Les données semi-structurées

Chapitre 3: DTD

3ème Année Licence Informatique
Ingénierie des Systèmes d'Information et du Logiciel (ISIL)
Préparé par M. L. FOUGHALI

Ressources utilisées :

1. https://www.w3schools.com/xml/xml_syntax.asp
2. **XML, Ed TITTEL, SCHAUMS, 2004.**
3. **XML par la pratique Bases indispensables, concepts et cas pratiques (3ième édition), Thierry BOULANGER, Editions-ENI, 2015.**
4. **XML, Gilles CHAGNON & Florent NOLOT, PEARSON, 2007.**

Validateur XML

- Un document XML avec une syntaxe correcte est dit **bien formé** (Well Formed.)
- Les règles de syntaxe ont été décrites précédemment:
  - Les documents XML doivent avoir un élément racine
  - Les éléments XML doivent avoir une balise de fermeture
  - Les balises XML sont sensibles à la casse
  - Les éléments XML doivent être correctement imbriqués
  - Les valeurs d'attribut XML doivent être entre guillemets
- Les erreurs dans les documents XML arrêteront vos applications XML.
- La spécification XML du W3C stipule qu'un programme doit arrêter de traiter un document XML s'il trouve une erreur. La raison en est que les logiciels XML doivent être petits, rapides et compatibles.
- Les navigateurs HTML sont autorisés à afficher des documents HTML contenant des erreurs (comme des balises de fin manquantes).

Avec XML, les erreurs ne sont pas autorisées.

Documents XML valides

- Un document XML «bien formé» n'est pas la même chose qu'un document XML «valide».
- Un document XML "valide" doit être bien formé. De plus, il doit être conforme à une définition de type de document.
- Il existe deux définitions de type de document différentes qui peuvent être utilisées avec XML:
  - DTD - La définition du type de document d'origine
  - XML Schema - Une alternative basée sur XML à DTD
- Une définition de type de document définit les règles, les éléments juridiques et les attributs d'un document XML.

Qu'est-ce qu'une DTD?

- Un document XML avec une syntaxe correcte est appelé "Well Formed".
- Un document XML validé par rapport à une DTD est à la fois «bien formé» et «valide».
- **C’est quoi une DTD:**
  - DTD est l'acronyme de Document Type Definition.
  - Une DTD définit la structure et les éléments et attributs juridiques d'un document XML.

Documents XML valides

- Un document XML "valide" est "bien formé » et conforme aux règles d'une DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

Note.dtd

- La déclaration DOCTYPE ci-dessus contient une référence à un fichier DTD.

Quand utiliser/NE PAS utiliser une DTD?

- **Quand utiliser une DTD?**
  - Avec une DTD, des groupes de personnes indépendants peuvent accepter d'utiliser une DTD standard pour échanger des données.
  - Avec une DTD, vous pouvez vérifier que les données que vous recevez du monde extérieur sont valides.
  - Vous pouvez également utiliser une DTD pour vérifier vos propres données.

- **Quand NE PAS utiliser une DTD?**
  - XML ne nécessite pas de DTD.
  - Lorsque vous expérimentez avec XML, ou lorsque vous travaillez avec de petits fichiers XML, la création de DTD peut être une perte de temps.
  - Si vous développez des applications, attendez que la spécification soit stable avant d'ajouter une DTD. Sinon, votre logiciel pourrait cesser de fonctionner en raison d'erreurs de validation.

DTD - Blocs de construction XML

- Les principaux éléments constitutifs des documents XML et HTML sont des éléments.
- Du point de vue DTD, tous les documents XML sont constitués des blocs de construction suivants: Éléments, Attributs, Entités, PCDATA et CDATA:
  - Éléments, Attributs, Entités : ***connus!***
  - **PCDATA**
    - Signifie des données de caractères analysées.
    - **PCDATA est un texte qui sera analysé par un analyseur** . **Le texte sera examiné par l'analyseur pour les entités et le balisage** .
    - Les balises à l'intérieur du texte seront traitées comme du balisage et les entités seront développées.
    - Cependant, les données de caractères analysées ne doivent contenir aucun caractère &, <ou>; ceux-ci doivent être représentés par le & amp; & lt; et & gt; entités, respectivement.
  - **CDATA**
    - Signifie des données de caractères.
    - **CDATA est un texte qui ne sera PAS analysé par un analyseur**: les balises à l'intérieur du texte ne seront PAS traitées comme du balisage et les entités ne seront pas développées.

Les déclarations d'éléments

Déclaration	Expression régulière
Elément	`<!ELEMENT element-name category>` `<!ELEMENT element-name (element-content)`
Elément vide	`<!ELEMENT element-name EMPTY>`
Éléments avec PCDATA	`<!ELEMENT element-name (#PCDATA)>`
Éléments avec n'importe quel contenu.	`<!ELEMENT element-name ANY>`
Éléments avec enfants (séquences)	`<!ELEMENT element-name (child1)>` `<!ELEMENT element-name (child1,child2,...)>`
Une seule occurrence d'un élément	`<!ELEMENT element-name (child-name)>`
Au moins une occurrence d'un élément	`<!ELEMENT element-name (child-name+)>`

Les déclarations d'éléments (bis)

Déclaration	Expression régulière
Zéro ou plusieurs occurrences d'un élément	`<!ELEMENT element-name (child-name*)>`
zéro ou une occurrence d'un élément	`<!ELEMENT element-name (child-name?)>`
L'un ou l'autre	`<!ELEMENT note (to,from,header,(message|body))>`
Contenu mixte	`<!ELEMENT note (#PCDATA|to|from|header|message)*>`

Déclaration des attributs

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

```
DTD   =>      <!ATTLIST payment type CDATA "check">
XML   =>      <payment type="check" />
```

Type d'attribut	Description
CDATA	The value is character data
(*en1*|*en2*|..)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
xml:	The value is a predefined xml value

Déclaration des attributs (bis)

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

DTD => `<!ATTLIST payment type CDATA "check">`

XML => `<payment type="check" />`

Type d'attribut	Description
CDATA	The value is character data
(*en1*|*en2*|..)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
xml:	The value is a predefined xml value

Déclaration des attributs (ter)

Valeur d'attribut	Explication
value	The default value of the attribute
#REQUIRED	The attribute is required
#IMPLIED	The attribute is optional
#FIXED *value*	The attribute value is fixed

Attribut	DTD:	XML
PAR DEFAUT	<!ELEMENT square EMPTY> <!ATTLIST square width CDATA "0">	<square width="100" />
OBLIGATOIRE	<!ATTLIST person number CDATA #REQUIRED>	<person number="5677" />
IMPLICITE	<!ATTLIST contact fax CDATA #IMPLIED>	<contact fax="555-667788" />
FIXÉ	<!ATTLIST sender company CDATA #FIXED "Microsoft">	
		<sender company="Microsoft" />

Déclaration des attributs (quater)

- **Valeurs d'attribut énumérées**
- `<!ATTLIST element-name attribute-name (en1|en2|..) default-value>`

- **Exemple DTD:**

```
<!ATTLIST payment type (check|cash) "cash">
```

- **Exemple XML :**

```
<payment type="check" />
```

or

```
<payment type="cash" />
```

- Utilisez des valeurs d'attribut énumérées lorsque vous souhaitez que la valeur d'attribut fasse partie d'un ensemble fixe de valeurs légales.

Éléments VS Attributs

- En XML, il n'y a pas de règles concernant l'utilisation des attributs et l'utilisation des éléments enfants.
- Les données peuvent être stockées dans des éléments enfants ou dans des attributs.
- **Exemples:**

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Quand évitez d'utiliser des attributs?

1. Certains des problèmes liés aux attributs sont:
   - les attributs ne peuvent pas contenir plusieurs valeurs (les éléments enfants peuvent)
   - les attributs ne sont pas facilement extensibles (pour les modifications futures)
   - les attributs ne peuvent pas décrire les structures (les éléments enfants peuvent)
   - les attributs sont plus difficiles à manipuler par le code du programme
   - les valeurs d'attribut ne sont pas faciles à tester par rapport à une DTD
2. Si vous utilisez des attributs comme conteneurs pour les données, vous vous retrouvez avec des documents difficiles à lire et à gérer. Essayez d'utiliser des **éléments** pour décrire les données.
3. Utilisez les attributs uniquement pour fournir des informations qui ne sont pas pertinentes pour les données.

Déclaration des Entités

1. **Une déclaration interne**

```
<!ENTITY entity-name "entity-value">
```

- **DTD :**

```
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">
```

- **XML :**

```
<author>&writer;&copyright;</author>
```

- **Remarque:** une entité se compose de trois parties: une esperluette (&), un nom d'entité et un point-virgule (;).

2. **Une déclaration externe**

```
<!ENTITY entity-name SYSTEM "URI/URL">
```

- **DTD :**

```
<!ENTITY writer SYSTEM "https://www.w3schools.com/entities.dtd">
<!ENTITY copyright SYSTEM "https://www.w3schools.com/entities.dtd">
```

- **XML :**

```
<author>&writer;&copyright;</author>
```

DTD - Programme TV

```
<!DOCTYPE TVSCHEDULE [

<!ELEMENT TVSCHEDULE (CHANNEL+)>
<!ELEMENT CHANNEL (BANNER,DAY+)>
<!ELEMENT BANNER (#PCDATA)>
<!ELEMENT DAY (DATE,(HOLIDAY|PROGRAMSLOT+)+)>
<!ELEMENT HOLIDAY (#PCDATA)>
<!ELEMENT DATE (#PCDATA)>
<!ELEMENT PROGRAMSLOT (TIME,TITLE,DESCRIPTION?)>
<!ELEMENT TIME (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>

<!ATTLIST TVSCHEDULE NAME CDATA #REQUIRED>
<!ATTLIST CHANNEL CHAN CDATA #REQUIRED>
<!ATTLIST PROGRAMSLOT VTR CDATA #IMPLIED>
<!ATTLIST TITLE RATING CDATA #IMPLIED>
<!ATTLIST TITLE LANGUAGE CDATA #IMPLIED>
]>
```

DTD - Article de journal

```
<!DOCTYPE NEWSPAPER [

<!ELEMENT NEWSPAPER (ARTICLE+)>
<!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT BYLINE (#PCDATA)>
<!ELEMENT LEAD (#PCDATA)>
<!ELEMENT BODY (#PCDATA)>
<!ELEMENT NOTES (#PCDATA)>

<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>

<!ENTITY NEWSPAPER "Vervet Logic Times">
<!ENTITY PUBLISHER "Vervet Logic Press">
<!ENTITY COPYRIGHT "Copyright 1998 Vervet Logic Press">

]>
```

Vos Questions!