NF92
Traitement automatique de l'information

Pavol BARGER

XML

Motivations

- Modéliser des données (semi-)structurées
  - textes littéraires
  - bases de données bibliographiques
  - dictionnaires
  - paramètres de configuration
  - échange de données entre applications
  - cartographie des gènes
- Publier les données sur le web (HTML, XML/XSL, XHTML)
- Séparer les structures sémantiques de la présentation (SGML, XML)
- Intégrer des données hétérogènes (XML)

Motivations

- Intégration croissante dans une large gamme d'applications
  - Usage industriel et applicatif
    - édition, communication, web, ...
    - bases de données
    - intégration de données multimédia
    - Internet (catalogue de produits et E-Commerce)
  - Usage scientifique et monde de la recherche
    - cartographie du génome humain
    - codage de données scientifiques (chimie, maths, ...)
    - représentation des connaissances (logique, IA, ...)
    - linguistique computationnelle (dictionnaires, traduction, ...)
- Indépendance
  - de toute langue, toute plate-forme et de tout fabricant de logiciel

XML ≠ HTML

- HTML (*HyperText Markup Language,* 1990)
  - application SGML
  - ensemble de balises figées
  - afficher texte, liens hypertextes et images (pages Web)
- XML
  - pas de balises prédéfinies
  - permet de spécifier des balises en fonction des données
    - « META-langage »
  - centré sur la structuration du contenu (et non pas la forme)
- Actuellement :
  - HTML est devenu une application XML (XHTML)

Qu'est-ce que c'est donc ?

- Une recommandation du W3C
  - http://www.w3.org/XML/
- Un langage de description extensible
  - « méta-langage » qui permet de définir d'autres langages
  - ≠ HTML, = SGML)
- Une simplification de la norme SGML
- Centré sur les structures
  - logiques et sémantiques
  - ≠ présentationnelles

XML

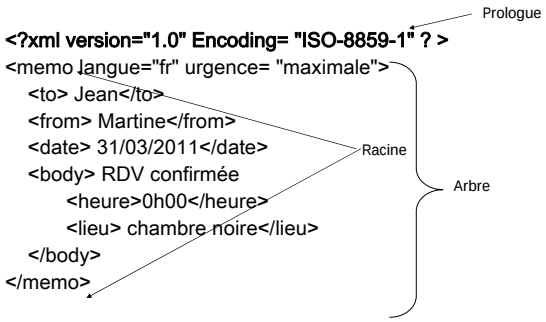
eXtensible Markup Language (XML)

- Une nouvelle application de SGML pour le Web et …
- Né : fin 96
- Père : W3C
- Petit-fils de SGML (ISO-1986)
- Cousin d'HTML
- Reconnu le : 10/02/98 – version 1.0
- Descendance – XHMTL, MathML, …

Document XML

- Données binaires
- Données textuelles
- Norme Unicode
- Marques :
  - Balises de début, de fin, références, commentaires, instructions de traitement
- Définition du Type de Document

Exemple XML



Structure document XML

- Un prologue (préambule)
  - Facultatif mais conseillé
  - Contient l'information de la version XML
  - Contient la DTD
- Un arbre d'éléments avec un élément racine
  - Le contenu propre
- Commentaires et instructions de traitement
  - Facultatifs, peuvent être dans le prologue ou dans le corps

Documents bien-formés

- Syntaxe correcte
  - balises fermées ?
  - guillemets autour des attributs ?
- Mais sémantique non vérifiable
  - nom des éléments ?
  - enchâssement ?
  - valeurs des attributs ?
- Pour vérifier tout cela :

Document Type Definition (DTD)

Règles syntaxiques

1. Commencer par une déclaration XML
2. Balisage sensible à la case
3. La valeur des attributs doit être quotée
4. Balises non vides appariées
</br>
5. Balises vides fermées
6. Les éléments ne doivent pas se chevaucher
   <jour> <mois> </jour> </mois> interdit
7. Un élément doit encapsuler tous les autres
8. Ne pas utiliser les caractères < et & seuls

Définition du Type de Document

- Permet de valider un document XML
  - Un document XML bien formé respect les règles de syntaxe
  - Un document XML valide doit être bien formé et contenir un seul arbre de données en respect de la DTD
- Structure et grammaire
- La DTD commence par

  <!DOCTYPE ... >

Documents valides

- Valide = bien-formé + conforme à une DTD
  - Définition d'une DTD
    - contraintes sur les noms des éléments, des attributs
    - description du contenu des éléments
      - enchâssement d'éléments, texte, ...
    - attachement d'attributs à un élément donné
    - type des attributs
      - facultatifs, obligatoire
    - valeur des attributs
      - numériques, alpha-numériques, liste fermée
- Principes de bases = approche SGML traditionnelle, mais
  - DTD optionnelle en XML et syntaxe simplifiée
  - « production en valide et distribution en bien-formé »

La structure des documents

- Analyse attentive des données à traiter !!!
- Mettre à jour les règles générales sur la structuration des données...
  - composantes nécessaires → éléments ?
  - propriétés pertinentes → attributs ?
    - valeurs spécifiques de certains attributs ?
    - type des attributs ?
  - enchâssement des éléments → séquences d'éléments ?

Ecrire une DTD

```
<!ELEMENT MEMBRE
 (LOGIN, NOM, PRENOM?, MEL, TEL+, FAX*, EQUIPE)>
<!ELEMENT LOGIN EMPTY>
  <!ATTLIST LOGIN
      ID #REQUIRED>
<!ELEMENT NOM (#PCDATA)>
<!ELEMENT PRENOM (#PCDATA)>
<!ELEMENT MEL( #PCDATA)>
<!ELEMENT TEL (#PCDATA)>
<!ELEMENT FAX (#PCDATA)>
<!ELEMENT EQUIPE (#PCDATA)>
  <!ATTLIST EQUIPE
      LAB CDATA #REQUIRED>
```

Élément

- Composant de base
- Identifié par un nom
- Délimité par une balise ouvrante et une balise fermante

<AUTEUR> Victor Hugo </AUTEUR>

- Ou élément vide

<PHOTO Source= "victor.gif" />

- Contenu textuel, éléments ou mixte

…et aussi

- <!-- des commentaires -->
- Des entités externes/internes, analysables ou non

  & " > <
- <![CDATA [

  …tout et n'importe quoi…

  …0x01265423deadbeef49653453462…

  ]]>

Déclaration des éléments

```
<!ELEMENT MEMBRE
 (LOGIN, NOM, PRENOM?, MEL, TEL+, FAX*, EQUIPE)>
<!ELEMENT LOGIN EMPTY>
  <!ATTLIST LOGIN
      ID ID #REQUIRED>
<!ELEMENT NOM (#PCDATA)>
<!ELEMENT PRENOM (#PCDATA)>
<!ELEMENT MEL( #PCDATA)>
<!ELEMENT TEL (#PCDATA)>
<!ELEMENT FAX (#PCDATA)>
<!ELEMENT EQUIPE (#PCDATA)>
  <!ATTLIST EQUIPE
      LAB CDATA #REQUIRED>
```

nom de l'élément

mot-clé ELEMENT

Contenu des éléments

```
<!ELEMENT MEMBRE
 (LOGIN, NOM, PRENOM?, MEL, TEL+, FAX*, EQUIPE)>
<!ELEMENT LOGIN EMPTY >
  <!ATTLIST LOGIN
      ID ID #REQUIRED>
<!ELEMENT NOM (#PCDATA)>
<!ELEMENT PRENOM (#PCDATA)>
<!ELEMENT MEL (#PCDATA)>
<!ELEMENT TEL (#PCDATA)>
<!ELEMENT FAX (#PCDATA)>
<!ELEMENT EQUIPE (#PCDATA)>
  <!ATTLIST EQUIPE
      LAB CDATA #REQUIRED>
```

éléments enfants

élément vide

données textuelles

Séquençage des éléments enfants

```
<!ELEMENT MEMBRE
 (LOGIN, NOM, PRENOM?, MEL, TEL+, FAX*, EQUIPE)>
```

1

< = 1

> = 1

> = 0

OU exclusif

```
<!ELEMENT PAYMENT (CASH|CREDIT_CARD)>
```

factorisation

```
<!ELEMENT ADRESSE (RUE+,(VILLE|CODE_POSTAL)*)>
```

Spécifications éléments

(#PCDATA)	Parsed Character DATA
(ELT)	1 fois ELT
(ELT1,ELT2)	Séquence
(ELT1|ELT2|…)	Choix
ELT?	0 ou 1 fois ELT
ELT+	au moins 1 fois ELT
ELT*	0 ou plusieurs fois ELT
()	groupe de sous éléments
ANY	n'importe quoi
EMPTY	rien

Les attributs

- Inclus dans la balise ouvrante d'un élément
- Composé d'un nom et d'une valeur

<AUTEUR NE="1802" MORT="1885" > Victor Hugo </AUTEUR>

Déclaration des attributs

```
<!ELEMENT MEMBRE
 (LOGIN, NOM, PRENOM?, MEL, TEL+, FAX*, EQUIPE)>
<!ELEMENT LOGIN EMPTY>
  <!ATTLIST LOGIN
      ID #REQUIRED>
<!ELEMENT NOM (#PCDATA)>
<!ELEMENT PRENOM (#PCDATA)>
<!ELEMENT MEL( #PCDATA)>
<!ELEMENT TEL (#PCDATA)>
<!ELEMENT FAX (#PCDATA)>
<!ELEMENT EQUIPE (#PCDATA)>
  <!ATTLIST EQUIPE
      LAB CDATA #REQUIRED>
```

mot-clé ATTLIST

nom de l'élément

nom de l'attribut

type de l'attribut

valeur implicite

Types d'attributs

CDATA	données textuelles	`<!ATTLIST EQUIPE LAB CDATA #REQUIRED>`
ID	nom unique	`<!ATTLIST LOGIN ID ID #REQUIRED>`
énumerat°	liste fermée	`<!ATTLIST FIGURE COLOR (RED|BLUE|BLACK)>`
(7 autres)	...	...

Plusieurs attributs

```
<!ATTLIST EQUIPE LAB CDATA #REQUIRED>
<!ATTLIST EQUIPE Date CDATA #IMPLIED>

<!ATTLIST EQUIPE
      LAB CDATA #REQUIRED
      Date CDATA #IMPLIED
>
```

attributs : présence et valeurs

"default_value"	valeur par défault	`<!ATTLIST TEXT LANGUE CDATA "ANGLAIS">`
#REQUIRED	saisie obligatoire	`<!ATTLIST EQUIPE LAB CDATA #REQUIRED>`
#IMPLIED	saisie facultative	`<!ATTLIST PERSON PHONE CDATA #IMPLIED>`
#FIXED	valeur prédéfinie	`<!ATTLIST EQUIPE LAB CDATA #FIXED "TAL">`

??? `<!ATTLIST LOGIN IDENT ID #FIXED "M">`

Spécifications d'attributs

CDATA	données textuelles
NMTOKEN	nom XML valide
NMTOKENS	noms XML valides
(val-1|val-2|…val-n)	liste de valeurs
ID	identificateur unique
IDREF	valeur d'un ID
IDREFS	valeurs d'Ids
ENTITY	entité externe non analysable
ENTITIES	entités externes non analysables

Utilisation et partage d'une DTD

- DTD interne au document XML

```
<!DOCTYPE MEMBRE [                    } déclaration doctype
<!ELEMENT MEMBRE … >                  )
…                                     ) déclaration DTD
]>                                    )
<MEMBRE TYPE="IE" ID="M28">           )
 …                                    ) document XML
</MEMBRE>                             )
```

- DTD déclarée au début du document XML (entre crochets)
- avantage : proximité
- inconvénient : partage de la même DTD entre plusieurs documents XML

Utilisation et partage d'une DTD

- DTD externe au document XML

```
<!ELEMENT MEMBRE … >      } déclaration DTD (fichier .dtd)
…                         )
```

```
<!DOCTYPE MEMBRE SYSTEM "http://…/MEMBRE.dtd">
<MEMBRE TYPE="IE" ID="M28">
 …
</MEMBRE>
```

- DTD déclarée dans un fichier à part
- avantage : partage de la même DTD pour plusieurs docs XML
- inconvénient : rétro-compatibilité

Application

- Ecrire une DTD pour une entrée de dictionnaire :

```
<p><b>Accomodation</b>,<i>f.</i>, faculté de l'œil humain permettant de maintenir une vision nette des objets quelle que soit leur distance. <BR>
<b><I>En stéréoscopie</b></i>, faculté des yeux humains d'obtenir la vision stéréoscopique par superposition de deux images.</p>
```

Accomodation, *f.*, faculté de l'oeil humain permettant de maintenir une vision nette des objets quelle que soit leur distance.
En stéréoscopie, faculté des yeux humains d'obtenir la vision stéréoscopique par superposition de deux images.

Exemple d'une DTD simple

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE dico [
<!-- Coucou -->
<!ELEMENT dico (article+)>
<!ELEMENT article (vedette, gramm, sens+)>
<!ELEMENT vedette (#PCDATA)>
<!ELEMENT gramm (categorie, (genre | nombre | transitivité)*)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT transitivité (#PCDATA)>
<!ELEMENT sens (domaine?, def, exemple*, sens*)>
<!ATTLIST sens n CDATA #IMPLIED>
]>
```

Exemple de DTD

```
<?xml version="1.0" Encoding= "ISO-8859-1" ? >
<!DOCTYPE memo [
    <!ELEMENT memo (to, from, date, body)>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT date (#PCDATA)>
    <!ELEMENT body (#PCDATA|jour|heure|lieu)>
    <!ELEMENT jour (#PCDATA)>
    <!ELEMENT heure (#PCDATA)>
    <!ELEMENT lieu (#PCDATA)>
    <!ATTLIST memo langue CDATA "Alsacien">
    <!ATTLIST memo urgence CDATA "normal">
]>
```

Exemple complet

```
<?xml version="1.0"?>
<!DOCTYPE FILM
    [
        <!ELEMENT FILM (TITRE, (STAR | NARRATEUR | ASSISTANT))>
        <!ATTLIST FILM Classe (fiction | documentaire) "fiction">
        <!ELEMENT TITRE (#PCDATA)>
        <!ELEMENT STAR (#PCDATA)>
        <!ELEMENT NARATTEUR (#PCDATA)>
        <!ELEMENT ASSISTANT (#PCDATA)>
    ]
>

<FILM Classe="documentaire">
    <TITRE>Cours XML</TITRE>
    <STAR>Pino Chio</STAR>
</FILM>
```

Modifications

```
<?xml version="1.0"?>
<!DOCTYPE FILM
    [
        <!ELEMENT FILM (TITRE+, (STAR, NARRATEUR, ASSISTANT)*)>
        <!ATTLIST FILM Classe (fiction | documentaire) "fiction">
        <!ELEMENT TITRE (#PCDATA)>
        <!ELEMENT STAR (#PCDATA)>
        <!ELEMENT NARATTEUR (#PCDATA)>
        <!ELEMENT ASSISTANT (#PCDATA)>
    ]
>

<FILM Classe="documentaire">
    <TITRE>Cours XML</TITRE>
    <STAR>Pino Chio</STAR>
</FILM>
```

Exemple

- Voici la DTD d'un document :

```
<?xml version="1.0" ?>
<!DOCTYPE annuaire[
  <!ELEMENT annuaire (personne*)>
  <!ELEMENT personne
     (nom+,prenom*,adresse)>
  <!ELEMENT nom    (#PCDATA)>
  <!ELEMENT prenom (#PCDATA)>
  <!ELEMENT adresse (ville|pays)>
  <!ELEMENT ville (#PCDATA)>
  <!ELEMENT pays (#PCDATA)>
]>
```

```
<annuaire >
<personne>
<nom>Maigret</nom>
<prenom>Marie</prenom><prenom>Anne
</prenom>
<pays>Suisse</pays>
</personne>
</annuaire>
```

Exemple

- Voici la DTD d'un document :

```
<?xml version="1.0" ?>
<!DOCTYPE annuaire[
  <!ELEMENT annuaire (personne*)>
  <!ELEMENT personne
     (nom+,prenom*,adresse)>
  <!ELEMENT nom    (#PCDATA)>
  <!ELEMENT prenom (#PCDATA)>
  <!ELEMENT adresse (ville|pays)>
  <!ELEMENT ville (#PCDATA)>
  <!ELEMENT pays (#PCDATA)>
]>
```

```
<annuaire >
<personne>
<nom>Dupont</nom>
<prenom>Jean</prenom>
<adresse><ville>Paris</ville><pays>France</pays></adresse>
</personne>
</annuaire>
```

Exemple

- Voici la DTD d'un document :

```
<?xml version="1.0" ?>
<!DOCTYPE annuaire[
  <!ELEMENT annuaire (personne*)>
  <!ELEMENT personne
     (nom+,prenom*,adresse)>
  <!ELEMENT nom    (#PCDATA)>
  <!ELEMENT prenom (#PCDATA)>
  <!ELEMENT adresse (ville|pays)>
  <!ELEMENT ville (#PCDATA)>
  <!ELEMENT pays (#PCDATA)>
]>
```

```
<annuaire >
<personne>
<nom>Valas</nom><nom>Durand</nom>
<prenom>Céline</prenom>
<adresse><ville>Madrid</ville></adresse>
<adresse><pays>Espagne</pays></adresse>
</personne>
</annuaire>
```

Exemple

- Voici la DTD d'un document :

```
<?xml version="1.0" ?>
<!DOCTYPE annuaire[
  <!ELEMENT annuaire (personne*)>
  <!ELEMENT personne
     (nom+,prenom*,adresse)>
  <!ELEMENT nom    (#PCDATA)>
  <!ELEMENT prenom (#PCDATA)>
  <!ELEMENT adresse (ville|pays)>
  <!ELEMENT ville (#PCDATA)>
  <!ELEMENT pays (#PCDATA)>
]>
```

```
<annuaire >
<personne>
</personne>
</annuaire>
```

XML et CSS

Inventaire.css

```
LIVRE
        {display: block;
        margin-top:12pt;
        }
AUTEUR
        {font-weight:bold}
```

Inventaire.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css"
href="Inventaire.css"?>
<LIVRE>
<TITRE>Cours XML</TITRE>
<AUTEUR>Manu Chao</AUTEUR>
</LIVRE>
```

Technologies basées XML

- Schema
- XPath
- xhtml
- XSL
- XML + SQL ⇨ XQuery
- XML + UML ⇨ XMI
- XML + IHM ⇨ XUL

- Fichiers de configuration

Conception d'un site

- HTML
- XML
- CSS
- PHP/mySQL
- Java & comp.
- cookies
- sessions
- sécurité

Outils de conception de sites
CMS : Wordpress, Joomla
CSS editors
template editors
site editors