Les règles syntaxiques de XML

Alexandre Roulois (Université Paris-Cité, IRIF, CNRS)

Table of contents

Présentation

XML : *Extensible Markup Language*

26 novembre 2008 : XML 1.0
https://www.w3.org/TR/REC-xml

16 août 2006 : XML 1.1 (pour des cas très spécifiques)
https://www.w3.org/TR/xml11

- format personnalisé (vocabulaire, grammaire)
- format normalisé : HTML, SVG, MathML…
- format tiers : TEI, TMX…

Analyse d'un fichier XML

API : *Application Programming Interface*

Deux API principales :

- **DOM :** *Document Object Model*
- **SAX :** *Simple API for XML*

Traitement d'un fichier XML :

1. parseur lit le document comme texte brut
2. modélisation par API (DOM, SAX…)
3. validation syntaxique
4. validation grammaticale
5. interrogation par un programme

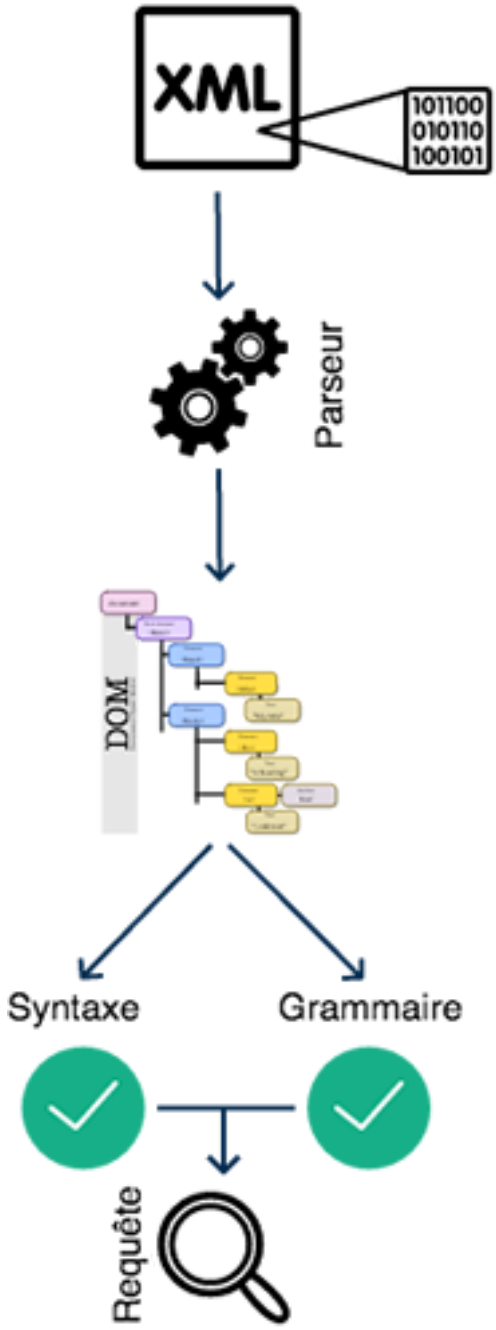


Figure 1: Analyse d'un fichier XML

Syntaxe

7 blocs de construction :

- Prologue XML
- Éléments
- Attributs
- Données textuelles (CDATA)
- Références d'entités
- Commentaires
- Instructions de traitement

Prologue XML

Facultatif… mais chaudement recommandé !

Indique aux applications :

- format (XML)
- version (1.0)

```
<?xml version="1.0"?>
```

Préciser l'encodage si différent de l'ASCII (255 caractères) :

```
<?xml version="1.0" encoding="utf-8"?>
```

Éléments

Composant central

Au minimum présent une fois

Premier élément obligatoirement unique : élément racine

Composé de trois parties : 1. Balise d'ouverture (`<sentence>`) 2. Contenu (tout bloc à l'exception de la déclaration) 3. Balise de fermeture (`</sentence>`)

```
<!-- Full example -->
<sentence>Lorem ipsum dolor sit amet.</sentence>
```

Cas particulier d'un élément vide (ou autonome) :

```
<!-- This element is empty -->
<sentence/>
```

Nom personnalisable, à l'exception de la chaîne `XML` :

```
<!-- An element starting with 'XML' is forbidden -->
<XMLsomething>Cette balise est interdite</XMLsomething>
```

- **Début :** lettre, nombre, underscore ou deux-points (déconseillé)
- **Suite :** lettres, nombres, tirets, underscores, deux-points, virgules

Sensible à la casse :

```
<docXML>Erreur de syntaxe</docXml>
```

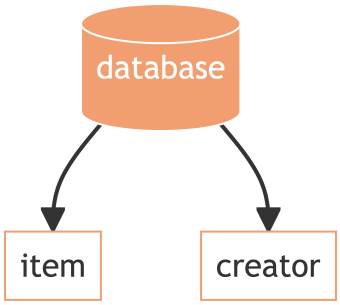
Respecter l'imbrication des balises :

```
<!-- This will provoke a syntax error -->
<database>
  <item>Mirror glass</database>
  <creator>Paul Smith</creator>
</item>
```

Impossible de représenter avec un graphe !

```
<!-- Hierarchy has been respected -->
<database>
  <item>Mirror glass</item>
  <creator>Paul Smith</creator>
</database>
```

Représentation possible avec un graphe :



Attributs

Attributs placés dans la balise d'ouverture

Autant d'attributs différents que souhaité

```
<!-- Not recommanded -->
<text author="Jacob Grimm" author="Wilhelm Grimm">
  <title>Le vaillant petit tailleur</title>
</text>
```

Exprimés sous forme de paire :
`name="value"`

Caractères interdits (réservés par le langage) :
`< & ' "`

Ordre indifférent

```
<book pages="240" author="Éric Chevillard" publisher="Minuit">
  <title lang="fr">Le vaillant petit tailleur</title>
</book>
```

Texte

Texte brut contenu dans un élément ou attribut

Caractères formellement interdits : `<` `&`

- Utiliser une référence d'entité (`<` `&`…)

```
<!-- Usage example of entity 'lower than' -->
<text>L'âge de Simon est strictement &lt; au coût de la vie.</text>
```

- Ou une section CDATA

```
<!-- CDATA section -->
<text>
  <![CDATA[
    L'âge de Simon est strictement < au coût de la vie.
  ]]>
</text>
```

Entités

Références d'entités pour insérer les caractères réservés : `&…;`

Entité	Caractère
`<`	<
`>`	>
`&`	&
`&apos;`	'
`&quot;`	"

Correspondances Unicode reconnues :

- Décimale : `&#…;`
- Hexadécimale : `&#x…;`

Possibilité de déclarer ses propres entités dans une grammaire

---

Commentaires

Respectent la syntaxe des commentaires HTML

```
<!-- Éric Chevillard's version of the 'Vaillant petit tailleur' is without a doubt the most
```

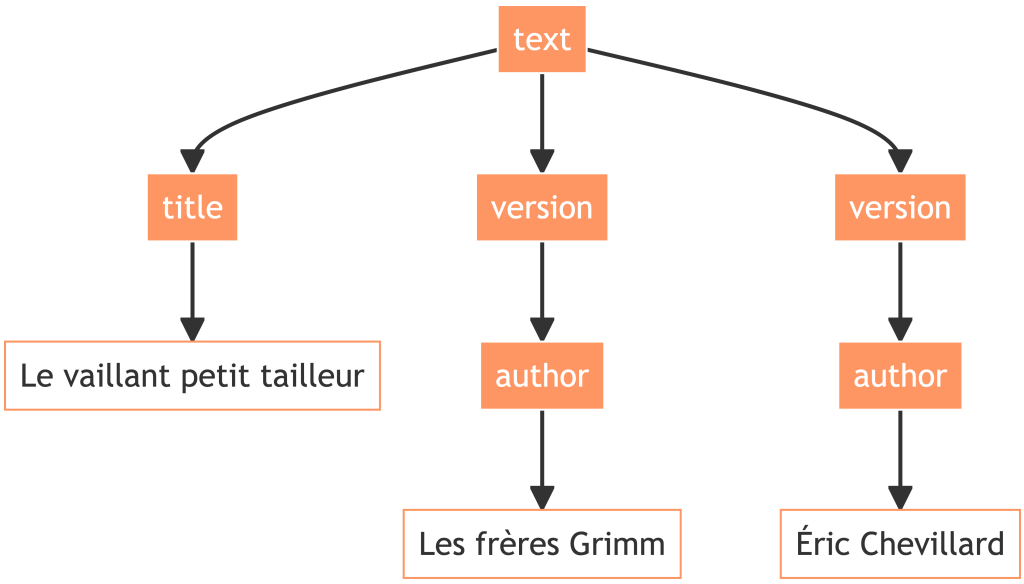
Séquence `--` interdite dans les commentaires

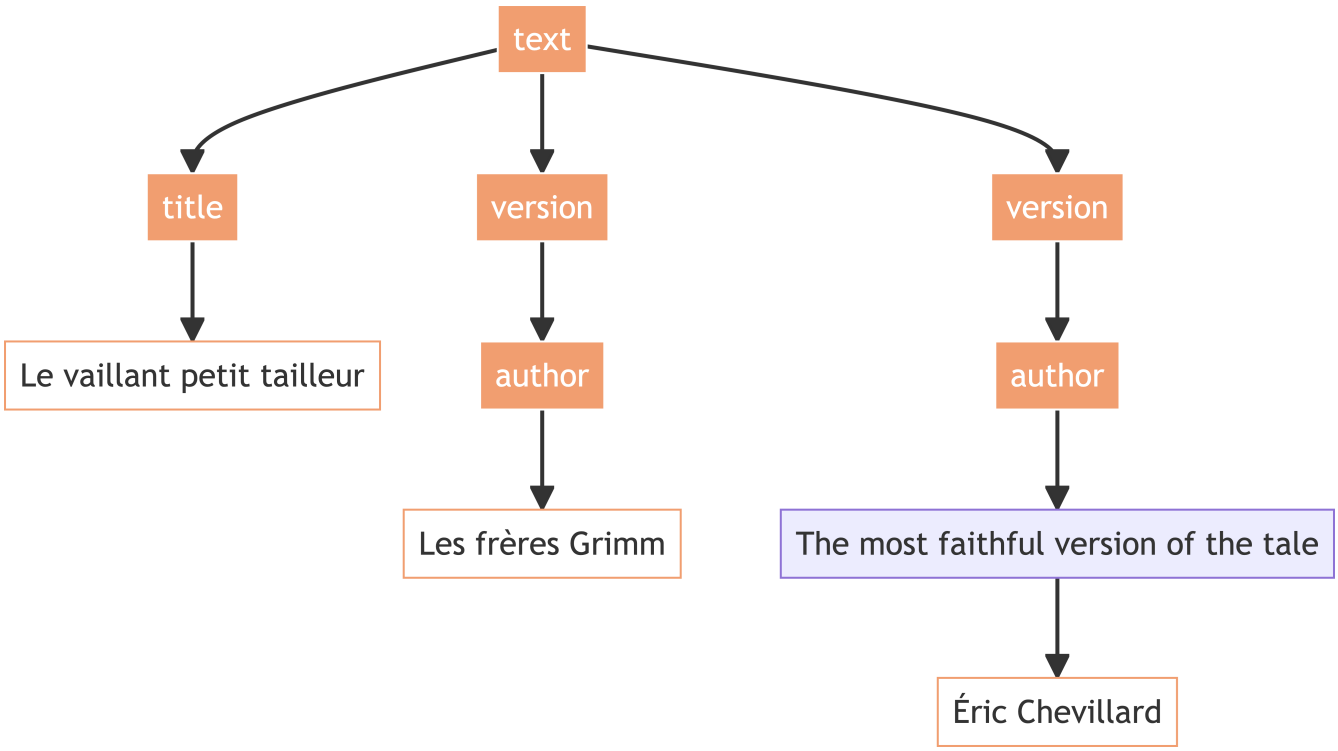
```
<!-- Include two -- will provoke a syntax error -->
```

Attention ! Traitement différent selon le parseur :

```
<text>
  <title>Le vaillant petit tailleur</title>
  <version>
    <author>Les frères Grimm</author>
  </version>
  <version>
    <!-- The most faithful version of the tale -->
    <author>Éric Chevillard</author>
  </version>
</text>
```

Selon le parseur, deux interprétations différentes :





Instructions de traitement

Pour indiquer une action à une application de traitement

Composée de quatre parties :

1. Balise ouvrante : `<?`
2. Cible du traitement (ne peut débuter par la chaîne `XML`)
3. Chaîne d'information complémentaire
4. Balise fermante : `?>`

```
<!-- A transformation sheet -->
<?xml-stylesheet type="text/xml" href="styles.xsl"?>
```

Espaces de nommage

Ajoute des fonctionnalités à la spécification XML 1.0

Permet la coexistence de plusieurs vocabulaires XML

Déclaration obligatoire :

- dans l'élément qui l'utilisera (attention à la portée)
- avec une URI comme valeur de l'attribut `xmlns` (*XML namespace*)

```
<!-- 'xlink' namespace is declared in this text element -->
<text xmlns:xlink="http://www.w3.org/1999/xlink">
  Un lien vers l'infini et au-delà !
</text>

<!-- This one is out of reach! -->
<text xlink:href="lien.xml"/>
```

```
<!-- 'xlink' namespace is declared in the parent node -->
<data xmlns:xlink="http://www.w3.org/1999/xlink">
  <text>Un lien vers l'infini et au-delà !</text>
  <lk xlink:href="resource.xml"/>
</data>
```

XML design

Données stockées indifféremment dans des éléments ou des attributs

Modèle par éléments

```
<?xml version="1.0" encoding="UTF-8"?>
<cheeses>
  <cheese>
    <name>Beaufort</name>
    <milk>
      <type>Cru</type>
      <animal>Vache</animal>
    </milk>
  </cheese>
  <cheese>
```

```
    <name>Crottin de Chavignol</name>
      <milk>
        <type>Cru</type>
        <animal>Chèvre</animal>
      </milk>
  </cheese>
</cheeses>
```

Modèle par attributs

```
<?xml version="1.0" encoding="UTF-8"?>
<cheeses>
  <cheese name="Beaufort" typeMilk="cru" animal="vache" />
  <cheese name="Crottin de Chavignol" typeMilk="cru" animal="chèvre" />
</cheeses>
```

- plus concis
- moins lisible avec un volume de données très important
- modèle mixte possible !

Comment justifier l'un ou l'autre des modèles ?

- sensibilité personnelle
- expérience

Questions pour orienter le design :

- principe de la donnée fondamentale
- principe de l'information structurée
- principe de dépendance entre l'élément et ses attributs
- principe de lisibilité des données

Le contenu est-il fondamental ou accessoire ?

La réponse à cette question doit aboutir à une distinction essentielle entre donnée et métadonnée. Dans le premier cas, il convient d'utiliser un élément ; dans le second, un attribut.

```
<!-- The genre of the book is regarded here as non-essential information -->
<book genre="roman">
  <!-- Titles should always been stored in elements -->
  <title>Moby Dick</title>
</book>
```

La donnée répond-elle à une norme, un format ?

Toutes les informations qui suivent un formatage défini mériteraient d'être encodées dans des attributs.

```
<!-- Publication date complies with ISO-8601 standard -->
<book genre="roman" pubDate="1851-11-14">
  <title>Moby Dick</title>
</book>
```

Le contenu est-il composé d'une donnée et de son modificateur ?

L'une des distinctions entre attributs et éléments réside dans l'utilité des premiers pour modifier ou affiner la précision des seconds.

```
<book genre="roman" pubDate="1851-11-14">
  <title>Moby Dick</title>
  <!-- The attribute retains the unit that modifies the stored numeric value -->
  <pages unit="p">379</pages>
</book>
```

Le contenu sera-t-il lu par un agent humain ?

Les données sont soit destinées à être consultées par un agent humain, soit destinées à être traitées par un programme informatique. Caractériser leur sort permet de déterminer s'il est préférable de les encoder dans des éléments ou avec des attributs.

```
<book genre="roman" pubDate="1851-11-14">
  <title>Moby Dick</title>
  <pages unit="p">379</pages>
  <!-- The summary of a book is intended more for display than for processing -->
  <summary>Attiré par la mer et le large, Ismaël, le narrateur, décide de partir à la chasse
</book>
```

XML comme base de données

Conversion directe d'un document n'est pas le but de XML

- extraction de données afin de séparer le fond de la forme
- identifier les segments à stocker
- normaliser les données

Catalogue de bibliothèque

Moby Dick

Genre : Roman

Date de publication : 14 novembre 1851

Nombre de pages : 379

Résumé : Attiré par la mer et le large, Ismaël, le narrateur, décide de partir à la chasse à la baleine. Il embarque sur le Péquod, baleinier avec son nouvel ami Queequeg…

```
<catalog>
  <books>
    <book genre="roman" pubDate="1851-11-14">
      <title>Moby Dick</title>
      <pages unit="p">379</pages>
      <summary>Attiré par la mer et le large, Ismaël, le narrateur, décide de partir à la cha
    </book>
  </books>
</catalog>
```