Introduction à la TEI : Exploiter son édition numérique

Floriane Chiffoleau, Doctorante en Humanités Numériques, ALMAnaCH/Le Mans Université

URFIST de Rennes, 24-25 novembre 2022

Repository GitHub : https://github.com/FloChiff/Introduction_TEI_2022

Fouiller dans ses données (XPath et XQuery)

Qu’est-ce que le XPath ?

XPath ou XML Path Language

- Langage de requête qui permet de rechercher des informations au sein d’un fichier XML
- S’utilise avec le XQuery (extraction) et la XSLT (transformation)

Requête par chemin de localisation

- Similaire au système de navigation des ordinateurs
- Navigation au sein de la hiérarchie de l’arbre XML
- Fonctionne à l’aide d’étapes, séparées par un slash “/” → un seul signifie que l’on se déplace vers une branche directe et deux signifie que l’on saute des étapes dans la hiérarchie
- Filtrage possible à l’aide de prédicats présentés entre crochet []

Comment fonctionne le XPath ?

Deux options pour l'écriture d'une requête en XPath

- Utilisation des noms de balises (quand on a connaissance de l'arbre)
- Utilisation d'axes de navigation

Les axes de navigation en XPath

- parent ou ancestor → remonter dans l'arbre ; élément directement au-dessus (parent) ou éléments plus hauts dans la hiérarchie (ancestor)
- child ou descendant → descendre dans l'arbre ; élément directement en dessous (child) ou éléments plus bas dans la hiérarchie (descendant)
- preceding ou following → élément juste avant ou juste après dans l'arbre
- preceding-sibling ou following-sibling → Pour un élément avec un parent spécifique, tous les enfants précédents ou suivants de ce parent
- attribute/@ → attributs de l'élément appelé de l'arbre

Exemples du XPath

```xml
<bibliography>
    <book type="fantasy">
        <title xml:lang="fr">Le Seigneur des Anneaux</title>
        <title xml:lang="en">Lord of the Rings</title>
        <author>JRR Tolkien</author>
        <date>1954-1955</date>
    </book>
    <book type="fantasy">
        <title xml:lang="fr">Le Monde de Narnia</title>
        <title xml:lang="en">The Chronicles of Narnia</title>
        <author>C. S. Lewis</author>
        <date>1950-1956</date>
    </book>
    <book type="fantasy">
        <title xml:lang="fr">Percy Jackson</title>
        <title xml:lang="en">Percy Jackson and The Olympians</title>
        <author>Rick Riordan</author>
        <date>2005-2015</date>
    </book>
    <book type="polar">
        <title xml:lang="fr">Mr Brown</title>
        <title xml:lang="en">The Secret Adversary</title>
        <author>Agatha Christie</author>
        <date>1922</date>
    </book>
    <book type="polar">
        <title xml:lang="fr">Le Crime de l'Orient-Express</title>
        <title xml:lang="en">Murder on the Orient Express</title>
        <author>Agatha Christie</author>
        <date>1934</date>
    </book>
</bibliography>
```

- **//book/title** → tous les titres de livre
- **//book/title[@xml:lang="fr"]** → tous les titres de livre en français
- **//book/@type** → les types de chacun des livres
- **//book[@type="fantasy"]/author** → les auteurs des livres de type “polar”
- **//book/date[preceding::author="Agatha Christie"]** → les dates des livres où l’auteur est Agatha Christie
- **//book[@type="fantasy"]/title[@xml:lang="en"]** → les titres anglais de livres fantastiques
- **//author/following-sibling::element()** → ce qui suit l’auteur dans le parent livre, soit la date

Exercice en XPath

Ouvrez le fichier “index_places” placé dans le dossier “xpath_xquery” du dépôt du cours. En vous aidant des exemples précédemment cités et de la documentation, trouvez les éléments suivants :

- Les noms de lieux qui sont des villes
- Les coordonnées géographiques des lieux qui sont des pays
- Les identifiants des lieux qui sont des mers
- Les noms de lieux des régions françaises contenus dans l’index

Qu'est-ce que le XQuery ?

XQuery ou XML Query Language

- Langage de requête pour l'extraction d'information dans un arbre XML

Plusieurs options pour les requêtes

- FLOWR → initiales des cinq clauses qui constituent la syntaxe du XQuery
- XPath → langage à utiliser pour sélectionner les éléments voulus dans l'arbre
- Fonctions → options pour le filtrage ou l'affichage des résultats (Exemples: *concat()* pour avoir une combinaison de plusieurs résultats)

Comment fonctionne le XQuery ?

FLOWR

- For : itération sur une séquence → `for $item in Expr`
  - Permet de parcourir une séquence par l'intermédiaire d'une variable
  - La variable *$item* prend successivement la valeur de chaque élément de la séquence
- Let : définir une constante → `let $const := Expr`
  - Permet de définir une constante (variable invariable) qui prend pour valeur une séquence, un fichier, etc.
  - *$const* est la variable de stockage. := est l'opérateur d'assignation.
- Order by : Trier une séquence → `order by Expr descending|ascending`
  - Permet de trier une séquence selon un ou plusieurs critères.
  - **Order by** s'utilise dans une instruction **for**
  - **ascending** permet un tri croissant, **descending** un tri décroissant
- Where : Filtrer une séquence → `where Expr`
  - Permet de filtrer une séquence selon un ou plusieurs critères.
  - **where** s'utilise dans une instruction **for**
- Return : clore une expression FLOWR → `return Expr`
  - Permet de choisir les éléments que l'on veut voir afficher dans les résultats

Exemples de XQuery

- Requête 1 → Types de lieux (A-Z) avec des valeurs uniques et les "_" compris dans @type sont remplacés par un espace

```
let $doc := doc("/Users/fchiffol/Downloads/index_place.xml")
for $type in distinct-values($doc//place/@type)
order by $type ascending
return replace($type, '_', ' ')
```

- Requête 2 → Nom/prénom (A-Z) des femmes françaises

```
let $doc := doc("/Users/fchiffol/Downloads/index_person.xml")
for $person in $doc//person
where $person/sex/@value="2" and $person/nationality="French"
order by $person/persName/text ascending
return $person/persName[2]/text()
```

- Requête 3 → Lieux et de leur type (Z-A), sous forme d'une phrase

```
let $doc := doc("/Users/fchiffol/Downloads/index_place.xml")
for $place in $doc//place
order by $place/@type descending
return $place/placeName[1] || " is a " || $place/@type
```

Exemples de XQuery

- Requête 4 → Personne française avec leur première occupation et leur lieu de naissance (A-Z)

```
let $doc := doc("/Users/fchiffol/Downloads/index_person.xml")
for $person in $doc//person
where $person/nationality = "French"
order by $person/persName[1]/text() ascending
return concat($person/persName[1], " was a ", $person/occupation[1], " and was born in ",$person
/birth/placeName)
```

- Requête 5 → Dans les cas où un lieu de l'index des lieux correspond au lieu de naissance d'une personne de l'index des personnes, une phrase nous donnera le nom de cette personne, avec sa première occupation et le pays de son lieu de naissance

```
let $doc := doc("/Users/fchiffol/Downloads/index_place.xml")
let $doc2 := doc("/Users/fchiffol/Downloads/index_person.xml")
for $place in $doc//place
for $person in $doc2//person
where $place/placeName[1]=$person/birth/placeName
return $person/persName[1] || " was a " || $person/occupation[1] || ", born in " || $place/
country
```

Exercice en XQuery

Après avoir ouvert l'application BaseX, ouvrez les fichiers "index_places" et "index_person" placé dans le dossier "xpath_xquery" du dépôt du cours (Database>New>Browse). Avant d'appuyer sur "OK", vérifier dans "Parsing" que l'option "Strip namespaces" a bien été cochée. En utilisant les exemples précédemment donnés et la documentation, faites les requêtes suivantes :

- Depuis l'index de personnes, récupérer le nom de toutes les personnes qui étaient journalistes
- Depuis l'index des lieux, récupérer le nom de toutes les villes allemandes, classé par ville de A à Z
- Depuis l'index des personnes, récupérer les personnes qui ont reçu le prix Nobel ("Nobel Peace Prize"), en faisant une phrase qui donne leur nom et leur nationalité
- En joignant les deux index, récupérer les instances où le lieu de mort correspond à un des lieux de l'index de lieux et faites une phrase qui donne le nom de la personne, sa date de mort et son lieu (ville + pays) de mort, en classant le résultat de la mort la plus ancienne à la plus récente

Transformer ses données (XSLT)

Qu’est-ce que la XSLT ?

XSLT ou eXtensible Stylesheet Language Transformation

- Langage de programmation, et plus précisément langage de transformation, qui consiste à transformer un document XML en un autre format
- Fonctionne à partir de XPath

Exemples de format de sortie :

- XML : d’un fichier XML à un autre, par exemple pour modifier certains éléments de l’arbre pour le mettre à jour
- HTML : d’un fichier XML à un fichier HTML, pour créer des pages web qui affichent le contenu de l’arbre
- LaTeX : d’un fichier XML à un document LaTeX, pour permettre une sortie avec une mise en page maitrisée, pour être ensuite possiblement transformé en PDF

Comment fonctionne la XSLT ?

Il existe de multiples règles de transformation en XSLT, qui permettent d'arriver à divers résultats en fonction de ce qui est requêté (vous pouvez les retrouver avec des explications ici) :

- Requêter au sein de l'arbre :
  - Se situer dans l'arbre XML : <xsl:template>
  - Appliquer les règles de transformations définies pour un élément : <xsl:apply-templates>
  - Appliquer les règles de transformations définies pour un cas donné d'une partie de l'arbre : <xsl:call-templates>
- Création d'un mode pour faire apparaitre plusieurs fois la même partie de l'arbre XML, mais selon divers critères : @mode
- Faire apparaitre du contenu :
  - Chercher le contenu d'une partie de l'arbre XML, balises et textes : <xsl:copy-of>
  - Chercher le texte contenu dans les balises appelées : <xsl:value-of>
- Appel d'une boucle pour itérer sur les éléments, les uns après les autres : <xsl:for-each>
- Affichage d'un élément sous certaines conditions : <xsl:if>/<xsl:choose>/ <xsl:when>/<xsl:otherwise>

Exercice en XSLT

En vous aidant de l'exemple fourni avec les fichiers XML, XSL et HTML de "Orgueil et Préjugés" et en vous servant du *template* XSL fourni pour "Le Tour du Monde en 80 jours" et de votre fichier d'encodage, faites apparaitre les informations suivantes:

- Créer une variable et un fichier de sortie qui permettront de produire le fichier HTML "Le_tour_du_monde_en_80_jours.html" qui aura pour titre HTML "Encodage"
- Faites apparaitre, sous forme de liste labellisée, toutes les données contenues dans le <fileDesc>
- Faites apparaitre le texte après un titre "Texte" où les sauts de lignes apparaissent (balise
 en HTML), où les entités nommées sont en italiques (balise <i> en HTML) et où la foliation est visible comme un titre (balise <h6> en HTML)
- Afficher le lien du facsimilé avec l'option de cliquer dessus pour y accéder (balise <a> en HTML)

En découvrir plus avec la XSLT

À partir du fichier “Le Tour du monde en 80 jours” que vous avez encodé et que vous venez de tester, on peut également aller plus loin et produire plusieurs pages HTML en même temps, ce qui pourra nous donner les pages suivantes :

- Une page d’accueil qui contient les différentes pages
- Une page de métadonnées reprenant les informations que vous avez encodées précédemment
- Une page d’index qui présente les personnes et lieux mentionnés dans le texte, avec des liens vers des fichiers d’autorité
- Deux versions du texte : une de lecture avec le lien vers le facsimilé et une diplomatique qui reprend les fins de ligne et qui affiche également le facsimilé

Publier son corpus (TEI Publisher)

Qu’est-ce qu’est TEI Publisher ?

TEI Publisher → outil de publication numérique de corpus

- Dépend de *exist-db*, un système de gestion de base de données basé sur la technologie XML
- Conforme aux TEI Guidelines et s’appuyant sur le *TEI Processing Model*, qui documente le modèle de traitement pour un élément donné
- Transformation de fichiers XML pour un affichage web et une exportation sous divers formats (*LaTeX*, *ePUB*, *PDF*)

Génération d’une application web

- ODD : règles de transformation des éléments de l’arbre XML TEI (condition d’affichage, mode de transformation, rendu de sortie du contenu, etc.)
- Template : modèle de présentation de données, mise en page de l’affichage, diverses propositions de templates, faites par TEI Publisher

Quelques exemples concrets

- Une collection de fichiers de démonstration sur TEI Publisher
- La possibilité d'importer ses propres documents pour tester des ODD dans une aire de jeux ou pour travailler son annotation d'entités nommées
- Des applications entièrement développées à l'aide de TEI Publisher

When the Wall Came Down

Van Gogh Letters

Shakespeare's Plays

Early English Books

Digital Scholarly Editions
(Faite par moi)

Exercice pratique avec TEI Publisher

Pour cette exercice, il vous faudra avoir téléchargé le fichiers “la_reine_margot.xml” situé dans le dossier “teipublisher” du dépôt du cours et vous connecter à TEI Publisher avec les identifiants qui sont donnés sur le site.

Tester l’annotation

- ❏ Aller dans “Annotation Samples”
- ❏ Importer le fichier XML de *La Reine Margot* puis aller dans le fichier
- ❏ Encoder une des instances de “Charles IX” et dites-moi ce que vous observez
- ❏ Encoder les autres entités présentes dans le texte

Exercice pratique avec TEI Publisher

Pour cette exercice, il vous faudra avoir téléchargé le fichier “orgueil_et_prejuges.xml” situé dans le dossier “teipublisher” du dépôt du cours et vous connecter à TEI Publisher avec les identifiants qui sont donnés sur le site.

Tester l’ODD

- ❏ Créer une ODD en mettant comme nom “odd_urfist” et comme titre “ODD URFIST” puis ouvrez cette ODD
- ❏ Ajouter un élément “persName”, ajouter un *model* dedans et mettre dans “Renditions” : “color: red;”, et enregistrez
- ❏ Aller dans “Aire de jeux”
- ❏ Importer le fichier XML d'*Orgueil et Préjugés* puis aller dans le fichier
- ❏ Tester différentes ODD (Dantiscus, Graves, VG), ainsi que la nôtre, et dites-moi ce que vous observez
- ❏ Aller dans l’ODD, ajouter un élément “pb”, supprimer le *model* existant et en créér un nouveau
- ❏ Mettre dans template “[Page [[number]]]” avec “@facs” comme valeur du paramètre ‘facs’ et “@n” comme valeur du paramètre ‘number’ et enregistrez
- ❏ Aller voir le fichier XML et tester ce qui est proposé

En découvrir plus avec TEI Publisher

À partir du fichier “formation_urfist.xar” que vous aurez téléchargé depuis le dossier “teipublisher”, vous pourrez observer diverses affichages du fichier que vous avez encodé auparavant.

- ❏ Connectez-vous en tant qu’admin sur votre instance TEI Publisher et importez l’application .xar
- ❏ Entrez les identifiant et mot de passe dans l’application et ajoutez votre fichier encodé du *Tour du monde en quatre-vingts jours*
- ❏ Changez le template d’affichage du fichier avec la ligne suivante (**<?teipublisher template="nomdetemplate.html"?>**) en début de fichier pour voir à quoi ressemble le texte avec un facsimilé, des modes ou un index

Documentation

Ressources

- XPath Tutorial : https://www.w3schools.com/xml/xpath_intro.asp
- XQuery Tutorial : https://www.w3schools.com/xml/xquery_intro.asp
- XSLT Tutorial : https://www.w3schools.com/xml/xsl_intro.asp
- Documentation TEI Publisher : https://teipublisher.com/exist/apps/tei-publisher/doc/documentation.xml?odd=docbook.odd