis 1995, les moteurs de recherche ont démontré la stupéfiante capacité de recherche d'information rendue possible par le WWW.

Ainsi, avant l'apparition de XML, existaient :

- un langage de balisage normalisé, riche en sémantique mais relativement lourd à mettre en oeuvre et inadapté au Web : **SGML**
- 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 : **HTML**

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

XML 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.

Ainsi, XML répond parfaitement aux nouveaux besoins du Web comme la gestion de contenu, l'interopérabilité des systèmes d'information hétérogènes, le commerce électronique et la personnalisation de la relation client.

Définition du langage XML :

XML signifie *eXtensible Markup Language,* c'est-à-dire langage de balisage extensible. XML propose en effet de décrire et de structurer les documents à l'aide de balises (*tags*). XML est cependant plus qu'un simple langage de balisage : c'est un langage utilisable pour créer d'autres langages. Il est donc extensible, à la différence d'HTML qui définit de façon rigide comment décrire la présentation de certains types de documents.

L'idée centrale d'XML est qu'il apporte de la valeur ajoutée si les trois aspects fondamentaux d'un document que sont son contenu, sa structure et sa présentation sont séparés pour celui qui le rédige comme pour la machine qui va l'interpréter et le délivrer à différents lecteurs.

Un document HTML mélange le contenu et la présentation. En ce qui concerne la structure, le concept est à peine présent ou en tout cas, elle est difficile à modifier.

L'approche XML propose de rédiger un document en adoptant un paradigme très différent. Vous créez un document en vous focalisant sur l'information qu'il contient et sur la façon dont elle est structurée ; les aspects de présentation sont quant à eux traités séparément.

XML permet donc de différencier le contenu et la structure d'un document de sa présentation. Pour résonner par analogie, XML est un langage permettant de décrire et de structurer des mots (les informations) grâce à une syntaxe (les balises), elle-même susceptible d'être adaptée et enrichie lorsqu'elle est utilisée par différentes communautés ou industries.

La philosophie d'XML consiste à bien séparer les données/documents (le fichier XML proprement dit) des traitements/présentations. Un document donné sera, lors de sa création, balisé *uniquement* en fonction de son contenu (sa sémantique) intrinsèque et *indépendamment*

de sa restitution future (papier, écran, terminal Braille, synthèse vocale ou autre) — comme d'ailleurs de tout autre traitement automatique qui pourra lui être appliqué.

Cette indépendance par rapport aux applications qui vont le traiter en général, et par rapport à celles chargées de sa restitution en particulier, lui confère :

- une très grande *interopérabilité* (le *même* document XML va pouvoir être affiché sur le Web et/ou produit en version papier, alimenter un SGBD, etc.)
- et une très grande *durabilité/réutilisabilité* (le document ne deviendra pas obsolète avec l'évolution des techniques informatiques ; il pourra sans difficulté être incorporé, en tout ou partie, dans des documents de nature très différente, être traité par des applications non prévues, voire non-existantes au départ...).

XML est un langage plus riche qu'HTML, non son remplaçant : XML permet de décrire les données et HTML permet de décrire leur présentation.

XML est un outil doué d'ubiquité car il se montre à l'aise aussi bien sur des documents peu structurés comme les écrits (présentations, articles...) que sur des documents très structurés comme des bases de données.

XML est extensible car l'ajout de nouvelles balises est à tout moment possible pour prendre en compte un nouvel élément d'information dans un document.

Les règles du jeu XML

Elles sont extrêmement simples.

- Les informations doivent être :
  - soit encadrées par des balises ouvrantes(ex. <LIVRE>) *et* fermantes (ex. </LIVRE>) — contrairement à HTML où ses ces dernières n'étaient pas toujours obligatoires). On parle alors d'*éléments*. Les éléments doivent s'imbriquer proprement les uns dans les autres : aucun chevauchement n'est autorisé. Les éléments vides sont permis, selon le format <ELEMENTVIDE/>.
  - soit incluses à l'intérieur même des balises : on parle alors d'*attributs*. Exemple: <LIVRE SUJET="XML">. Ici l'attribut *SUJET* de l'élément *LIVRE* a la valeur "XML". En XML, contrairemet à HTML, les valeurs des entités doivent toujours être encadrées par des guillemets (simples ou doubles).
  - soit encore définies sous forme d'*entités*. Les entités sont des abréviations. Par ex; si "Extensible Markup Language" est déclaré comme une entité associée à la notation "xml"; cette chaîne de caractères pourra être abrégée en "&xml;" dans tout le fichier XML. Une entité peut aussi représenter un fichier XML externe tout entier. Ainsi un même fichier XML (par exemple un fichier bibliographie) pourra être utilisé par plusieurs pages XML différentes.

La définition de types de données

Il existe deux façons pour définir des types de données :

- A l'aide d'une DTD (Définition de type de document)
- A l'aide d'un schéma

La DTD (Définition de Type de Document)

La structure arborescente du document XML (intitulé des balises, imbrications des balises, caractère obligatoire ou facultatif des balises et de leur ordre de succession…) peut être déclarée formellement dans le corps du document XML ou dans un fichier à part. Cette déclaration s'appelle une Définition de Type de Document (DTD). Elle s'effectue selon un formalisme particulier défini lui-aussi dans la spécification XML. En XML cette déclaration est *facultative*, ce qui donne une grande souplesse aux développeurs. On n'écrira donc une DTD que lorsqu'il y aura vraiment intérêt à le faire (par exemple pour contraindre la saisie/mise à jour du document XML)

Lorsqu'un document XML possède une DTD associée et la respecte, on dit qu'il est *valide*. Lorsqu'il respecte seulement les règles de la grammaire XML (balises fermées, correctement imbriquées…) on dit qu'il est *bien formé*.

La DTD est écrite dans un langage qui lui est propre, non conforme aux règles XML.

Beaucoup applications XML ont été définies à l'aide de DTD, mais cela tend à changer puisque les DTD sont remplacées par le langage de XML Schema. Ceci est dû au fait que les DTDs ont de nombreuses limitations :

- Trop contraignantes (tout doit être défini)
- Pas de contraintes sur le contenu (date, par exemple)
- Syntaxe différente de la syntaxe de balisage

Schéma XML

L'extensibilité de XML reposait sur la définition de nouvelles balises regroupées dans les DTD (Definition type document) des documents. Difficiles à maîtriser, ces dernières présentaient des limites quant à l'emploi de XML comme vecteur de données. Le consortium W3C a standardisé une nouvelle façon de décrire les données en XML : des schémas qui autorisent l'emploi de vrais types de données, eux-mêmes définis en XML, et sur lesquels il est possible de fixer des contraintes. La définition des schémas XML fait l'objet d'un large consensus au sein de l'industrie informatique.

Les schémas offrent plusieurs avantages par rapport aux DTD :

- Ils utilisent une syntaxe XML et peuvent ainsi être produits et modifiés à l'aide d'outils standards XML.
- Ils possèdent un système de types de données plus détaillé que la DTD.
- Les utilisateurs peuvent définir leurs propres types de données, y compris des types structurés.

- Les schémas XML sont plus modulaires, plus lisibles et plus faciles à réutiliser.
- Ils sont orientés objets et permettent de faire de l'héritage et du polymorphisme.

Avec les schémas, XML devient plus facile à employer pour les échanges de données interentreprises.

Le Schéma XML est un formalisme qui doit permettre de définir des contraintes en matière de syntaxe, de structure et de valeurs applicables à une classe de documents HTML. Il va permettre entre autres choses d'effectuer des contrôles de validité lors de la saisie/mise à jour de documents XML (exactement comme pour la saisie/mise à jour d'une bases de données).

Les feuilles de styles XSL

XSL est un langage qui permet d'exprimer des feuilles de styles ; il consiste en deux parties :

- un vocabulaire de mise en forme de documents.
- un langage de transformation de documents.

Une feuille de styles XSL permet de spécifier la présentation d'un document XML source en décrivant comment il peut être transformé en un document XML résultat qui puisse utiliser un vocabulaire de mise en forme.

Une feuille de styles XSL est constituée d'un ensemble de règles de transformation qui spécifient comment, grâce à certaines actions, un document XML source doit être converti en un document résultat mis en forme.

Séparer le style d'un document de sa structure/contenu

- Sans modifications, un même document peut être présenté de différentes façons :
  - Selon le lecteur
  - Selon les caractéristiques de l'appareil de restitution
- Un ensemble de documents peuvent être présentés de façon homogène

Avantages : maintenabilité des sites Web, indépendance des plates-formes, performances

Le langage normalisé de feuille de style XSL (*Extensible Style Language*) va permettre ensuite de spécifier comment un type de document (= une DTD "orientée restitution") donné va être restitué sur un support donné. C'est à ce niveau que seront réglés les problèmes du type "saut de page", notes présentées en bas de page ou en fin de chapitre, etc., que les liens de navigation seront fabriqués (hyperliens pour les versions électroniques, renvoi à un n° de page ou de paragraphe ou de note pour les versions papier...) (Voir la spécification à http://www.w3.org/TR/WD-xsl)

Une feuille de style XSL est appelée à partir d'un document XML par une "processing instruction" (PI) selon l'exemple suivant :

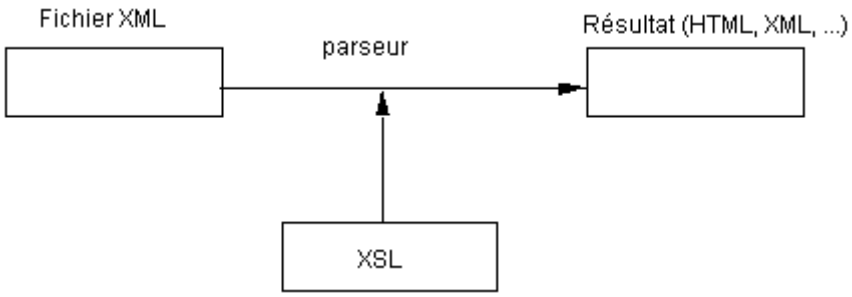
```
<?xml-stylesheet href="biblio.xsl" type="text/xsl" ?>
```

Le langage normalisé de feuille de style CSS (Cascading Style Sheets) déjà utilisé avec HTML, pourra également être utilisé concurremment où à la place de XSL.

Une feuille de style CSS est appelée à partir d'un document XML par une "processing instruction" (PI) selon l'exemple suivant :

```
<?xml-stylesheet href="biblio.css" type="text/css" ?>
```

C'est une technologie très jeune (Avril 1999 draft W3C).
En fait XSL est beaucoup plus qu'un simple langage pour le formattage de documents XML (ce que pourrait très bien faire CSS Cascading Style Sheets). Il permet de retraiter un document XML et ainsi de réarranger sa structure. En fait XSL permet de transformer un document XML et un autre document souvent XML mais pas forcément par exemple en HTML, TeX, RTF, PostScript, etc.

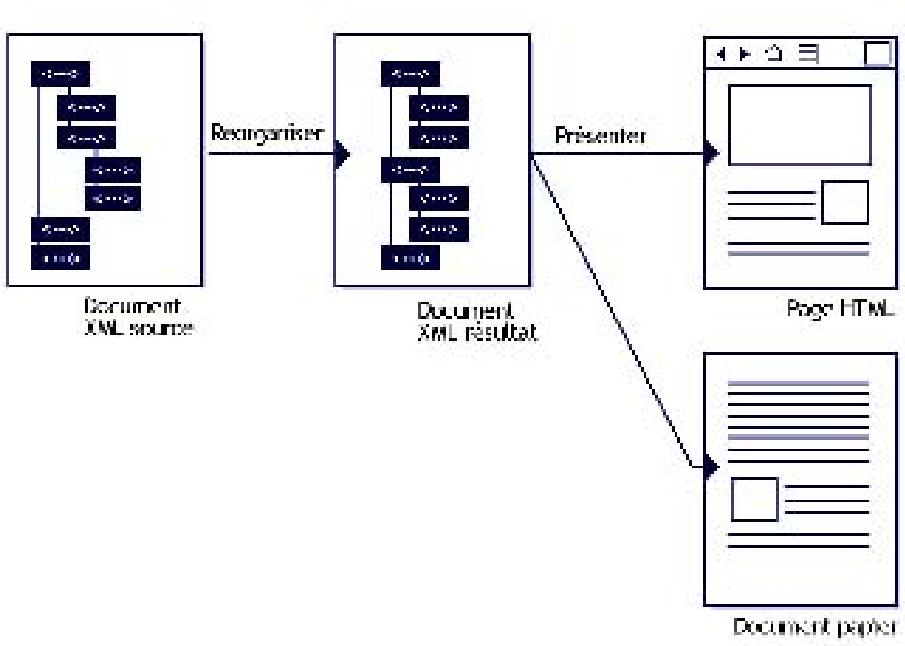


Un fichier XSL est constitué d'une suite de règles à appliquer sur un fichier XML. Ces règles sont appelées template. Elles comportent un élément susceptible d'être rencontré dans un fichier XML avec sa traduction associé.

Les objets constituant le document XML source utilisés par une feuille de styles XSL forment un arbre.

La présentation d'un document XML grâce à XSL se fait en deux étapes, comme l'illustre la figure suivante :

- la construction d'un arbre résultat à partir d'un arbre source,
- l'interprétation de l'arbre résultat pour produire une sortie mise en forme sur un écran, du papier, ou tout autre média.

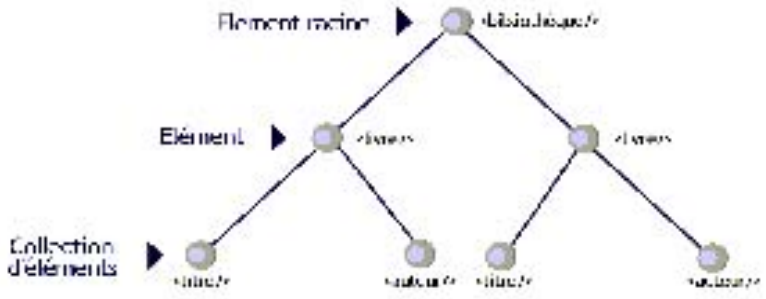


Présentation d'un document XML en deux étapes

Analyse des documents XML

DOM (Document objet Model)

Le DOM spécifie un jeu d'interfaces de programmation pour les documents XML et HTML. Il définit la structure logique d'un document ainsi que la façon dont une application peut y avoir accès et le manipuler. Il présente un document comme une hiérarchie de nœuds (*nodes*) qui implémentent d'autres interfaces plus spécialisées.



Représentation d'une bibliothèque grâce au DOM

Grâce au DOM, des développeurs peuvent construire des documents, parcourir leur structure, ajouter, modifier ou détruire certains de leurs éléments et de leur contenu.

Le nom *Object Model* a été choisi au sens de la conception orientée objet : les documents sont modelés à partir d'objets et le modèle couvre non seulement la structure d'un document, mais aussi le comportement de ce document et des objets qui le composent. En d'autres termes, les

nœuds du diagramme ci-dessus ne représentent pas une structure de données mais des objets, qui ont une identité et des fonctions. Comme modèle objet, le DOM identifie :

- les interfaces et les objets utilisés pour représenter et manipuler un document ;
- la sémantique de ces interfaces et objets (incluant à la fois des attributs et des méthodes) ;
- les relations et les collaborations entre ces interfaces et objets.

La structure des documents SGML a traditionnellement été représentée par un modèle de données abstrait, non par un modèle objet. Un modèle de données abstrait est centré autour des données. Dans les langages de programmation orientés objet, les données elles-mêmes sont encapsulées et " masquées " dans des objets ; elles sont ainsi protégées des manipulations externes directes. Les fonctions associées à ces objets déterminent comment ces derniers peuvent être manipulés et font partie intégrante du modèle.

Un des objectifs fondamentaux du DOM était de fournir une interface de programmation standard qui pouvait être utilisée dans différents environnements par une grande variété d'applications. Le DOM a été conçu pour être utilisé par tout langage de programmation. Pour cela, les concepteurs du DOM ont choisi de l'exprimer dans l'IDL (Interface Definition Language) de l'OMG (Object Management Group). En plus de la spécification IDL de l'OMG, le DOM fournit des *bindings* pour l'ECMAScript (un standard de l'industrie basé sur JavaScript et Jscript) et Java.

Le DOM prend racine comme une spécification pour permettre à des scripts écrits en JavaScript et des programmes écrits en Java d'être portables sur tous les navigateurs Web.
