Initiation XML-TEI

URFIST – BORDEAUX
07/12/2016 – 09/12/2016

Supports et exercices provenant de différentes pérégrinations professionnelles
incluant les rencontres suivantes :
Bibliothèques Virtuelles Humanistes (CESR, Tours), Reliures-BNF (Bibliothèque Nationale de France)

Lou Burnard, Nicole Dufournaud, Mathieu Duboc, Jorge Fins, Alexei Lavrentiev

2- Exercice (Documentation)

- Copier-coller à l'intérieur le <schemaSpec> du fichier **ODD1.xml** dans le <back>

- Lancer la transformation **TEI ODD XHTML**

- Vous obtenez la documentation liant votre manuel, aux balises déclarées dans votre ODD.

3- ODD : ajouter une définition personnalisée

- **<desc>** : permet d'ajouter une définition personnalisée d'un élément ou d'un attribut

- **<exemplum>** : permet d'ajouter un exemple personnalisé

Exercice : créer une ODD à partir d'un corpus

- On va créer un **ODD personnalisé** à partir d'un corpus (d'un ensemble de fichiers XML)
- On va ensuite ajouter des **descriptions** de certaines éléments. Et des **exemples** personnalisés
- On va générer enfin la **documentation** de cet ODD depuis Roma

3- Exercice (Personnalisation)

- Ouvrez le fichiers XML **2_front_corr.xml** de votre dossier **sources_odd**
- Cliquez sur **Document > Transformation > Configurer les scénarios de transformation**
- Cliquez sur **Nouveau** et choisir dans le menu "**XML Transformation with XSLT**"
- **Nommez** votre transformation ("oddbyexample" par ex)
- Laissez XML URL tel quel et cliquez sur le petit dossier à côté du champ XSL URL pour pointer vers **oddbyexample.xsl**

3- Exercice (Personnalisation)

- Choisissez le transformateur : **Saxon-HE** 9.6.0.7
  - Template("-it"): **main**
- Selectionnez dans **Sortie** "Enregistrer sous" : **../oddbyexample.xml** (par ex)
  - Cochez "**Ouvrir dans un éditeur**"
  - Cochez "**Afficher comme : XML** »
- Lancez la transformation en cliquant sur **OK** puis **Appliquer les Scénarios Associés**
- Si tout s'est bien passé, le fichier oddbyexample.xml devrait être apparu dans le dossier Travaux **4_ODD**
- **Ouvrir** ce fichier dans oXygen

Donner une définition à la valeur « Douzains »

- Pour cela, développer l'élément <valItem ident="douzains"> et ajouter une balise **<desc>** avec votre définition à l'intérieur
- Lancer la transformation **TEI ODD XHTML**
- Vous voyez votre description apparaître :

Legal values are:

douzains
Une suite de douze vers
quatrain
refrain

Donner une définition à la balise <stage>

- Pour cela, ajouter un **<desc>** juste après <elementSpec ident="stage" mode="change">
- Lancer la transformation **TEI ODD XHTML**

Donner un exemple à l'utilisation de <lg>

- Pour cela, ajouter un **<exemplum>** juste après </attList> du <elementSpec ident="lg">
- Ajouter à l'intérieur la balise **<egXML xmlns="http://www.tei-c.org/ns/Examples">**
- Lancer la transformation **TEI ODD XHTML**

```
<lg rhyme="abab" type="quatrain">
 <l>Es-tu l'œil du ciel borgne?</l>
 <l>Quel chérubin cafard</l>
 <l>Nous lorgne</l>
 <l>Sous ton masque blafard?</l>
</lg>
```

Initiation XML-TEI

MANUSCRIT

Lauranne Bertrand – 08/12/2016
(avec certaines diapos de Mathieu Duboc, IE BVH, CESR)

Introduction

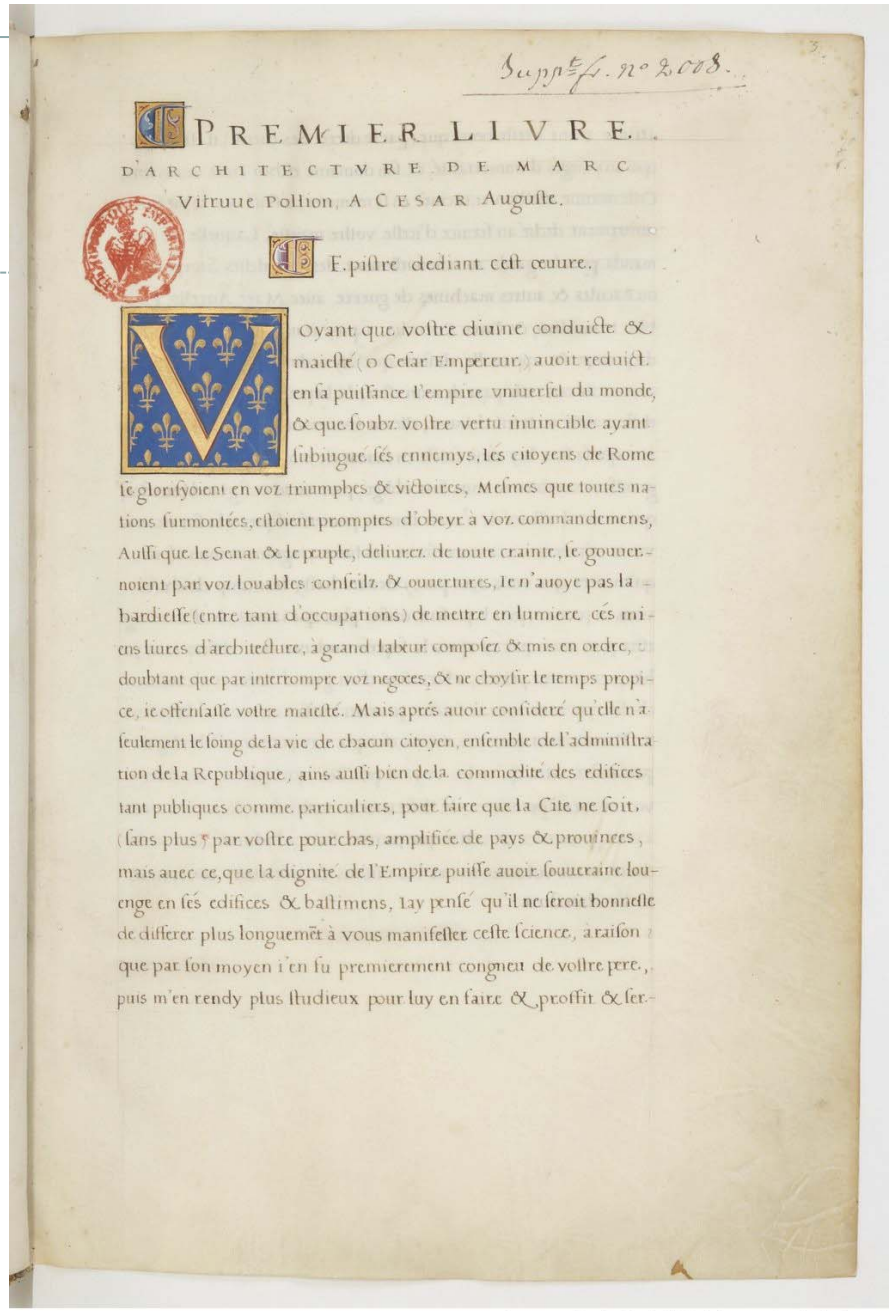
- Les manuscrits sont des documents **spécifiques**
  - Ils sont uniques
  - Ils témoignent parfois d'un seul texte mais avec de nombreuses variantes
  - Ils sont évolutifs, vivants : biffés, modifiés par d'autres mains que leur auteur
  - Ils peuvent donner lieu à plusieurs interprétations

Objectifs de l'encodage :

- Structurer en fonction du type de manuscrit
- Garder toutes les spécificités :
  - L'état de l'oeuvre
  - Biffures, additions, corrections
  - Abréviations
  - Marques d'insertions
  - Différenciation des mains

Manuscrits :

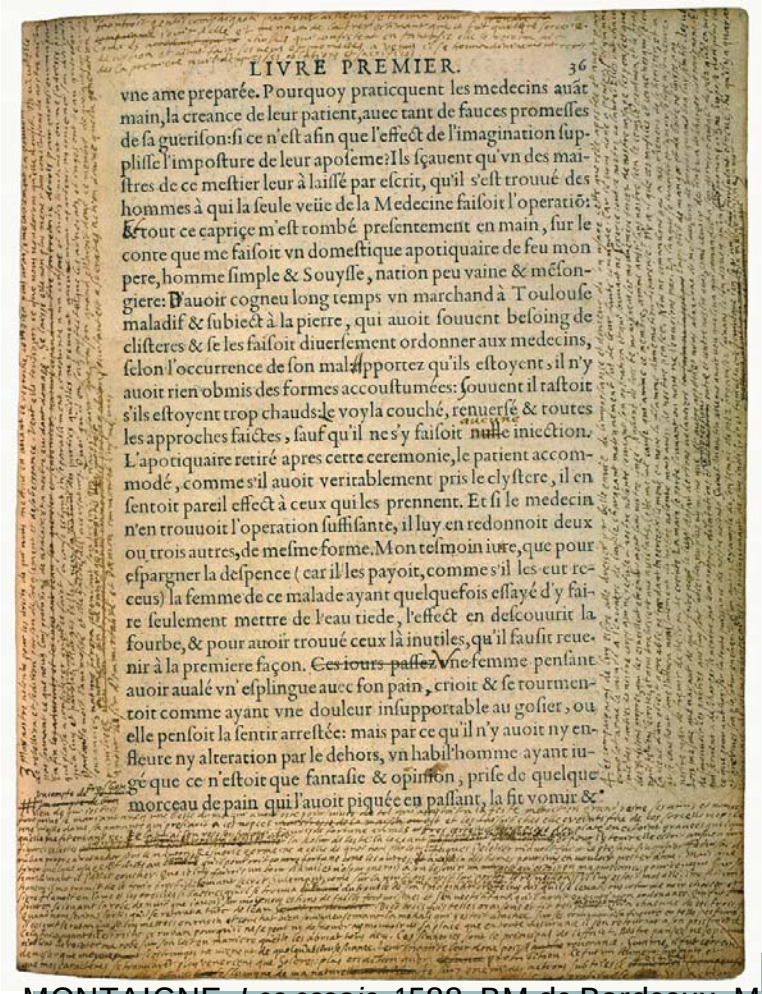
COLAS Jacques, *Suputation nouvellement faicte de la valeur des monnoyes et abuz dicelles*, 1557. BM Orléans, Ms. 629.



VITRUVE, *Premier livre d'architecture*, XVIe. BNF, Fr. 12338,

tion

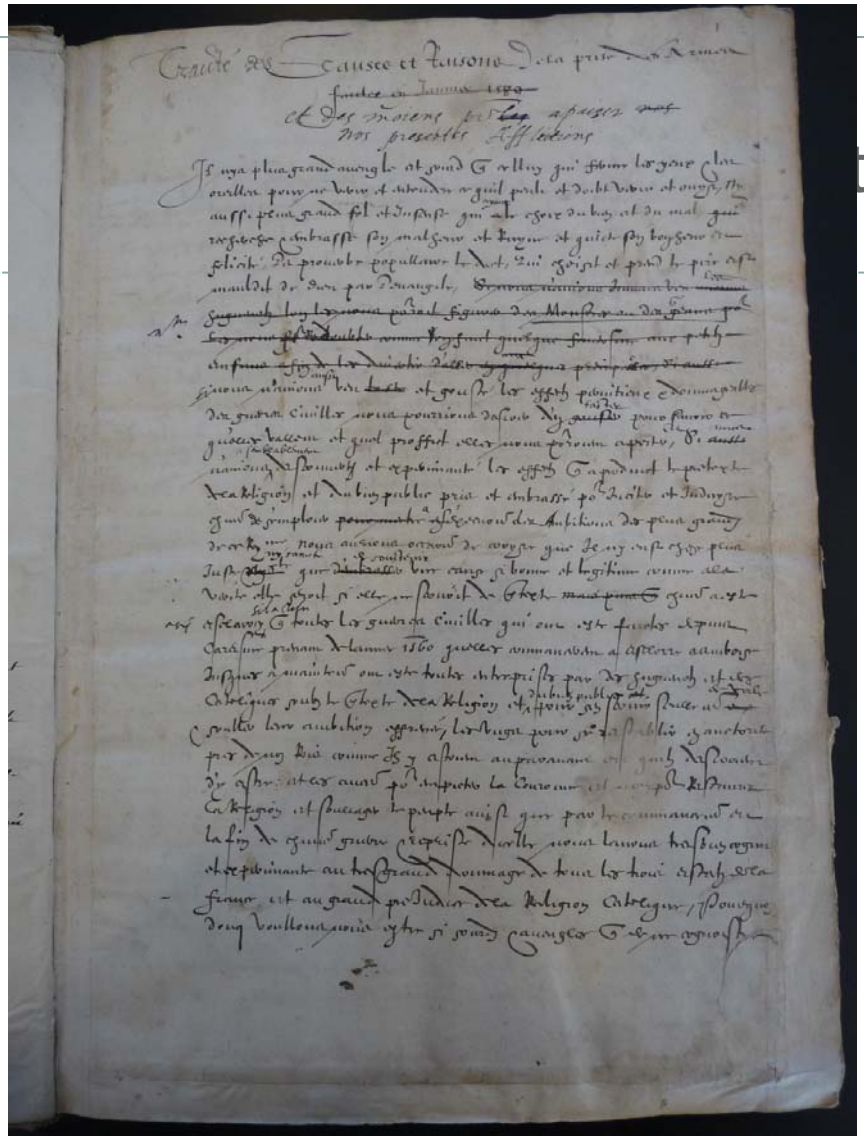
Manuscrits d'auteurs :



LIVRE PREMIER. 36

vne ame preparée. Pourquoy praticquent les medecins auãt main, la creance de leur patient, auec tant de fauces promesses de sa guerison: si ce n'est afin que l'effect de l'imagination supplisse l'imposture de leur aposeme? Ils sçauent qu'vn des maistres de ce mestier leur à laissé par escrit, qu'il s'est trouué des hommes à qui la seule veüe de la Medecine faisoit l'operatiõ: & tout ce capriçe m'est tombé presentement en main, sur le conte que me faisoit vn domestique apotiquaire de feu mon pere, homme simple & Souysse, nation peu vaine & mesongiere: Il auoit cogneu long temps vn marchand à Toulouse maladif & subiect à la pierre, qui auoit souuent besoing de clisteres & se les faisoit diuersement ordonner aux medecins, selon l'occurrence de son mal: Apportez qu'ils estoyent, il n'y auoit rien obmis des formes accoustumées: souuent il tastoit s'ils estoyent trop chauds: le voyla couché, renuersé & toutes les approches faictes, sauf qu'il ne s'y faisoit nulle iniection. L'apotiquaire retiré apres cette ceremonie, le patient accommodé, comme s'il auoit veritablement pris le clystere, il en sentoit pareil effect à ceux qui les prennent. Et si le medecin n'en trouuoit l'operation suffisante, il luy en redonnoit deux ou trois autres, de mesme forme. Mon tesmoin iure, que pour espargner la despence (car il les payoit, comme s'il les eut receus) la femme de ce malade ayant quelquefois essayé d'y faire seulement mettre de l'eau tiede, l'effect en descouurit la fourbe, & pour auoir trouué ceux là inutiles, qu'il fausit reuenir à la premiere façon. ~~Ces iours passez~~ Vne femme pensant auoir aualé vn' esplingue auec son pain, crioit & se tourmentoit comme ayant vne douleur insupportable au gosier, ou elle pensoit la sentir arrestée: mais par ce qu'il n'y auoit ny enfleure ny alteration par le dehors, vn habil'homme ayant iugé que ce n'estoit que fantasie & opinion, prise de quelque morceau de pain qui l'auoit piquée en passant, la fit vomir &

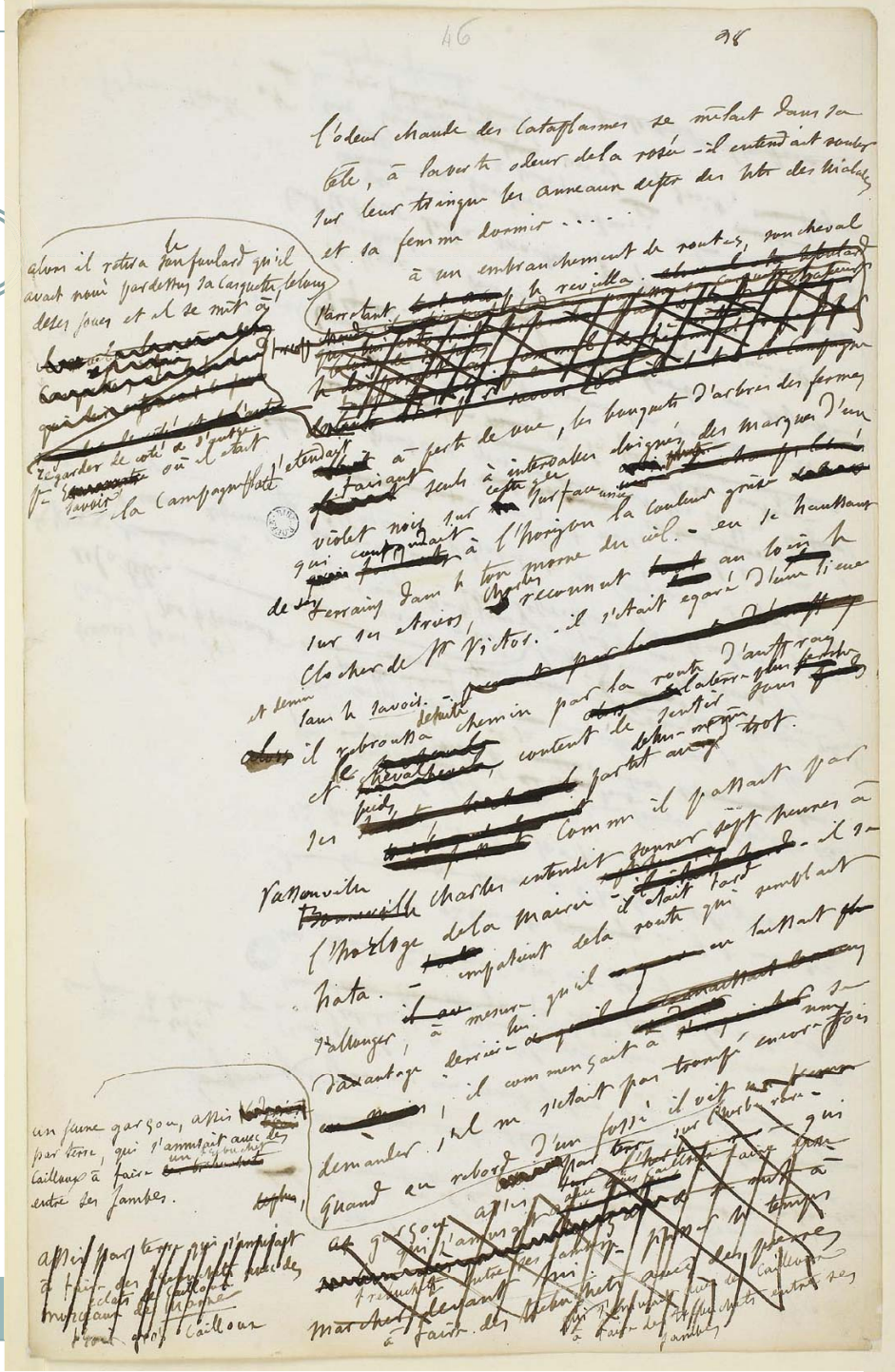
MONTAIGNE, *Les essais*, 1588. BM de Bordeaux, Ms. S1238.



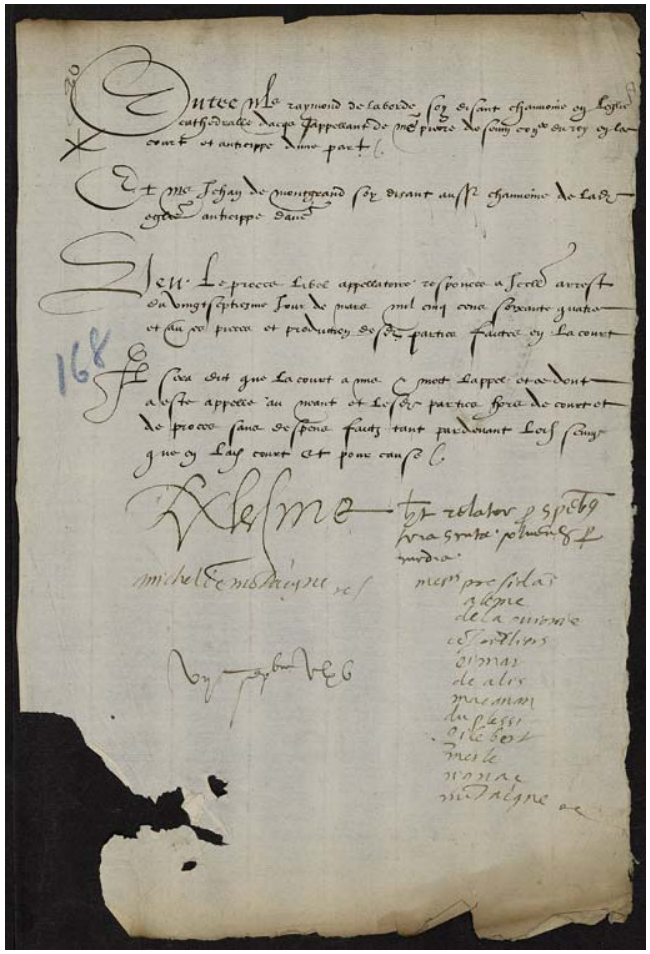
GONZAGUE Louis de, *Traicté des causes et raisons de la prise des armes faicte en janvier 1589*, 1590.
Médiathèque de Nevers, NM 1331.

Autres manuscrits d'auteurs :

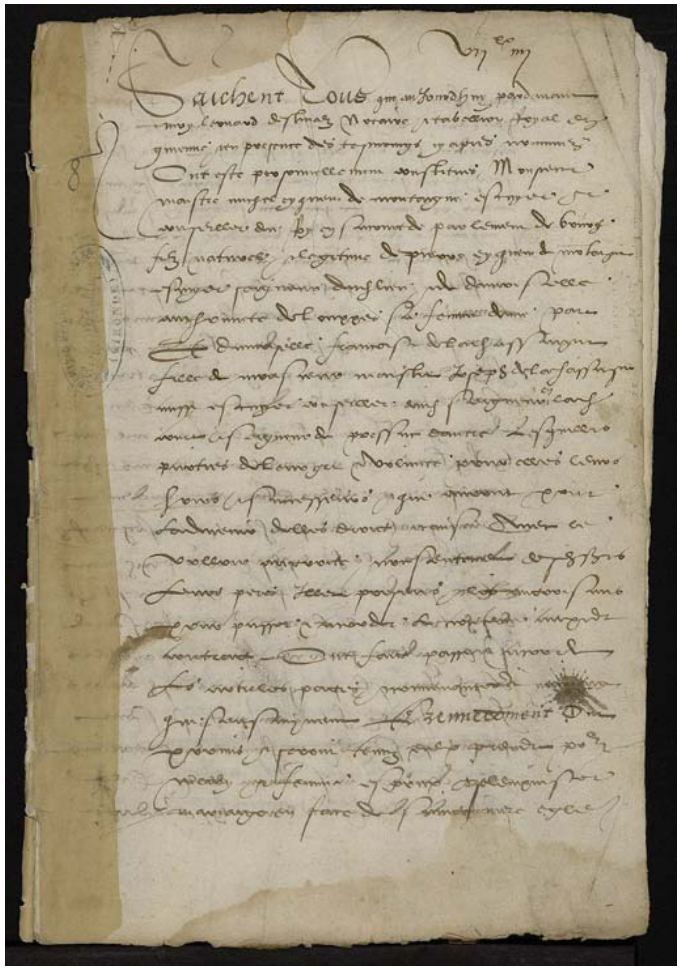
Madame Bovary, Flaubert



Documents d'archive



Arrêt du parlement de Bordeaux du 7 septembre 1565 Archives Départementales de la Gironde, 1B 284 168



Contrat de mariage de Michel de Montaigne avec Françoise de la Chassaigne, 23 septembre 1565 Archives Départementales de la Gironde, 3E 4457

Structure physique

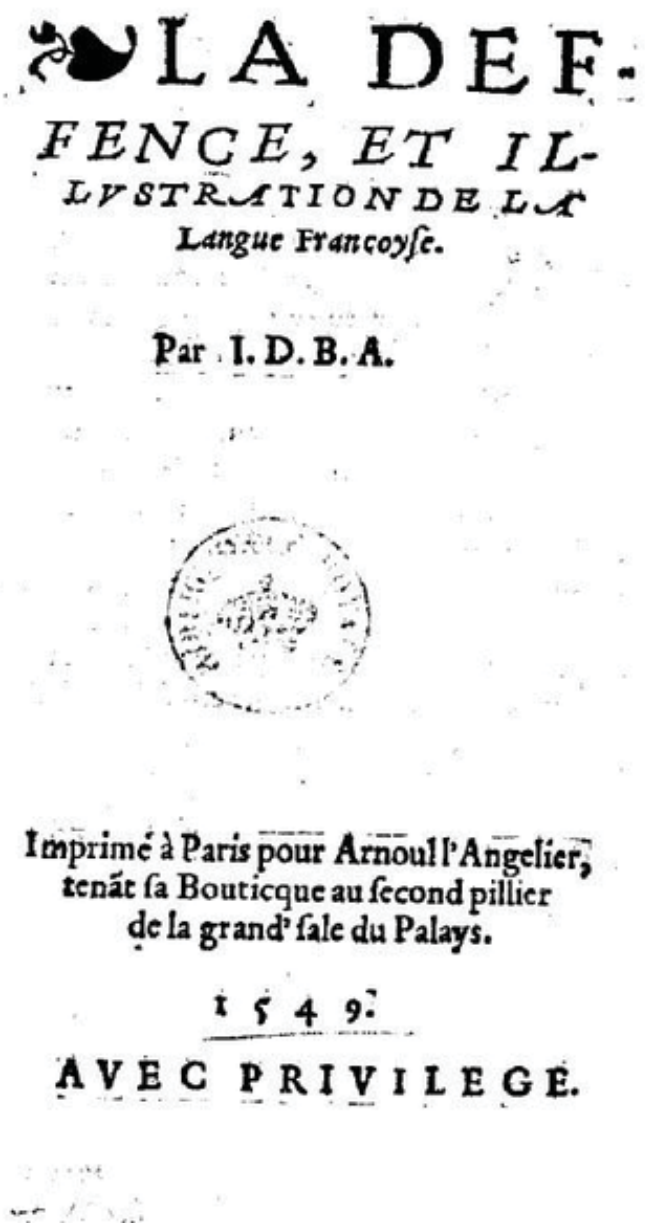
- Structure physique du document :
  - Ce n'est pas un texte linéaire, mais il est inféodé malgré tout à ces contraintes :
    - Feuillet ou page
    - Dégradations : déchirures, trous de vers…
    - Reliure

Cette structure physique interfère dans l'interprétation du texte et donc dans la manière de l'encoder.

Transcription : plusieurs niveaux

- Couche paléographique : caractères et graphèmes reconnus
- Couche documentaire ou diplomatique : ce qui a été écrit
- Couche éditoriale ou sémantique : ce qu'il faut lire

Page de titre imprimée



LA DEF-
FENCE, ET IL-
LVSTRATION DE LA
Langue Francoyse.

Par I. D. B. A.

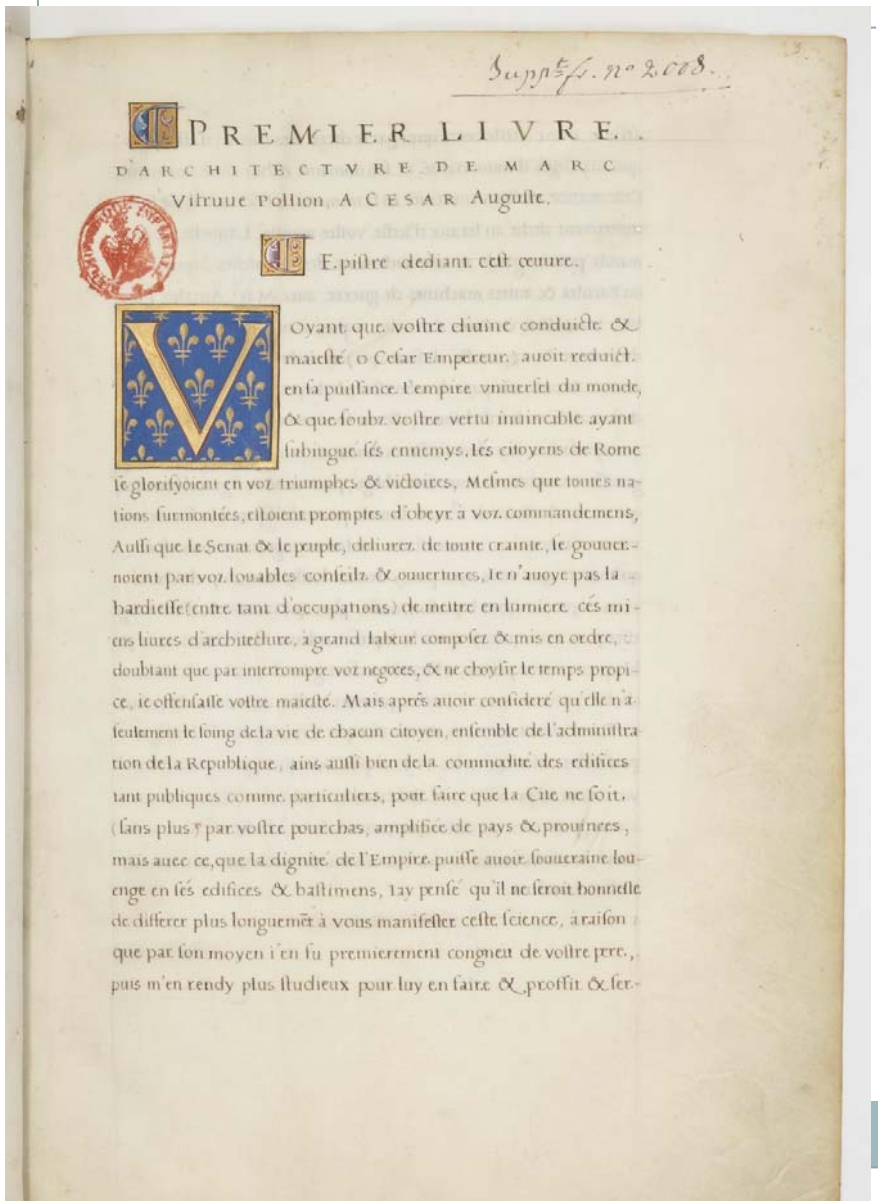
Imprimé à Paris pour Arnoul l'Angelier,
tenãt sa Bouticque au second pillier
de la grand' sale du Palays.

1549.

AVEC PRIVILEGE.

- Titre du livre
- Auteur
- Nom de l'imprimeur et lieu d'impression
- Date d'impression
- Privilège

Première page chez Vitruve



Suppt. fr. n° 2.008.

PREMIER LIVRE.

D'ARCHITECTVRE DE MARC Vitruue Pollion, A CESAR Auguste.

EPISTRE dediant cest œuure.

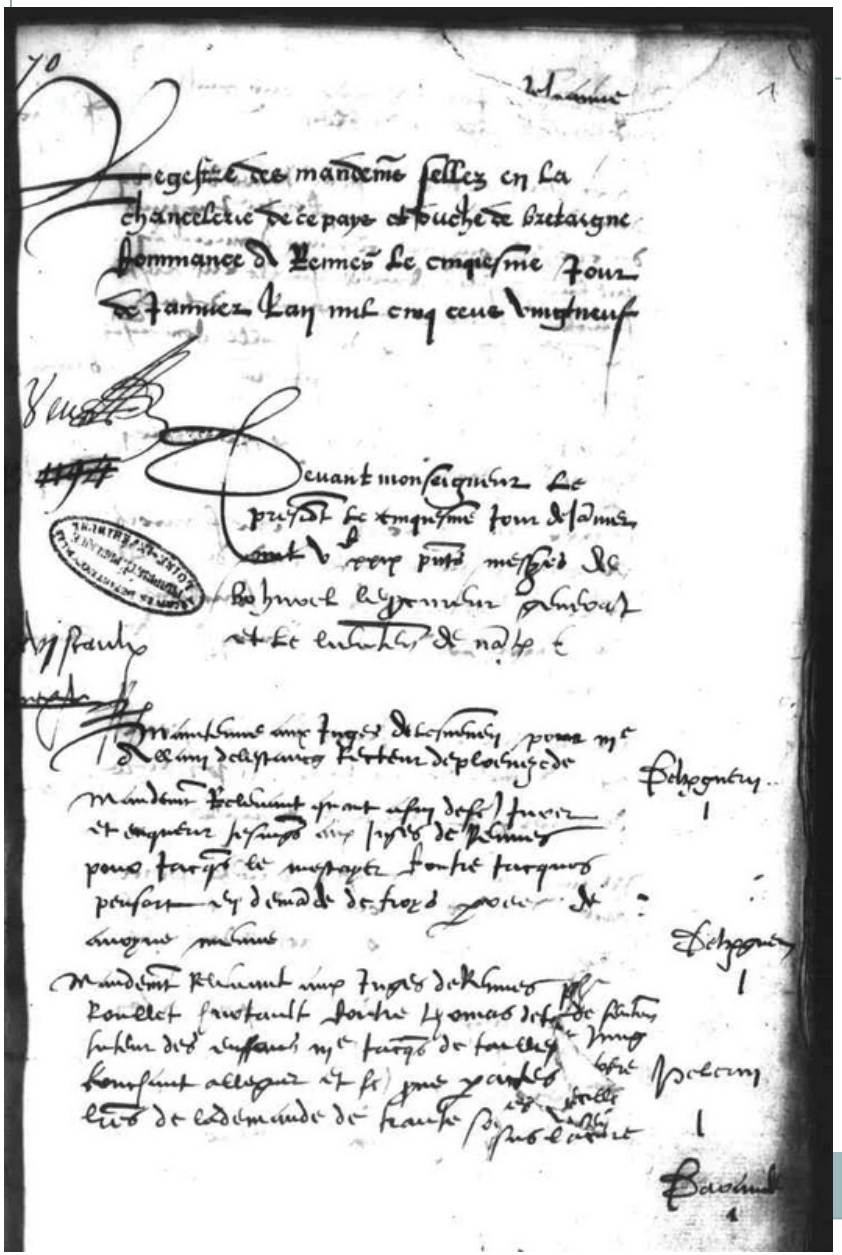
VOyant que vostre diuine conduicte & maiesté (o Cesar Empereur) auoit reduict en sa puissance l'empire vniuersel du monde, & que soubz vostre vertu inuincible ayant subiugué ses ennemys, les citoyens de Rome se glorifyoient en voz triumphes & victoires, Mesmes que toutes nations surmontées, estoient promptes d'obeyr à voz commandemens, Aussi que le Senat & le peuple, deliurez de toute crainte, se gouuernoient par voz louables conseilz & ouuertures, Ie n'auoye pas la hardiesse (entre tant d'occupations) de mettre en lumiere ces miens liures d'architecture, à grand labeur composez & mis en ordre, doubtant que par interrompre voz negoces, & ne choysir le temps propice, ie offensasse vostre maiesté. Mais aprés auoir considéré qu'elle n'a seulement le soing de la vie de chacun citoyen, ensemble de l'administration de la Republique, ains aussi bien de la commodité des edifices tant publiques comme particuliers, pour faire que la Cité ne soit, (sans plus) par vostre pourchas, amplifiée de pays & prouinces, mais auec ce, que la dignité de l'Empire puisse auoir souueraine louenge en ses edifices & bastimens, Iay pensé qu'il ne seroit honnesté de differer plus longuemẽt à vous manifester ceste science, à raison que par son moyen i'en fu premierement congneu de vostre pere, puis m'en rendy plus studieux pour luy en faire & profit & ser-

Titre du livre

Titre de la division

Corps du texte de l'épitre dédicatoire

Première page d'un registre de chancellerie



Titre du registre

Titre de la division du registre

Corps du texte du registre

Structure du registre

- Description du registre :
  - Fabrication du registre, état du registre
  - Son contenu (durée, différence des pièces…)

- Typologie des actes : évocation, mandement, placet, répit, sauvegarde, acte exécutoire, lettre d'office, relief, lettre de pardon, commutation de peine, abolition…

```
<text>
  <body>
    <div type="registre">
      <div type="acte">
        <ab>Texte</ab>
      </div>
      <div type="lettres_patentes">
        <div type="lettre_1">
          <ab>Texte</ab>
        </div>
        <div type="lettre_2">
          <ab>Texte</ab>
        </div>
      </div>
      <div type="acte">
        <ab>Texte</ab>
      </div>
    </div>
  </body>
</text>
```

Les éléments et attributs privilégiés pour l'encodage du manuscrit

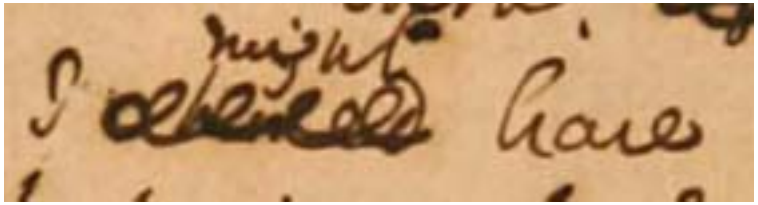
- Pour sa structure :
  - Eléments communs à l'imprimé : , <pb/>, <p>, <lb/>
  - <ab> pour un bloc anonyme : remplace <p> qui a une valeur plus sémantique.

Les éléments et attributs privilégiés pour l'encodage du manuscrit

- L'état de l'œuvre :
  - **<damage>** pour un dommage physique (déchirure, trou de ver…)
    attribut @agent
  - **<gap/>** pour une partie manquante
    attribut @reason
  - **<unclear>** pour les approximations
    attribut @reason
  - **<supplied>** pour les parties manquantes restituées
    attributs *@source, @resp, @cert*

Exemple : <unclear>

- La surcharge produit ici une difficulté à lire ce qui a été barré. Il est probable qu'il s'agisse de « should »
- Pour transcrire ce mot, avant même de signifier qu'il est barré, il faut exprimer cette incertitude de lecture :
  - <unclear>should</unclear>



Exercice 1

- Fichiers à ouvrir :
  - ex_jeanne.xml
  - jeanne_la_devineresse_1.jpg
  - jeanne_la_devineresse_2.jpg
  - jeanne_la_devineresse_3.jpg

- Commencer par encoder la structure :
  - , <pb/>, <ab>, <lb/>
  - attributs @*n*
- Placer l'approximation [incertitude] avec <unclear>

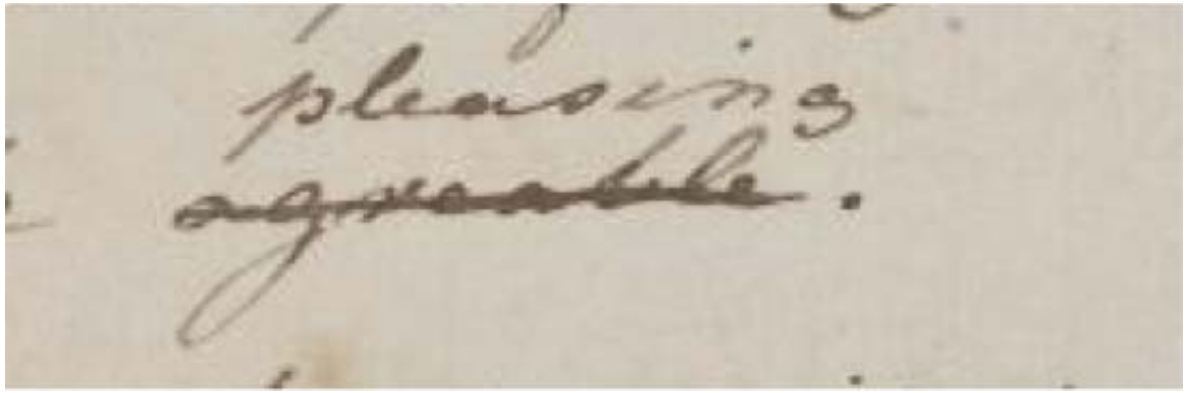
- Les ajouts/suppressions :

  - **<add>** pour les additions
    attribut @place
  - **<del>** pour les suppressions
    attribut @rend
  - **<subst>** pour les substitutions. Cette structure nécessite obligatoirement les éléments <add> et <del> :

    Il a mangé des <subst>~~nouilles~~<add>pâtes</add></subst>

Exemple : Substitution

- Le mot agréable a été raturé et le mot pleasing a été écrit au-dessus.
- Ici, vu la position des deux mots, il s'agit d'un remplacement, qu'on peut encoder avec **<subst>** :
  - **<subst>~~**agreable**~~<add>**pleasing**</add></subst>**



Suite exercice 1

- Encoder les suppressions [suppr] avec **<del>**

Les abréviations :

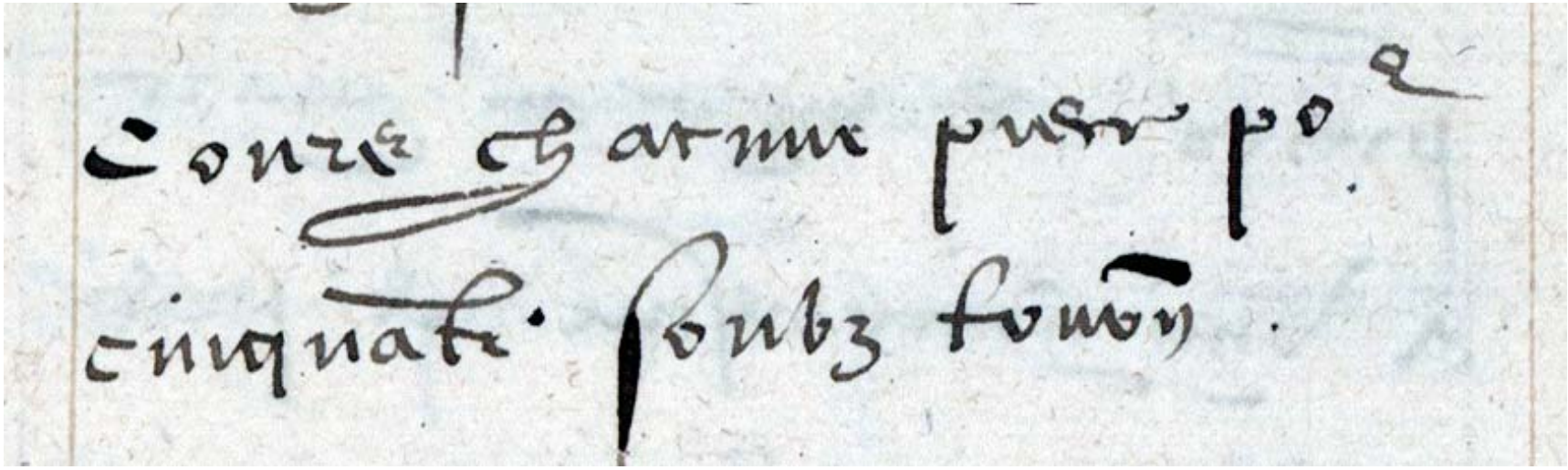
- **<abbr>** pour les mots abrégés
- **<expan>** pour les mots développés
  avec <choice> pour éventuellement regrouper les deux éléments
- **<ex>** pour les seules lettres abrégées (évite le recours au <choice>)

```
<choice><abbr>Mr</abbr><expan>Monsieur</expan></choice>

<abbr>M<ex>onsieu</ex>r</abbr>
```

Exemple : Abréviation/Désabréviation

<lb/>Cours chacune piece **<expan>**pour**</expan>**
<lb/>
<expan>cinquante**</expan>** soubz
<expan>tournois**</expan>**.



Éléments / Attributs ms

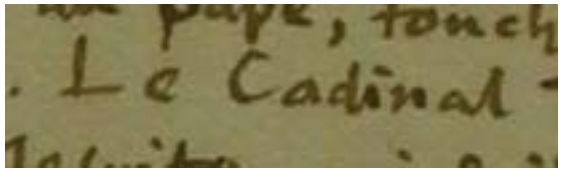
- Les coquilles :
  - **<sic>** pour la coquille
  - **<corr>** pour la correction

    avec <choice> pour éventuellement regrouper les deux éléments

Exemple coquille/correction

- Le mot « cardinal » a été mal écrit, il apparaît donc comme « Cadinal »
- On peut rétablir la bonne orthographe au moyen de sic/corr

  - **<choice>**
    - **<sic>Cadinal</sic>**
    - **<corr>Cardinal</corr>**
  - **</choice>**



Suite exercice 1

- Encoder toutes les abréviations développées [désabr] dans la transcription avec **<expan>**

- Mentionner les coquilles [sic] et ajouter une correction avec : **<choice> / <sic> + <corr>**

Éléments communs manuscrits et imprimés

- Les formes particulières :
  - **<orig> pour la forme originale**
  - **<reg> pour la forme régularisée**

    avec <choice> pour éventuellement regrouper les deux éléments

```
<choice>
        <orig>preuenir</orig>
        <reg>prévenir</reg>
</choice>
```

Éléments communs manuscrits et imprimés

- Les listes
  - <list> pour l'ensemble de la liste
  - <item> pour chaque objet, de cette liste

```
<list>
        <item>un</item>
        <item>deux</item>
        <item>trois</item>
</list>
```

Éléments communs manuscrits et imprimés

- Les langues :
  - <foreign>

    avec l'attribut @xml:lang

```
<foreign xml:lang="lat">Veni vidi vici.</foreign>
```

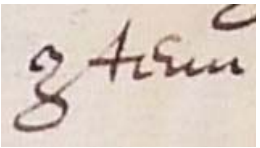
Remarque : les codes de langues sont établis selon la norme internationale ISO-639 (plus précisément ici 639-3 pour les codes à 3 lettres).

Suite exercice 1

- Encoder les passages en langue étrangère avec **<foreign>** et l'attribut **@xml:lang**
  - valeurs (iso) : lat, ell

Rester au plus près de la source

- Les abréviations
  - Le choix retenu ici est de les développer directement avec <expan>
  - Nous aurions très bien pu encoder les brévigraphes avec la balise <g> pour glyph avec l'attribut @ref (déclaré dans le TEI header)

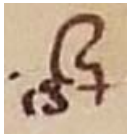


```
<g ref="#abr_con">con</g>tenu
```

  - Il est également possible d'utiliser un caractère unicode :

```
<choice><abbr>&#42863;tenu</abbr><expan>contenu</expan></choice>
```

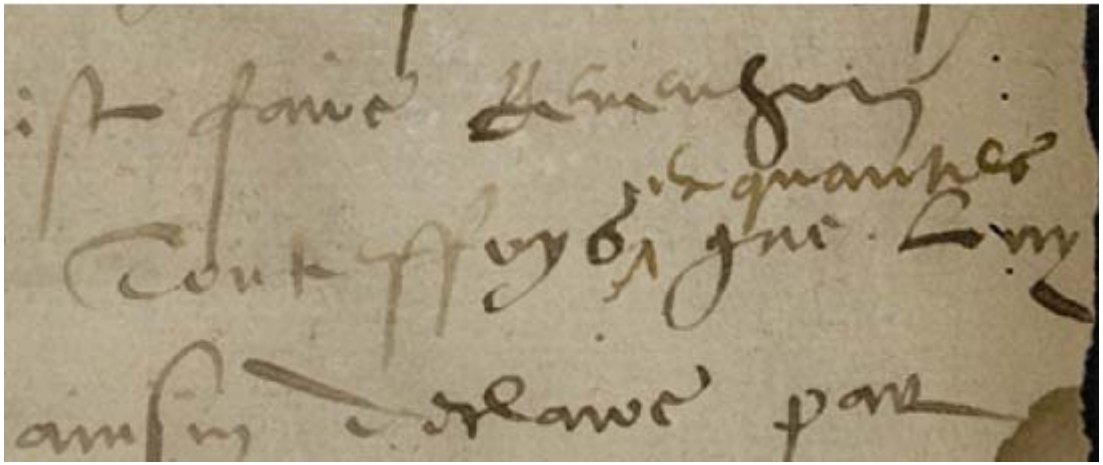
  - Nous pouvons également encoder certaines graphies de la même manière telle que la ligature st :



Attention la plupart de ces symboles ont un caractère unicode, mais bon nombre ne sont pas supportés par toutes les polices, ce qui peut poser des problèmes d'affichage.

Additions insérées

- Certaines additions peuvent dépendre d'un signe d'insertion indiquant la place où elles s'insèrent dans le texte :



- On utilise la balise <**metamark**> qui peut comporter l'attribut @*target* pointant vers l'endroit où se trouve l'addition.

Éléments / Attributs ms

- Les mains :
  - <handShift/> pour un changement de main
    attribut @new
  - L'attribut *@hand* peut également être utilisé sur un conteneur (<ab> par exemple)

Exercice : Arrêts du Parlement de Bordeaux

- Fichiers à ouvrir :
  - ex_arret.xml
  - arret_1B260_218.pdf

- Commencer par structurer le document :
  - ** , <ab>, <lb/>, <list>, <item>**

Exercice : Arrêts du Parlement de Bordeaux

- Continuez par les particularités de ce manuscrit :
  - abréviations : **<expan>**
  - additions : **<add>** @place
  - marque d'insertion : **<metamark>**
  - changement de main : @hand sur un conteneur ou **<handShift>** @new
  - langue : **<foreign>** @xml:lang

Problèmes de ce genre de sources

- Ces méthodes d'encodage sont souvent compliquées à appliquer
- Il y a des blocs qui se chevauchent, par exemple des additions continuant sur d'autres pages
- On peine à reconnaître l'ordre des interventions
- La mise en forme et la signification associée sont trop liées et difficile à distinguer

Le chevauchement

- Les éléments **<addSpan>** et **<delSpan>** délimitent une section de texte au moyen de pointeurs plutôt qu'en encadrant une section de texte
- Ils sont utiles si un ajout ou une suppression chevauche une autre portion de texte
- L'attribut **@spanTo** indique où se situe la fin d'une portion de texte commençant par l'élément qui porte l'attribut.

Le chevauchement

<l> .... **<delSpan spanTo="#id4"/>** ... </l>
<l> ... **<anchor xml:id="id4"/>** ... </l>
<l> .... </l>

- Ici on a une suppression commençant au milieu d'un vers et se terminant au milieu du vers suivant
- À cause des <l>, si on avait utilisé <del>, on aurait eu un cas interdit par le XML, le chevauchement :
  - <l>.... <del>.....</l>
  - <l>.....</del>....</l>

La correspondance

- Éléments
  - **<opener>** regroupe les éléments de début de lettre
    - **<salute>** pour la formule de salutation
    - **<address>** pour l'adresse
    - **<dateline>** pour le lieu et la date
  - **<closer>** regroupe les éléments qui terminent la lettre
    - **<salute>**
    - **<signed>** pour le signataire
    - **<dateline>**
  - Au milieu de ces deux blocs, on a en général **<ab>**

Exercice : Correspondance

- Fichiers à ouvrir :
  - ex_vinet.xml
  - ex_vinet.jpg

- Balises proposées pour l'encodage :
  - <opener>, <ab>, <closer>, <signed>, <salute>, <dateline>

Exercice suite

- Ajouter la couche sémantique
  - <sic> pour les coquilles
  - <add> pour les ajouts (interlignes)
  - <del> pour les ratures
  - <unclear> pour les passages peu lisibles
  - <foreign> pour les passages en langues étrangères
  - <choice> avec <abbr> et <expan> pour les abréviations
  - <choice> avec <orig> et <reg> pour la régularisation des graphies (si vous le souhaitez)

Variantes

- Lorsqu'on produit l'édition d'un manuscrit, on a bien souvent à intégrer plusieurs lectures du même mot ou passage. Pour cela, on utilise l'apparat critique : **<app>** qui indique une **variante** textuelle
- **<app>** se combine avec les balises :
  - **<lem>** (passage de la source originale)
  - **<rdg>** (autre lecture, dont on précise la source au moyen de **@wit**)

Variantes

... monsieur de l’Escut &

```
<app>
        <lem>sa</lem>
        <rdg wit="#LAL">la</rdg>
        <rdg wit="#E82_p14">sa</rdg>
</app>
```

trouppe...

Exercice : Variantes

- **Encoder en priorité les variantes dans le Manuscrit du Cinquième Livre**

- ex_rabelais.pdf
- ex_rabelais.xml

- Variante insérée :
  - <app><lem/><rdg wit="#source">variante</rdg></app>

- Variante classique :
  - <app><lem>pivot</lem><rdg="#source">variante</rdg></app>

Exercice suite

- Encoder le reste :
  - sic/corr
  - unclear
  - gap
  - abbr/expan

Initiation XML-TEI

OUTILS

Lauranne Bertrand – 08/12/2016

Introduction

Des outils, parce qu'il ne suffit pas d'encoder :

- Il faut pouvoir **gérer** les fichiers produits (création, modification, validation)
- Il faut pouvoir **produire une édition** (transformer le XML, le visualiser dans un navigateur)
- Il faut pouvoir en tirer des **données utiles à la recherche** (analyse et fouille de textes)
- Il faut pouvoir **pérenniser** son fichier (stockage, archivage, moissonnage)

Introduction

Nous avons vu qu'en ce qui concerne la gestion des fichiers encodés, **oXygen** offre un environnement de travail adapté, permettant de **valider** un fichier contre un schéma, de **modifier** le code et de le manipuler.

Pour information, ces éditeurs sont également compatibles avec le XML, et permettent parfois de transformer avec XSLT et de valider selon un schéma, comme oXygen :

- JEdit
- Editix
- XML Copy Editor
- XMLMind
- Xpontus
- D'autres éditeurs listés sur le wiki : http://wiki.tei-c.org/index.php/Editors

Introduction

Concernant la transformation XSLT de XML, il est tout à fait possible de lancer ce genre de transformation en ligne de commandes, en s'affranchissant d'un environnement :

- Voir directement la technologie utilisée :

http://saxon.sourceforge.net/

Introduction

Car pour **l'édition** d'un fichier XML en ligne, il faut en passer par XSLT. Plusieurs outils permettent de faire des transformations déjà paramétrées, sans avoir à connaître XSLT, cependant, il sera difficile de s'en passer tout à fait.

Transformer automatiquement :

- Utiliser les scénarios de transformation inclus dans oXygen
- Utiliser oXgarage : http://www.tei-c.org/oxgarage/

Démonstration oXgarage

http://www.tei-c.org/oxgarage/

- Essayer de transformer un fichier **TEI P5** vers **xHTML** ou vers **PDF**
- Vous voyez que le résultat n'est pas toujours idéal. Voilà pourquoi, il faut en général, **modifier** (ou créer) une feuille XSLT pour obtenir le résultat souhaité
- Tester également la transformation **Word** vers **TEI P5**
- Tous les scénarios XSLT qui concernent la TEI se trouvent ici : https://github.com/TEIC/Stylesheets

Chaîne éditoriale MRSH Caen

- http://www.unicaen.fr/recherche/mrsh/document_numerique/projets/chaine_editoriale
- Chaîne éditoriale permettant de publier du XML-TEI en EPUB ou au format papier
- Utilise XXE (XML XMLMind Editor) et ses templates (formulaires)
- **Démo** : Formulaire paramétré pour Reliures

Démonstration TEI boilerplate

http://dcl.ils.indiana.edu/teibp/

- Il s'agit d'un outil permettant d'afficher directement un fichier XML dans un navigateur.
- La transformation vers HTML se fait **à la volée**
- Plusieurs éléments TEI sont gérés par défaut comme :
  -
  - <list> et <item>
  - <choice> et sous-balises
  - <pb> et @facs pour afficher une imagette de la page
- **Démo** : afficher le code source de la page
- Le principal problème actuel étant que les navigateurs modernes bloquent la transformation permettant de visualiser le code, il faut donc utiliser un serveur

TAPAS

http://beta.tapasproject.org/about

- Plateforme communautaire permettant de stocker et d'afficher ses projets en XML-TEI.

- Exemple de texte hébergé sur TAPAS : http://beta.tapasproject.org/captainsingleton/files/life-adventures-and-pyracies-famous-captain-singleton

Synopsx

http://ahn.ens-lyon.fr/synopsx

- Système léger permettant de publier des fichier en XML-TEI (avec une BDD entièrement en XML articulée par BaseX)
- Permet d'articuler EAD avec la TEI

- Dépôt Github : https://github.com/synopsx/synopsx

Démonstration XTF

http://xtf.cdlib.org/

- Plateforme permettant d'**interroger** et **d'afficher** des éditions XML (dont TEI)
- Framework java basé sur des transformations XSLT qui génèrent à la volée les requêtes

- http://www.marktwainproject.org/
- http://dlib.indiana.edu/collections/vwwp/

Démonstration Versioning Machine

http://v-machine.org/

- Plateforme permettant d'afficher **plusieurs** versions d'un même texte en parallèle, afin de les **comparer**
- http://v-machine.org/samples/prophecy_of_merlin.xml

Démonstration Juxta Commons

http://juxtacommons.org/shares/yY6Wrj

- Outil en ligne permettant de comparer plusieurs témoins d'un même texte, afin d'afficher les **variantes** automatiquement

- Possibilité d'importer ses propres textes dans un espace de travail, et d'exporter les résultats
- Support pour la visualisation de la TEI (plusieurs textes ou encodage avec <app>)

- Plusieurs visualisations/outils : tout en un, face à face, VM

Démonstration TEI Critical Edition Toolbox

http://ciham-digital.huma-num.fr/teitoolbox/

- Outil accompagnant la création des éditions critiques
  - Vérifie l'encodage
  - Affiche les versions parallèles
- Test sur **ex_rabelais_corr.xml**

Démonstration Philologic

https://artfl-project.uchicago.edu/philologic4

- Philologic est un outil de fouille de texte intégrant des commandes **linguistiques** sur les corpus fouillés
  - Concordances
  - Recherche par similarité
  - Recherche dans une partie du texte (head)

- http://www.bvh.univ-tours.fr/Epistemon/philologic.asp

Démonstration TXM

TXM est une plateforme (bureau et web) permettant d'importer un corpus textuel (en XML mais aussi dans d'autres formats) et de l'interroger :

- Concordances
- AFC
- Création de sous-corpus
- Partitions

- Exemple d'un portail sous TXM : http://txm.bfm-corpus.org/
- (nécessite une inscription)
- **Démo** : Sélection de textes via les descripteurs

SIG Tools

- Plusieurs « Special Interest Groups » travaillent autour de la TEI dont un sur les outils :

- http://www.tei-c.org/Activities/SIG/Tools/
- http://wiki.tei-c.org/index.php/SIG:Tools
- http://wiki.tei-c.org/index.php/Category:Tools

Initiation XML-TEI

XSLT

Lauranne Bertrand – 08/12/2016

Introduction

- XSLT : eXtensible Stylesheet Language Transformations
- Comme son nom l'indique, il s'agit d'un langage utilisé pour effectuer des transformations. Mais de quoi ?
- Bien souvent XML-TEI > HTML (donc un langage XML vers un autre langage XML).
- Le XSLT est lui même un langage XML.

<xsl:template>

...

</xsl:template>

template = gabarit

En XSLT, une template est un modèle de transformation qui en général cible un "**noeud**" pour le transformer en autre chose.

Le "noeud" en question peut être du texte mais souvent, il s'agit **d'une balise qu'on transforme en une autre balise**.

Espaces de nom

- Lorsqu'une balise est précédée d'un nom suivi de ":" on indique son espace de nom (namespace) de rattachement. Ici, chaque balise xsl est définie par le namespace **xmlns:xsl="http://www.w3.org/1999/XSL/Transform"**

- C'est **l'espace de nom** qui regroupe en fait l'ensemble des balises appartenant à un même langage en XML.

- C'est ainsi qu'au sein d'une feuille xslt, on peut utiliser diverses balises, rattachées à leur espace de nom respectif comme celui de la TEI (**xmlns:tei="http://www.tei-c.org/ns/1.0"**).

Exemple :

```
<xsl:template match="truc">
<bidule>
<xsl:apply-templates/>
</bidule>
</xsl:template>
```

Ici on transforme l’élément "truc" en l’élément "bidule".

C'est à dire que :

<truc>un paragraphe de texte</truc>

Deviendra

<bidule>un paragraphe de texte</bidule>

Qu’en est-il du reste de la template ?

- L’attribut **@match** : cible un nœud

- La balise **<xsl:apply-templates/>** : dit à la feuille d'appliquer toutes les autres templates possibles au niveau du noeud courant. Quand comme dans notre exemple, il n’y a plus rien à faire, la transformation **récupère le texte contenu** dans la balise ciblée.

Opérations plus complexes

Une template en xslt ne contient pas qu'une seule opération (remplacer une balise par une autre) mais en général plusieurs.

En résumé : on cible avec l'attribut **@match**, et on applique tout ce qu'on veut faire à ce niveau.

- **Remplacer la balise**
- **Copier la balise et son contenu** (c'est à dire tous ses enfants)
- **Adjoindre du texte**
- **Adjoindre d'autres balises**...

Exercice 1

Nous allons paramétrer une transformation XSLT étape par étape, en partant d'un squelette déjà existant.

Le fichier que nous allons transformer en édition HTML est : **ex_rabelais_corr.xml** utilisé précédemment.

- **Ouvrir** ce fichier XML dans oXygen.
- Vous voyez tout en haut qu'il est lié à la XSLT "ex_rabelais_corr.xsl"
  - <?xml-stylesheet type="text/xsl" href="ex_rabelais_corr.xsl"?>

Lancez une première transformation en appuyant sur la flèche rouge.

- Le logiciel demande si vous souhaitez lancer la transformation par défaut, répondez « oui »

- *[Selon configuration d'oXygen : montrer comment paramétrer une sortie vers HTML par défaut]*

Problème :

- La transformation obtenue n'est qu'un bloc de texte non formaté.
- Nous allons essayer de le mettre en forme

- **Ouvrez** la feuille XSLT : **ex_rabelais_corr.xsl**

Ajouter des retours à la ligne à chaque <lb/>

- Il faut remplacer la balise <lb/> par la balise

```
<xsl:template match="tei:lb">
    <br/>
</xsl:template>
```

Ajouter des paragraphes :

- Selon le même modèle que précédemment, remplacer la balise <p> par <p>

- À votre tour.

Remarque : Si vous obtenez un texte quasiment entièrement supprimé, c'est que vous avez oublié une partie de votre template.

En effet, écrire :

```
<xsl:template match="tei:p">
    <p></p>
</xsl:template>
```

Équivaut à transformer la balise en sans rien à l'intérieur.

Voilà pourquoi il faut absolument, dans les balises englobantes, prévoir **<xsl:apply-templates/>** au milieu, qui permet de dire à la feuille de soit :

- **Appliquer les autres templates** à cet endroit là (par exemple, à l'intérieur de <p>, il y a des <lb/>, il faut donc que les deux templates soient appliquées)
- Et **récupérer le contenu textuel** une fois fini

Exercice 1

```
<xsl:template match="tei:p">
        <p>
                <xsl:apply-templates/>
        </p>
</xsl:template>
```

Gérer de la même manière <lg> et <l> :

- La balise <lg> devient <p>
- La balise <l> devient <br/> avant que le contenu ne soit récupéré
- **À vous.**

Exercice 1

Remarque : Pour qu'un contenu de balise soit copié après l'insertion d'une autre, il suffit encore une fois d'ajouter **<xsl:apply-templates/>** au bon endroit.

Mentionner les sauts de page

- Ici, nous n’allons pas transformer <pb/> en une autre balise, mais dans du texte.

<xsl:text>

Balise permettant d'inclure du texte.

```
<xsl:template match="balise">
  <balise>
        <xsl:apply-templates/>
        <xsl:text>mon texte</xsl:text>
  </balise>
</xsl:template>
```

Ici, j'ajoute du texte à ce qui est contenu par ma balise.
Ex : <balise>je voulais ajouter ici </balise>
Devient : <balise>je voulais ajouter ici mon texte</balise>

Pour <pb/> cependant, c'est plus simple, il n'y a rien à récupérer, la balise est vide. Il suffit donc

- De cibler <pb/>
- De placer du texte à la place, comme **[Saut de page]**
- De placer avant et après ce texte des sauts de ligne

Supprimer du contenu

Une petite astuce permet de supprimer du contenu, c'est de cibler la balise englobante mais de ne rien mettre dans la template.

```
<xsl:template match="balise"/>
```

Exercice 1

- Supprimez le contenu du <teiHeader>

Arbre et XPath

- Nous aimerions à présent, faire en sorte que le premier titre devienne un titre de niveau **<h1>** et que le second titre devienne un titre de niveau **<h2>**
- Les deux sont concernés par la balise <head>, on devrait normalement tout traiter avec **une seule** template. Il faut donc naviguer dans l'arbre.

- XPath est une forme d'expression permettant d'interroger les structures en arbre, comme le XML.

- Ex : **div/head/lb** est une expression qui cible un <lb/> enfant de <head> lui même enfant de
- Ex : **head[parent::tei:div]**

- On a deux titre. Les deux sont enfants de : **tei:head[parent::tei:div]**
- Une particularité cependant, un @type a une valeur de @type="prologue" et l'autre "livre"

- On pourrait donc dans la solution la plus simple, faire deux templates :
  - L'une @match **tei:head[parent::tei:div[@type="livre"]]**
  - L'autre @match **tei:head[parent::tei:div[@type="prologue"]]**

- Sur le même modèle, XPath propose plusieurs autres expressions :

parent:: tout élément direct englobant le noeud courant
ancestor:: tout élément englobant le noeud courant, à tout niveau
child:: tout élément enfant direct du noeud courant
descendant:: tout élément enfant du noeud courant, à tout niveau
following:: tout élément suivant le noeud courant, à tout niveau
following-sibling:: tout élément suivant le noeud courant, au même niveau (frère direct)
preceding:: tout élément précédant le noeud courant, à tout niveau
preceding-sibling:: tout élément précédent le noeud courant, au même niveau (frère direct)

<xsl:choose>

- La solution la plus simple, pour mes titres serait donc deux templates distinctes, mais on pourrait très bien gérer chaque cas de <head> en **une seule template**

<xsl:choose>

<xsl:choose> donne le choix entre plusieurs conditions (qui s'excluent entre elles : condition 1 **ou** condition 2 **ou** condition 3...)

Cette balise s'utilise avec plusieurs sous-balises :

<xsl:when>

<xsl:otherwise>

<xsl:choose>

```
<xsl:template match="balise">
<xsl:choose>
<xsl:when test="@type='bleu'">
<bleu>
<xsl:apply-templates/>
</bleu>
</xsl:when>
<xsl:when test="@type='rouge'">
<rouge>
<xsl:apply-templates/>
</rouge>
</xsl:when>
<xsl:otherwise>
<rien><xsl:apply-templates/></rien>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

<xsl:choose>

Ce qui donne le schéma :

OU bleu

OU rouge

OU bien autre chose qui n’est ni bleu, ni rouge

Au secours !

Récapitulons un peu ce qui concerne nos titres

- Une seule template qui match **tei:head**
- Un choix entre
  - Soit un head dont le parent (parent::tei:div) est une div qui a un **@type="livre"**
  - Soit un head dont le parent (parent::tei:div) est une div qui a un **@type="prologue"**

Attention vous allez être obligés d'utiliser deux types de guillemets : " et '

Une suite possible à cet exercice

- Supprimer soit <orig> soit <reg>
- Afficher le <lem> dans une couleur particulière
- Afficher le <rdg> entre crochets et en plus petit
- Afficher les <note> entre crochets et en italique
- …

Quelques pistes supplémentaires

D’autres balises XSLT :

- <xsl:value-of/> @select
- <xsl:if> @test

Et des fonctions :

- concat(XPath,XPath,...)
- contains(XPath,'string')
- matches(XPath,'regex')

Tester différentes expressions XPath sur le fichier

/TEI
/TEI//title
//text/body/div[1]
//div[parent::body]
//p[ancestor::div]

Rappel :

/balise = balise racine (/TEI)
/balise/balise2 = balise 2 enfant de balise racine
// = n'importe où dans l'arbre

Initiation XML-TEI

MÉTADONNÉES

Lauranne Bertrand – 09/12/2016

Introduction

Qu'est-ce qu'une métadonnée ?

- Une métadonnée est une **donnée numérique** servant à représenter ou décrire une autre donnée (numérique ou non).

- Elle peut contenir des informations sur la **source** d'un document, sa **nature**, son **contenu**, sa **localisation**, son **histoire**, son **statut juridique**, etc.

- Ces informations gagnent en général à être normalisées.

- Elles sont encodées dans le **<teiHeader>**

Introduction

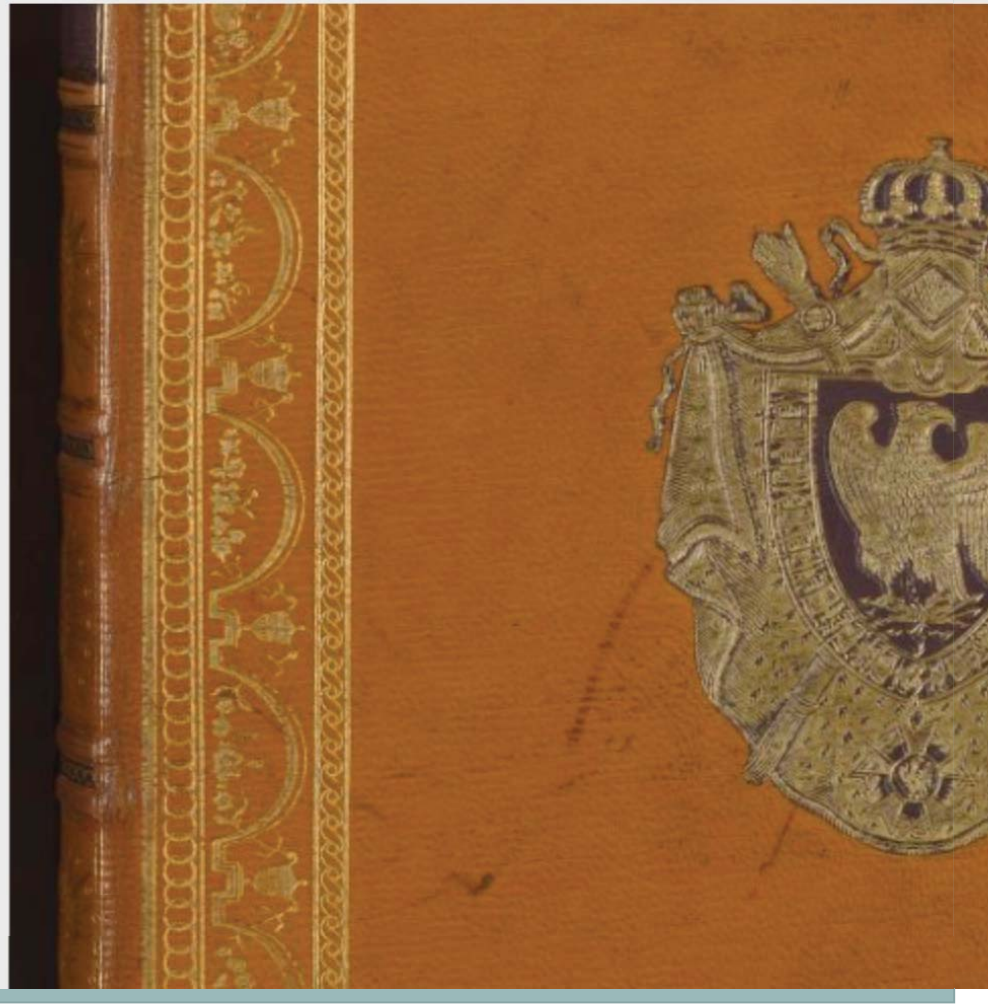
eil > Consultation

Reliure de : Poésies galliques, traduites sur l'anglois de M. Mac...

Informations détaillées ×

NOTICE

Titre : Reliure de : Poésies galliques, traduites sur l'anglois de M. Macpherson, par M. Le Tourneur
Auteur : Ossian (auteur supposé)
Éditeur : Musier fils (Paris)
Date d'édition : 1777
Contributeur : Le Tourneur, Pierre-Prime-Félicien (1737-1788). Traducteur
Type : monographie imprimée
Langue : Français
Format : 2 tomes en 1 vol. : fig. ; gr. in-8
Format : application/pdf
Description : Numérisation partielle de reliure
Droits : domaine public
Identifiant : ark:/12148/btv1b73003472
Source : Bibliothèque nationale de France, département Réserve des livres rares, RES-YN-4
Relation :



À quoi servent les métadonnées ?

- Elles jouent un rôle **d'indexation** : une fois indexées par un logiciel ou une plateforme numérique, elles permettent d'améliorer la recherche d'information et d'accéder ainsi au document souhaité.

- **Normalisées** grâce à des standards, elles servent à **l'interopérabilité**, c'est-à-dire à l'échange de données.

- Par exemple : les notices catalographiques sont en général **exportables** dans un format de données permettant d'être réutilisées dans le cadre d'un autre catalogue (**MarcXML**). Il en va de même pour les données d'un teiHeader.

Représentation d'un même document

Notice bibliographique

Notice Au format public

Auteur(s) : Ossian (auteur supposé)

Titre(s) : Ossian,... Poésies galliques, traduites sur l'anglois de M. Macpherson, par M. Le Tourneur [Texte imprimé]

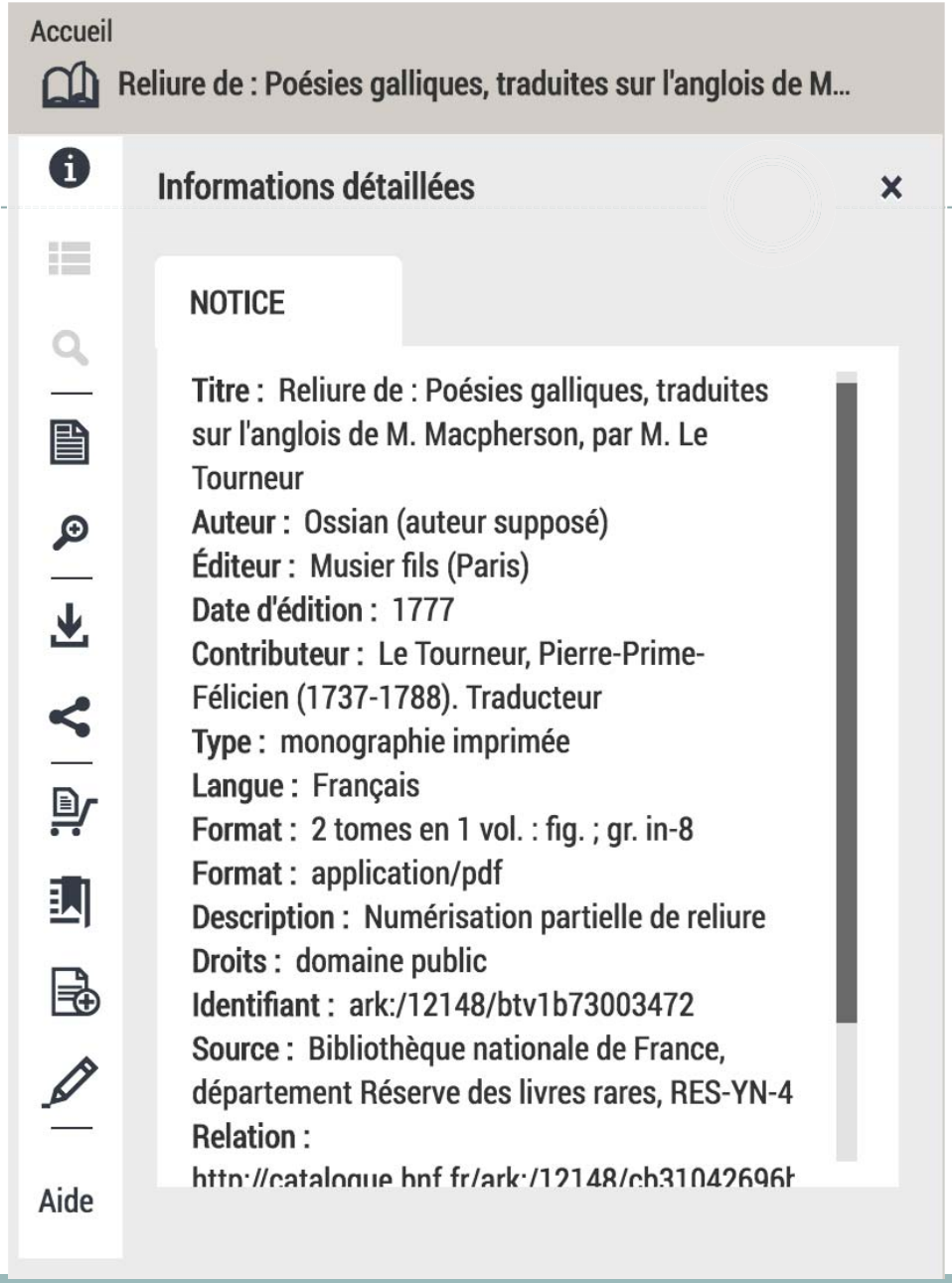
Publication : Paris : Musier fils, 1777

Description matérielle : 2 tomes en 1 vol. : fig. ; gr. in-8

Autre(s) auteur(s) : Le Tourneur, Pierre-Prime-Félicien (1737-1788). Traducteur

Notice n° : FRBNF31042696

Introduction



Introduction

Imprimer la page | Télécharger les données en rdf

plat supérieur

plat inférieur

Reliure en maroquin citron à décor de bordure néo-classique, aux armes de Napoléon Ier, Paris, atelier de Jean-Claude Bozerian, vers 1805

Paris. Bibliothèque nationale de France. Réserve des livres rares. RES-YN-4

Mots clés	Reliure à décor ♦ Plein cuir ♦ Maroquin ♦ Bordure ♦ Dorure • Fers pleins • Mosaïque de cuirs • Roulettes ornées
Dimensions :	279 x 211 x 46 mm

Contenu

Ossian. *Poésies galliques*. Paris : Musier fils, 1777. In-4.
Notice dans le Catalogue général de la BnF

+ de détails sur l'exemplaire relié

Reliure

Des liens entre les métadonnées

MARC 21	UNIMARC	DC	HTML	TEI
Machine Readable Cataloguing	UNIversal MARC	Dublin Core	Hypertext Markup Language	Text Encoding Initiative
Auteur : 100$a	700$a	<dc:title>	<title>	<title>
Titre : 210$a	200$a	<dc:creator>	<meta name="author">	<author>

Text Encoding for INTERCHANGE

- XML facilite **l'échange** des données et métadonnées.
- Pour échanger des métadonnées, il est parfois indispensable de passer d'un langage (TEI) à un autre langage (par ex. Dublin Core ou RDF).
- Il existe des **recommandations** pour la compatibilité des standards, par ex. MARC vers TEI, ou TEI vers DC.
- Comme dit précédemment la TEI est vue comme un format **pivot** permettant à partir **d'un seul fichier** de générer plusieurs sorties.

Parenthèse : Mapper les données avec XSLT

XSLT : eXstensible Stylesheets Language Transformations

- Comme son nom l'indique, il s'agit d'un langage utilisé pour effectuer des transformations. Mais de quoi ?
- XML-TEI vers un autre XML (HTML, DC, MarcXML...)
- Le XSLT est lui même un langage XML
- Cette transformation permet de “mapper” des balises avec d’autres balises, c’est à dire de les faire correspondre.

XSLT : visualisation notice

Le mapping permet deux points principaux :

- Récupérer les données dans un autre système (TEI > BDD)
- Afficher le contenu (TEI > HTML)
  - Fichier XML-TEI
  - Sortie Notice

Interroger des métadonnées

Le processus précédent (transformation XSLT) permet aussi l'indexation des données dans des plateformes.

- **Indexer** des données c'est permettre leur interrogation
- En gros, on dit au système d'indexation quelle information faire correspondre à un champ particulier.
  - ex : <title> correspond au titre de l'oeuvre, <author> correspond à l'auteur principal. Ce qu'un logiciel/plateforme ne peut savoir d'office sans que ce soit programmé pour. Voilà pourquoi les métadonnées qu'on souhaite interroger doivent avoir été encodées en amont.

Exemple d'un formulaire d'interrogation

Exemple d'un autre formulaire d'interrogation

Rechercher
Par mots :

saisir votre recherche

dans toute la reliure

ok

Type de reliure

- Reliure à décor
- Reliure armoriée
- Reliure courante
- Reliure de dédicace
- Reliure d'édition
- Reliure de libraire
- Reliure de présentation
- Reliure objet

Technique de reliure

- Bradel
- Demi-reliure
- Plein cuir
- Pleine peau
- Plein tissu
- Reliure à encadrement
- Reliure à la grecque
- Reliure à passes

Matériau de couvrure

<text>

<index indexName="binding_type">

<index indexName="binding_technique">

Exemple concret : interrogation / affichage

- Vers XTF : base Epistemon
- Vers Reliures BNF

- (cependant, l'exemple de Reliures-BNF est un peu différent, dans le sens où les « métadonnées » deviennent corps du texte, vu que le document numérique n'est qu'une description sans transcription)

Structurer les métadonnées

Rappel : structure globale

```
<TEI>
    <teiHeader>...</teiHeader>
    <text>
     <front>...</front>
     <body>...</body>
     <back>...</back>
    </text>
</TEI>
```

<teiHeader>

- L'en-tête TEI présente une description **structurée** des données contenues dans le fichier XML/TEI.
- Autrement dit, cette balise contient les **métadonnées** du fichier XML/TEI.
- Certaines balises du teiHeader sont **obligatoires**.

Structurer les métadonnées

<teiHeader> : contenu (habituel)

<fileDesc> (file description) : contient la description bibliographique du **fichier numérique** (fichier + sa source) **[OBLIGATOIRE]**

<encodingDesc> (encoding description) : contient la relation du fichier électronique avec sa source, c'est à dire les choix d'encodage

<profileDesc> (text-profile description) : donne une description des aspects non bibliographiques du texte, par exemple le langage du texte, et certains thesaurus

<revisionDesc> (revision description) : donne une liste de tous les changements qui ont été effectués sur le fichier, c'est une documentation interne

Structurer les métadonnées

<teiHeader> minimal : avec un unique fileDesc

```
<teiHeader>
    <fileDesc>
        <titleStmt>
            <title><!-- title du fichier numérique --></title>
        </titleStmt>
        <publicationStmt>
            <p><!-- Information sur la publication du fichier --></p>
        </publicationStmt>
        <sourceDesc>
        <p><!-- Information sur la source du fichier numérique --></p>
        </sourceDesc>
    </fileDesc>
</teiHeader>
```

Structurer les métadonnées : fileDesc

<titleStmt> (title statement) : contient l'ensemble des informations sur le titre du document numérique et ceux qui l'ont créé.

<title> : titre du document numérique
<author> : auteur du document numérique (si document natif)
<editor> : responsabilité secondaire du document numérique (si document natif)
<respStmt> : responsabilité intellectuelle (= collaborateurs sur le fichier numérique, comme le transcripteur, l'encodeur etc)
<name> : nom de la personne concernée
<resp> : fonction
<date> : date

Balisage des métadonnées

- Ouvrir l'exercice **01_Header_Test.xml**
- Partez du principe que vous êtes l'auteur du document TEI. Et que les autres responsabilités sont des personnes de la même équipe que vous.
- Remplissez le **titleStmt** avec les balises
  <title>, <author>,
  <respStmt> (et sous-balises : **<resp>** + **<name>**)

Structurer les métadonnées : fileDesc

<editionStmt> (edition statement) : regroupe les informations concernant la version de l'édition (optionnel)

<edition> : décrit les particularités de l'édition présente

Structurer les métadonnées : fileDesc

```
<titleStmt>
...
</titleStmt>
<editionStmt>
    <edition n="P2">Second draft, substantially
    extended, revised, and corrected.</edition>
</editionStmt>
```

Balisage des métadonnées

- Reprendre l'exercice
- Remplir le **<editionStmt>** en décrivant l'édition que vous souhaitez publier. Par exemple "Édition scientifique réalisée dans le cadre de l'ANR 3210".

<editionStmt>

    <edition>

Structurer les métadonnées : fileDesc

<publicationStmt> (publication statement) : contient les informations sur les lieux et l'éditeur de la ressource numérique. **(obligatoire)**

<publisher> : nom de l'éditeur ou du responsable de la publication numérique

<pubPlace> (publication place) : lieu d'édition

<availability> : décrit les droits de la ressource numérique

<p>

<date> : date d'édition de la ressource numérique

Structurer les métadonnées : fileDesc

```
<titleStmt>...</titleStmt>
<editionStmt>...</editionStmt>
<publicationStmt>
    <publisher>Oxford University Press</publisher>
    <pubPlace>Oxford</pubPlace>
    <date>1989</date>
    <availability><p>Copyright 1989, Oxford University
    Press</p></availability>
</publicationStmt>
```

Balisage des métadonnées

- Reprendre l'exercice
- Remplir le **<publicationStmt>** en ajoutant les informations éditoriales qui concernent l'organisme de publication. Dans le cadre d'un projet, il s'agit en général de votre tutelle.

<publisher>, **<pubPlace>**, **<date>**
<availability>
    <p>

Structurer les métadonnées : fileDesc

<sourceDesc> (source description) : décrit la source originale du document numérique. S'il n'y a pas d'original et que le document est un document numérique "natif", on le précise ici. **(obligatoire)**

<p> : pour une description simplifiée de la source, ou une mention "born digital"

<bibl> : pour une source bibliographique simple
<author>, **<title>** : auteur, titre
<publisher>, **<pubPlace>**, **<date>** : imprimeur/éditeur, lieu de publication, date.

Structurer les métadonnées : fileDesc

```
<titleStmt>...</titleStmt>
<editionStmt>...</editionStmt>
<publicationStmt>...</publicationStmt>
<sourceDesc>
    <bibl>
        <author>Eugène Sue</author>
        <title>Martin, l'enfant trouvé</title>
        <pubPlace>Bruxelles et Leipzig</pubPlace>
        <publisher>C. Muquardt</publisher>
        <date when="1846">1846</date>
    </bibl>
</sourceDesc>
```

Balisage des métadonnées

- Reprendre l'exercice
- Remplir le **<sourceDesc>** partant du principe que l'objet de votre édition est l'exemplaire suivant. Vous devez d'abord supprimer le <p>, et ajouter la balise **<bibl>** et les sous-éléments suivants :

<title> Titre : **Andromaque**
<author> Auteur : **Jean Racine**
<date> Date : **1668** (possibilité d'utiliser @when)
<publisher> Éditeur : **Th. Girard**
<pubPlace> Lieu d'édition : **Paris**

Structurer les métadonnées - encodingDesc

<teiHeader> : rappel

<teiHeader>

<fileDesc>...</fileDesc> *(description bibliographique du fichier numérique : fichier + source)*

<encodingDesc>...</encodingDesc> *(contient la relation du fichier électronique avec sa source, c'est à dire les choix d'encodage)*

<profileDesc>...</profileDesc> *(donne une description des aspects non bibliographiques du texte)*

<revisionDesc>...</revisionDesc> *(donne une liste de tous les changements qui ont été effectués sur le fichier)*

</teiHeader>

Structurer les métadonnées : encodingDesc

<projectDesc> (project description) : décrit pourquoi le fichier a été encodé et comment il a été encodé

<p> : description simple

ex :

```
<projectDesc>
    <p>Projet d'édition universitaire. Source choisie parce
    qu'il s'agit d'un exemplaire unique.</p>
</projectDesc>
```

Structurer les métadonnées : encodingDesc

<editorialDecl> (editorial practice declaration) : donne les détails du niveau d'encodage, des principes éditoriaux, ou encore du niveau de normalisation du texte

<correction> : explicite le niveau des corrections apportées à la transcription

<normalization> : explicite le niveau de normalisation de la langue

<hyphenation> : a-t-on respecté ou non les sauts de lignes dans l'original et les mots coupés ?

<punctuation> : a-t-on respecté la ponctuation originale, a-t-elle été rétablie ?

<p> : balise <p> pour chaque élément ci-dessus

Structurer les métadonnées : encodingDesc

```
<projectDesc>...</projectDesc>
<editorialDecl>
        <correction>
        <p>Errors in transcription controlled by using the
        WordPerfect spelling checker.</p>
    </correction>
    <normalization source="
http://szotar.sztaki.hu/webster/">
        <p>All words converted to Modern American spelling
    following
        Websters 9th Collegiate dictionary.</p>
    </normalization>
</editorialDecl>
```

Structurer les métadonnées : encodingDesc

Balisage des métadonnées

- Reprendre l'exercice
- Ajouter la balise **<encodingDesc>**
- Ajouter à l'intérieur un **<projectDesc>** et la sous-balise **<p>**.
- Puis utiliser un **<editorialDecl>**, ainsi que la sous-balise **<normalization>** + <p>

Structurer les métadonnées - profileDesc

<teiHeader> : rappel

<teiHeader>

<fileDesc>...</fileDesc> *(description bibliographique du fichier numérique : fichier + source)*

<encodingDesc>...</encodingDesc> *(contient la relation du fichier électronique avec sa source, c'est à dire les choix d'encodage)*

<profileDesc>...</profileDesc> *(donne une description des aspects non bibliographiques du texte)*

<revisionDesc>...</revisionDesc> *(donne une liste de tous les changements qui ont été effectués sur le fichier)*

</teiHeader>

Structurer les métadonnées : profileDesc

<langUsage> (language usage) : décrit l'ensemble des langues présentes dans la transcription

<language> : avec l'attribut **@ident**, liste une langue utilisée

ex :

```
<langUsage>
    <language ident="frm">Français</language>
    <language ident="lat">Latin</language>
    ...
</langUsage>
```

Balisage des métadonnées

- Reprendre l'exercice
- Placer la balise **<profileDesc>**
- Ajouter à l'intérieur un **<langUsage>**
- Lister les langues ainsi que leurs codes ISO : latin (lat), grec (ell), moyen français (frm) au moyen de la balise **<language>** et de l'attribut **@ident**

Structurer les métadonnées - revisionDesc

<teiHeader> : rappel

<teiHeader>

<fileDesc>...</fileDesc> *(description bibliographique du fichier numérique : fichier + source)*

<encodingDesc>...</encodingDesc> *(contient la relation du fichier électronique avec sa source, c'est à dire les choix d'encodage)*

<profileDesc>...</profileDesc> *(donne une description des aspects non bibliographiques du texte)*

<revisionDesc>...</revisionDesc> *(donne une liste de tous les changements qui ont été effectués sur le fichier)*

</teiHeader>

Structurer les métadonnées : revisionDesc

<revisionDesc> (revision description) : donne une liste des révisions du fichier pour conserver un historique du projet

<change> : chaque changement peut être documenté, avec l'attribut **@who** et **@when**

ex :

```
<revisionDesc>
    <change when="2015" who="LBe">Création du fichier en XML-TEI</change>
    <change when="2015-11-09">Ajout des métadonnées dans le TEIHeader</change>
...
</revisionDesc>
```

Structurer les métadonnées : revisionDesc

Balisage des métadonnées

- Reprendre l'exercice
- Ajouter la balise **<revisionDesc>**
- Ajouter à l'intérieur quelques **<change>** et utiliser les attributs **@who** et **@when**

Métadonnées plus complexes : msDesc

- Les documents anciens ont en général besoin d'un niveau supplémentaire à l'intérieur de la balise **<sourceDesc>** (rappel : à l'intérieur de <fileDesc>).
- Tout à l'heure, on a juste utilisé **<bibl>** avec des informations assez simples, mais rien ne nous permettait en l'état d'indiquer où était **conservé** l'ouvrage, quel était son **format**, s'il y avait une ressource numérique existante en ligne etc.
- Pour préciser ce genre de choses, il faut ajouter la couche **<msDesc>**, ainsi que toutes les sous-balises associées

<teiHeader> rappel structure

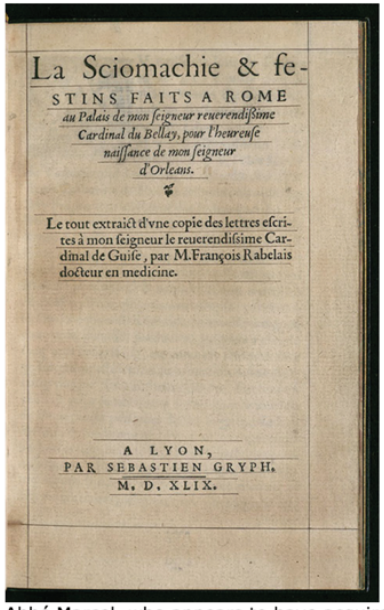
```
<teiHeader>
    <fileDesc>
        <titleStmt>...</titleStmt>
        <editionStmt>...</editionStmt>
        <publicationStmt>...</publicationStmt>
        <sourceDesc>
            <msDesc>...</msDesc>
        </sourceDesc>
    </fileDesc>
    <encodingDesc>...</encodingDesc>
    <profileDesc>...<profileDesc>
    <revisionDesc>...<revisionDesc>
</teiHeader>
```

Exemple Header complexe

François Rabelais, *La Sciomachie et festins faits à Rome*, Lyon, 1549

Suggérer une modification

Accéder au document (Mode texte/image)


La Sciomachie & fe-
STINS FAITS A ROME
au Palais de mon seigneur reuerendissime
Cardinal du Bellay, pour l'heureuse
naissance de mon seigneur
d'Orleans.

Le tout extraict d'vne copie des lettres escri-
tes à mon seigneur le reuerendissime Car-
dinal de Guise, par M. François Rabelais
docteur en medicine.

A LYON,
PAR SEBASTIEN GRYPH.
M. D. XLIX.

Titre long : La Sciomachie & festins faits à Rome au Palais de monseigneur reverendissime Cardinal du Bellay pour l'heureuse naissance de monseigneur d'Orléans
Adresse typographique : A Lyon, Sebastien Gryph. M. D. XLIX.

Auteur : Rabelais (François)
Auteur secondaire : Du Bellay (Jean)

Publication : 1549

Lieu d'impression : Lyon
Libraire : Gryphe (Sébastien)
Imprimeur : Tournes (Jean de)

Format : 8°
Collation : [1, 1 bl.], 3-31, [1 bl.] p. (sig. A-B^{8})
Domaine : Littérature | Genre : Recit
Langue(s) : Français, Latin.

Localisation : Bayerische Staats Bibliothek, BSB / Außenmagazin , Munich, Allemagne | Cote : 960727

État physique : (Tours :) Exemplaire lavé et réglé. Dans cet exemplaire, à l'emplacement du nom du dauphin (p. 3, l. 5) est laissé un espace vide, où le nom de "Louis" est inscrit au crayon de bois à une date relativement récente. Un minuscule trou à la première ligne du titre sur le caractère "h".
Reliure : (Tours :) Reliure en maroquin bleu (XIXe siècle ; signée Masson-Debunnelle). Dos à 5 faux-nerfs. Tranches dorées.
Provenance : (Tours :) Au contreplat et à la garde supérieurs, ex-libris estampés sur cuir : "EX MUSEO DOUBLE" ; "Bibliothèque E. M. Bancel" (XIXe siècle : Étienne-Marcel Bancel) ; "Pro captu lectoris" ; ex-libris gravé sur papier : "Ex libris Anat. France", "Quidquid erit pati" (Anatole France) ; ex-libris estampé "Ex libris Jean Tannery". "Bequeathed by the Abbé Marcel, who appears to have acquired it at the Bry sale, 5 December 1966, no. 179, for 6000 fr." (Michel de Bry ; Raymond Marcel).
Une note marginale au crayon de bois à la page 20 ; inscription au crayon à la garde inférieure : "EE1. III. 54".
Note(s) : L'édition numérique de cette « sciomachie », terme savant que Rabelais traduit par « simulacre de bataille », a été établie à partir de l'édition unique de 1549 (Lyon, Sébastien Gryphe) d'après l'exemplaire de la Bayerische Staatsbibliothek (Ital. 448), qui montre deux R majuscules, et sur l'exemplaire de la Bibliothèque Municipale de Tours (Fonds Marcel Rés. 3542, ex libris d'Anatole France). Cet exemplaire, distinct par deux légères différences graphiques corrigées sur l'exemplaire de Munich, serait antérieur à ce dernier. Apparaît au crayon et d'une écriture moderne le nom du deuxième fils du roi, Louis, dont la naissance est célébrée dans le texte, nom laissé en blanc par l'imprimeur. Rabelais offre, sous la forme d'une prétendue lettre au cardinal de Guise, le récit d'une fête donnée à Rome pour cet événement en mars 1549 (Louis d'Orléans mourra en octobre), et à laquelle il a dû assister, même si la manipulation du témoignage est probable. Très inspirée par l'entrée de Henri II à Lyon en 1548, mais aussi par les diableries des mystères avec ses feux d'artifice, elle a été précédée de deux versions en français d'un premier récit italien publié la même année par Antonio Buonaccorsi, source de Rabelais. Elle est marquée par de nombreux italianismes (voir la notice de François Moreau dans Rabelais, *Œuvres*, Pléiade, 1994, p. 1725-1726, et R. Cooper, *Rabelais et l'Italie*, 1991, p. 70-71).

Exemple Header complexe

Numérisation(s) :
Bayerische Staats Bibliothek Digital (XBSB_Rabelais_Sciomachie1549)
BVH - Bibliothèque Municipale de Tours (Rés. 3542 FM)

Bibliographie :
- La Charité (Cl.), *« La Sciomachie (1549) de Rabelais : la 'juste quantité d'une epistre' ou l'alibi épistolaire de la propagande épidictique »*, Tangence, n° 72, 2003, p. 111- 126
- Rawles (St.) et Screech (M.), *New Rabelais Bibliography*, Genève, 1987, n° 103
- Plan (P.-P.), *Bibliographie rabelaisienne*, Paris, 1904, p. 235-236, n° XV
- Baudrier (H.-L.) & Baudrier (J.), *Bibliographie lyonnaise. Recherches sur les imprimeurs libraires, relieurs et fondeurs de lettres de Lyon au XVIe siècle*, Lyon, 1895-1921, t. VIII, p. 232.
- Gültlingen (S. Von), *Bibliographie des livres imprimés à Lyon au seizième siècle*, Baden-Baden, t. V, 1997, "Sébastien Gryphe", p. 176, n° 1096 ; t. IX, 2004, p. 157, "Jean de Tournes", n° 151
- Cartier (A.), Bibliographie des éditions des De Tournes, imprimeurs lyonnais, Paris, 1937 (réimpression : Genève, 1970), n° 151
- Cooper (R.), Rabelais et l'Italie, Genève, 1991, Texte XII (p. 183-211)
- La Sciomachie (1549) sur ReNom

Corpus : Epistemon / Rabelais
Responsable BVH CESR, Relecture, Éditeur scientifique : Marie-Luce Demonet (2015)
Transcription : Prestataire (1996)
Relecture de la transcription : Jonas Kurscheidt (2014)
Encodage : Prestataire (2014)
Encodage : Marie Olivron (2015)
Contrôle de l'encodage : Marie Olivron (2015)
Contrôle de l'encodage : Lauranne Bertrand (2015)
Révision de la notice : Toshinori Uetani (2015)
Normes éditoriales :
- *État du texte :* Non dissimilé et non désabrégé
- *Normalisation* :
  La transcription est dissimilée et désabrégée. Les s long sont normalisés.
- *Gestion des césures (hyphénation)* :
  La division des mots en fin de ligne respecte l'original.
- *Informations éditoriales :*
  Cette édition suit deux exemplaires de référence.

URL de référence :
http://xtf.bvh.univ-tours.fr/xtf/view?docId=tei/XBSB_Sciomachie1549/XBSB_Sciomachie1549_tei.xml;doc.view=notice
Crédits : Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 (CC BY-NC-SA 4.0). 2015
Mise en ligne : 19/10/2015

Exercice : msDesc

Ouvrir l'exercice : 02_header_msDesc.xml

- Vous devez procéder par étape
- Des instructions sont contenues dans les éléments <msIdentifier>, <msContents>, <physDesc>, <history> et <additionnal>
- Deux types d'instructions
  - du texte pour vous aider à intégrer le fonctionnement des balises, qui doit être peu à peu supprimé
  - certaines balises déjà placées que vous ne devez pas modifier (il y a leurs balises fermantes)
  - certaines balises listées, sans leur balise fermante, que vous devez compléter avec les infos entre guillemets

Exercice 2 : header

Ouvrir l'exercice : 03_header.txt

En vous inspirant de l'ensemble des slides ou des exercices déjà effectués, ouvrir l'exercice et créer un header de toute pièce, en faisant Fichier > Nouveau > ... > All

Et si les métadonnées devenaient document numérique ?

- Revenons sur le projet de description des Reliures, précédemment évoqué : http://reliures.bnf.fr/

- Il s'agit, dans le cadre de ce projet, non pas d'encoder le document, mais de **décrire** la reliure afin d'obtenir une notice.

- L'unité documentaire devient donc un ensemble de métadonnées, qui décrivent un objet.

Et si les métadonnées devenaient document numérique ?

- Structure :
  - teiHeader : métadonnées du fichier numérique (born digital, la notice n'existait pas avant le projet)
  - corps du fichier TEI : notice descriptive de la reliure <body> qui contient
    - une unique qui représente la notice
    - puis à l'intérieur bookDescription [msDesc]
      - bookIdentifier [msIdentifier] : localisation de l'oeuvre
      - bookContents [msContents] avec bookItem [msItem] : description du livre
      - physDesc : description physique de l'ouvrage
        - bindingDesc : reliure, histoire, matériaux etc

Et si les métadonnées devenaient document numérique ?

- Démonstration sur la notice n° 17.
- FR-751131011_RELIURE_NOT_00017.xml

Initiation XML-TEI

CITATIONS, DISCOURS, RÉFÉRENCES

Lauranne Bertrand (+ inspiration N. Dufournaud) – 09/12/2016

Introduction

L'encodage permet de mettre en valeur le sens du texte, et d'en extraire certains passages, dès lors que les balises s'y attachent. Nous allons voir comment encoder :

- Les **citations** au sein du texte
- Les **dialogues** et prises de parole au sein d'une œuvre (discours direct / indirect)
- Les **références** à d'autres ouvrages (encodage bibliographique)

Introduction

- Jusqu'alors, on a surtout vu l'encodage de la structure : imprimés, manuscrits, oral

- Dès à présent, il s'agit de faire ressortir **la sémantique** du texte.

- C'est un encodage souvent coûteux en temps mais essentiel pour l'analyse profonde du texte encodé (la véritable plus-value de la TEI par rapport à un autre XML)

Introduction

- Différents niveaux de couches textuelles :
  - Discours, directs ou indirects au sein d'une œuvre
  - Citations non bibliographiques, lorsqu'un auteur rapporte un autre écrit mot pour mot.
  - Les renvois vers d'autres sources externes au texte via des pointeurs
  - Les citations bibliographiques
- Vous vous en doutez, chaque couche textuelle a sa ou ses balises associées

Citations

- **<cit>** (cited quotation) : pour une citation provenant d'un autre document, ainsi que sa référence
  - **<cit>**
    <quote>Alea jacta est</quote>
    <bibl><author>Jules César</author></bibl>
    </cit>
- **<quote>**(quotation) : imbriqué dans <cit> pour une référence complète, ou utilisé seul pour un passage sans référence associée
  - **<quote>**Je vous l'avais prédit qu'en dépit de la Grèce, de votre sort encor vous seriez la maîtresse**</quote>**

Citations

- **<cit>** utilisera obligatoirement **<quote>**

- Mais **<quote>** peut être utilisé tout seul
  - **@source** : pour spécifier la source de la citation si elle est déclarée quelque part

Discours

Pour les différents discours (direct ou indirect) on a deux choix possibles :

- **<said>** (speech or thought) : discours ou pensée
  - <said who="#pant">Tu es sûr de cela, Panurge ?</said>
- **<q>** (quoted) : tout passage qui est distingué du texte qui l'englobe par une marque ou une autre indication : incluant les discours, les pensées, les termes techniques, les citations...
  - <q>- S'il vos pleust, fait Galaad, je m'en entrasse en cele a senestre, car si com je pens je m'en getasse mielz que vos.</q>

- **<said>** comporte deux attributs spécifiques
  - **@aloud** : le discours est-il pensé ou dit à haute voix ? Valeurs obligatoires ici : **true, false, inapplicable, unknown**
  - **@direct** : discours direct ou indirect, avec les valeurs : **true, false, inapplicable, unknown**
  - <said aloud="true" direct="true">Tu es sûr de cela, Panurge ?</said>
  - @who : n'est pas spécifique à <said> mais il est recommandé de l'utiliser.

- **<q>** peut être spécifié au moyen de
  - **@type** : avec les valeurs suggérées **spoken, thought, written, soCalled, foreign, distinct, term, emph, mentioned**
  - **@who**
  - **<q type="spoken" who="#gal">**S'il vos pleust,**</q>** fait Galaad, **<q type="spoken" who="#gal">**je m'en entrasse en cele a senestre, car si com je pens je m'en getasse mielz que vos.**</q>**

Discours

- Pour une analyse des discours direct/indirect, l'élément **<said>** semble donc le plus approprié.

- **<q>** est devenu un élément beaucoup plus générique qu'il ne le fut.

Exercice : Discours et Citations

Deux exercices en un : **disc_cit.xml**

- Première : encoder les discours direct et indirects avec **<said>**
  - Utiliser @direct="false" quand il s'agit du discours indirect (milieu du texte) (pour éviter d'avoir à placer l'attribut sur chaque balise)
  - Utiliser @who pour désigner qui parle
  - N'hésitez pas à englober les formules telles que « dit-il » lorsqu'elles se trouvent au milieu du discours

Exercice : Discours et Citations

Deux exercices en un : **disc_cit.xml**

- Deuxième : encoder les citations avec **<cit>** et/ou **<quote>**. On peut aussi utiliser <q> pour les mots mis en valeur.
  - NB : Pour les <cit> avec en plus de la citation, la référence à un ouvrage, englober l'ensemble de la référence avec la balise **<bibl>** que nous verrons en détail ensuite.
  - Vous avez le choix que de laisser les guillemets ou non (on pourrait les rétablir à l'affichage grâce au balisage par ex)

Références : liens et pointeurs

On a vu jusqu'alors, le fonctionnement des @xml:id et des pointeurs (valeurs précédées de #). Mais certaines balises sont dédiées aux liens :

<ref> : détermine un lien vers une source interne ou externe, permet un contenu

<ptr/> : détermine un lien vers une source interne ou externe, aucun contenu de balise

@target : attribut qui permet de déclarer le lien.

- Différence entre <ref> et <ptr/>

<**ref** target="http://www.tei-c.org/release/doc/tei-p5-doc/en/html/CO.html#COXR">Guidelines</ref>

<**ptr** target="#n21.3"/>

Dans le premier cas, j'ai bien un lien, et le texte du lien.
<ref> est un élément englobant.
Dans le second cas, je n'ai qu'un pointeur, ici vers un identifiant, la balise est vide.

Les deux peuvent contenir des URI.

Liens / Pointeurs

- Référence interne :

```
<div xml:id="n3.21">...</div>
....
<ref target="#n3.21">...</ref>
```

- Référence externe :

```
<ref target="https://fr.wikipedia.org/wiki/
Uniform_Resource_Identifier">Wikipedia : URI</ref>
```

Liens / Pointeurs

- Définir un identifiant : **@xml:id**

- Les attributs pointeurs :

  - @target (<ref>, <ptr/>, <note>, ...) : lien cible
  - @who (<sp>, <said>, <change>...) : identité
  - @ref (<name>, <persName>, <placeName>...) : référence externe
  - @source (<choice>, <supplied>...) : source
  - @resp (<note>, <supplied>, <add>...) : responsabilité

- Beaucoup de ces attributs sont permis sur de nombreuses balises. Pour vérifier, une seule ressource : Les Guidelines.
  - http://www.tei-c.org/release/doc/tei-p5-doc/en/html/ref-att.canonical.html

RAPPEL : Le chevauchement

<l> .... **<delSpan spanTo="#id4"/>** ... </l>
<l> ... **<anchor xml:id="id4"/>** ... </l>
<l> .... </l>

- Ici on a une suppression commençant au milieu d'un vers et se terminant au milieu du vers suivant
- À cause des <l>, si on avait utilisé <del>, on aurait eu un cas interdit par le XML, le chevauchement :
  - <l>.... <del>.....</l>
  - <l>.....</del>....</l>

Références : Bibliographie

Encoder une référence bibliographique, méthode simple :

- **<bibl>** (bibliographic citation)

  - **<bibl>**Blain, Clements and Grundy: Feminist Companion to Literature in English (Yale, 1990)**</bibl>**

Bibliographie

Structuration classique :

<bibl>
<author>Blain**</author>**,
<title>Clements and Grundy: Feminist Companion to Literature in English**</title>**
(**<pubPlace>**Yale**</pubPlace>**,
<date>1990**</date>**)
</bibl>

Structuration plus complète :

<biblFull> (fully-structured bibliographic citation)

```
<biblFull>
        <titleStmt>
                <title>Les Oeuvres de Clément Marot...</title>
                <author>Clément Marot</author>
        </titleStmt>
        <publicationStmt>
                <publisher>Étienne Dolet</publisher>
                <pubPlace>Lyon</pubPlace>
                <date>1538</date>
        </publicationStmt>
</biblFull>
```

Structuration encore plus complexe : **<biblStruct>** (structured bibliographic citation)

Bibliographie

```
<biblStruct>
    <monogr>
        <title>Sensuivent Les Blasons anatomiques du corps femenin...</title>
        <author>Clément Marot</author>
        <author role="secondaire">Jean de Vauzelles</author>
        <author role="secondaire">Maurice Scève</author>
        <imprint>
          <pubPlace>Paris</pubPlace>
          <date>1543</date>
            <respStmt>
              <resp>Libraire</resp>
              <name>Charles L'Angelier</name>
            </respStmt>
        </imprint>
        <extent>16°</extent>
        <biblScope unit="page">22-25 ff.</biblScope>
    </monogr>
</biblStruct>
```

Exercice : Bibliographie

- Structurer le fichier suivant : **bibl.xml**

  - **<bibl>** pour chaque référence
    - **<author>** auteur
    - **<title>** titre de l'article, titre du volume
    - **<editor>** éditeur
    - **<date>** date
    - **<biblScope>** @unit pour les numéros de page, ou de volumes
    - **<ref>** @target pour les liens

Initiation XML-TEI

NOMS PROPRES ET ENCODAGE SÉMANTIQUE

Lauranne Bertrand – 09/12/2016

Introduction

Pourquoi ?

- Par amour de l'encodage ?
- Pour mettre en valeur le sens de mon texte ?
- Pour générer automatiquement des index : noms de personne, de lieux, etc... ?
- Pour améliorer la finesse des requêtes dans les outils de fouille de textes ?

Introduction

- Une balise très générique est disponible en TEI, qui signifie «nom propre» : il s'agit de la balise :
  - <name>...</name>
- Elle n'offre que peu d'information sur le terme qu'elle contient : est-ce une personne, un lieu ?

<persName>

- **<persName>** : nom de personne
  - <forename> : prénom
  - <surname> : nom de famille, patronyme
  - <roleName> : titre officiel, rang, rôle
  - <nameLink> : particule
  - <genName> : attribut générationnel (Pline «le Jeune», Henri «IV»)
  - <addName> : nom additionnel comme un surnom ou un alias

persName

- ex : Geoffroy Ier Grisegonelle
- <persName>
  <forename>Geoffroy</forename>
  <genName>Ier</genName>
  <addName>Grisegonelle</addName>
  </persName>

Exercice : persName

- Ouvrir Semantique.xml.
- Encoder les noms de personne avec la balise **<persName>** et ses sous-balises :
  - **<forename>, <surname>, <nameLink>, <genName> et <roleName>**

- *Ne vous occupez pas des noms de lieux.*
- *Vous avez la possibilité de typer les persName avec @type="mythologie" ou @type="historique" par exemple (comme toujours, vous avez le choix des valeurs qui vous semblent les plus pertinentes)*

<orgName>

- **<orgName>** sert à encoder le nom d'une organisation ou d'une institution.
  - Ex : Chapitre de Saint Léon

```
<orgName>
  Chapitre de
  <persName>
    <roleName>Saint</roleName> Léon
  </persName>
</orgName>
```

<placeName>

- **<placeName>** : nom de lieu absolu ou relatif, c'est-à-dire nom d'un bâtiment, village, ville ou pays.
  - ex : Église Saint-Christophe à Tours

```
<placeName type="eglise">
   Église
   <persName>
      <roleName>Saint</roleName>-Christophe
   </persName>
</placeName>
à <placeName type="ville">Tours</placeName>
```

<geogName>

- **<geogName>** : nom ayant une caractéristique géographique (La Loire, le Mont Sinai...)
  - Cette balise est associée à **<geogFeat>** qui sert à encoder le nom commun caractérisant le nom géographique (mont, fleuve, vallée, montagne...)
  - Ex : le Mont Sinai

```
<geogName>
  Le
  <geogFeat>Mont</geogFeat>
  Sinai
</geogName>
```

Exercice : placeName, geogName

- Reprenez l'exercice Semantique.xml
- Encodez les noms de lieux avec les balises :

- **<placeName>** avec son attribut **@type**
  - @type="ville"
  - @type="pays"
  - @type="continent"
  - @type="region"....

- **<geogName>** et sa sous-balise **<geogFeat>**.

Topographie actuelle

- Nous n'utilisons pas ces balises dans le corps de nos textes anciens mais dans le teiHeader :
  - <country> : pays
  - <settlement> : ville
  - <region> : région ou département
- Ce sont des balises adaptées à notre découpage administratif actuel.

Dates

- Les attributs suivant accompagnent la balise **<date>** (corps du texte) ou **<docDate>** (page de titre). Leurs valeurs sont normalisées selon la norme iso : **aaaa-mm-jj**.
- **@when** : Date exacte.
  - ex : <date **when="2016-01-20"**>20 janvier 2016</date>
  - ou <date **when="1552"**>M.D.LII</date>

Dates

- **@notBefore** : indique la date la plus antérieure possible
  - ex : c'était <date **notBefore="2001"**>un ou deux ans</date> avant sa naissance en <date **when="2003"**>2003</date>.
- **@notAfter** : indique la date la plus postérieure possible
  - ex : Nous partirons d'ici <date **notAfter="2012-03-31"**>deux mois maximum</date> à partir de ce <date **when="2012-01-31"**>jour</date>.
- Ces deux attributs peuvent bien entendu être utilisés ensemble pour définir un intervalle.

Exercice complet

- Encoder sémantiquement le texte de l'exercice Semantique2.xml

- **<persName>**
  - <forename>, <surname>, <roleName>, <addName>, <nameLink>, <genName>
- **<placeName>**
  - @type
- **<geogName>**
  - <geogFeat>
- **<date when="aaaa-mm-jjTHH:MM:SS">**
  - @when, @notBefore, @notAfter

<offset>

- Parfois, on a une mention qui indique soit un rapport au temps relatif, ou bien un repère spatial
  - Avant 12h / Derrière le Mont Blanc
- Ces mentions peuvent être encodées dans **<offset>**
  - <offset>Derrière</offset> <geogName>le <geogFeat>Mont</geogFeat> Blanc</geogName>
  - <offset>Avant</offset> <time when="12:00:00">12h</time>

<measure>

- Dans la même lignée, **<measure>** permet d'indiquer un mot ou plusieurs mots qui renvoient à une quantité
  - 12 €
  - <measure type="monnaie">12 €</measure>

Démonstration ReNom

- ReNom : http://renom.univ-tours.fr/fr

Génération d'un index avec XSLT

- Comme dit en introduction, un balisage sémantique sert souvent à la génération d'index. Vu que vous avez manipulé XSLT la dernière fois, vous pouvez ouvrir la feuille **Index.xsl** qui va vous permettre de générer un index en TEI.

- Mais avant tout parlons des balises servant à marquer des entrées d'index.

Index

- Un index peut être auto-généré grâce à certaines transformations XSLT (packages disponibles sur tei-c). Le principe est le suivant :
  - On balise dans le texte les mots à indexer grâce aux balises **<index>** et **<term>** :
    - Machin disait souvent que
      <index>
      <term>Bidule</term>
      </index>
      avait de drôles de manies...

Index

- Après avoir indexé certains termes au fil du texte, on place dans le <back> la balise vide suivante :

  - <back>
    <divGen type="index"/>
    </back>

- C'est la transformation qui ensuite ira chercher toutes les balises **<index>** et listera les valeurs automatiquement à la place de **<divGen/>**

Index

- Petit test : Encoder dans un fichier d'exercice quelques mots avec les balises **index/term**
- Ajouter la balise <back> avec à l'intérieur **<divGen type="index"/>**
- Lancer la transformation **XHTML** par défaut

Index

- Cependant, dans notre exercice, nous indiquons déjà les entités nommées grâce à un balisage fin si bien que nous n'avons pas besoin d'utiliser la balise <index>.

- Revenez à la feuille XSLT **Index.xsl** et regardez un peu ce qu'il s'y passe. Reconnaissez vous les balises de l'index en fin de document ? Il s'agit des balises **<list>** et **<term>**.

Index

- Vous voyez aussi qu'il vous reste du travail, car la feuille n'est pour l'instant paramétrée que pour générer un index des Dates.
  - Nous allons commenter cette feuille ensemble
  - Puis paramétrer la transformation pour les lieux et les noms de personnes
  - Puis lier cette feuille avec notre fichier XML pour tout transformer