Représenter des données en XML

Typer des documents avec une DTD

BBM12

Année 2007-08

Introduction

Qu'est-ce que XML ?

- eXtensible Mark-up Language
- défini par le W3C
- permet la définition de familles de langages de balises
- fait aussi référence à une famille de technologies

```
<fiche-identite>
  <nom>Parrain</nom>
  <prenom>Anne</prenom>
</fiche-identite>
```

Introduction

A quoi sert XML ?

Représenter des données pour les manipuler, les échanger, les interroger.

Exemples

- Documents de bureautique : OpenOffice
- Documents texte : DocBook,. . .
- Données informatiques : configurations. . .
- Données échangées : XHTML, jabber, web services,. . .
- Données stockées : bases de données XML
- beaucoup d'autres choses nouvelles, chaque jour ou presque !

Exemple de document XML

```
<?xml version="1.0" encoding="iso-88-59-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
                "http ://www.w3.org/TR/xhtml11/DTD/xhtml11.
<html>
<head>
  <title>Master Pro ILI</title>
</head>
<body>
<div class="entete">
<div class="titre">
<h1>Master Professionnalisé ILI</h1>
<h1>Ingéniérie Logicielle pour l'Internet</h1>
<h2>Master Sciences : Mention Mathématiques-Informatique</h
<h2><a href="http ://www.univ-artois.fr">Université d'Arto
à l'UFR des Sciences (Lens)</a></h2>
        div></body></html>
```

Introduction

Avantages de XML :

- Favoriser l'interopérabilité, l'échange.
- Rendre pérennes les données.
- Les rendre manipulables à la fois par les hommes et les machines :
  - transformations de documents (fusion, réorganisation, ...)
  - extraction d'informations
  - consultation, modification par programmes ad hoc

Quelles applications ? – Exemple

```
<agenda>
  <evt><date>10/10/2005</date>
       <concerne>parrain@cril.univ-artois.fr</concerne>
       <heure-debut>16h</heure-debut>
       <heure-fin>18h20</heure-fin>
       <objet>Cours SI-BD-Internet</objet>
  </evt>
  <regulier><date-debut>20/09/2005</date-debut>
            <date-fin>15/12/2005</date-fin>
            <periode unite="semaine">12</periode>
            <concerne>parrain@cril.univ-artois.fr</concer
            <heure-debut>11h</heure-debut>
            <heure-fin>15h45</heure-fin>
            <objet>Cours ILI3</objet>
  </regulier>
</agenda>
```

Quelles applications ?

- afficher un emploi du temps hebdomadaire ;
- envoyer un mail aux personnes concernées ;
- calculer des heures d'enseignement
- ...

XML : concepts fondamentaux

Éléments fondamentaux de XML :

- éléments
- attributs
- entités

XML : concepts fondamentaux

```
<?xml version="1.0" encoding="iso-8859-1"?>
<message priorité="important">
<destinataire>M. Dupont</destinataire>
<expediteur>Melle. Dumoulin</expediteur>
<objet>alimentation du chat</objet>
<corps>
<para>Conformément à vos instructions, je donne
<emphase>trois</emphase> rations de croquettes
par jour à <nomduchat />.
</para>
<para><nomduchat /> a cependant
pris <emphase>deux</emphase> kilos pendant
les vacances.</para>
</corps>
<formulepolitesse style="simple"/>
re>Melle. Dumoulin</signature>
e>
```

XML : prologue de document

Le prologue du document contient :

- la déclaration XML
- la déclaration du type de document

Le prologue est facultatif, mais important car il précise des informations importantes pour les processeurs XML.

XML : prologue de document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="macss.css" type="text/css"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http ://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

XML : prologue de document

Une Définition de Type de Document (DTD)

- sous-langage XML (ex : XHTML, MathML, MusicXML, DocBook ...)
- précise la grammaire que doit suivre le document.

XML : prologue de document

Une DTD peut-être :

- publique : diffusée par une institution, accessible via le web par un identifiant public ;
- spécifique à une application : accessible via une URL
- décrite directement dans le document
- un peu de tout !

XML : Éléments

Les éléments sont les balises qui peuvent apparaître dans un document XML. Ils peuvent être qualifiés par des attributs.

```
<nom_elt attribut="valeur" [attribut="valeur"]>
</nom_elt>
```

ou bien (élément vide)

```
<nom_elt attribut="valeur" [attribut="valeur"]/>
```

Exemple :

```
<message priorité="important">
<destinataire>M. Dupont</destinataire>
<expediteur>Melle. Dumoulin</expediteur>
...
<formulepolitesse style="simple"/>
<signature>Melle. Dumoulin</signature>
</message>
```

XML : Éléments

Restrictions syntaxiques :

- nom d'élément
  - commence par une lettre ou un _
  - contient des lettres (symboles définis par le codage utilisé), des chiffres, des tirets, des soulignés, des points
- séparateur d'éléments : espace, retour à la ligne, tabulation, =, ", '

XML : Éléments

Restrictions syntaxiques :

- une balise de fin arrive après une balise de début
- les balises de début et de fin apparaissent à l'intérieur du même élément parent
- entre la balise de début et celle de fin : données textuelles ou éléments
- chaque document XML a un seul élément racine

XML : Attributs

Les attributs qualifient les éléments sur lesquels ils portent.

```
<nom_elt attribut="valeur" [attribut="valeur"]>
```

Exemple

```
<exception type="msg='erreur'" />
```

Pour un même élément :

- l'ordre n'a pas d'importance
- une seule occurrence d'un même attribut

XML : Attributs

Possibilité de donner plusieurs valeurs à un même attribut :

```
<copains filles="josette ginette colette"
     garçons="alain sylvain govain">
```

Attention ! La bonne solution pourrait être :

```
<copains>
  <fille>josette</fille>
  <fille>ginette</fille>
  <fille>colette</fille>
  <garçon>alain</garçon>
  ...
</copains>
```

XML : Attributs

Attributs particuliers : id et idref pour connecter des ressources

id : une même valeur ne peut être attribuée qu'une seule fois dans le document

idref : il doit exister un élément qui possède un attribut id de même valeur

Typer les documents

Intérêts

- avoir des documents conformes à un modèle
- pouvoir automatiser des traitements sur les documents

Comment

- avec une définition de type de document DTD
- avec un schéma XML
- ...

Document Type Definition

Les éléments sont définis à l'aide d'expressions rationnelles :

```
<!ELEMENT nom_elt exp_rat>
```

Opérateurs utilisés :

```
, | * +?
  ( )
#PCDATA
```

mistigri.xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<message priorité="important">
<destinataire>M. Dupont</destinataire>
<expediteur>Melle. Dumoulin</expediteur>
<objet>alimentation du chat</objet>
<corps>
<para>Conformément à vos instructions, je donne
<emphase>trois</emphase> rations de croquettes
par jour à <nomduchat />.
</para>
<para><nomduchat /> a cependant
pris <emphase>deux</emphase> kilos pendant
les vacances.</para>
</corps>
<formulepolitesse style="simple"/>
re>Melle. Dumoulin</signature>
e>
```

La DTD de mistigri.xml

```
<!ELEMENT message (destinataire,expediteur?,objet?,
                   corps,formule_politesse,
                   (signature|sign_complete))>
<!ELEMENT destinataire (#PCDATA)>
<!ELEMENT expediteur (#PCDATA)>
<!ELEMENT objet (#PCDATA|nomduchat|emphase)*>
<!ELEMENT sign_complete (lieu,date,signature)>
<!ELEMENT signature (#PCDATA)>
<!ELEMENT formule_politesse EMPTY>
<!ELEMENT corps (para+)>
<!ELEMENT para (#PCDATA|nomduchat|emphase)*>
<!ELEMENT nomduchat EMPTY>
<!ELEMENT emphase (#PCDATA)>
```

XML : vers la conception d'une DTD

Concevoir une DTD :

- importance de l'analyse ;
- choisir des noms d'éléments comme un typage
- différentes catégories d'éléments
  - ne contiennent que d'autres éléments, ou que du texte : blocs
  - ont un contenu mixte : leurs éléments fils sont des éléments en ligne
- attention à la position (ordre des éléments significatif)
- attention à la hiérarchie
- éviter les instructions de traitement

XML : vers la conception d'une DTD

Choisir entre un élément et un attribut :

- Utiliser un élément :
  - contenu relativement gros
  - ordre important
  - le contenu fait partie de l'information du document
- Utiliser un attribut :
  - le contenu modifie le traitement de l'information
  - contrôle sur les valeurs
  - le contenu est un identifiant

Définition des attributs

```
<!ATTLIST nom_elt nom_attr1 type_attr1 desc_attr1 ...>
```

- `type_attribut` : type de l'attribut ou liste des valeurs possibles ;
- `desc_attribut` : comportement de l'attribut (obligatoire, optionnel, valeur par défaut, etc . . .)

Comportement des attributs

- spécification d'une valeur par défaut

```
<!ATTLIST message
          priorite (importante|courante|basse)
          "courante">
```

- attribut obligatoire

```
<!ATTLIST feuTricolore couleur
(rouge|orange|vert) #REQUIRED>
```

- attribut optionnel, sans valeur par défaut

```
<!ATTLIST div class NMTOKEN #IMPLIED>
```

- attribut à valeur fixée (cas rare !) `#FIXED valeur`

Les différents types d'attributs

- CDATA : donnée textuelle. Type le plus permissif

```
<!ATTLIST message  texte CDATA #REQUIRED>
<message texte="Salut machin, truc &amp; Cie !">
... </message>
```

- NMTOKEN : lexème nominal (commence par une lettre, suivi de lettres, de chiffres ou de . :-...)
- NMTOKENS : suite de lexèmes nominaux séparés par des espaces

```
<!ATTLIST copains filles NMTOKENS #REQUIRED>
<copains filles="Gisèle Marcelle Danièle">
... </copains>
```

Les différents types d'attributs

- ID : identifiant unique. Même syntaxe que NMTOKEN

```
<!ATTLIST garçon nom ID #REQUIRED>

<garçon nom="Paulo">...</garçon>
```

- IDREF : référence d'identifiant

```
<!ATTLIST copains meilleur IDREF #IMPLIED>
<copains meilleur="Paulo"
         filles="Colette Lucette Perette">
...</copains>
```

- IDREFS : liste de références d'identifiants, même syntaxe que NMTOKENS

Les différents types d'attributs

- énumération de valeurs : liste de mots-clés

```
<!ATTLIST message
          priorite (importante|courante|basse)
          "courante">
<!ATTLIST feuTricolore couleur (rouge|orange|vert)
          #REQUIRED>
```

Valider un document XML

- Il existe plusieurs outils