

ÉCOLE NATIONALE DES CHARTES  
UNIVERSITÉ PARIS, SCIENCES & LETTRES

---

**Kelly Christensen**

*licenciée ès enseignement musical*

*diplômée de master musicologie*

*diplômée de doctorat musicologie*

# Modélisation des transcriptions ALTO avec la TEI

En complétant le pipeline du projet  
*Gallic(orpor)a*

Mémoire pour le diplôme de master  
« Technologies numériques appliquées à l'histoire »

2022



# Résumé

Quand des modèles OCR (*Optical Character Recognition*) et HTR (*Handwriting Text Recognition*) extraient les données d’une ressource textuelle numérisée, les informations relatives à la structure physique de l’image risquent de se perdre. Un schéma XML standardisé qui s’appelle ALTO (*Analyzed Layout and Text Object*) a été créé afin de conserver et structurer ces données non-textuelles et géométriques en les tenant en relation avec le contenu textuel. La plupart des modèles OCR et HTR compte sur ce schéma. Cependant ALTO ne convient pas bien à l’édition numérique ni aux traitements automatique du langage. Les éditeurs et les chercheurs en lettres attendent un schéma XML plus courant dans le monde des humanités numériques : la TEI (*Text Encoding Initiative*). Il faut donc un mapping pour transformer un fichier ALTO en fichier TEI sans perdre aucune donnée lors du processus. Cette transformation automatisée permet à conserver les données particulières au schéma ALTO, telles que celles sur la segmentation et sur la structure physique du document numérisé, ainsi qu’à exploiter le contenu textuel de la ressource textuelle. La flexibilité de la TEI et son usage très répandu rendent le schéma idéal pour mieux valoriser les données produites par les modèles OCR et HTR.

Dans le cadre du stage pour obtenir le diplôme de Master 2 « Technologies numériques appliquées à l’histoire », ce mémoire porte sur la modélisation de la transformation de ALTO en TEI. Cette modélisation a été réalisée dans le cadre du projet *Gallic(orpor)a*, financé par la BnF (Bibliothèque nationale de France) lors d’un stage qui a eu lieu au sein du laboratoire ALMA<sub>na</sub>CH (Automatic Language Modelling and Analysis & Computational Humanities) entre avril et juillet 2022.

**Mots-clés :** HTR, OCR, ALTO, TEI, TAL, édition numérique.

**Informations bibliographiques :** Kelly Christensen, *Modélisation des transcriptions ALTO avec la TEI. En complétant le pipeline du projet Gallic(orpor)a*, mémoire de master « Technologies numériques appliquées à l’histoire », dir. [Noms des directeurs], École nationale des chartes, 2022.



# Remerciements

M<sup>Es</sup> remerciements vont tout d'abord à...



# Introduction





# Première partie

## Présentation du projet



# Chapitre 1

## Qu'est-ce que l'HTR ?

### 1.1 L'objectif de l'HTR

Décrire les enjeux de la reconnaissance automatique des caractères de texte sur une image. Donner des exemples de cette tâche avec une petite figure qui montre le processus.

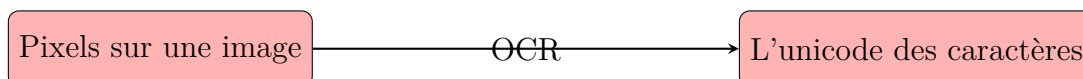


FIGURE 1.1 – figure

### 1.2 L'histoire et l'évolution de la technologie

Décrire l'évolution de cette technologie, en commençant avec l'OCR.

### 1.3 Les deux approches actuelles

Expliquer qu'il y a actuellement deux approches : OCR et HTR.

#### 1.3.1 L'OCR

Expliquer comment cette technologie plus ancienne compte sur les polices des caractères et a besoin des technologies de traitement automatique des langues.

#### 1.3.2 L'HTR

Expliquer comment l'HTR réussit à s'entraîner sur les courbes des caractères écrits et peut donc compléter ses tâches sans besoin d'un modèle TAL derrière. (Au moins, c'est actuellement ce que j'ai compris des explications d'Ariane Pinche) Justifie pourquoi l'HTR a été privilégié dans le projet Gallic(orpor)a, même pour les imprimés.



## Chapitre 2

# Au commencement, il y avait les *guidelines SegmOnto*

### 2.1 La problématique

Décrire le problème d'une manque de cohérence dans la manière par laquelle la communauté scientifique caractérise les composants d'un texte numérisé.

### 2.2 Les *guidelines* de *SegmOnto*

Présenter la solution proposée par le projet *SegmOnto*.

### 2.3 L'avantage

Expliquer comment le lexique systématisé de *SegmOnto* m'a permis à programmer un script qui sait faire le liaison entre le lexique de *SegmOnto* et le lexique de la TEI.



# Chapitre 3

## Le rêve du projet *Gallic(orpor)a*

### 3.1 Le contexte du projet

Présenter les institutions qui soutiennent le projet (École nationale des chartes, Inria, Université de Genève, DataLab de la BnF) et les projets qui l'ont précédé et sur lesquels *Gallic(orpor)a* compte.

### 3.2 L'objectif du projet

Le projet a pour but d'arriver des images numérisées à un document numérique sans besoin d'un éditeur. L'idée est qu'avec le produit du pipeline, qui sera un texte prédit et pré-éditorialisé, un individu qui n'est pas forcément spécialisé dans l'informatique peut facilement le prendre et effectuer des analyses et/ou éditer le texte du document dont les images de ses pages ont été traités.

### 3.3 Le pipeline

Présenter le pipeline. Réutiliser l'information qu'on a générée pour le présenter à la BnF.





## Deuxième partie

### Exposition de la préparation et du travail d'analyse



# Chapitre 4

## Un pipeline visant à tout rassembler

### 4.1 Les données d’entraînement HTR

J’ai généré des données d’entraînement sur eScriptorium en suivant les mêmes conseils donnés aux vacataires chargés à segmenter et transcrire les images d’un document selon les normes de *SegmOnto*. Je me suis familiarisée avec leur travail et la création du corpus d’or qui serviraient à l’entraînement du modèle HTR. J’ai aussi surveillé la discussion entre les vacataires et les chefs du projet, afin de poser et de répondre aux questions généralisées. Grâce à ces tâches, je comprends mieux le défi d’harmoniser la création d’un corpus d’or par en groupe.

### 4.2 Les données d’entraînement TAL

Certains vacataires ont travaillé sur les textes extraits des transcriptions que les autres vacataires ont faites sur eScriptorium, et ils ont créé des données textuelles qui serviraient à l’entraînement des modèles TAL. Afin de les aider dans la création de ce deuxième corpus d’or, j’ai créé un workflow automatisé sur les dépôts GitHub du projet qui extrait automatiquement les lignes de texte des fichiers ALTO et divise le texte en segments selon les signes de ponctuation. La création d’un workflow sur GitHub m’a appris comment le faire plus tard pour l’application `alto2tei`.

### 4.3 La fin du pipeline : TEI Publisher

J’ai assisté à un atelier sur le logiciel TEI Publisher afin de mieux comprendre les objectifs downstream de l’application `alto2tei`.



# Chapitre 5

## L'analyse des structures des données XML

### 5.1 XML-ALTO

#### 5.1.1 Qu'est-ce qu'est le format ALTO ?

Décrire la création, le suivi, et l'objectif du format XML ALTO : enregistrer les infos sur la structure d'une image segmentée.

#### 5.1.2 La structure des fichiers XML-ALTO

Montrer la structure des données d'un fichier ALTO dans les deux formats qui sortent de Kraken : (1) ligne de texte encodé dans la balise `<TextLine>`, qui sort de Kraken via l'interface d'eScriptorium, et (2) ligne de texte encodé au niveau du glyph, qui sort directement de la ligne de commande de Kraken.

### 5.2 XMI-TEI

#### 5.2.1 Qu'est-ce qu'est la TEI ?

Décrire la création, le suivi, et l'objectif de la TEI.

#### 5.2.2 Les éléments de base de la TEI

Expliquer qu'il y a deux éléments essentiels de la racine, le `<teiHeader>` et le `<body>`. Ensuite expliquer l'utilité de l'élément facultatif `<sourceDoc>` et expliquer pourquoi il convient bien aux données de structure d'un fichier ALTO.



# Chapitre 6

## À la recherche des métadonnées

### 6.1 Uniquement l'essentiel

#### 6.1.1 Documents de divers types et de plusieurs époques

Expliquer le défi de modéliser un `<teiHeader>` qui est à la fois assez généralisé pour convenir aux divers documents et assez précisé pour servir aux utilisateurs et à la recherche.

#### 6.1.2 Exemples des métadonnées souhaitées

Donner un exemple des métadonnées d'un imprimé (cf. fig. 6.1) :

```
1 <sourceDesc>
2   <biblStruct>
3     <monogr>
4       <author xml:id="author">
5         <persName>
6           <surname>Balzac</surname> <!-- auteur -->
7           <forename>Honoré</forename>
8         </persName>
9       </author>
10      <title>The Wild Ass's Skin</title> <!-- titre -->
11      <editor role="translator">Ellen Marriage</editor>
12      <editor role="preface">George Saintsbury</editor>
13      <pubPlace key="FR">Paris</pubPlace>
14      <imprint>
15        <pubPlace>London</pubPlace> <!-- lieu de publication -->
16        <publisher>Dent</publisher> <!-- éditeur -->
17        <date when="1906">1906</date> <!-- date de publication -->
18      </imprint>
19    </monogr>
20  </biblStruct>
```

FIGURE 6.1 – Exemple des métadonnées d'un imprimé encodées en TEI (emprunté de [teibyexample.org](http://teibyexample.org) – à changer)

Et donner un exemple d'un incunable (cf. fig. 6.2) :

```

1 <sourceDesc>
2   <msDesc>
3     <msContents>
4       <biblStruct>
5         <monogr>
6           <author xml:id="author">
7             <persName>
8               <surname>Tory</surname> <!-- auteur -->
9               <forename>Geoffroy</forename>
10            </persName>
11          </author>
12          <title>Champ fleury</title> <!-- titre -->
13          <imprint>
14            <pubPlace>Paris</pubPlace> <!-- lieu de publication / lieu d'
apparition -->
15            <publisher>
16              <persName>
17                <surname>Gourmont</surname> <!-- éditeur -->
18                <forename>Gilles de</forename>
19              </persName>
20            </publisher>
21          </imprint>
22        </monogr>
23      </biblStruct>

```

FIGURE 6.2 – Exemple des métadonnées d'un incunable encodées en TEI (emprunté du cours TEI de J-B Camps 2015 – à changer)

Expliquer comment l'objectif du projet *Gallic(orpor)a* de traiter des documents d'une diachronie longue pose un défi à la récupération et encodage généralisée des métadonnées.

## 6.2 Où se trouvent les métadonnées des sources de Gallica

Présenter les deux sources de métadonnées ciblées par l'application `alto2tei`.

### 6.2.1 L'IIIF Image API

L'API du manifest IIIF contient des données rudimentaires sur le document. Elles sont envoyées dans un format JSON. Donner un exemple (cf. fig. 6.3) :

### 6.2.2 L'API SRU de la BnF

L'API du catalogue général de la BnF contient des données bien précises sur le document. Elles sont envoyées dans un format XML-Unimarc. Donner un exemple (cf. fig. 6.4).



```

1 {"Metadata":
2   {
3     "Label": "Title",
4     "Value": "The Wild Ass's Skin", # titre
5
6     "Label": "Creator",
7     "Value": "Honoré Balzac" # auteur
8   }
9 }

```

FIGURE 6.3 – Exemple des métadonnées envoyées par l'API IIIF

```

1 <mx:datatype tag="200" ind1="1" ind2=" ">
2   <mx:subfield code="a">The Wild Ass's Skin</mx:subfield> <!-- titre --
3   >
4   <mx:subfield code="b">Texte imprimé</mx:subfield>
5 </mx:datatype>
6 <mx:datatype tag="210" ind1=" " ind2=" ">
7   <mx:subfield code="a">Londres</mx:subfield> <!-- lieu de publication
8   -->
9   <mx:subfield code="c">Dent</mx:subfield> <!-- éditeur -->
10  <mx:subfield code="d">1906</mx:subfield> <!-- date de publication -->
11 </mx:datatype>
12 [...]
13 <mx:datatype tag="700" ind1=" " ind2="|">
14   <mx:subfield code="a">Balzac</mx:subfield> <!-- auteur -->
15   <mx:subfield code="b">Honoré</mx:subfield>
16 </mx:datatype>

```

FIGURE 6.4 – Exemple des métadonnées envoyées par l'API SRU de la BnF

## 6.3 Une solution

Présenter les métadonnées du document qu'on a déterminé d'être essentielle / assez généralisées parmi les divers types de document.



## Troisième partie

### Mise en opérationnelle du projet



# Chapitre 7

## La génération du <teiHeader>

### 7.1 La récupération des données

Parler de la récupération des données depuis les deux APIs discutés (cf. fig. 7.1) ainsi que le fichier YAML de configuration.

#### 7.1.1 Du manifest IIIF à un dictionnaire Python

Montrer le mapping des données du manifest au dictionnaire Python.

#### 7.1.2 De l'Unimarc à un dictionnaire Python

Montrer le mapping des données du catalogue au dictionnaire Python.

### 7.2 L'analyse des données

Parler de la stratégie d'atténuation des risques en sélectionnant les données fiables. Les métadonnées du catalogue général de la BnF sont utilisées uniquement si le même exemplaire physique du document numérisé sur Gallica a bien été trouvé. Sinon, on risque de mettre les données d'un autre exemplaire de l'oeuvre que celui qui a été transcrit et donc introduire des fausses données, tel que le cote ou même l'éditeur et la date de publication de l'exemplaire.

### 7.3 Le modèle du <teiHeader>

Montrer le mapping des données au <teiHeader>.

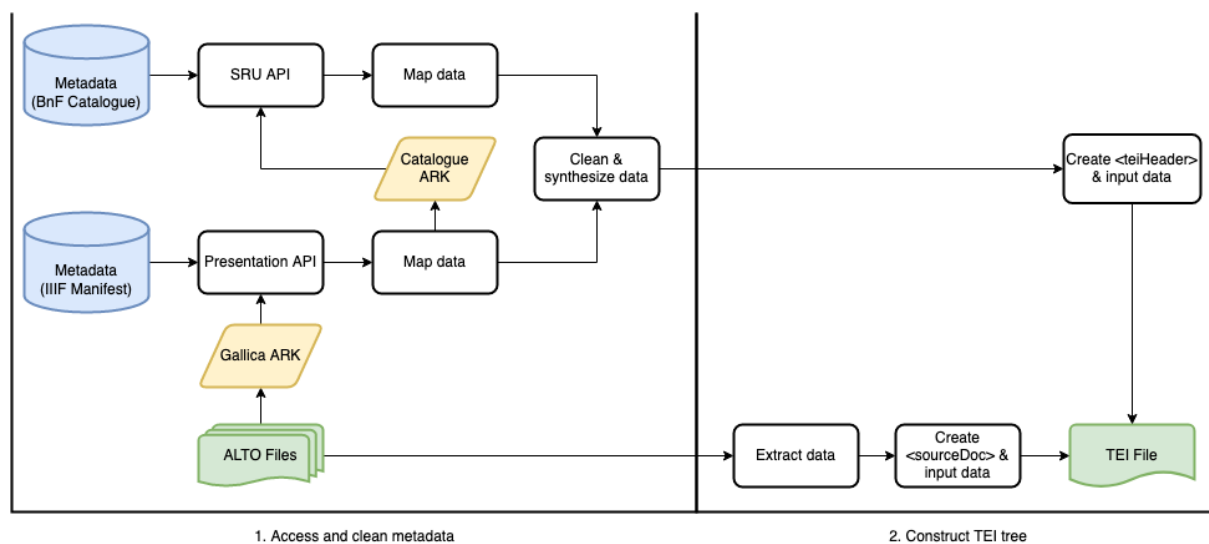


FIGURE 7.1 – Workflow

# Chapitre 8

## La modélisation de la <sourceDoc>

### 8.1 Le modèle du <sourceDoc>

Montrer le mapping des données prédites par les modèles HTR et segmentation vers les éléments TEI de la <sourceDoc>.

#### 8.1.1 Niveau de la ligne de texte

Des exemples / tables ...

#### 8.1.2 Niveau d'un mot ou d'une espace

Des exemples / tables ...

#### 8.1.3 Niveau du glyphe

Des exemples / tables ...

### 8.2 Documenter ce modèle dans l'ODD

Présenter le travail d'avoir écrit l'ODD *SegmOnto-Gallicorpora*.





# Chapitre 9

## Les données textuelles produites

### 9.1 La génération du `<body>` grâce au lexique *SegmOnto*

Expliquer comment j'ai généré le `<body>` en prenant les lignes de texte de la `<sourceDoc>` selon leurs étiquettes.

### 9.2 L'analyse linguistique

Parler d'un travail futur qui pourra prendre le fichier XML-TEI que j'ai créé et extraire les lignes de texte du `<body>` et les passer aux modèles TAL pour faire des analyses linguistiques. Montrer un exemple de ces données et comment cela marcherait avec un travail préliminaire que j'aurai fait d'ici la fin de stage et/ou le travail d'un ancien stagiaire que j'ai appliquée aux données Gallic(orpor)a.



# Conclusion



# Annexe A

## Données

Données du projet.









# Table des figures

1.1	figure . . . . .	3
6.1	Exemple des métadonnées d'un imprimé encodées en TEI . . . . .	15
6.2	Exemple des métadonnées d'un incunable encodées en TEI . . . . .	16
6.3	Exemple des métadonnées envoyées par l'API IIIF . . . . .	17
6.4	Exemple des métadonnées envoyées par l'API SRU de la BnF . . . . .	17
7.1	Workflow . . . . .	22



# Liste des tableaux



# Table des matières

Résumé	i
Remerciements	iii
Introduction	v
<b>I Présentation du projet</b>	<b>1</b>
<b>1 Qu'est-ce que l'HTR ?</b>	<b>3</b>
1.1 L'objectif de l'HTR . . . . .	3
1.2 L'histoire et l'évolution de la technologie . . . . .	3
1.3 Les deux approches actuelles . . . . .	3
1.3.1 L'OCR . . . . .	3
1.3.2 L'HTR . . . . .	3
<b>2 Au commencement, il y avait les <i>guidelines SegmOnto</i></b>	<b>5</b>
2.1 La problématique . . . . .	5
2.2 Les <i>guidelines</i> de <i>SegmOnto</i> . . . . .	5
2.3 L'avantage . . . . .	5
<b>3 Le rêve du projet <i>Gallic(orpor)a</i></b>	<b>7</b>
3.1 Le contexte du projet . . . . .	7
3.2 L'objectif du projet . . . . .	7
3.3 Le pipeline . . . . .	7
<b>II Exposition de la préparation et du travail d'analyse</b>	<b>9</b>
<b>4 Un pipeline visant à tout rassembler</b>	<b>11</b>
4.1 Les données d'entraînement HTR . . . . .	11
4.2 Les données d'entraînement TAL . . . . .	11
4.3 La fin du pipeline : TEI Publisher . . . . .	11

<b>5</b>	<b>L'analyse des structures des données XML</b>	<b>13</b>
5.1	XML-ALTO . . . . .	13
5.1.1	Qu'est-ce qu'est le format ALTO ? . . . . .	13
5.1.2	La structure des fichiers XML-ALTO . . . . .	13
5.2	XMI-TEI . . . . .	13
5.2.1	Qu'est-ce qu'est la TEI ? . . . . .	13
5.2.2	Les éléments de base de la TEI . . . . .	13
<b>6</b>	<b>À la recherche des métadonnées</b>	<b>15</b>
6.1	Uniquement l'essentiel . . . . .	15
6.1.1	Documents de divers types et de plusieurs époques . . . . .	15
6.1.2	Exemples des métadonnées souhaitées . . . . .	15
6.2	Où se trouvent les métadonnées des sources de Gallica . . . . .	16
6.2.1	L'IIIF Image API . . . . .	16
6.2.2	L'API SRU de la BnF . . . . .	16
6.3	Une solution . . . . .	17
<b>III</b>	<b>Mise en opérationnelle du projet</b>	<b>19</b>
<b>7</b>	<b>La génération du &lt;teiHeader&gt;</b>	<b>21</b>
7.1	La récupération des données . . . . .	21
7.1.1	Du manifest IIIF à un dictionnaire Python . . . . .	21
7.1.2	De l'Unimarc à un dictionnaire Python . . . . .	21
7.2	L'analyse des données . . . . .	21
7.3	Le modèle du <teiHeader> . . . . .	21
<b>8</b>	<b>La modélisation de la &lt;sourceDoc&gt;</b>	<b>23</b>
8.1	Le modèle du <sourceDoc> . . . . .	23
8.1.1	Niveau de la ligne de texte . . . . .	23
8.1.2	Niveau d'un mot ou d'une espace . . . . .	23
8.1.3	Niveau du glyphe . . . . .	23
8.2	Documenter ce modèle dans l'ODD . . . . .	23
<b>9</b>	<b>Les données textuelles produites</b>	<b>25</b>
9.1	La génération du <body> grâce au lexique <i>SegmOnto</i> . . . . .	25
9.2	L'analyse linguistique . . . . .	25
	<b>Conclusion</b>	<b>27</b>
<b>A</b>	<b>Données</b>	<b>29</b>