Résumé XML

XML, eXtensible Markup Language

1-Origine de XML

On trouve l'origine des langages à balises dans la nécessité **d'échanger des informations** dont la structure est trop **complexe** pour être simplement mise sous forme de colonnes délimitées par un caractère de séparation.
Le **succès du langage à balise HTML** a offert l'opportunité de pousser plus loin l'utilisation de ce type de langages. La **norme de langage XML** *(définie par le consortium W3C)* va ouvrir de nouveaux horizons pour aller au-delà d'un simple transfert de données. On a en effet aujourd'hui, un certain nombre de **dialectes[1] héritant du langage XML** *(le langage est extensible et les noms des balises y sont définis pour un usage précis, comme HTML*) qui ont ouvert de multiples voies à son utilisation, bien au-delà du 'transport' de données.

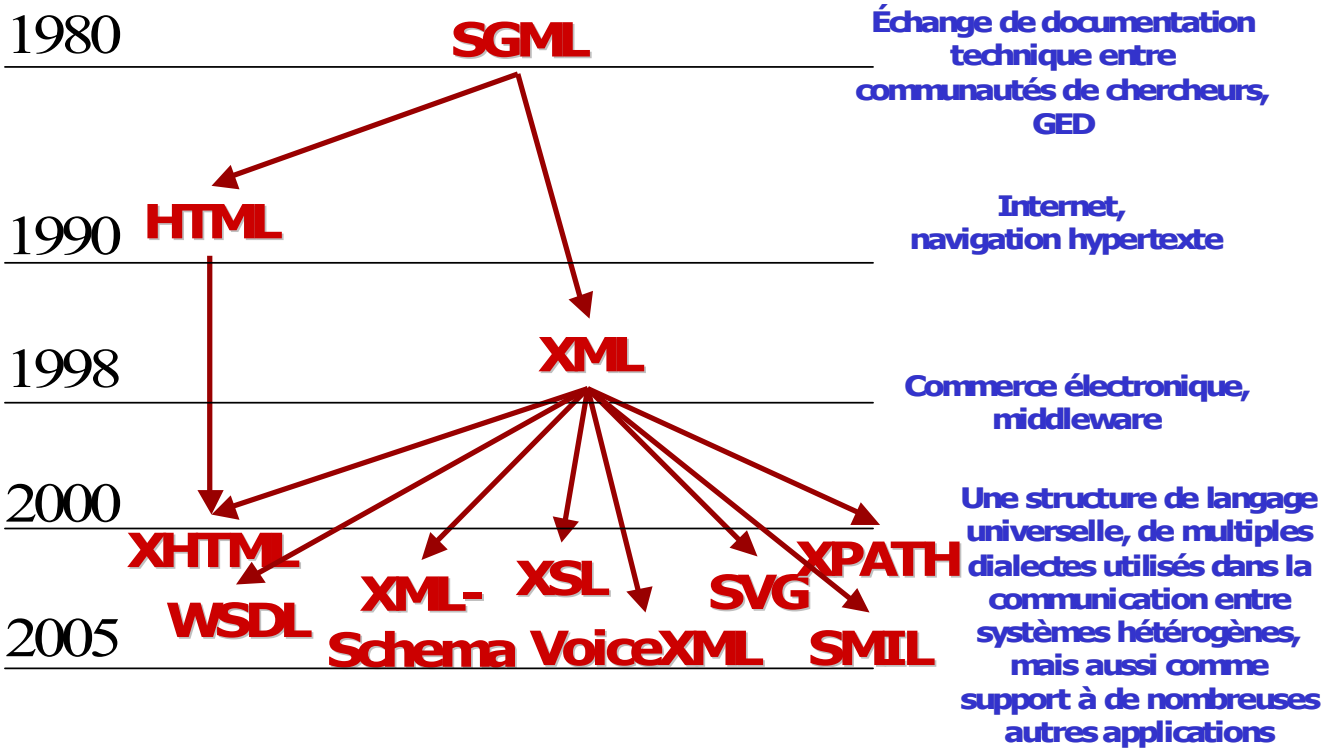

Figure 1 : évolution des langages à balises

2-Syntaxe XML

Un **document XML** permet de '**transporter' un ensemble de données ayant une structure complexe**. Il possède une structure hiérarchique décrivant le lien de composition entre un élément racine et les données qu'il porte, celles-ci pouvant être à leur tour des éléments composés d'autres éléments, etc., jusqu'à l'obtention des éléments simples porteurs des données élémentaires.
Un élément possède un nom ; il peut être **décrit par des attributs** porteurs de valeur et **son contenu est encadré** par un balise de début et une balise de fin :

[1] http://fr.wikipedia.org/wiki/Cat%C3%A9gorie:Dialecte_XML

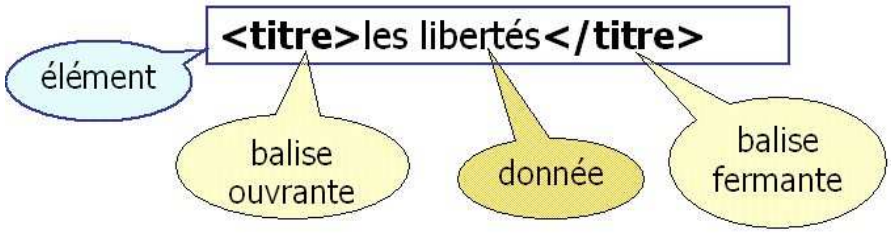

Figure 2 : un élément XML, balisé, porteur de données

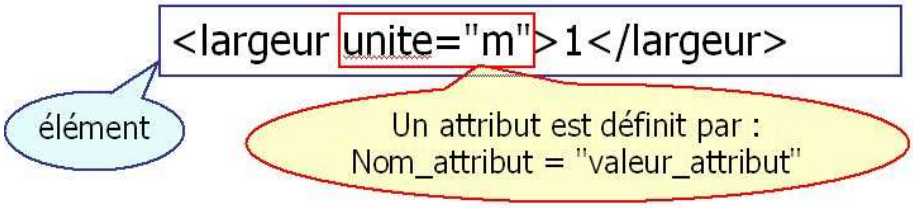

Figure 3 : un élément XML avec un attribut

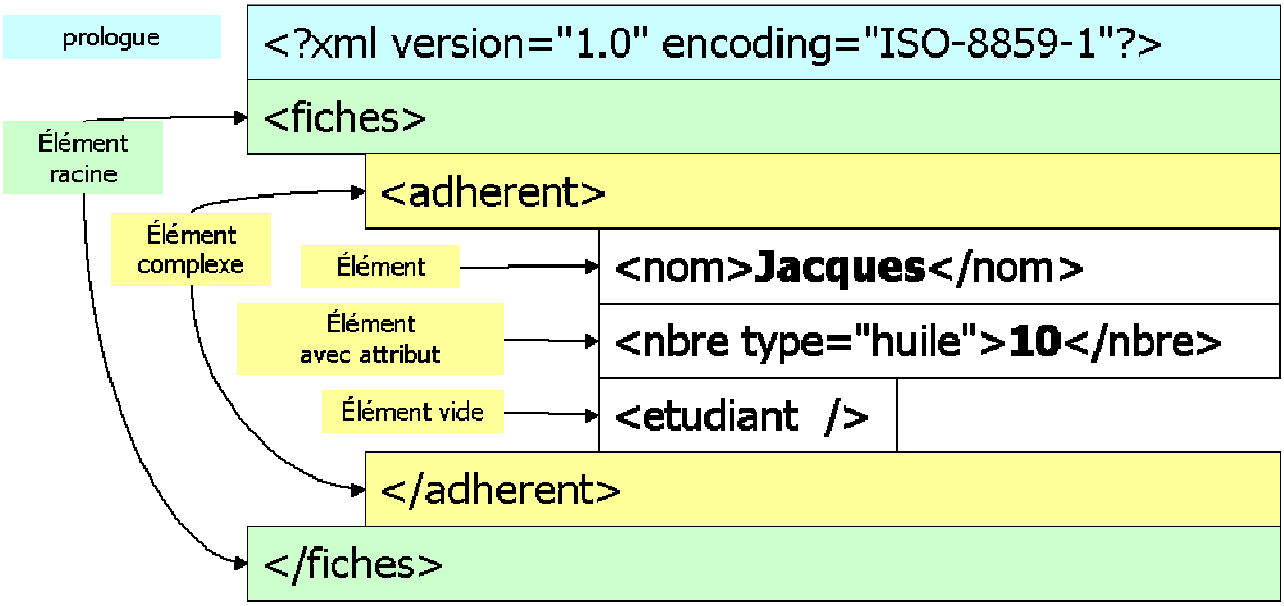

Figure 4 : structure général d'un document XML

3-Un document XML 'BIEN FORME' ou 'CONFORME'

Un document XML est dit '**BIEN FORME**' ou 'CONFORME' (*anglais : well formed document*) lorsqu'il respecte scrupuleusement les règles définies ci-dessous :

- **Un élément racine** encadre tous les autres éléments
- Les **noms des éléments sont corrects**: lettres, chiffres, tiret, trait de soulignement, point et deux-points (dans certains cas) (**pas d'espaces dans le nom, commence par une lettre**)
- Les **noms XML** sont **sensibles à la casse** (majuscule / minuscule)
- **Chaque balise de début d'élément possède une balise de fin d'élément** (ou bien un indicateur de fin dans le cas d'un élément vide)
- **Les balises des éléments ne se chevauchent pas**
- **Les données ne contiennent pas de caractères réservés**

4-Un document XML 'VALIDE'

Un document XML est dit '**VALIDE**' lorsque lui est **associé une description qu'il respecte**. La description est spécifiée soit sous forme d'une DTD ou d'un Schéma XML.

A-DTD, Document Type Definition, ou définition d'un type de document

La **DTD décrit la structure arborescence d'un document XML** de manière précise, en précisant les liens de composition en terme du nombre d'occurrences d'un élément fils pour un élément père. La présence d'attributs est également décrite.
Cependant ce langage, encore très utilisé aujourd'hui, est encore un nouveau langage et ne permet pas de décrire de manière assez précise les données.

B-XML Schéma (XSD)

Le **schéma XML va étendre les possibilités de description de la DTD** en les exprimant **dans le langage XML**, ce qui fait tout son intérêt. Cela va en effet permettre d'adjoindre à chaque élément des informations plus précises relatives au nombre d'occurrences et au type des données transportées.

5-Exploiter un document XML

La production d'un document XML est peu complexe mais son exploitation, pour de multiples usages, va nécessiter 2 types d'outils : le parseur et le render.

A-le 'parser'

Le 'parser' (parseur) est le programme qui sait lire un document XML et **identifier sa structure**, soit la totalité de **l'arborescence du document** (DOM, Document Object Model), soit l'indication des **début et fin de chaque élément** (SAX, Simple API for XML).
Certains parseurs sont capables de valider un document en utilisant la DTD, ou le schéma XML, qui lui est attaché.

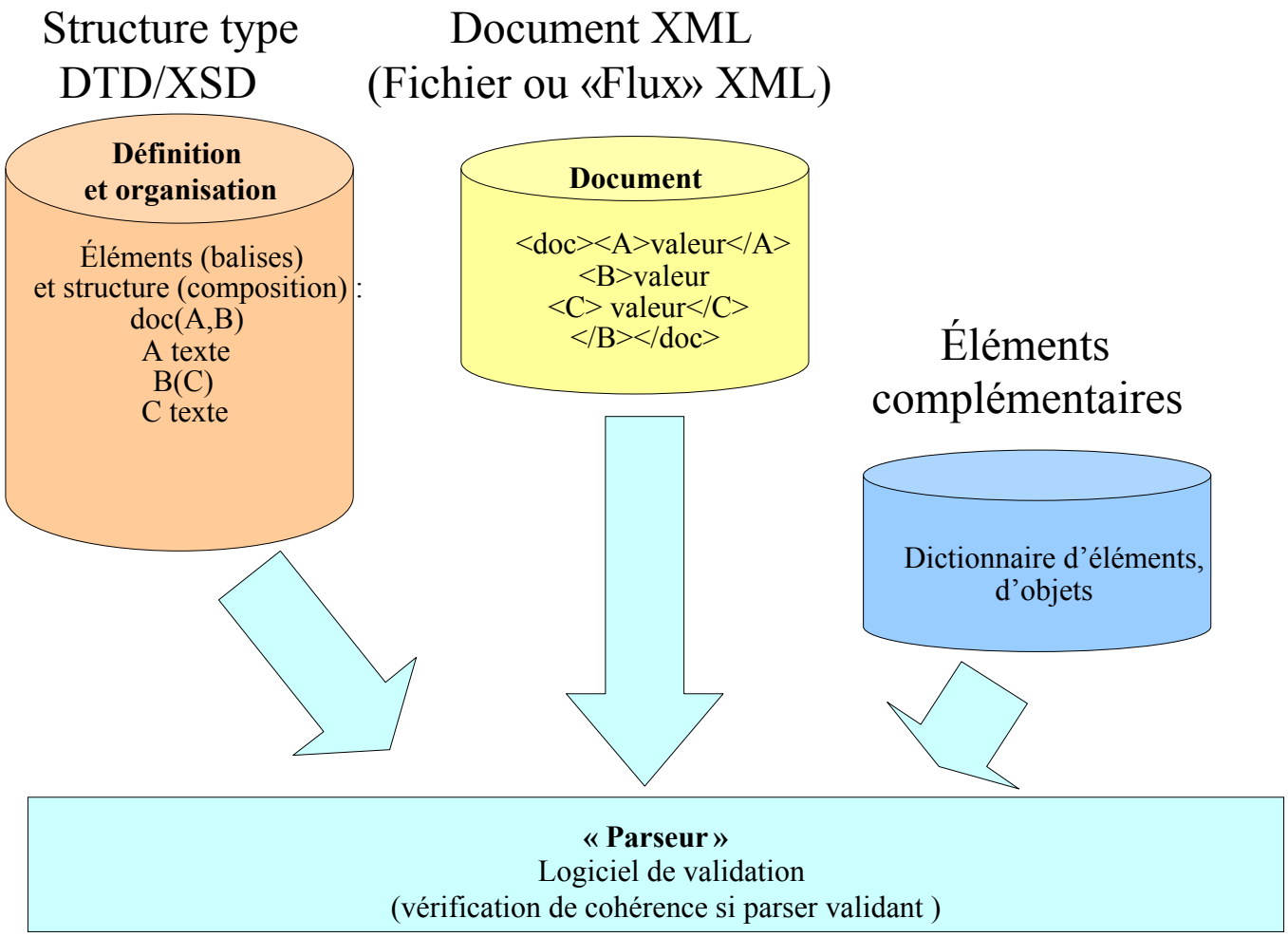

Figure 5 : le parseur

B-le 'render', ou 'moteur de rendu'

Une fois le document XML valide chargé, le 'moteur de rendu' va être capable d'**assurer un rendu en fonction du dialecte XML** utilisé. On trouvera donc autant de programmes de rendus que de dialectes XML.

6-Exemples d'utilisation

Parmi les dialectes XML les plus en vogue aujourd'hui :

- **XHTML** : le HTML avec la rigueur de XML (render = IE6, Mozilla Firefox,etc.)
- **SVG** : format d'images vectorielles (render = logiciels de dessin comme GIMP)
- **SMIL** : format d'animation (render = lecture de flux audio/video comme RealPlayer)
- **XSL** : langage de transformation de document XML en d'autres types de documents (textes, XML, HTML, PDF, etc.) (render = le moteur XSL ou XSL-Formatting Objects)

7-Complément - CDATA

Il est possible de saisir une données avec des caractères spéciaux dans un document XML, en précisant que ces données ne seront pas des PCDATA (parsed characters data) mais des CDATA (characters data), ces dernières n'étant pas analysées par le parseur.
C'est ce qui permet de définir d'une manière 'propre' certains éléments d'un code XHTML.
On pourra saisir dans un document, encadré par **<![CDATA[**. . . **]]>**, par exemple :

```
. . .
<script>
<![CDATA[
. . .
du code Javascript avec utilisation des caractères spéciaux : <, >,
etc.
. . .
]]>
</script>
```

8-Références sur Internet

http://www.w3c.org : les normes relatives à XML et ses dialectes, des liens vers des traductions françaises et vers les outils qui exploitent les dialectes XML.

Annexe : un document XML bien formé et valide

Exemple d'une compilation de CD :

Le document XML (fichier compil.xml) :

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<COMPILATION>
     <ALBUM>
          <NOM>The Silent Force </NOM>
          <TITRE>Memories </TITRE>
          <DUREE>03min51s </DUREE>
          <INTERPRETE>Within Temptation </INTERPRETE>
     </ALBUM>
     <ALBUM>
          <NOM>At Sixes and Sevens </NOM>
          <TITRE>Meridian </TITRE>
          <DUREE>06min22s </DUREE>
          <INTERPRETE>Sirenia </INTERPRETE>
     </ALBUM>
     <ALBUM>
          <NOM>Wishmaster </NOM>
          <TITRE>SheIsMySin </TITRE>
          <DUREE>04min46s </DUREE>
          <INTERPRETE>Nightwish </INTERPRETE>
     </ALBUM>
     <ALBUM>
          <NOM>Satellite </NOM>
          <TITRE>Messenjah </TITRE>
          <DUREE>04min19s </DUREE>
          <INTERPRETE>POD </INTERPRETE>
     </ALBUM>
</COMPILATION>
```

➔ **le document XML est bien formé** (Cf. 3)

Le fichier DTD suivant définit les règles d'agencement des éléments :

Le document DTD (fichier compil.dtd) :

```
<!ELEMENT COMPILATION (ALBUM*)
<!ELEMENT ALBUM (NOM, TITRE, DUREE, INTERPRETE)>
<!ELEMENT NOM (#PCDATA)>
<!ELEMENT TITRE (#PCDATA)>
<!ELEMENT DUREE (#PCDATA)>
<!ELEMENT INTERPRETE (#PCDATA)>
```

➔ **le document XML est conforme à cette DTD** (Cf. 4)

L'élément racine est COMPILATION ; il comporte plusieurs éléments ALBUM ; l'élément ALBUM comporte les éléments NOM, TITRE, DUREE, INTERPRETE, une et une seule fois ; chacun des ces éléments contient des données caractères.