Retour sur la DTD et présentation de l’en-tête TEI

Séance 4

Master 2 - UVSQ - 2024-25

Retour sur la DTD

Contraindre le XML

Un document valide

Un document XML est "valide" lorsqu'il est non seulement bien formé, mais qu'il respecte aussi les contraintes définies par une DTD ou un autre schéma de validation.

La DTD définit quelles balises sont permises, comment elles peuvent être imbriquées, et quels attributs sont autorisés ou obligatoires. Si le document respecte toutes ces règles, il est considéré comme valide.

Déclaration de la DTD

La déclaration de la DTD peut être faite au **début du document** XML dont elle contraint l'encodage, ou dans un **fichier externe** dont il est fait référence dans le document XML

Déclaration de la DTD Interne

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

<!DOCTYPE book [
  <!ELEMENT book (title, author)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
  <!ATTLIST author ref CDATA #IMPLIED>
]>

<book>
  <title>Les Misérables</title>
  <author ref="https://fr.wikipedia.org/wiki/Victor_Hugo">Victor Hugo</author>
</book>
```

Contenu d’une DTD

Une DTD contient des déclarations concernant :

- **Éléments**: Définit les éléments qui peuvent être utilisés, leur nombre, leur imbrication, leur ordre, etc.
- **Attributs**: Définit les attributs autorisés pour un élément précis, leur valeurs et le type de valeur autorisé
- **Entités**: Définit les entités qui pourront être utilisées dans le document XML

L’<élément> dans la DTD

1. Définir un Élément dans une DTD

Syntaxe Générale :

```
<!ELEMENT nom_de_l_element type_de_contenu>
```

Exemple :

```
<!ELEMENT title (#PCDATA)>
```

Donc par exemple dans le document XML-TEI :

```
<title>Les Misérables</title>
```

L'<élément> dans la DTD

2. Types de contenus des éléments

Expression	Signification
`EMPTY`	Contenu vide
`ANY`	Contenu quelconque
`(#PCDATA)`	Contenu textuel
`(elt)`	Un seul élément
`(elt1, elt2, ..., eltn)`	Séquence d'éléments pris dans cet ordre
`(elt1 | elt2 | ... | eltn)`	Un des éléments au choix

L’<élément> dans la DTD

3. Les opérateurs d’occurence

Opérateur	Signification
`?`	0 ou 1 occurrence
`*`	0 ou plusieurs occurrences
`+`	1 ou plusieurs occurrences
(rien)	1 seule occurrence

L’attribut dans la DTD

1. Définir un Attribut dans une DTD

Syntaxe Générale :

```
<!ATTLIST nom_de_l_element nom_de_l_attribut type_de_l_attribut
valeur_par_défaut_ou_contrainte>
```

Exemple :

```
<!ELEMENT title (#PCDATA)>
<!ATTLIST title lang CDATA #IMPLIED>
```

Exemple de Document XML-TEI :

```
<title lang="fr">Les Misérables</title>
```

L’attribut dans la DTD

2. Type de l’attribut

Expression	Signification
CDATA	chaîne de caractères ne comprenant pas de balises
(val1 | val2 | ...)	liste de valeurs à utiliser
ENTITY / ENTITIES	entité déclarée dans la DTD (ou liste séparée par des espaces)
ID	pour identifier l’élément
IDREF / IDREFS	ID d’un autre élément (ou liste séparée par des espaces)

L’attribut dans la DTD

3. Contrainte et valeur

Opérateur	Signification
`#REQUIRED`	valeur requise dans l’élément
`#IMPLIED`	valeur facultative
`#FIXED "valeur"`	valeur fixe pour l’attribut
`"valeur"`	valeur par défaut de l’attribut (on peut la remplacer)

L’entité dans la DTD

1. Définir un Entité dans une DTD

Syntaxe Générale :

```
<!ENTITY nom_de_l_entité "définition_de_l_entité">
```

L'entité dans la DTD

2. Utiliser des entités prédéfinies

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

Exemple avec l'esperluette :

```
<text>Il est important de comprendre les signes < et > dans un document XML.
Utilisez &amp; pour représenter le caractère &.</text>
```

L'entité dans la DTD

3. Définir un Entité dans une DTD

DTD :

```
<!ENTITY uvsq "Université de Versailles Saint-Quentin-en-Yvelines">
<!ELEMENT p (#PCDATA)>
```

Exemple de Document XML

Document XML :

```
<!DOCTYPE university [
  <!ENTITY uvsq "Université de Versailles Saint-Quentin-en-Yvelines">
]>
<p>Bienvenue à &uvsq; !</p>
```

Le résultat affiché sera :

```
<p>Bienvenue à Université de Versailles Saint-Quentin-en-Yvelines !</p>
```

Exercice : proposez une DTD pour le fichier séance3/exemple-complexe.xml

Présentation de l’en-tête TEI

Présentation de l’en-tête TEI

Ce que fait le teiHeader

- Identifie et décrit le texte
- Documente l'encodage
- Fournit un contexte
- Gère les droits et la publication

Structure d’un document XML-TEI

Un document XML-TEI a une structure simple et organisée, comprenant principalement deux parties principales :

1. **teiHeader** : Cette section se trouve au début du document et contient toutes les métadonnées. Elle décrit le texte, son auteur, l'encodage, les sources, les personnes impliquées, et d'autres informations contextuelles.

2. **text** : Cette section contient le contenu principal du document, c'est-à-dire le texte lui-même. Le texte peut être structuré en plusieurs sous-sections :

   - **front** (facultatif) : Pour un éventuel préambule, introduction ou autre contenu avant le corps du texte.

   - **body** : La partie principale du texte.

   - **back** (facultatif) : Pour des annexes, notes, ou autres contenus après le corps du texte.

Structure d’un document XML-TEI

```xml
<TEI xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
        <!-- Informations sur le texte (métadonnées) -->
    </teiHeader>
    <text>
        <body>
            <!-- Contenu principal du texte -->
        </body>
    </text>
</TEI>
```

Le <teiHeader>comporte quelques éléments obligatoires.

1. **fileDesc** (Description du fichier) :

C'est l'unique élément obligatoire du teiHeader. Il doit contenir des informations sur le titre, l'auteur, la publication, et la source du texte.

- Sous-parties principales :
  - **titleStmt** : Contient le titre de l'œuvre et les noms des personnes ou institutions responsables.
  - **publicationStmt** : Donne les informations sur la publication du document, comme l'éditeur et la date de publication.
  - **sourceDesc** : Décrit la source d'origine du texte encodé, qu'il s'agisse d'un livre, d'un manuscrit, ou d'une autre forme de document.

```xml
<TEI xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title>Titre de l'œuvre</title>
                <author>Nom de l'auteur</author>
            </titleStmt>
            <publicationStmt>
                <publisher>Nom de l'éditeur</publisher>
                <date when="2024-08-29">Date de publication</date>
            </publicationStmt>
            <sourceDesc>
                <biblStruct>
                    <monogr>
                        <title>Titre de la source originale</title>
                        <author>Auteur de la source</author>
                        <imprint>
                            <pubPlace>Lieu de publication</pubPlace>
                            <publisher>Éditeur original</publisher>
                            <date when="2000">Date de publication originale</date>
                        </imprint>
                    </monogr>
                </biblStruct>
            </sourceDesc>
        </fileDesc>
    </teiHeader>

    <text>
        <body>
            <p>Contenu principal du texte.</p>
        </body>
    </text>
</TEI>
```

Exemple du <sourceDesc>

Comment citer un livre

Il existe différents niveaux de détails :

Expression	Signification
`bibl`	Référence bibliographique peu structurée dont les sous-composants peuvent être ou non balisés
`biblFull`	Référence bibliographique totalement structurée
`biblStruct`	Une référence bibliographique structurée en sous-éléments bibliographiques dans un ordre déterminé
`msDesc`	Description d'un unique manuscrit
`monogr`	Sous-éléments bibliographiques à insérer dans `biblStruct`
`analytic`	Sous-éléments bibliographiques à insérer dans `biblStruct`

Exemple du <sourceDesc>

Comment citer un livre?

Référence bibliographique non-structurée :

Il s'agit de la méthode la plus simple, elle est à éviter en général.

```
<bibl>Notre-Dame de Paris, Texte établi par Paul Meurice, Librairie Ollendorff, 1904, [volume 1]
[Section A.] Roman, tome II. (p. 43-84)</bibl>
```

Il est mieux de préciser les éléments apportés:

```
<bibl>
    <title type="main">Notre-Dame de Paris</title>
    <title type="subtitle">Livre Deuxième</title>
    <author>
        <forename>Victor</forename>
        <surname>Hugo</surname>
    </author>
    <editor>Paul Meurice</editor>
    <publisher>Librairie Ollendorff</publisher>
    <distributor facs="https://fr.wiki...">
        Wikisource
    </distributor>
</bibl>
```

Exemple du <sourceDesc>

Exercice

En vous appuyant sur le template fourni, renseignez au maximum le **<sourceDesc>** pour une édition de Notre Dame de Paris.

Source pour l'encodage du **<sourceDesc>** :

- https://gallica.bnf.fr/ark:/12148/bpt6k1264641j/f12.image
- https://fr.wikisource.org/wiki/Notre-Dame_de_Paris/Texte_entier

Pour des informations sur la balise **<biblFull>**: https://tei-c.org/release/doc/tei-p5-doc/en/html/HD.html#HD3

Présentation d’un <teiHeader> minimaliste

La taille du `teiHeader` varie selon le niveau de détail souhaité. Nous présentons ici un `teiHeader` peu développé.

Quatre niveaux de description

1. `fileDesc` **(Description du fichier)** :
    - C'est la partie la plus importante. Elle inclut des informations sur le titre, l'auteur, la publication, et la source du texte.
    - Sous-parties principales :
        - `titleStmt` : Contient le titre de l'œuvre et les noms des personnes ou institutions responsables.
        - `publicationStmt` : Donne les informations sur la publication du document, comme l'éditeur et la date de publication.
        - `sourceDesc` : Décrit la source d'origine du texte encodé, qu'il s'agisse d'un livre, d'un manuscrit, ou d'une autre forme de document.
2. `encodingDesc` **(Description de l'encodage)** :
    - Donne des informations sur le processus d'encodage, les règles et conventions utilisées.
3. `profileDesc` **(Description du profil)** :
    - Fournit des informations sur le contenu du texte, les participants, les langues utilisées, les lieux mentionnés, etc.
4. `revisionDesc` **(Historique des révisions)** :
    - Enregistre l'historique des modifications apportées au document encodé.

Présentation d’un <teiHeader> minimaliste

Voir le document header-min.xml

Exercice

Compléter au maximum le document **template-header.xml** afin qu'il corresponde à un encodage fictif des *Mystères de Paris* d'Eugène de Sue en vous appuyant sur la notice de catalogue, en supprimant les éléments non pertinents pour ladite édition.

Lien

Vers les Mystères de Paris d'Eugène de Sue : https://gallica.bnf.fr/ark:/12148/bpt6k4458735.item

Vers le catalogue de la Bibliothèque nationale de France : https://catalogue.bnf.fr/ark:/12148/cb39294634r

Comprendre les composantes du <profileDesc>

Pour rappel, le **<profileDesc>** est utilisé pour décrire différents aspects du texte, tels que l'utilisation des langues, les classifications thématiques, les personnes et lieux mentionnés, ainsi que les événements historiques évoqués. Cette section du TEI header permet de contextualiser le texte, enrichissant ainsi sa valeur pour la recherche et l'analyse.

Comprendre les composantes du <profileDesc>

1. `<langUsage>` : Utilisation des Langues

Cet élément vous permet de spécifier les langues utilisées dans le texte. Dans un document multilingue, cela est utile pour indiquer quelles parties sont écrites dans quelles langues.

Exemple :

```
<langUsage>
    <language ident="fr">Français</language>
    <language ident="en">Anglais</language>
</langUsage>
```

Comprendre les composantes du <profileDesc>

2. `<textClass>` : Classification du Texte

L'élément `<textClass>` permet de catégoriser le texte selon des systèmes de classification standards ou personnalisés, ce qui permet d'organiser et retrouver des documents dans de grandes collections.

Exemple :

```
<textClass>
    <keywords>
        <term>Correspondance</term>
        <term>Analyse littéraire</term>
        <term>Épistolaire</term>
    </keywords>
</textClass>
```

Comprendre les composantes du <profileDesc>

3. `<particDesc>` : Description des Personnes

Cet élément permet de décrire les personnes mentionnées dans le texte, en fournissant des informations détaillées sur leur identité, rôle, et autres caractéristiques pertinentes.

Exemple :

```
<particDesc>
    <listPerson>
        <person xml:id="p01" sex="1" role="auteur">
            <persName>Jean Dupont</persName>
            <birth when="1975-05-15" place="Paris, France"/>
            <occupation>Écrivain</occupation>
            <note>Jean Dupont est l'auteur principal de la correspondance encodée.</note>
        </person>
        <person xml:id="p02" sex="2" role="destinataire">
            <persName>Marie Durand</persName>
            <birth when="1980-09-10" place="Lyon, France"/>
            <occupation>Historienne</occupation>
            <affiliation>Université de Lyon</affiliation>
            <note>Marie Durand est la destinataire de la majorité des lettres encodées.</note>
        </person>
    </listPerson>
</particDesc>
```

Comprendre les composantes du <profileDesc>

4. `<settingDesc>` : **Description des Lieux**

L'élément `<settingDesc>` est utilisé pour décrire les lieux mentionnés ou impliqués dans le texte, souvent essentiel pour comprendre le contexte géographique.

Exemple :

```
<settingDesc>
    <setting>
        <place>
            <placeName ref="#Paris">Paris</placeName>
            <placeName ref="#Lyon">Lyon</placeName>
            <note>Les lettres échangées mentionnent fréquemment ces deux villes.</note>
        </place>
    </setting>
</settingDesc>
```

Comprendre les composantes du <profileDesc>

5. `<listEvent>` : **Description des Événements**

Cet élément permet de lister et décrire les événements historiques ou contextuels mentionnés dans le texte.

Exemple :

```
<listEvent>
    <event xml:id="ev01">
        <eventName>Révolution de 1848</eventName>
        <date when="1848"/>
        <note>Événement historique mentionné dans plusieurs lettres.</note>
    </event>
</listEvent>
```

Compléter un <profileDesc>

À l'aide de ce qui précède, complétez le **<profileDesc>** pour l’extrait de Notre-Dame(2).xml