Groupe *e*Fa

Introduction à XML

Par

Joel Amoussou

Consultant XML

jamoussou@efagroup.com

(514) 341-7836

Objectifs

- Concepts de base de XML
- Positionnement de XML par rapport à DHTML et SGML
- Aperçu du Langage Extensible de Feuille de Style (XSL), du Langage Extensible de Liens Hypertextes (XLink) et du Langage Extensible d'adressage XPointer

Concepts de Base

- XML: Langage Extensible de Balisage de Documents
- XML 1.0 (10 Février 1998) est une Recommandation du W3C
- Langage de description de la structure logique des documents, adapté aux applications web
- Syntaxe simplifié: 26 pages vs 500 pour SGML

XML vs HTML

- Documents HTML:
  - Balises prédéfinies decrivant le formatage des donnees sur le browser (<a>, <b>, <I>, < p>, etc.)
  - Java Script
  - Programmes CGI (Common Gateway Interface) ou ASP (Active Server Pages)
- Inconvénient: pauvreté sémantique de HTML
- Inadapté à des applications complexes du type EDI comme X12, EDIFACT et HL7

La Notion de Document

- Le document électronique aujourd'hui:
  - Tables de base de données relationnelle
  - Bon de commande (EDI)
  - Formulaire électronique
  - Manuel de procédures administratives
  - etc.
- XML permet de décrire la structure et le contenu de tout type de document

Structure Logique et Physique du Document

- Structure hiérarchique et imbrication des éléments structurels
- Relation et dépendance entre les éléments structurels
- Exemple: un livre contient une introduction suivi d'un ou plusieurs chapitres pouvant contenir à leur tour des sous-chapitres
- XML décrit:
  - la structure logique du document et des contraintes sur la structure du documents
  - la structure physique du document (les entités du documents)

Balisage Procédural

- Balisage procédural:
  - codes de formatage (gras, italiques) des traitements de texte traditionnels
  - Codes mélangés au contenu
  - spécifique à un logiciel et à une version du logiciel
- Échange difficile entre applications hétérogènes

XML utilise le balisage descriptif

- Décrit la sémantique du contenu
- Basé sur la structure hiérarchique du document
- Sépare le contenu des instructions de traitement (y compris le formatage)
- Permet validation et navigation de la structure du document

Le Balisage en XML

- <GI>contenu</GI>
- Balise d'ouverture: <GI>
- Balise de fermeture: </GI>
- GI = identificateur générique: identifie le type d'information entre la paire de balise
- Balise d'ouverture peut contenir des attributs qui qualifient l'élément

<prénom sexe="masculin">Dominique</prénom>

XML: Pierre Rosetta du e-commerce

- Découverte en 1799, la Pierre Rosetta sur laquelle étaient inscrites trois versions d'un même texte (hiéroglyphe, démotique et grecque) a permis de percer le secret de la signification des hiéroglyphes et partant de l'histoire de l'Egypte ancienne
- XML= langage neutre, ouvert et universel de description de données sur le web
- XML aidera l'interopérabilité des applications

Langage Ouvert et Extensible

- XML utilise le jeux de caractère Unicode (ISO 10646)
- XML est un méta-langage qui permet d'inventer des jeux de balises et les règles syntaxiques d'utilisation de ces balises
- Support des grands éditeurs de logiciels

XML et le DHTML

- Les balises HTML ont été conçues pour l'affichage des données et non pour leur traitement
- XML = langage de description de données
- Le Dynamique HTML (DHTML) restera le langage de définition d'interface
- XML sera transformé en HTML sur le browser avec XSL
- Îlots XML dans les documents HTML

Composantes d'une application XML

- Le document ou instance XML comprenant:
  - le prologue pouvant pointer vers une définition de type de document (DTD) (optionnel)
  - le texte balisé selon la structure définie par la DTD
- Une feuille de style XSL qui transforme le document XML en HTML pour l'affichage sur le browser web

Groupe *e*Fa

Bon de Commande au Format XML

```
<?xml version=“1.0” encoding=“UTF-8” standalone=“no”?>
<?xml-stylesheet href=“bon-commande.xsl” type=“text/xsl”?>
<!DOCTYPE  bon-commande SYSTEM “dtds/bon-commande.dtd”>
<bon-commande xmlns:edi=“http://ata-eco.org/spec” no-commande="172563">
      <date-commande dt::type=“dateTime”>1998-04-07T18:39:09</date-commande>
      <fournisseur nom=“Aero Hardware Inc.” cage=“83639”/>
      <acheteur nom=“Air Québec Inc.” cage=“83759”/>
      <livreur nom="Québec Express Inc." cage="64549"/>
      <commande item-no=“001”>
            <no-pièce> NAS1104-10D</no-pièce>
            <description>Verrou</description>
            <quantité dt:type="number">1</quantité>
            <date-livraison dt:type=“dateTime”>1998-04-07T21:30:00</date-livraison>
      </commande>
</bon-commande>
```

Déclaration XML

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

- version="1.0": version de la spécification XML utilisée
- encoding="UTF-8": le jeu de caractère utilisé (ASCII est un sous-ensemble de UTF-8)
- standalone="no": déclaration standalone indiquant la présence de déclarations externes (dans le fichier contenant la définition de type de document) nécessaires au traitement du document XML

La Référence à la feuille de style

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

- La feuille de style XSL est référencée par le URL en utilisant l'attribut href
- Un processeur XSL sur le client ou le serveur reçoit le document XML et la feuille de style XSL en entrée et génère du HTML pour l'affichage sur le fureteur web

La déclaration de type de document

<!DOCTYPE bon-commande SYSTEM “dtds/bon-commnde.dtd”>

- bon-commande: nom de la DTD et de l'élément racine du document XML
- SYSTEM: nom réservé
- “dtds/bon-commande.dtd”: identificateur système indiquant le Uniform Resource Locator (URL) de la DTD du bon de commande

La Déclaration du Namespace

<bon-commande xmlns:edi="http://ata-eco.org/spec" no-commande="172563">

- xmlns:edi: définition du préfix contenant le préfix du namespace (edi) utilisé pour associer les éléments à ce namespace dans l'envergure (descendance) de l'élément bon-commande
- "http://ata-eco.org/spec": nom du namespace référençant le schema (dictionnaire) qui définit l'élément
- Utilisé pour éviter les conflits de noms et mélanger les vocabulaires

Déclaration de Type de Données

```
<date-commande dt::type="dateTime">1998-04-07T18:39:09</date-commande>
```

- L'attribut dt::type="dateTime" attaché a l'élément date-commande indique que la date doit être spécifiée selon le format ISO8601

La Syntaxe d'une DTD

- Décrit rigoureusement la structure d'un document spécifique à l'aide des déclarations suivantes:
  - Éléments
  - Attributs
  - Entités générales et de paramètres
  - Commentaires
  - Instructions de traitement

Déclaration des Éléments

<!ELEMENT elm (el1, el2, …,elj)>

- ELEMENT: mot réservé de XML
- elm: élément que l'on définit
- (el1, el2, …,elj): modèle de contenu de l'élément déclaré. elm composé de j éléments
- Exemple:

<!ELEMENT nom (prénom, nom_de_famille)>

Les Symboles d'Occurrence

- + Élément obligatoire, éventuellement répétitif (1 ou plus)
- ? Élément optionnel qui ne peut être répétitif
- * Élément optionnel, éventuellement répétitif (0 ou plus)
- Pas de signe: élément obligatoire, apparaît une seule fois

Les Connecteurs

- (el1, el2, …,elj)>: les j éléments doivent obligatoirement être présents dans l’élément décrit et dans l’ordre indiqué
- (el1 | el2 | …| elj)>: un et un seul des j éléments doit être présent dans l’élément décrit

Exemple

DTD

```
<!ELEMENT   commande  ( no-pièce+, decription*, quantité, date-livraision?)>
```

Document XML conforme à la DTD ci-dessus

```
<commande>
        <no-pièce> NAS1104-10D</no-pièce>
        <description>Verrou</description>
        <quantité>1</quantité>
        <date-livraison>1998-04-07T21:30:00</date-livraison>
</commande>
```

Type de Contenu des Éléments

- #PCDATA (parsed character data): mot réservé qui signifie que les éléments et les entités sont reconnus
- ANY: mot réservé qui signifie que l'élément peut contenir tout autre élément définit dans la DTD
- Éléments à contenu mixte. Exemple:

```
<!ELEMENT  titre  (#PCDATA | nip)*>
```

Élément à Contenu Vide

- Syntaxe de déclaration d'un élément vide:

  <!ELEMENT elem_vide EMPTY>

- Le balisage se fait selon la syntaxe:

  <élément_vide/>

- Exemple:
  - Dans la DTD, on déclare:

    <!ELEMENT logo EMPTY>

  - Dans l'instance: **<logo/>**

Caractères de Balisage

- Les caractères & et < sont des caractères de balisage et doivent être remplacés par les entités & et < respectivement
- Balisage incorrect:

<titre>Le budget de R&D</titre>

- Balisage correct:

<titre>Le budget de R&D</titre>

Les Attributs

<!ATTLIST elem attribut1 type1 valeur_par_defaut1
attribut2 type2 valeur_par_defaut2
attributn typen valeur_par_defautn>

- ATTLIST: mot réservé XML annonçant la définition d'une liste d'attributs.
- Elem: élément dont on définit les attributs.
- attribut-k: nom du k-ième attribut déclaré.
- type-k: nature du contenu du k-ième attribut.
- La valeur de l'attribut doit toujours être entre quottes

Les Types d'Attributs

- NMTOKEN(S): chaîne de caractères ne commençant pas nécessairement par une lettre
- ENTITY(IES): Un nom (ou des noms) d'entité (s) générale(s)
- CDATA: Une chaîne de caractères
- ID: Identificateur (un nom unique à un élément)
- ID ou IDREFs: Égale à la valeur de l'attribut ID d'un autre élément du document
- Liste de noms: valeurs explicitement énumérées

Les Valeurs par Défaut

- #REQUIRED: la valeur de l'attribut doit obligatoirement être indiquée
- #IMPLIED: le système définira une valeur pour l'attribut si aucune valeur n'est indiquée
- #FIXED: l'attribut à une valeur fixe qui est unique
- "value": s'il n'est pas spécifié, l'attribut prend une valeur donnée

Exemple

```
<!ELEMENT para (#PCDATA)>
<!ATTLIST   para   sécurité   (non_classifié
                              | secret_de_polichinelle
                              | top_secret)   non_classifié>
```

Les Espaces Blancs

- Attribut XML-SPACE attaché a l’ élément:

XML-SPACE (DEFAULT | PRESERVE) #IMPLIED

Exemple:

<adresse xml-space=“preserve”>
2754 Ste-Catherine
Montréal, Québec
Canada, H5D 2P8</adresse>

La Déclaration Notation

- Identifie les types de données non-xml (graphiques, son, vidéo) référencées dans le document. La syntaxe est:

<!NOTATION nom_notation ident_notation>

- Exemple:

<!NOTATION cgmbin SYSTEM “http://www.qc.ca/apps/cgmviewer.exe”>

Entité Paramètre

- Regroupe des déclarations réutilises plusieurs fois dans la DTD (similaire à un macro):

<!ENTITY % nom_entité “contenu référencé”>

- Exemple:

<!ENTITY % adresse “(no-rue, ville, province, pays, code-postal)”>

- Dans la DTD au point voulu, on insère l’appel d’entité %adresse;

Les Entités Générales

- Syntaxe de déclaration:

<!ENTITY nom_de_l'entité "contenu_référencé">

A l'intérieur du document, l'insertion se fait avec l'appel d'entité: &nom_de_l'entité;

Entité Générale Interne

- Définies pour du texte réutilisé plusieurs fois dans un document (exemple: notice de droits d'auteur)
- Dans la DTD, on déclare:

<!ENTITY xml "Extensible Markup Langage">

- Dans l'instance, on insère au point voulu:

&xml;

Entité Générale Externe

<!ENTITY fig-1 SYSTEM "C:/tms/tm59/fig1" NDATA cgmbin>

- La déclaration entité dans ce cas pointe vers un fichier graphique au format CGM (entité graphique)
- Un élément ayant un attribut de type ENTITY sera utilisé dans le document XML pour insérer le graphique au point voulu:
  - **<illustration titre="fig-1"/>**
  - **L'attribut titre est de type ENTITY**

Entités Internes Prédéfinies

- XML définit les entités internes suivantes:
  - < produit le caractère <
  - > produit le caractère >
  - & produit le caractère &
  - ' produit le caractère ‘
  - " produit le caractère “

Les Commentaires

- Synataxe: <!--Commentaires-->
- Peut s'insérer dans la DTD ou l'instance
- Ne peut contenir la chaîne de caractères "--"
- Exemple de commentaire:

<!--Ceci est un commentaire-->

Les Sections CDATA

- Signale au parser d'ignorer les caractères de balisage et les délimiteurs
- Exemple:

<![CDATA[Le Budget de R&D.]]>

- Ne peut contenir la chaîne de caractères]]>

Les Instructions de Traitement

- <?nom_instr_traitement?>
- Fournit de l'information à une application XML
- Exemple:

  <?XML version="1.0"?>

XML Bien Formé vs XML Valide

- Documents XML bien formés:
  - Sans DTD
  - Contient au moins un élément
  - Imbrication correcte des balises
- Documents Valides:
  - Possède une DTD
  - L'instance XML suit les règles de la DTD

Le Rôle du Parser

- Parser valideur:
  - Vérifie que l'instance est conforme à la DTD et à la spécification XML
- Parser non-valideur:
  - Vérifie que l'instance est bien formée

Référence à la Feuille de Style

- Feuille de style référencée dans le document XML par son URL:

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

- Permet de présenter le document XML sur un browser web sous format HTML
- Le Processeur XSL peut être sur le client ou sur le serveur

Exemple de Feuille de Style XSL

```
<?xml version="1.0"?>
<TABLE xmlns:xsl="http://www.w3.org/TR/WD-xsl" STYLE="border:5px solid green;
    cellpadding:17pt">
     <TR  STYLE="font-size:12pt; font-family:Verdana; font-weight:bold; text-
    decoration:underline">
            <TD>NUMÉRO DE LA PIÈCE</TD>
            <TD>DESCRIPTION</TD>
            <TD>PRIX</TD>
    </TR>
    <xsl:for-each select="catalogue/pièces/pièce">
            <TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
             <TD><xsl:value-of select="numero"/></TD>
                        <TD><xsl:value-of select="description"/></TD>
                        <TD><xsl:value-of select="prix"/></TD>
   </TR>
  </xsl:for-each>
</TABLE>
```

Le Langage XPointer

- Langage XML Pointer

<DISCOURS ID="a27"><SUJET>XML</SUJET>

<CONFÉRENCIER>A. Mondeau</ CONFÉRENCIER >,
<CONFÉRENCIER>G. Nehru</ CONFÉRENCIER > et
<CONFÉRENCIER>Z. Biogera</ CONFÉRENCIER > nous parlerons de leurs expériences.</DISCOURS>

- Pour créer un lien avec le dernier élément “CONFÉRENCIER” du document XML:

<lien href=“http://abc.com/discours/descendant(-1, CONFÉRENCIER)”>lien avec Z. Biogera</lien>

- Pour créer un lien avec le second élément de DISCOURS:

<lien href=“http://abc.com/discours/id(a27).child(2,#element)>lien avec A. Mondeau</lien>

Déclarations de Liens Xlink Simples

- Liens simples uni-directionnels:

```
<!ELEMENT mylink (#PCDATA)>
<!ATTLIST    mylink
                xml:link        CDATA          #FIXED "simple"
                href            CDATA          #REQUIRED
                content-role    CDATA          #IMPLIE    >
```

- Le balisage se fera comme suit:

```
<mylink xml:link="simple" title="Citation"
    href="http://www.xyz.com/xml/foo.xml" show="new"
    content-role="Référence">Tel que mentionné dans
   Smith(1997)</mylink>
```

Les Liens Xlink Multidirectionnels

- Liens Multidirectionnels:

```
<annotations xml:link="extended" inline="false">
   <locator href="pr.xml" role="Procédure de Réparation">
   <locator href=lise.xml" role="annotation"/>
   <locator href=jacques.xml" role="annotation"/>
</annotations>
```

Le Comportement des Liens

Attribut “Show”

- Embed
- Replace
- New

Attribut “Actuate”

- auto
- user

Groupe de Liens Étendus

```
<!ELEMENT group (document*)>
<!ATTLIST   group
     xml:link      CDATA            #FIXED "group"
     steps         CDATA            #IMPLIED>

 <!ELEMENT document EMPTY>
 <!ATTLIST    document
     xml:link      CDATA            #FIXED "document"
     href          CDATA            #REQUIRED >
```

Groupe *e*Fa

Questions?

Veuillez adresser vos questions à Joel Amoussou à l'adresse:

jamoussou@efagroup.com

(514) 341-7836