



HAL
open science

**Enrichissement d'une base de données et encodages en
XML-TEI. L'exemple du projet Antonomaz :
imprimeurs et repères**

Zoé Cappe

► **To cite this version:**

Zoé Cappe. Enrichissement d'une base de données et encodages en XML-TEI. L'exemple du projet Antonomaz : imprimeurs et repères. Sciences de l'Homme et Société. 2022. dumas-04570509

HAL Id: dumas-04570509

<https://dumas.ccsd.cnrs.fr/dumas-04570509>

Submitted on 7 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE NATIONALE DES CHARTES
UNIVERSITÉ PARIS, SCIENCES & LETTRES

Zoé Cappe

licenciée en langues scandinaves

Enrichissement d'une base de données et encodages en XML-TEI

**L'exemple du projet Antonomaz :
imprimeurs et repères**

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

2022

Résumé

Le présent mémoire a été rédigé suite à un stage effectué au sein du projet Antonomaz (« Analyse auTOMatique et NumérisatiON des MAZarinades »), dans le cadre de master « Technologies numériques appliquées à l'histoire ». Ce projet de recherche a pour objectif de rassembler et mettre en ligne le corpus des mazarinades - libelles imprimés en masse pendant la Fronde (1648-1653) - ainsi que d'en permettre l'exploration et l'exploitation. Ce mémoire consiste en la description des réalisations effectuées au cours du stage, et leur contextualisation.

En vue de la mise en valeur des données du projet, des textes brefs, nommés « repères », qui donnent des informations sur le contexte des productions des mazarinades, ont été encodés en XML-TEI. Un ODD a été rédigé afin d'en définir les règles d'encodage et d'en assurer la validité.

Dans le même objectif, une base de données des imprimeurs de la Fronde pré-existant au stage a été enrichie avec les informations de leurs notices d'autorité d'IdRef, à l'aide d'une requête SPARQL et d'un script Python. Elle a ensuite été encodée automatiquement en XML-TEI selon les règles de l'ODD du projet Imprinters, en utilisant un script Python et la librairie *lxml*.

Mots-clés : Text Encoding Initiative (TEI) ; Python ; lxml ; Fronde ; mazarinades ; SPARQL ; imprimeurs.

Informations bibliographiques : Zoé Cappe, *Enrichissement d'une base de données et encodages en XML-TEI. L'exemple du projet Antonomaz : imprimeurs et repères*, mémoire de master « Technologies numériques appliquées à l'histoire », dir. Thibault Clé-ric, École nationale des chartes, 2022.

Remerciements

MES remerciements vont tout d'abord à Karine Abiven et Gaël Lejeune, et tout particulièrement Alexandre Bartz, pour leur accueil et leur suivi tout au long du stage. Je voudrais aussi remercier toute la promotion 2022, ainsi que Thibault Clérice et toute l'équipe pédagogique, qui ont rendu ces deux années passionnantes et mémorables. Enfin, je remercie surtout ma famille, sans qui je ne serais pas arrivée jusque là.

Introduction

La deuxième année du master « Technologies numériques appliquées à l’histoire » prend fin avec un stage de plusieurs mois, afin de mettre en pratique les connaissances acquises au cours de la formation. J’ai donc effectué un stage qui s’est déroulé dans le cadre du projet Antonomaz, créé en 2019, encadrée par Karine Abiven, enseignante-chercheuse en langue et littérature françaises, et Gaël Lejeune, enseignant-chercheur en informatique, responsables du projet, et Alexandre Bartz, ingénieur, à la Maison de la recherche de Sorbonne Université (Serpente).

C’est par intérêt pour le sujet que j’ai participé au projet Antonomaz (« Analyse automatique et NumérisatiOn des MAZarinades ») : en effet, celui-ci porte sur les mazarinades, textes imprimés pendant la période de la Fronde au XVII^e. L’objectif du projet est de créer une collection numérique rassemblant toutes les mazarinades dont on connaît l’existence. Ces textes, très nombreux, sont éparpillés dans un grand nombre de lieux de conservation, tels que la Bibliothèque Mazarine, la Bibliothèque nationale de France, la bibliothèque royale de Copenhague, ou même la bibliothèque de l’Université de Tokyo. Les regrouper en un seul endroit permettrait de faciliter la recherche des documents et l’exploitation du corpus, et le faire connaître à un public plus large.

« La nécessité de relier ces objets éphémères entre eux, avec d’autres imprimés et d’autres savoirs, apparaît consubstantielle à leur appréhension. Ces brochures ont été l’objet, par le passé, de nombreuses formes de recueils et de reliures : on souhaite proposer, avec les moyens du numérique, de les *relier* autrement et virtuellement. »¹

Pour rassembler ces documents, un corpus a été constitué par les membres du projet. Il est composé à la fois des numérisations d’un exemplaire par mazarinade, des informations qui leurs sont liées et permettent de les contextualiser, et de leur texte brut (sans mise en forme). Une fois ce corpus constitué, l’étape suivante consiste en sa mise à disposition sur un site Web propre au projet. Il s’agit de permettre aux utilisateurs d’explorer le corpus et de l’interroger, en effectuant des recherches dans le texte des mazarinades ou des recherches plus spécifiques (par auteur, par date, etc.).

1. Abiven, Karine, Bartz, Alexandre, Lejeune, Gaël et Tanguy, Jean-Baptiste, “Vers une collection numérique des libelles de la Fronde, ou comment relier les mazarinades”, *Le Verger - bouquet XXIII* (, mai 2022)

Le stage que j'ai effectué s'insère dans cette étape du projet. Le corpus était déjà constitué, et l'application du site en cours de développement. Il était alors intéressant d'envisager plusieurs façons de valoriser les données collectées au cours du projet. Deux axes se sont alors dessinés : le premier consistait en la mise en valeur des numérisations des documents sur le site Web du projet, en travaillant avec la IIIF (International Image Interoperability Framework).² Cet aspect du projet est celui sur lequel a travaillé une autre étudiante du master, Margaux Faure, qui y a elle aussi effectué son stage³. Le deuxième axe consistait tout d'abord en l'enrichissement des ressources du site Web avec la création de courtes notices, ou « repères », qui permettent de préciser le contexte de production des mazarinades ; et ensuite en l'enrichissement et l'encodage d'une base de données sur les imprimeurs de la Fronde, afin de la mettre à disposition des utilisateurs sur le site et créer des liens entre les imprimeurs et les mazarinades. J'ai donc travaillé sur ce deuxième axe au cours de mon stage, et ce sont ces deux réalisations qui seront présentées dans ce mémoire.

2. L'*International Image Interoperability Framework* est une infrastructure qui permet la description, la manipulation, l'annotation et la diffusion d'images de haute résolution sur le Web de façon interopérable. <https://iiif.io/>

3. Faure, Margaux, *Rassembler et valoriser les mazarinades. Traiter les numérisations et les métadonnées d'une collection d'imprimés du XVIIe siècle pour sa mise à disposition numérique*, mémoire de master « Technologies numériques appliquées à l'histoire », dir. Thibault Clérice, École nationale des chartes, 2022

Première partie
Contexte et projet

Chapitre 1

La période historique du projet

1.1 La Fronde

Le projet Antonomaz a pour cadre historique la Fronde, qui a eu lieu de 1648 à 1653. C'est une période de troubles intenses qui se déclenchent pour diverses raisons. Toutes les catégories de population (peuple et bourgeois de Paris, aristocratie et princes de sang) ont quelque chose à reprocher à Mazarin, et il devient alors l'incarnation de tous les problèmes. Depuis un certain temps déjà, le pays connaît la guerre. Les prélèvements fiscaux sont lourds et la paix devient alors souhaitable. Le gouvernement du cardinal Mazarin tente alors de créer de nouvelles taxes pour les Parisiens qui y échappaient jusqu'alors en grande partie pour financer la guerre - c'est là la cause du mécontentement des bourgeois parisiens contre Mazarin. Les princes de sang, eux, lui reprochent son rôle important dans le gouvernement - il est le principal ministre et gouverne avec Anne d'Autriche -, qui les empêche d'exercer leur rôle de conseillers du prince. Les membres du Parlement de Paris et des autres corps constitués voient aussi d'un mauvais œil ce cardinal qui détient autant de pouvoir à lui seul, et qui, de plus, se trouve aussi être un étranger. La seule solution qui semble s'imposer à tous, c'est de l'écarter du pouvoir.

Le pays entre en révolte pendant cinq ans, révolte qui prend de nombreuses formes, allant même jusqu'à ce que certains historiens décrivent comme une guerre civile¹. On la décompose généralement en quatre périodes : la Fronde parlementaire (1648-1649), la Fronde des princes ou Fronde princière (1650), l'union des frondes (1651) et la Fronde condéenne (1651-1653). Ces périodes servent cependant surtout de points de repères chronologiques, car leurs contours sont en réalité bien moins nets. La Fronde est un enchaînement d'intrigues et d'évènements complexes dont le fil est parfois difficile à démêler. Mazarin, qui n'est pas sans avoir commis un certain nombre d'erreurs en cette période, est la cible de tous les reproches et sa présence au gouvernement contribue à alimenter la Fronde : chacun défend ses propres intérêts et tous pointent du doigt le cardinal. Il sera

1. Pernot, Michel, *La Fronde : 1648-1653*, Éditions Tallandier, Paris, 2019

contraint de quitter la cour à plusieurs reprises et même à l'exil. Après ces cinq années, aucun des partis n'a su faire d'alliance de façon durable, chacun pensant à son propre intérêt, et le pays est las de ce temps de révolte qui n'a plus de sens. La situation s'inverse, et pour mettre un terme au conflit et retrouver une stabilité, une seule solution est trouvée : le retour de Mazarin.

L'une des particularités de la Fronde, qui nous intéresse plus particulièrement dans le cadre de ce projet, est le rôle qu'ont joué les imprimeurs et les mazarinades.

1.2 Les mazarinades

Les mazarinades sont des textes qui sont imprimés et diffusés en masse pendant cette période, à Paris principalement, mais aussi dans d'autres villes qui ont participé à la Fronde, comme Bordeaux ou Rouen. Hubert Carrier parle même d'un « déluge de libelles »². On en conserve aujourd'hui entre 5 500 et 6 000 éditions, et il a pu en exister plus encore qui ont été perdues. Elles ont été imprimées de façon plus ou moins importante selon le cours de événements et l'intensité de l'actualité politique : les périodes d'impression les plus denses étant en 1649 et en 1652.

Que sont les mazarinades ? Le terme de « mazarinade » s'installe avec *La Mazarinade*, pamphlet attribué à Paul Scarron et publié en 1651, et est repris par les historiens et bibliographes pour désigner l'ensemble des textes imprimés et diffusés pendant la Fronde, aussi appelés libelles³ ou pamphlets.

Les mazarinades ne peuvent être définies par un genre littéraire commun : en effet, on y trouve toutes sortes de textes, comme des chansons, pamphlets, publications des parlements (arrêts)... Les sujets sont tout aussi variés, bien que la personne de Mazarin en soit l'un des principaux. Ce n'est pas leur genre littéraire qui permet d'en faire un ensemble, mais plutôt la période en elle-même qui sert à en définir le cadre.

Elles pouvaient prendre des formes diverses. Elles consistent pour la plupart en livrets de petits formats, généralement in-4, mais aussi souvent en in-8 ou in-16. Elles circulaient aussi sous forme de tracts, de placards affichés dans la rue ; elles pouvaient être lues à haute voix ou chantées dans le cas des chansons, ce qui implique une importante circulation orale dont l'ampleur n'est pas aisée à mesurer, ou encore sous forme manuscrite, pour une diffusion à destination de cercles moins larges, pour plus de discrétion (elles pouvaient cependant être imprimées par la suite).

2. Carrier, Hubert, *La presse de la Fronde (1648-1653) : les mazarinades. La conquête de l'opinion*. Droz, vol. I, Genève, 1989

3. « Le terme de "libelle", malgré sa neutralité étymologique (*libellus*, "petit livre"), est dès le XVI^e siècle employé pour désigner une importante réalité de la communication imprimée, ces produits auxquels on reconnaît à peine le statut de "livre", et qui ont pour objet l'invective, la polémique ou la satire, en relation avec une actualité particulière. », Sordet, Yann, *Histoire du livre et de l'édition : production et circulation, formes et mutations*, Albin Michel, Paris, 2021, pp.341-342

Elles pouvaient être trouvées chez les imprimeurs eux-mêmes, et étaient souvent vendues par des libraires dans des lieux très fréquentés tels que le Palais, où se trouvaient le siège du Parlement et les galeries marchandes, ou sur le Pont-Neuf et les quais. Les autorités ont tenté d'en limiter la vente, en particulier sur le Pont-Neuf, mais sans grand succès. De plus, les mazarinades se vendaient aussi d'une façon plus difficile à contrôler par l'intermédiaire de colporteurs qui circulaient dans les rues.

Leur bas prix (elles coûtaient moins cher qu'une livre de pain blanc⁴) est aussi l'une des raisons de l'importance des ventes et de leur diffusion très large dans la population. Leur coût seul n'explique pas le nombre de publications pendant la Fronde : c'est que les textes publiés étaient profondément liés à l'actualité, et servaient à la fois à faire connaître le cours des événements, mais aussi l'avis et la prise de position de chacun vis-à-vis desdits événements. Ainsi, en publiant et en lisant des textes, on devenait acteurs du cours des choses et non plus simples spectateurs.

Les auteurs et les grands acteurs de la Fronde avaient alors bien compris l'impact que pouvaient avoir les textes sur le cours des événements. Des personnes comme Jean-François Paul de Gondi, cardinal de Retz en 1652, employaient des auteurs pour écrire des textes qui soutenaient leur parti et lui-même a plus d'une fois pris la plume. Bien que certains auteurs aient choisi de révéler leur identité dans leurs textes, beaucoup d'auteurs des mazarinades ont fait le choix de rester anonymes, voire d'utiliser un pseudonyme, afin de pouvoir publier leurs textes tout en se préservant des éventuels risques que pouvait entraîner leur diffusion. Cet anonymat rend difficile leur identification, et l'absence fréquente d'informations sur les imprimeurs la complique.

1.3 Les imprimeurs

De nombreuses mazarinades n'ont, en effet, pas d'imprimeur identifié : beaucoup, en raison du caractère séditieux des textes qu'ils impriment, préfèrent ne pas y indiquer leur nom, voire ne donner aucune indication d'adresse, ou choisissent d'utiliser des pseudonymes ou des fausses adresses. On estime qu'un peu plus d'un cinquième des mazarinades ont été publiées sans lieu d'impression⁵.

Ces imprimeurs, beaucoup plus difficiles à identifier, n'ont pas de protecteur et prennent un risque en diffusant ces textes. Ils peuvent avoir leur propre officine, mais il peuvent aussi être des apprentis n'ayant pas encore été reçus maîtres (ils n'ont donc pas le droit d'imprimer eux-mêmes des textes), des textes auraient ainsi été imprimés à l'insu des maîtres dans leur propre imprimerie.

4. « En général, six derniers les quatre pages, un sou les huit, quand la livre de pain blanc coûte un sou et demi (huit deniers) » Jouhaud, Christian, *Mazarinades : la Fronde des mots*, Aubier, Paris, 2009, p.29

5. Carrier, Hubert, *La presse de la Fronde (1648-1653) : les mazarinades. La conquête de l'opinion....*, p.77

En effet, au XVII^e siècle, les imprimeurs sont organisés en communauté de métier, avec des maîtres, apprentis et compagnons. Les imprimeries sont, dans la plupart des cas, des entreprises familiales, et les associations entre elles sont fréquentes. Les imprimeurs sont regroupés dans une même zone de la ville, à Paris dans le quartier de l'université et de la montagne Sainte-Geneviève, et au Palais. On en dénombre environ 250 qui auraient participé à l'impression des mazarinades à Paris, et une soixantaine en province (comme à Bordeaux ou à Rouen).

D'autres imprimeurs ont pris le risque d'indiquer leur nom et adresse sur les textes, ce qui a pu avoir d'importantes conséquences pour eux, comme dans le cas de Claude Morlot, condamné à mort pour avoir imprimé la *Custode de la reine* en 1649. Il parvient néanmoins à échapper à son sort en étant libéré par un groupe d'émeutiers.

Certains imprimeurs ont pu être protégés et obtenir un statut officiel auprès de personnalités importantes, ce qui leur a permis d'imprimer les mazarinades sans avoir à s'en cacher. On connaît, par exemple, la veuve Guillemot qui est devenue l'imprimeuse officielle du duc d'Orléans à partir de décembre 1651, ou Nicolas Vivenay à partir de l'été 1651 et Guillaume de la Court à Bordeaux pour le prince de Condé.

Des imprimeurs ont aussi travaillé pour le gouvernement et pour le parti de Mazarin – bien que le nombre de textes prenant leur défense existe en un nombre bien plus limité –, comme Théophraste Renaudot et ses fils. Ils impriment la *Gazette*, créée en 1631, un journal officiel qui a pris le parti, d'abord de Richelieu, puis de Mazarin. Son activité continue pendant la Fronde, et Théophraste Renaudot suit la cour lors de son départ à Saint-Germain-en-Laye, où il imprime des libelles pro-Mazarin. Ses fils, cependant, continuent à imprimer la *Gazette* à Paris en s'éloignant du parti de Mazarin. Après le retour de la cour à Paris, les affaires se compliquent, puis l'exil de Mazarin cause la perte de sa protection.

La Fronde connaît un équivalent de la *Gazette*, *Le Courrier français*, imprimé par Rolin de la Haye pendant l'éloignement de Théophraste Renaudot, et qui connut un succès certain jusqu'à la paix de Rueil, le 11 mars 1649. D'autres formes de presse furent ensuite créées, comme le *Courrier bordelais*, imprimé en neuf numéros pendant les quelques mois qui suivirent la paix.

Du côté du gouvernement se trouvent aussi les imprimeurs, dits "ordinaires", du roi tels que Julien Courant, qui sont chargés d'imprimer les textes liés à l'administration du royaume, comme des lois ou des édits royaux. En contrepartie, ils obtenaient des avantages tels que des revenus et des privilèges. Ils sont plusieurs à occuper cette charge, à Paris et en province, ce qui implique aussi une forme de concurrence.

Ils se distinguent en cela de l'imprimerie royale créée en 1640 par le cardinal de Richelieu, et dirigée par Sébastien Cramoisy et ses frères. Elle est chargée de l'impression, outre des éditions prestigieuses de textes historiographiques, juridiques ou scientifiques, de textes officiels ou commandés par le gouvernement, tels que le *Mascurat* de Gabriel Naudé (1649)

ou les libelles du Père François Faure.

Malgré les risques encourus, imprimer des mazarinades était une activité très rentable, ce qui explique en grande partie l'explosion du nombre des impressions. Les imprimeurs sont alors souvent en situation précaire : il y a peu d'imprimeurs par presse, et ceux-ci impriment souvent des textes qui rapportent peu et le coût des matériaux est élevé. La Fronde est une période prospère pour les imprimeurs, malgré des répressions, car c'est alors une source intarissable de textes de toutes sortes à faire paraître, avec du papier de mauvaise qualité et peu d'illustrations (donc moins coûteux) et ils se vendent particulièrement bien.

Un texte intitulé *Remerciement des imprimeurs au cardinal Mazarin* a même été publié en 1649 :

« Nous n'auions ny argent ny viures, toutesfois graces à Dieu, & à vofre Eminence Monsieur Saint Iules noftre fecond patron, nous fommes aujourd'huy des mieux accomodez, & nous craignons plus de manquer d'ancre & de papier, que de pain & de vin, ny de viande. »

Chapitre 2

Le projet Antonomaz

2.1 Réalisations techniques

Rassembler les mazarinades et leurs métadonnées en une seule collection est un travail qui a été réalisé en plusieurs étapes¹.

L'ensemble a été construit à partir de la liste des titres des mazarinades établie par Célestin Moreau dans sa *Bibliographie des mazarinades*², qui contient 4 082 entrées (ainsi que 299 titres complémentaires). Cette liste a été océrisée³ et peut être consultée et interrogée *via* le site Web du projet (encore en cours de construction)⁴. La même méthode a été appliquée au *Supplément à la Bibliographie des Mazarinades* d'Émile Socard, publié en 1876, et au *Nouveau supplément à la Bibliographie des mazarinades* d'Ernest Labadie, publié en 1904, afin de compléter la liste des titres.

Une fois cette liste obtenue, des requêtes ont été effectuées sur catalogue de la Bibliothèque nationale de France et Gallica, ainsi que Google Books, afin de collecter un fac-similé par titre. Cela implique parfois des lacunes et des doublons, mais a permis de récupérer 2 569 titres sur les 5 063. Le nombre a pu être accru grâce à la collaboration entre le projet et la Bibliothèque Mazarine, le projet ayant permis de financer des numérisations de mazarinades qui enrichissent aussi le Mazarinum, la bibliothèque numérique de l'institution, et récupérer des informations de la *Bibliographie des mazarinades*, projet en cours de développement à la Bibliothèque Mazarine, afin d'enrichir les métadonnées des textes.

1. Cette partie se base principalement sur les deux articles suivants, qui explique de façon détaillée les différentes étapes : Abiven, Karine, Bartz, Alexandre, Lejeune, Gaël, *et al.*, “Vers une collection numérique des libelles de la Fronde, ou comment relier les mazarinades”... et Abiven, Karine, Lejeune, Gaël et Tanguy, Jean-Baptiste, “Exploiter un corpus de données textuelles sans post-traitement : l'écriture burlesque de la Fronde”, *Humanités numériques*-4 (1^{er} déc. 2021), DOI : [10.4000/revuehn.2355](https://doi.org/10.4000/revuehn.2355)

2. Moreau, Célestin, *Bibliographie des Mazarinades*, Renouard, 3 t., Paris, 1850

3. L'océrisation consiste en la récupération du texte d'une image par un processus de reconnaissance automatique des caractères (Optical Character Recognition (OCR), abréviation dont dérive le verbe « océriser »).

4. Pour consulter la liste, suivre le lien suivant : https://antonomaz.humanum.fr/tools/Biblio_Moreau.html

Les numérisations ont ensuite pu être océrisées à leur tour, et les textes bruts obtenus ont été encodées de façon semi-automatique en XML-TEI. Cette océrisation a pu être réalisée avec Kraken et un modèle d’OCR entraîné pour le français du XVII^e siècle⁵. L’encodage permet aussi de les contextualiser en fournissant des informations telles que leur auteur, imprimeur ou encore leur date de publication. Ces informations proviennent en partie de la liste de la liste des titres, mais aussi des relectures effectuées par les membres du projet afin d’en compléter les métadonnées. Le corpus constitué est consultable et téléchargeable en accès libre sur le dépôt GitHub du projet⁶.

2.2 Projets similaires

Il existe des projets similaires au projet Antonomaz, dont le sujet est le corpus des mazarinades. Deux de ces projets en particulier ont retenu notre attention : la *Bibliographie des Mazarinades* et le « Projet Mazarinades ».

2.2.1 La Bibliographie des mazarinades

La *Bibliographie des mazarinades* est un projet porté par la Bibliothèque Mazarine depuis 2019. Son objectif est le recensement des différentes éditions et état des mazarinades, en se basant sur le travail de Célestin Moreau, Armand d’Artois⁷, Émile Socard, Ernest Labadie et Hubert Carrier. Le travail de comparaison et d’analyse des textes est réalisé sur les 25 000 exemplaires de la Bibliothèque Mazarine. Les entités identifiées, structurées au format Unimarc, peuvent être liées les unes aux autres et un identifiant unique « BM » leur est attribué. Il est possible d’effectuer des recherches dans la base de données en ligne⁸, que ce soit une recherche plein texte ou par recherche avancée (auteur, imprimeur-libraire, date, etc.).

Les données collectées sur certaines entités peuvent être reprises par le projet Antonomaz pour enrichir et compléter les métadonnées des mazarinades qui leur correspondent, en y incluant le numéro « BM » et le lien vers la notice de la *Bibliographie*.

2.2.2 Le Projet Mazarinades

Le Projet Mazarinades est développé par l’équipe « Recherches internationales sur les mazarinades » (RIM) depuis 2008. Il est très similaire au projet Antonomaz, puisqu’il

5. Gabay, Simon, Clérice, Thibault et Reul, Christian, *OCR17*, URL : <https://github.com/editions/OCR17/> (visité le 23/08/2022)

6. On peut le trouver ici : <https://github.com/Antonomaz/Corpus>

7. Vellet, Christophe, “Les mazarinades à l’affiche ? Armand d’Artois et la collection de la Bibliothèque Mazarine”, *Mazarinades, nouvelles approches*–12 (2016), p. 51-67, URL : <https://revues.droz.org/index.php/HCL/article/view/2442> (visité le 15/08/2022)

8. Voici le lien pour y accéder : <https://mazarinades.bibliotheque-mazarine.fr/>

a pour objectif de rassembler les numérisations et les textes de mazarinades dans un corpus numérique, qui est mis en ligne et consultable par les chercheurs comme par le grand public.

Ce projet s'est construit en deux étapes. En 2008, les 2 700 mazarinades de la bibliothèque de l'Université de Tokyo ont été numérisées, puis les textes ont tous fait l'objet d'une transcription diplomatique. Mis en ligne en 2011, le corpus est donc constitué des numérisations et des textes transcrits, et il continue à être mis à jour régulièrement.

Le site Web du projet⁹ continue à être mis à jour régulièrement, et en plus de partager l'actualité scientifique concernant les mazarinades, il permet d'effectuer des recherches plein texte et avancées (auteur, lieu de publication, etc.) dans le corpus.¹⁰

Il diffère cependant du projet Antonomaz en plusieurs façons. Tout d'abord, les transcriptions et l'édition des documents sont faites manuellement, par les membres du projet ou par des utilisateurs qui peuvent se créer un compte sur le site Web, tandis que les transcriptions du projet Antonomaz sont réalisées automatiquement avec l'OCR.

De plus, les textes bruts des transcriptions du projet Antonomaz sont en libre accès : si quelqu'un souhaite les télécharger et les réutiliser, il est libre de le faire. En revanche, les textes des transcriptions du Projet Mazarinades son accessibles sur demande uniquement. Seul un extrait peut être rendu visible dans le cas d'une recherche plein texte.

Enfin, les collections sur lesquelles se basent le Projet Mazarinades sont celles de la bibliothèque de l'Université de Tokyo, et celles du projet Antonomaz sont principalement celles de la Bibliothèque Mazarine, la Bibliothèque nationale de France, la Bibliothèque municipale de Lyon, ainsi que d'autres bibliothèques dont les numérisations ont été récupérées lors du requêtage de Google Books.

9. Le lien pour y accéder : <http://mazarinades.org/>

10. Ichimaru, Tadako, "Enjeux de la numérisation des mazarinades", *Histoire et civilisation du livre*–12 (2016), 12, p. 77-89

Deuxième partie

Les « repères »

Chapitre 3

Présentation des « repères »

3.1 Typologie des documents

Le projet Antonomaz a pour but de voir ses ressources utilisées par un public très large. Afin d’y parvenir, le site doit être accessible et les données qu’il valorise doivent pouvoir être comprises par tous. C’est pourquoi les membres du projet ont choisi de créer des « repères », c’est-à-dire des textes brefs qui peuvent servir à éclaircir les différents aspects de la période. Ceux-ci se trouveraient en ressources complémentaires sur le site Web, et seraient bien distincts du corpus des mazarinades.

Il s’agit donc de couvrir plusieurs thématiques concernant la période historique du projet, comme les évènements marquants de la Fronde et les personnages importants pour la partie historiques, ou encore les différents genres littéraires des mazarinades et leur caractéristiques matérielles.

Ces « repères » se présenteraient sous la forme de textes brefs, avec une limite de cinq cents mots environ : il s’agit d’une simple introduction à ces sujets. C’est pourquoi il nous a paru intéressant d’y joindre aussi une courte bibliographie, qui présenterait plusieurs sources qui ayant servi à la rédaction et qui peuvent aussi être utiles à un utilisateur qui souhaiterait approfondir ses connaissances sur le sujet.

Afin d’illustrer et clarifier le propos des « repères », la possibilité d’ajouter des images pour accompagner le texte était aussi envisagée.

Un autre point important était la question des liens dans les « repères ». En effet, il nous paraissait intéressant de créer deux types de liens : tout d’abord, rendre possible le renvoi vers des mazarinades en particulier, qui ayant servi d’exemple dans le texte ou qui sont en lien direct avec le sujet. Ce lien, une fois mis en place sur le site Web, renverrait donc vers la notice descriptive des mazarinades du projet, contenant leurs métadonnées, le texte et la numérisation. Ensuite, des renvois vers d’autres notices qui ont un sujet similaire pourraient aussi intéresser lecteur.

3.2 Un encodage en XML-TEI

Le site Web du projet est créé avec TEI Publisher, un outil qui permet de publier les ressources encodées en XML-TEI¹. Ce moyen permet de faciliter le traitement des mazarinades, elles-mêmes en XML-TEI, et c'est pourquoi il a été choisi de procéder de la même façon pour les « repères ». Les deux types de documents peuvent ainsi être traités de façon similaire, sans ajouter d'étape pour adapter les « repères » à TEI Publisher. La TEI est principalement utilisée pour des textes tels que des sources primaires (manuscrits, documents d'archives, etc.), mais s'adapte très bien à ce type de documents. L'objectif principal de leur encodage est d'apporter une structure aux différents éléments qui les composent, ce qui correspond à l'usage de la TEI.

Qu'est-ce que la TEI ?

« "La Text Encoding Initiative" (TEI) est un consortium qui développe et maintient collectivement un standard »² pour l'encodage des textes. Depuis sa création en 1987, ses recommandations (« *Guidelines* ») continuent à évoluer et en sont à leur cinquième version (Proposition 5), publiée en 2007³. L'encodage en XML-TEI repose sur un principe de balisage du texte, qui permet d'en identifier précisément les différents composants et métadonnées, selon les recommandations établies par le consortium TEI. Le balisage, reposant sur le principe du XML (« eXtensible Markup Language », soit « Langage à balises étendu ») se fait à l'aide d'éléments (représentés ainsi <element>) qui encadrent ces composants et métadonnées, et d'attributs (qui s'insèrent ainsi dans les éléments : <element attribut="valeur d'attribut">, et que l'on cite ainsi : @attribut). Ces éléments et attributs sont spécifiés par la TEI P5, et répondent à des règles précises (ordre des éléments, occurrences...) qui sont définis dans un schéma général (« TEI All »)⁴.

Fonctionnement des schémas

Ce schéma est prévu pour couvrir autant que possible toute les différentes façons d'encoder un texte : il est donc beaucoup trop large, et il n'est pas recommandé de l'utiliser pour réaliser des éditions. Il est donc possible de créer une sorte de « sous-schéma », qui obéit aux mêmes règles de validation que la TEI All mais avec des éléments, attributs et règles qui peuvent être précisées selon les besoins d'une édition⁵.

1. Pour une présentation plus détaillée de TEI Publisher, voir le chapitre 5

2. Il est présenté sur la page d'accueil du site Web dédié : <https://tei-c.org/>

3. Consortium TEI, *TEI P5 : Guidelines for Electronic Text Encoding and Interchange*, URL : <https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html> (visité le 17/08/2022)

4. Id., *Elements Available in ALL TEI Documents*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/CO.html> (visité le 22/08/2022)

5. *Why Is Schematron Different ?*, URL : https://schematron.com/home/whyisschematrongifferent_.html (visité le 19/08/2022)

Ces schémas sont appelés ODD, « One Document Does it All ». Les *Guidelines* en documentent l'utilisation et la structure⁶. Ils fonctionnent sur le principe d'un appel des éléments qui font partie de « modules » afin de les insérer ou les exclure de l'encodage voulu. Les éléments ajoutés peuvent ensuite être modifiés : on peut choisir quels éléments ils peuvent contenir (ou non), de même pour les attributs. On peut aussi en modifier l'ordre et le nombre d'occurrences (par exemple : tel élément peut apparaître deux fois, et seulement à la suite de tel autre élément...). Il en va de même pour les attributs, auquel on peut aussi, par exemple, imposer une liste de valeurs.

Il est aussi possible de pousser plus loin les règles de l'encodage en créant des « contraintes », en utilisant Schematron, un standard qui permet d'établir ces contraintes à l'aide d'expressions XPath, et qui s'exprime en XML. Les requêtes XPath se font sous la forme de chemins dans le XML (de la même façon que des chemins de fichiers) qui permettent de localiser certaines informations. Avec Schematron, on vérifie si ces informations sont vraies ou fausses dans les fichiers, et selon la contrainte imposée, soit le fichier sera validé, soit une erreur apparaîtra (un message d'erreur peut être ajouté à la contrainte pour expliquer la cause de l'erreur à l'utilisateur).

Exemple 1 : On modifie ici l'élément <titleStmt> : au sein du <titleStmt>, il est maintenant obligatoire d'avoir deux éléments <title>, suivis d'un élément <respStmt>.

```
<elementSpec ident="titleStmt" mode="change">
  <content>
    <elementRef key="title" minOccurs="2" maxOccurs="2"/>
    <elementRef key="respStmt"/>
  </content>
</elementSpec>
```

Exemple 2 : On modifie ici l'élément <title> de trois manières.

Dans un <constraintSpec>, on fait une assertion Schematron qui permet de vérifier s'il y a du texte dans l'élément <title>. S'il n'y en a pas, il y aura une erreur lors de la validation du fichier qui indiquera « Le titre doit être renseigné ».

Ensuite, on modifie les attributs. On impose d'abord une liste de valeurs (« main » ou « sub ») à l'attribut @type, et enfin on procède à la suppression d'attributs inutiles avec le @mode="delete" (la liste est ici abrégée).

```
<elementSpec ident="title" mode="change">
  <!-- contrainte Schematron sur le contenu du titre -->
  <constraintSpec ident="title_text" scheme="schematron">
    <constraint>
```

6. Id., *Documentation Elements*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/TD.html> (visité le 22/08/2022)

```

    <sch:rule context="tei:titleStmt/tei:title">
      <sch:assert test="./text()">Le titre doit être renseigné</sch:assert>
    </sch:rule>
  </constraint>
</constraintSpec>

<attList>
  <!-- ajout de valeurs à l'attribut @type -->
  <attDef ident="type" mode="change" usage="rec">
    <valList mode="add" type="closed">
      <valItem ident="main"/>
      <valItem ident="sub"/>
    </valList>
  </attDef>
  <!-- suppression d'attributs -->
  <attDef ident="calendar" mode="delete"/>
  <attDef ident="cert" mode="delete"/>
  <attDef ident="copyOf" mode="delete"/>
  <!-- [...] -->
</attList>

</elementSpec>

```

C'est un schéma Relax NG qui sert de lien entre l'ODD et les documents, et qui permet de procéder à leur validation⁷. Ce document peut être généré automatiquement à partir de l'ODD en utilisant le logiciel d'édition XML Oxygen.

Le choix d'XML-TEI imposait donc de créer un ODD (ainsi qu'un schéma Relax NG) avec des règles adaptées aux besoins des « repères », afin de les appliquer à chacun des documents produits par les membres du projet.

3.3 Cahier des charges

Il était donc nécessaire de prévoir un encodage léger des textes, qui permette de les manipuler facilement. En effet, en limiter la complexité permettrait de simplifier leur traitement pour leur mise en valeur sur le site Web. De plus, différentes personnes pourraient être amenées à rédiger des notices pour le projet, et participer à leur encodage : il fallait donc rendre son usage simple accessible, en prenant en compte que les utilisateurs pourraient ne pas avoir des connaissances approfondies en TEI.

7. RELAX NG, URL : <https://relaxng.org/> (visité le 29/08/2022)

La participation de plusieurs personnes à l'écriture et l'encodage des « repères » impliquait aussi de prendre en compte l'importance du suivi des versions des textes. Différentes étapes peuvent être prises en compte et impacter leur évolution : nous avons ainsi distingué la rédaction du texte des relectures et modifications. Ces étapes doivent être apparentes dans l'encodage, et l'auteur de chaque étapes doit être identifiable. C'est donc le rôle principal qui devait être donné aux métadonnées des « repères », qui devaient en outre contenir des informations contenant le projet et l'édition de ces textes, afin de les identifier clairement comme faisant partie du projet Antonomaz.

Les « repères » devaient pouvoir contenir, en plus des métadonnées avec le suivi des versions et le texte en lui-même, trois éléments bien distincts : les images, la bibliographie, et les renvois vers d'autres documents.

Il a été décidé de limiter l'usage des images, soit en n'en utilisant pas, soit en n'en utilisant qu'une seulement : le but étant ici d'illustrer le propos, et non de réaliser une exposition en lien avec le sujet. Cet usage limité des images avait aussi pour objectif de limiter la difficulté de leur traitement ensuite pour le site Web.

Le deuxième élément était les renvois vers d'autres documents du projet, comme les mazarinades et les autres « repères ». Il s'agissait de pouvoir y faire référence dans le texte, comme dans le cas où ils sont pris comme exemple, mais aussi en-dehors du texte, de façon à indiquer au lecteur qu'il existe d'autres sources qui pourraient l'intéresser, par exemple sous forme de liste. Ces renvois devaient donc pouvoir se faire sous forme de lien avec des documents internes au projet, de la même façon que pour des liens externes vers d'autres sites.

Enfin, le troisième élément concerne la bibliographie. Le choix a été fait de rester le plus simple possible dans le texte : le lecteur doit pouvoir accéder rapidement à l'information qu'il recherche, sans être dérangé par des informations superflues. C'est pourquoi il était préférable d'éviter l'usage des notes de bas de page. En revanche, une bibliographie doit pouvoir être proposée en-dehors du texte, pour tout lecteur qui souhaite aller plus loin. Les références bibliographiques pouvant être structurées de façon très détaillées en XML-TEI, il était souhaitable d'en limiter autant que possible la complexité, toujours dans l'optique d'un encodage léger et facile d'accès.

De façon à pouvoir appliquer le même encodage à tous les « repères », la rédaction d'un ODD spécifique à ces textes, avec des règles, des contraintes et des éléments qui leur sont adaptés s'est imposée.

Chapitre 4

Réalisation : ODD et choix d'encodage

4.1 Choix d'encodage des « repères »

4.1.1 Les métadonnées : le `teiHeader`

Le travail sur les « repères » a donc commencé par la création d'un encodage. Une fois les choix d'encodage déterminés, j'ai pu procéder à la rédaction de l'ODD. Celui-ci a ensuite progressivement évolué au cours du stage en fonction des besoins et des remarques de membres du projet, qui ont pu commencer à encoder des notices en parallèle. L'ODD et l'encodage sont encore susceptibles d'être modifiés à l'avenir selon les besoins du projet, les explications qui suivent sont donc celles sur lesquelles s'est terminé le stage et pourraient ne plus être tout à fait exactes par la suite.

La structure du document se fait dans un élément `TEI` qui englobe toutes les balises. À cet élément ont été attribués deux attributs : `@type` et `xml:id`.

Le premier sert à indiquer de quel type de « repère » il s'agit : cela a été prévu pour l'organisation des fichiers, en particulier pour leur traitement sur le site ensuite. On peut ainsi préciser s'il s'agit d'un « repère » concernant un personnage historique (`@type="personne"`), des événements historiques (`@type="histoire"`), des genres littéraires des mazarinades (`@type="litterature"`), des institutions (`@type="institution"`) ou les caractéristiques matérielles des mazarinades (`@type="forme_livre"`).

Le second attribut permet de donner un identifiant unique au « repère », qui doit correspondre exactement au nom du fichier : il permet de faciliter la création de liens entre les « repères ». Afin de rendre plus clair le nommage des fichiers et l'attribution des identifiants, ainsi pour les retrouver plus aisément, il a été décidé d'utiliser par exemple le nom courant de la personne quand le sujet est une personne, ou le nom sous lequel un événement est le plus connu pour un événement historique. Une limite a été établie à deux

voire trois mots pour composer ce nom.

*Exemple*¹ :

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" type="personne" xml:id="mazarin">
```

Dans cette exemple, nous avons donc une notice concernant un personnage historique d'après l'attribut `@type="personne"`. Le « repère » dont est issu l'exemple étant à propos du cardinal Mazarin, on lui a donné pour valeur d'`@xml:id` « mazarin », et son fichier est donc nommé « mazarin.xml » en conséquence.

L'élément `<teiHeader>` s'insère en première position au sein de l'élément `<TEI>`. Il contient toutes les métadonnées du document. Dans le cas des « repères », les métadonnées servent principalement à suivre les différentes versions du document, ainsi qu'à les insérer dans le cadre du projet : dans des éléments `<publicationStmt>` (mention de publication) et `<encodingDesc>` (description de l'encodage), on présente le projet, les conditions d'accès et d'utilisation et les choix éditoriaux concernant les « repères ». Ces valeurs sont les mêmes pour tous les fichiers.

Le suivi des versions se fait à plusieurs niveaux, et commence par l'introduction des différentes personnes susceptibles de participer à la rédaction dans le `<titleStmt>`, à l'intérieur d'un élément `<respStmt>` (mention de responsabilité). Chaque personne y encode les informations la concernant dans un élément `<persName>`. Le nom complet de la personne doit être indiqué en contenu textuel de la balise, et ses initiales dans l'attribut `@xml:id`. Son rôle est à renseigner dans l'attribut `@role`, et son numéro ORCID dans l'attribut `@ref`.

Exemple :

```
<respStmt>
  <resp>Chaîne de traitement</resp>
  <persName xml:id="KA" ref="orcid:0000-0001-9518-1040" role="Project_manager">Abiv
    Karine</persName>
  <!-- [...] -->
</respStmt>
```

Cette déclaration des personnes participant à la rédaction permet ensuite d'y faire référence dans les éléments qui servent à donner des informations sur la création et les modifications apportées aux fichiers. Dans un élément `<editionStmt>` (mention d'édition), on indique la date de création du « repère » dans un élément `<date>`, ainsi que la personne qui l'a créé dans un élément `<name>`, auquel on ajoute un attribut `@ref` qui permet de faire un lien direct entre cet élément `<name>` et l'élément `<persName>` qui correspond à la même personne dans la déclaration décrite ci-dessus.

1. Les exemples utilisés dans cette partie sont issus de plusieurs « repères ». Un exemple complet se trouve dans l'annexe B

Exemple :

```
<editionStmt>
  <edition>
    <date when="2022-07-07">7 juillet 2022</date>
  </edition>
  <respStmt>
    <resp>Rédaction de la notice</resp>
    <name ref="#AB">Alexandre Bartz</name>
  </respStmt>
</editionStmt>
```

On fait référence aux personnes de la même façon pour indiquer un changement dans le « repère ». Les changements sont notés dans l'élément `<revisionDesc>` (descriptif des révisions) : à chaque changement opéré, un nouvel élément `<change>` est à créer. Celui-ci prend pour attribut `@status`, qui peut avoir pour valeur « `not_modified` », « `modified` » ou « `relu` »². S'il n'y a pas eu de modification ou de relecture, l'élément `<change>` est une balise auto-fermante avec l'attribut `@status="not_modified"`. Dans le cas d'une relecture ou d'une modification, on change la valeur de l'attribut `@status` avec la valeur appropriée, et on indique dans un attribut `who` les initiales de la personne qui a procédé au changement, ainsi que la date à laquelle cela a été fait dans un attribut `@when`. Dans le cas d'une modification, on peut préciser en quoi consiste la modification, ce qui améliore le suivi des versions.

Exemple :

```
<revisionDesc>
  <change status="relu" when="2022-05-10" who="#MF"/>
  <change status="modified" when="2022-05-10" who="#AB">Ajout d'informations sur le
    intendants.</change>
</revisionDesc>
```

Le `<teiHeader>` contient deux autres éléments qui auront une utilité pour la publication des notices sur le site Web : le `<title>` qui est placé au même endroit que la déclaration des noms des personnes participant à la rédaction et aux relectures, et le `abstract` qui se trouve dans le `profileDesc` (description du profil).

Le premier contient le titre du « repères », et si le sujet est à propos d'une personne, on indique aussi son identifiant unique ISNI ou VIAF³ dans un attribut `@ref`.

Le second contient un résumé très bref du « repère », qui ne doit pas dépasser la longueur d'une phrase.

Exemple d'un <title> :

2. La valeur « `relu` » prend un « `e` » final car les textes étaient initialement nommés « `notices` ».

3. Une explication plus approfondie de ces identifiants dans le chapitre 6.2

```
<title type="main" ref="isni:0000000121223121">Mazarin</title>
```

Exemple d'un `<abstract>` :

```
<abstract>
```

```
  <p>Jules Mazarin est un homme d'État d'origine italienne qui a été principal mini-
    la mort de Richelieu à sa propre mort (1643-1661).</p>
```

```
</abstract>
```

4.1.2 Les images

Les images, le texte, les liens vers d'autres documents et les références bibliographiques se trouvent tous dans un élément `<text>` qui vient à la suite du `<teiHeader>`. A l'intérieur de cet élément, on trouve en premier lieu les images.

Pour l'intégration des images, il a été décidé d'utiliser l'élément `<figure>`, dans lequel se trouve un élément `<graphic>` qui permet d'intégrer les images dans le document. Pour ce faire, le lien de l'image doit y être renseigné dans son attribut `@url`. Afin de citer la source d'où provient l'image, il attribut `@source` a été ajouté.

L'image peut être décrite dans un élément `<head>`. Le premier choix avait été de renseigner le titre ou l'intitulé de l'image dans cet élément `<head>` et d'en ajouter si besoin une description dans un élément `<p>` (paragraphe). Les deux éléments étant cependant plus difficiles à cerner dans la pratique et leur utilisation variant selon les personnes, il a finalement été décidé de ne conserver que l'élément `<head>` et d'encadrer les titres ou intitulés mentionnés au sein de celui-ci d'un élément `<title>`.

Exemple :

```
<figure>
```

```
  <graphic source="Bibliotheque_Mazarine"
```

```
    url="https://mazarinum.bibliotheque-mazarine.fr/i/?IIIF=/af/f1/3e/6e/aff13e6e"
```

```
  <head><title>Recit véritable du duel arrivé entre deux soeurs proche de Bordeaux.
```

```
    Imprimé en 1650, ce placard affiché dans les rues attire l'attention à la fois par ses
    illustrations, son texte et la chanson (à gauche) qui pouvait ensuite être
    reprise.</head>
```

```
</figure>
```

4.1.3 Le corps de texte

À la suite du `<figure>` contenant les images vient le texte. Celui-ci est inséré dans un élément `<body>`. Cet élément ne contient que le texte des « repères », les renvois vers d'autres documents et la bibliographie se trouvant dans un autre élément bien distinct.

Dans le `<body>`, on trouve en premier lieu un élément `<head>` qui a pour attribut `@type="main"`

et qui sert à renseigner le titre principal : il s'agit du même titre que celui qui se trouve dans les métadonnées. Il est aussi possible d'ajouter, en second lieu, un deuxième élément `<head>` pour un sous-titre : celui-ci prend pour attribut `@type="sub"`.

Puis, deux structures uniquement sont possibles pour le texte : la première est la plus courante, il s'agit simplement d'enchaîner plusieurs paragraphes dans des éléments `<p>`. Les « repères » étant des textes très courts, il n'est souvent pas nécessaire de prévoir une structure plus complexe.

Exemple :

```
<head type="main">Libelles pro-Mazarin</head>
<head type="sub">Écrits qui défendent le principal Ministre et son gouvernement</head>
<p>On pourrait penser que les "mazarinades", du nom du libelle éponyme de Scarron qui
est d'une virulence et d'une obscénité rares contre un homme à la tête de l'État,
sont toutes toujours dirigées contre Mazarin. Pourtant environ 550 textes (sur le
quelque 5500 libelles repérés) prennent une position favorable au gouvernement,
parmi lesquels environ 150 ont pu être identifiés par H. Carrier comme des textes
"officiels", émanant du gouvernement en place.</p>
<!-- [...] -->
```

La deuxième structure sert dans les cas de « repères » un peu plus longs ou qui ont besoin d'une stratification un peu plus importante du texte afin d'en clarifier le propos. Il est alors possible de créer des parties avec des éléments `<div>` (divisions), qui contiennent chacune un titre dans un élément `<head>` (sans attribut) et du texte dans un ou plusieurs éléments `<p>`.

Exemple :

```
<head type="main">La fiscalité sous l'Ancien Régime</head>
<head type="sub">Présentation des impôts et taxes évoqués dans les mazarinades.</head>

<div>
  <head>Les aides</head>
  <p>À l'origine, au XIVe siècle, les aides correspondent à toute forme d'imposition
  vise à "aider" le roi à faire face aux dépenses du royaume. Au XVIIe siècle,
  des taxes perçues sur la circulation et la consommation de certains produits
  principalement collectées à l'entrée des villes. Elles sont particulièrement
  impopulaires et inégalitaires car elles font augmenter le prix des denrées et
  l'instar d'autres taxes, certains groupes sociaux (nobles, ecclésiastiques, une
  partie des bourgeois) en sont exemptés. Elles connaissent une forte augmentation
  le règne de Louis XIII.</p>
</div>
<div>
```

```

<head>Les aisés</head>
<p>La taxe d'aisés est un prélèvement qui vise principalement les riches parisiens.
Partiellement d'Hémery tente d'y avoir recours en 1644 pour financer le coût de
guerre, mais devant les plaintes des bourgeois et des financiers, il décide de
renoncer à cette idée.</p>
</div>
<!-- [...] -->

```

4.1.4 Liens et éléments bibliographiques : le back

Afin de répondre à l'ensemble du cahier des charges, un élément `<back>` a été ajouté après l'élément `<body>` : c'est à cet endroit que se trouvent les renvois et la bibliographie. Les deux formes de liens étant très similaires, il a été nécessaire de trouver une solution pour les différencier. C'est par l'utilisation d'un élément identique et d'attributs distincts que la séparation a pu être réalisée.

Les renvois

Des liens vers des mazarinades et des « repères » devaient être intégrés dans les documents, la forme qui a semblé la plus adaptée pour remplir ces conditions était la liste. Afin de distinguer les deux types de liens et, si besoin, de les traiter de façon différente sur le site, deux types de listes - avec des éléments `<list>` - ont été créés : une liste avec, pour attribut, `@type="notices"`⁴, et l'autre avec l'attribut `@type="mazarinades"`. Dans les deux types de listes, chaque élément vers lequel on veut renvoyer correspond à un seul élément `<item>`. Pour créer le lien, on l'encadre, à l'intérieur de l'`<item>`, d'un élément `<ref>`. Cet élément prend pour attribut `@target`, dans lequel on renseigne en valeur le nom du fichier vers lequel on souhaite renvoyer. Dans le cas de la liste pour les « repères », on ajoute aussi un attribut `@type`, dans lequel on indique de quel type de notice il s'agit (histoire, personne...), comme on le fait dans l'élément `<TEI>` de chaque fichier.

Exemple :

```

<div type="renvois">
  <list type="notices">
    <item>
      <ref type="forme_livre" target="formats_libelles.xml">Les formats des
        mazarinades</ref>
    </item>
    <item>

```

4. De même que pour l'élément `<change>`, on retrouve ici le nom de « notices » qui avait été initialement prévu pour les « repères ».

```

        <ref type="litterature" target="libelles_pro_mazarin.xml">Les libelles
            pro-Mazarin</ref>
    </item>
</list>
<list type="mazarinades">
    <item>
        <ref target="Moreau3suppl58_GALL.xml">Le Ministère victorieux de l'envie,
            du Faur, prédicateur du Roy.</ref>
    </item>
    <item>
        <ref target="Moreau856_Maz.xml">La custode de la Reyne qui dit tout.</ref>
    </item>
</list>
</div>

```

Les références bibliographiques

Les références bibliographiques se trouvent donc dans une division distincte des renvois. À l'intérieur de celle-ci, chaque référence se trouve dans un élément `<bibl>`. Celui-ci peut contenir de nombreuses informations, la TEI permettant d'être particulièrement précis pour cela. Dans le cadre des « repères », nous avons tenté d'en limiter la complexité, par exemple dans le cas des auteurs (`<author>`, le niveau de description pour aller jusqu'à distinguer les éléments du nom en prénom (`<forename>` et nom de famille `<surname>`, ce que nous avons choisi d'ignorer afin de faciliter l'encodage. En revanche, des éléments plus complexes comme les niveaux de titres (par exemple `@level="m"` pour les monographies) ont été conservés afin de faciliter la mise en page qui vient ensuite sur le site Web.

Exemple :

```

<div type="bibliography">
    <bibl>
        <author>Bernard Barbiche</author>
        <title level="m">Les institutions de la monarchie française à l'époque
            moderne</title>
        <pubPlace>Paris</pubPlace>
        <publisher>Presses Universitaires de France</publisher>
        <date>2012</date>
        <idno type="ISBN">978-2-13-060678-9</idno>
    </bibl>
    <bibl>
        <author>Lucien Bély</author>

```

```

<title level="a">Parlements</title>
<title level="m">Dictionnaire de l'Ancien Régime: royaume de France: XVIIe-XVIII
    siècle</title>
<pubPlace>Paris</pubPlace>
<publisher>Presses Universitaires de France</publisher>
<date>2010</date>
<biblScope unit="page" from="960" to="965"/>
<idno type="ISBN">978-2-13-058422-3</idno>
</bibl>
</div>

```

4.2 Rédaction de l'ODD

Il s'agissait donc de créer un ODD qui permette de contrôler les éléments qui peuvent ou non être utilisés dans l'encodage, et qui établisse des règles et des contraintes propres aux besoins de auteurs des « repères ». Cet ODD aurait pu être généré automatiquement, mais une rédaction à la main permet d'avoir une meilleur gestion des éléments et des règles.

Afin de simplifier l'encodage, il fallait rendre l'ODD très restrictif : en effet, plus on supprime d'éléments et d'attributs, moins d'options et de différences d'utilisations sont possibles. La plupart des éléments présents dans l'encodage on donc fait l'objet d'une analyse afin d'en limiter au maximum le nombre d'attributs, ainsi que le nombre d'éléments que l'on peut utiliser au sein de ceux-ci.

Dans un premier temps, on fait appel aux éléments que l'on souhaite utiliser dans l'encodage, ce qui permet de faire un premier filtre sur le nombre d'éléments existant en TEI. Les éléments appartiennent à des modules, c'est-à-dire à des groupes d'éléments selon leur rôle. Il convient donc de les appeler chacun dans le module qui leur correspond comme dans l'exemple qui suit :

```

<moduleRef key="tei"/>
<moduleRef key="textstructure" include="TEI text body div back"/>
<moduleRef key="header"
    include="teiHeader fileDesc titleStmt editionStmt edition publicationStmt availab
        licence sourceDesc encodingDesc projectDesc editorialDecl profileDesc
        langUsage language abstract revisionDesc change idno"/>

```

Ces éléments, une fois inclus, peuvent ensuite subir des modifications afin de les faire correspondre à nos attentes. Les modifications propres à l'ODD nous ont permis, par exemple, d'imposer des valeurs aux attributs comme le @type de l'élément <TEI> et ainsi conserver les mêmes noms de types de « repères ».

Exemple :

```
<attDef ident="type" mode="change" usage="req">
  <valList mode="add" type="closed">
    <valItem ident="personne">
      <desc>Pour les notices à propos de personnages historiques</desc>
    </valItem>
    <valItem ident="institution">
      <desc>Pour les notices à propos d'institutions</desc>
    </valItem>
    <valItem ident="histoire">
      <desc>Pour les notices à propos d'évènements historiques</desc>
    </valItem>
    <valItem ident="litterature">
      <desc>Pour les notices à propos de formes littéraires</desc>
    </valItem>
    <valItem ident="forme_livre">
      <desc>Pour les notices à propos des formes et formats des mazarinades</de
    </valItem>
  </valList>
</attDef>
```

Cette stabilité imposée dans les noms nous assure par exemple que tous les « repères » pourront être retrouvés lors de leur mise en ligne, ce qui n'aurait pas été le cas sans cette contrainte.

Les règles schematron ont elles aussi été utilisées afin de faciliter l'encodage et prévenir des oublis ou de erreurs éventuels. Dans le cas de l'élément `<biblScope>`, qui sert à indiquer des numéros de volumes, de pages, de chapitres, etc., il y avait un grand nombre de possibilités pour les choix d'attributs. Afin de simplifier le procédé, il a été décidé d'en supprimer un grand nombre, et de ne conserver que les attributs qui servent le plus souvent : `@unit`, qui sert à préciser l'unité bibliographique (page, volume, chapitre, numéro de revue, etc.), `@from` (à partir de), `@to` (jusqu'à) et `@n` (le numéro précis). D'après les règles schematron qui ont été définies dans l'ODD et que l'on peut voir ci-dessous, on peut donc avoir deux cas de figure : soit l'on décrit un intervalle de pages, et dans ce cas on devra l'encoder ainsi : `<biblScope unit="page" from="65" to="94"/>`, soit un numéro de page unique, et dans ce cas l'encodage se fera de cette façon : `<biblScope unit="page" n="19"/>`; ou alors il ne s'agit pas d'une page, et dans ce cas on a affaire à un numéro précis : `<biblScope unit="volume" n="2"/>`.

```
<elementSpec ident="biblScope" mode="change">
  <constraintSpec ident="pages" scheme="schematron">
```

```

    <constraint>
      <sch:rule context="tei:biblScope[@unit='page']">
        <sch:assert test="./@from and ./@to or @n" role="error">Si le biblScope
          il doit contenir ou les attributs @from et @to, ou l'attribut @n<
        </sch:rule>
      </constraint>
    </constraintSpec>
    <constraintSpec ident="biblScopeAttn" scheme="schematron">
      <constraint>
        <sch:rule context="tei:biblScope[@unit!='page']">
          <sch:assert test="@n" role="error">L'élément biblScope doit contenir
        </sch:rule>
      </constraint>
    </constraintSpec>
    <attList>
      <attDef ident="cert" mode="delete"/>
      <attDef ident="copyOf" mode="delete"/>
      <attDef ident="corresp" mode="delete"/>
      <!-- [...] -->
      <!-- liste des attributs qui sont supprimés -->
    </attList>
  </elementSpec>

```

L'ODD évolue aussi selon les besoins : des règles ont pu être établies pendant un certain temps et sont maintenant hors d'usage, soit parce que les choix d'encodage ont changé au fur et à mesure du stage, soit parce que des éléments contenaient des informations qui ont été modifiées et il était utile de faire apparaître des erreurs dans les fichiers qui contenaient encore cette information. Cela a par exemple été le cas pour la description du projet (<projectDesc>) et la présentation des « repères » (<editorialDecl>), qui ont été mis à jour alors que des « repères » avaient déjà été encodés : en vérifiant si le texte correspondait bien à la nouvelle version avec une contrainte Schematron, les fichiers qui n'avaient pas été mis à jour ont pu être repérés et corrigés de manière plus efficace. Ces règles ont été mises en commentaires dans l'ODD des « repères » afin d'en conserver une trace, et suivre l'évolution de l'encodage.

Exemple :

```

<constraintSpec ident="projectDesc" scheme="schematron">
  <constraint>
    <sch:rule context="tei:encodingDesc/tei:projectDesc/tei:p">
      <sch:assert test="./starts-with(., 'Le projet Antonomaz (« ANalyse auToma

```

```
    doit contenir "Le projet Antonomaz (« ANalyse auTOMatique et NumérisatiOn
</sch:rule>
</constraint>
</constraintSpec>
```

Dans le but d'éclaircir le rôle de chaque élément et des contraintes qui ont été définies, une documentation a été rédigée et incluse dans l'ODD. Une version HTML a pu être générée automatiquement afin de rendre la document aisément consultable. De plus, afin de faciliter l'encodage des documents et de le rendre accessibles à des personnes qui ne faisaient pas partie du projet au moment de la création de celui-ci, un manuel d'encodage, contenant un certain nombre d'exemples, a été rédigé en plus de la documentation⁵

5. Tous ces documents sont consultables dans les « livrables » du dépôt Github du mémoire (<https://github.com/zcappe/MemoireM2TNAH>). Le manuel d'encodage a aussi été ajouté en annexe A.

Chapitre 5

Vers TEI Publisher : Présentation et premiers pas

Les « repères » ont été créés dans le but d’enrichir le site Web du projet, et ce, si possible, dès sa mise en ligne. La rédaction de l’ODD coïncidait avec le début de la création du site avec TEI Publisher.

La création de TEI Publisher en 2015 a amené à la fondation de l’association internationale e-editions¹ en 2020, qui poursuit le développement et la maintenance de l’application, et la met régulièrement à jour.² Cet outil repose sur eXist-db, qui est un système de gestion de bases de données en XML, et Xquery, un langage de requête qui permet de manipuler des données encodées en XML.

TEI Publisher permet de transformer tout type de textes encodés en XML-TEI en site d’édition en ligne, en y intégrant différentes fonctionnalités comme la recherche ou la pagination. Il permet aussi l’export en d’autres formats comme LaTeX. Son principal avantage est d’être très accessible : il n’est pas nécessaire d’être développeur pour l’utiliser, des connaissances basiques en HTML peuvent suffire.

Son fonctionnement repose sur un système similaire à l’ODD : on crée un nouvel ODD spécifique à TEI Publisher, qui ne se concentre pas sur la structure du document mais sur sa transformation en site Web. Chaque élément existant dans les documents peut alors être traité pour sa transformation et son insertion dans la mise en page du site.

En ce qui concerne plus particulièrement les « repères », le stage a pris fin à cette étape. Des réflexions ont néanmoins pu commencer à être développées sur leur transformation.

Ils ont été intégrés dans la partie « Ressources » du site, et le choix a été fait de les séparer en catégories de notices : pour ce faire, ce sont les attributs `@type`, qui sont renseignés

1. « Its[e-editions] goal is to empower digital editions of cultural, scientific, and artistic works by promoting open standards related to digital editions and advancing open source applications based on them. » E-editions, URL : <https://e-editions.org/> (visité le 22/08/2022)

2. e-editions, *TEI Publisher - Application*, URL : <https://github.com/e-editions/tei-publisher-app> (visité le 22/08/2022)

dans l'élément racine <TEI> de chaque fichier, qui ont servi à créer ces divisions. On peut alors affiner la recherche d'informations sans passer par une barre de recherche, en proposant d'accéder aux « repères » soit sur « Les institutions », soit sur « Les individus », soit sur « Le contexte historique », soit sur « Les formes matérielles », soit sur les « Formes littéraires et musicales »³.

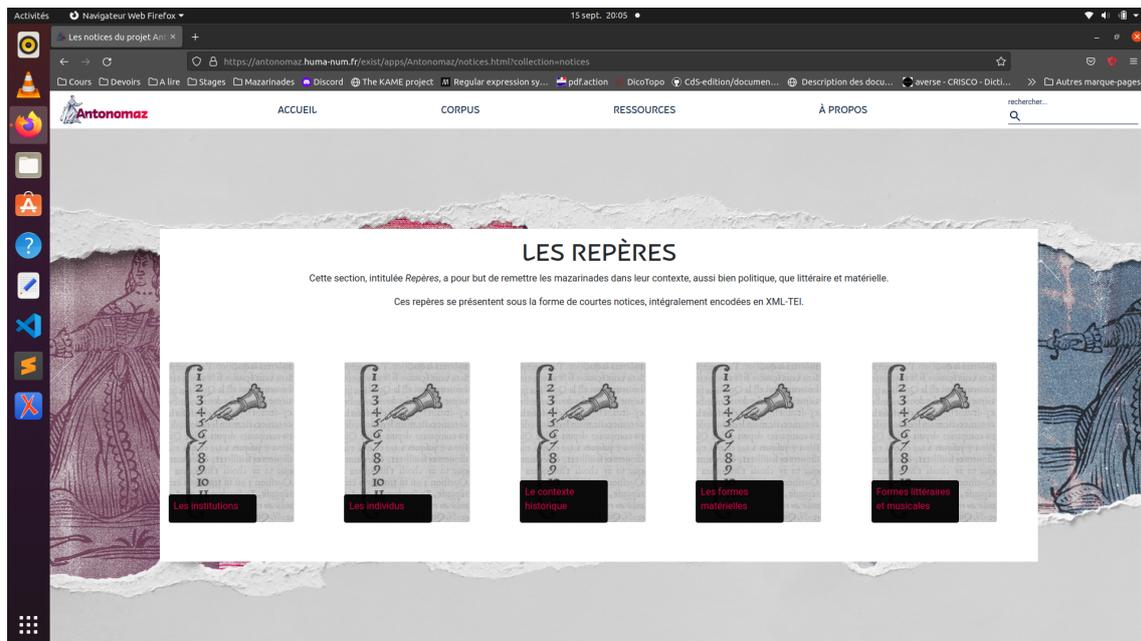


FIGURE 5.1 – Les catégories des « repères »

Une fois dans l'une de ces catégories, les « repères » apparaissent alors sous la forme d'une liste pour une meilleure visibilité. C'est leur titre qui est lisible, et qui est récupéré à partir de l'élément `<title @type="main">` du `<teiHeader>`. Une information complémentaire devait compléter les intitulés, mais son contenu était sujet à discussions : il pouvait s'agir soit du résumé du texte qui est renseigné dans un paragraphe (`<p>`) de l'élément `<abstract>`, soit le nom de l'auteur se trouvant dans `<name>` à l'intérieur du `<respStmt>` (les deux éléments se trouvant dans les métadonnées). Il a finalement été décidé de conserver le nom de l'auteur et non le résumé.

L'une des questions qui se sont alors posées, et qui restaient alors à régler, était le traitement des images : celles-ci pouvant avoir toutes sortes de dimensions, elles ne

3. On peut y accéder avec le lien suivant : <https://antonomaz.humanum.fr/exist/apps/Antonomaz/notices.html?collection=notices>

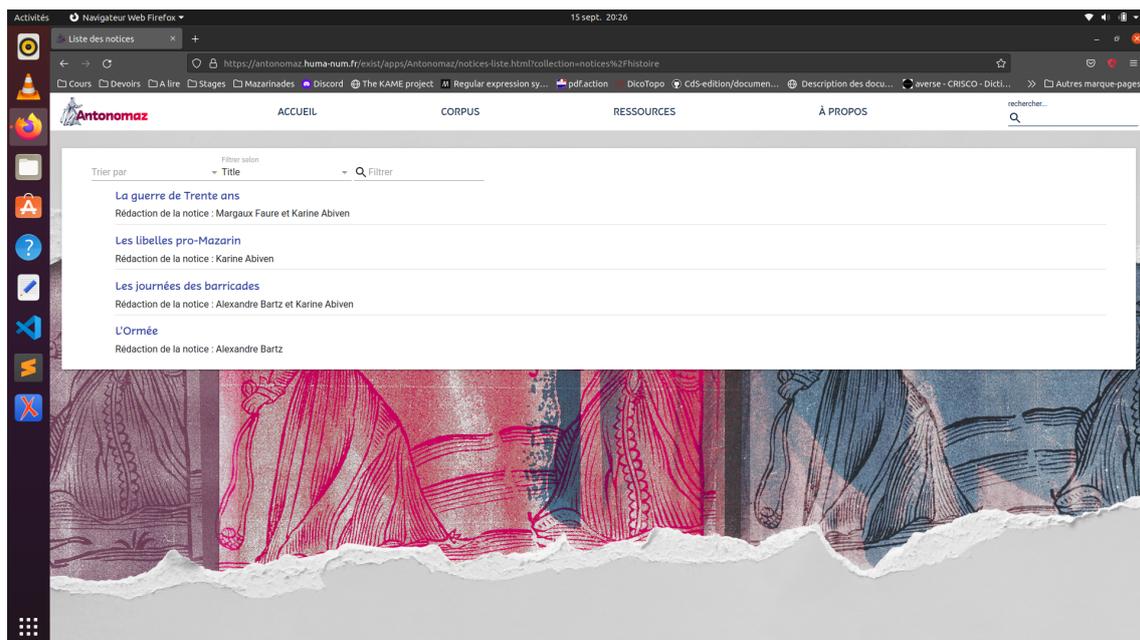


FIGURE 5.2 – Les listes de « repères », avec leur intitulé et le nom des auteurs.

pouvaient s'insérer de la même façon dans la mise en page. Il était alors envisagé de les insérer dans une `<div>` en HTML qui obéirait à des règles spécifiques pour que l'image s'adapte au format de la page du site quelles que soient ses dimensions.

Troisième partie
Base imprimeurs

Chapitre 6

Éléments d'introduction à la base des imprimeurs

6.1 La constitution de la première base

Une première base de données avait été constituée par les membres du projet avant le début du stage, sous la forme d'un fichier CSV¹. Celle-ci avait été créée à partir de la liste des imprimeurs pendant la Fronde, établie par Célestin Moreau dans sa *Bibliographie des mazarinades*². Elle contenait alors 5 colonnes : le nom de famille de l'imprimeur, son prénom, son identifiant, ses indications d'adresse et d'enseigne, et des notes ajoutées par les membres du projet, qui sont des informations complémentaires sur les imprimeurs. Ce fichier était donc complété à la main, soit pour l'ajout de notes, soit pour compléter des adresses lacunaires ou manquantes. L'objectif principal de ce travail manuel était néanmoins l'ajout d'identifiants aux imprimeurs : pour cela, ce sont des identifiants uniques de type ISNI ou VIAF qui étaient utilisés.

L'encodage des mazarinades en XML-TEI comprenant l'encodage du nom des imprimeurs, on pouvait alors leur ajouter un attribut `@ref` dans lequel on renseignait cet identifiant, et ainsi créer un lien avec l'imprimeur.

Exemple :

```
<publisher ref="isni:0000000110423830">
  <persName>
    <forename>Nicolas de</forename>
    <surname>La Vigne</surname>
  </persName>
</publisher>
```

1. CSV (Comma-separated values, est un fichier dans lequel les données sont séparées par des virgules. Cela le rend aisément manipulable, en particulier avec des langages de programmation comme Python.)

2. Moreau, Célestin, *Bibliographie des Mazarinades...*, vol.III, 1851, pp.283-294

C'est grâce à ces identifiants que la base de données a pu être ensuite enrichie.

6.2 Référentiels et identifiants pérennes : ISNI et VIAF

Le Web est en constante évolution, et cela a un impact sur les données et les liens entre celles-ci, qui sont alors instables. Afin de rendre ces données trouvables et identifiables de façon plus durable, on leur applique un identifiant unique, qui ne change pas. On peut alors les lier à l'aide d'un d'URI (Uniform Resource Identifier), auquel on accole l'identifiant unique : on obtient alors un lien pérenne vers la ressource (ou permalien). L'attribution des identifiants uniques permet aussi de limiter les ambiguïtés qui peuvent exister entre deux entités, comme dans le cas des homonymes.

C'est sur ce principe que fonctionne VIAF (Fichier d'autorité international virtuel), hébergé par le Online Computer Library Center (OCLC) qui regroupe et lie des fichiers d'autorité de bibliothèques nationales, d'institutions et d'organismes culturels. Il est utilisé par des chercheurs, des bibliothèques et des utilisateurs du monde entier. On peut trouver les ressources VIAF en créant un lien à l'aide de l'URI adapté et l'identifiant unique qui est attribué aux fichiers d'autorité mis en commun : par exemple, pour l'imprimeur Guillaume Barbier, son identifiant VIAF est « 225408084 », le lien qui permet de le lier à son fichier d'autorité est donc « <http://viaf.org/viaf/225408084> ».

De même, l'International Standard Name Identifier (ISNI) rassemble de notices d'autorité et leur attribue des identifiants uniques. Son objectif est le même que VIAF et il se concentre sur la désambiguïsation des homonymes. Il a aussi une portée internationale mais vise un public plus large. La qualité des ses données est notamment gérée par la Bibliothèque nationale de France et la British Library. Un lien vers une notice d'autorité ISNI prendrait alors la forme suivante : « <http://isni.org/isni/0000000364366936> ». Elle correspond à la notice d'autorité de l'imprimeur Robert Ballard, dont le numéro ISNI est « 0000000364366936 ».

Un identifiant de type ISNI ou VIAF a été attribué à chaque imprimeur, dans le cas où leurs fichiers d'autorité existaient. C'est grâce à ces identifiants que la base de données a ensuite pu être enrichie lors du stage : en utilisant un identifiant unique et pérenne, on peut récupérer automatiquement les informations des imprimeurs qui sont renseignées dans les notices d'autorité, et les associer à leurs autres identifiants s'ils existent, afin de lier leurs données.

6.3 Cahier des charges

La base de données de départ contenait donc peu d'informations sur les imprimeurs : suffisamment pour les identifier, mais pas assez pour la mettre en valeur sur le site Web.

Il s'agissait donc de l'enrichir et la rendre exploitable en l'enrichissant d'autant d'informations que possibles : tout d'abord des données concernant l'imprimeur lui-même, et ensuite des informations concernant ses productions pendant la Fronde.

En premier lieu, il s'agissait d'enrichir la base : les informations sur les imprimeurs étaient sommaires (nom, prénom, identifiant pérenne et éléments d'adresse s'il y en avait, et parfois des notes ajoutées par les membres du projet). Il a donc été décidé, pour ce faire, de récupérer les informations sur les notices d'autorité d'IdRef. Le but était de construire une base de données plus complète, comprenant des éléments biographiques tels que les dates de naissance, de mort, ou des indications sur la vie de ces imprimeurs.

En second lieu, il fallait créer une "fiche imprimeur" pour chacun des imprimeurs de la base, afin de pouvoir repérer et distinguer aisément chacun d'entre eux dans la section consacrée à la base sur le site Web du projet. Le projet Antonomaz ayant déjà collaboré avec E-ditiones pour la création d'un ODD qui correspond à ce type de données, c'est ce schéma qui a été retenu pour encoder la base en XML-TEI. Cet encodage comprenant la possibilité de créer des liens entre les imprimeurs et les textes qu'ils ont imprimé, il fallait aussi trouver une solution pour les lier à leurs mazarinades, et intégrer ces liens dans les fiches. Il était prévu de gérer cet encodage de façon entièrement automatique au moyen d'un script en Python.

Chapitre 7

Enrichissement de la base

7.1 SPARQL et IdRef

Le choix de recourir à IdRef pour enrichir la base de données se justifie par la richesse de ses notices d'autorité, tout d'abord en matière de données biographiques - à la différence de VIAF, par exemple, qui se concentre sur les liens entre les fichiers d'autorité -, et ensuite ce qu'elles regroupent tous les identifiants pérennes que l'on connaît pour une personne.

L'application IdRef

IdRef (Identifiants et Référentiels pour l'Enseignement supérieur et la Recherche) est une application qui a été créée en 2010 par l'Abes (Agence bibliographique de l'enseignement supérieur), aussi à l'origine du Sudoc (Système universitaire de documentation). Cette application rassemble des informations issues d'un certain nombre de bases de données, comme Calames (Catalogue en ligne des archives et des manuscrits de l'enseignement supérieur), ou theses.fr (base des thèses de doctorat françaises). Elle est toujours maintenue et développée, et sert à créer, enrichir et interroger des notices d'autorité, qui peuvent concerner différents types d'entités (personnes, organismes, lieux, etc.).

Dans notre cas, ce sont les notices de personnes qui nous intéressent particulièrement. Celles-ci contiennent une quantité importante d'informations biographique, en plus des différents identifiants uniques de chaque personne. Par ailleurs, tout comme pour ISNI et Viaf, il est attribué à chaque entité un identifiant unique et pérenne qui correspond à sa notice d'autorité sur IdRef, qui permet de l'identifier sans ambiguïté.

Il est possible d'utiliser la plateforme pour effectuer des recherches dans les notices, pour exporter des notices sous différents formats (comme XML ou RDF), mais aussi pour effectuer des requêtes à l'aide de SPARQL, et d'en récupérer les informations.

IdRef : triplestore, RDF et SPARQL

IdRef est un « triplestore », c'est-à-dire une base de données formée à partir de triplets (ou graphes)¹. Ces triplets sont une façon de décrire les ressources et leurs métadonnées qui correspond au modèle RDF (Resource Description Framework). La description sous forme de triplets fonctionne de la façon suivante : il y a un sujet, un prédicat, et un objet, qui peuvent être comparés à une phrase simple (« L'imprimeur imprime des mazarinades »). Le prédicat est décrit à l'aide de vocabulaires (ou ontologies) spécifiques, tels que SKOS (Simple Knowledge organization system) ou FOAF (Friend of a friend), qui permettent de les traiter universellement de la même façon. Cette représentation des données rend possible leur traitement automatique, par exemple pour la récupération d'informations à partir d'un triplestore comme IdRef.

Cela peut se faire avec le langage de requête SPARQL, conçu pour répondre au besoin d'accéder aux données en RDF.

Pour rédiger une requête SPARQL, la première étape consiste à lister les vocabulaires qui seront nécessaires à l'expression de la requête : chaque vocabulaire correspond à une ligne commençant par le mot `PREFIX`, puis vient le préfixe prévu, suivi de la référence à l'URI de l'ontologie. Ces préfixes serviront ensuite pour les prédicats. Dans l'exemple qui suit, `foaf` : est le préfixe du vocabulaire FOAF, et le lien est son URI. *Exemple* :

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

On déclare ensuite ce que l'on souhaite trouver dans des variables (qui fonctionnent comme des boîtes dans lesquelles on place l'information recherchée) à la suite d'un `SELECT`, au début d'une nouvelle ligne. Chacune de ces variables commence par un point d'interrogation, et le nom qu'on leur donne n'a pas d'incidence sur leur comportement dans la requête. Il est cependant préférable de leur donner des noms qui ont un sens en lien avec l'information que l'on récupère, afin de rendre la requête plus compréhensible. Dans l'exemple ci-dessous, deux variables sont déclarées : `?person` et `?nom`. *Exemple* :

```
SELECT ?person ?autreNom
```

On pose ensuite une condition qui permet de préciser l'information que l'on souhaite récupérer : par exemple, on peut vouloir l'identifiant IdReF d'une personne, et son nom. On l'inscrit dans une nouvelle ligne débutant par `WHERE` et à l'intérieur d'accolades. Le sujet est la variable `?person`. Le prédicat est `foaf:name`, car en RDF le nom d'une personne est encodé ainsi : `<foaf:name>Alliot, Gervais</foaf:name>`, c'est pourquoi il est nécessaire de préciser en début de requête le préfixe `foaf`. Pour un autre élément, il peut être nécessaire de faire appel à d'autres préfixes, et connaître le nom de la balise dans laquelle il se trouve. Enfin, l'objet est la variable `?nom`.

1. Abes, *Documentation du triple store d'IdRef*, URL : <https://documentation.abes.fr/aideidrefdata/index.html> (visité le 19/08/2022) et Id., *Les notices d'autorité en RDF*, URL : <https://documentation.abes.fr/aideidrefrdf/index.html> (visité le 19/08/2022)

```
WHERE {?person foaf:name ?nom}
```

Avec cette requête, on obtient à la fois le lien pérenne vers la notice d'autorité de personnes, et leur nom. On peut limiter le nombre de résultats en ajoutant une ligne commençant par `LIMIT`, suivi du nombre de lignes que l'on souhaite.

Il existe deux SPARQL Endpoints pour accéder aux données d'IdRef - c'est-à-dire une interface Web qui permet d'effectuer les requêtes. Le premier est YASGUI, il construit de façon à être accessible à tous, en proposant des exemples et des aides à la rédaction. Le second est le Virtuoso SPARQL Query Editor, il est conçu pour un usage professionnel et est utilisé par des personnes qui ont déjà des connaissances en SPARQL.

7.2 Réalisation de l'intégration des données

Afin de procéder à l'enrichissement de la base, une requête SPARQL a été écrite. Comme la seule organisation ayant un identifiant dans la base de données était l'Imprimerie royale, il a été décidé d'ajouter ses informations manuellement et de se concentrer sur les notices de personnes dans la requête. Pour la mettre en oeuvre, elle a été intégrée à un script Python, et son traitement a été permis grâce à la librairie *Sparqlwrapper*². La requête qui a été rédigée pour obtenir toutes les informations est la suivante :

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bio: <http://purl.org/vocab/bio/0.1/>
PREFIX rdau: <http://rdaregistry.info/Elements/u/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX bnf-onto: <http://data.bnf.fr/ontology/bnf-onto/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT ?person ?ident ?bnf ?nom ?autreNom ?naissance ?mort ?genre ?infos ?in
WHERE {
    ?person ?num <" + identifieur + ">.
    ?person a foaf:Person;
    foaf:name ?nom.
    OPTIONAL {?person skos:altLabel ?autreNom}
    OPTIONAL {?person owl:sameAs ?ident}
    OPTIONAL {?person bnf-onto:FRBNF ?bnf}
    OPTIONAL {?person bio:event [a bio:Birth ; bio:date ?naissance]}
    OPTIONAL {?person bio:event [a bio:Death ; bio:date ?mort]}
    OPTIONAL {?person foaf:gender ?genre}
    OPTIONAL {?person rdau:P60492 ?infos}
```

2. *SPARQL Endpoint interface to Python*, URL : <https://sparqlwrapper.readthedocs.io/en/latest/> (visité le 22/08/2022)

```
OPTIONAL {?person skos:note ?infos2}
}
```

On a commen   par appeler les ontologies que l’on utilise dans la requ  te :

- `foaf` pour FOAF (Friend of a friend),
- `bio` pour BIO (vocabulary for biographical description),
- `rdau` pour RDA (unconstrained properties),
- `owl` pour OWL (Web ontology language),
- `bnf-onto` pour la BnF,
- `skos` pour SKOS (Simple Knowledge organization system).

Les variables sont ensuite d  clar  es dans la ligne du `SELECT`. Il est suivi d’un `DISTINCT`, qui sert    ne conserver que les informations uniques et    ignorer les doublons. Les lignes commen  ant par `OPTIONAL` servent    r  cup  rer des informations qui ne sont pas syst  matiquement renseign  es dans les notices d’autorit   : si cette information optionnelle n’existe pas pour un imprimeur, la requ  te peut continuer. Par exemple, une date de naissance ou de mort peuvent ne pas   tre connues.

On obtient donc la liste de variables suivantes :

- `?person` correspond    l’entit   de l’imprimeur dont on souhaite r  cup  rer les informations. Il a donc le r  le de *sujet* et est r  p  t   pour chaque recherche d’information, afin de lier ces informations    cette unique entit  . Les autres variables sont ses objets.
- `?ident` permet de r  cup  rer la liste des identifiants ;
- `?bnf` permet de r  cup  rer les identifiants de la Bnf ;
- `?nom` permet de r  cup  rer le nom complet de l’imprimeur ;
- `?autreNom` permet de r  cup  rer ses variantes de nom ;
- `?naissance` permet de r  cup  rer sa date de naissance ;
- `?mort` permet de r  cup  rer la date de son d  c  s ;
- `?genre` permet de r  cup  rer son genre ;
- `?infos` permet de r  cup  rer le premier type d’informations biographiques (`<rdau:P60492>`) ;
- `?infos2` permet d’en r  cup  rer le deuxi  me type (`<skos:note>`), les deux balises   tant utilis  es de mani  re similaires par les r  dacteurs des notices).

Le passage `<"" + identifier + "">` fait partie du script Python. La liste des identifiants a   t   r  cup  r  e dans une fonction, et elle est int  gr  e    la requ  te SPARQL de cette fa  on : ainsi, la requ  te est r  it  r  e pour chaque imprimeur (soit chaque item, ou identifiant, dans la liste) et les informations lui sont associ  es, sans intervention manuelle.

Le script permet donc d’effectuer la requ  te pour l’ensemble des imprimeurs³. Une fois les informations r  cup  r  es, elles sont enregistr  es dans un fichier au format CSV.

3. Le script complet peut   tre consult   dans le dossier "livrables" du d  p  t GitHub

Cependant, ce fichier ne comprend que les imprimeurs dont les informations ont pu être récupérées : il y manque donc à la fois les imprimeurs qui n'ont pas d'identifiant, et ceux qui en ont un mais qui n'ont pas de notice d'autorité associée dans IdRef.

Il était donc nécessaire de former à nouveau une base de données unique, afin de pouvoir l'encoder ensuite en XML-TEI de façon uniforme et en le moins d'étapes possibles.

Pour ce faire, deux jointures ont été réalisées à l'aide de Dataiku, une application qui permet de gérer les bases de données. Il a fallu procéder en deux étapes : la première consistait en l'import des informations des imprimeurs qui avaient bien un identifiant. Cette fusion était assez aisée, puisque la correspondance entre les identifiants uniques dans les deux bases a permis d'ajouter les informations sans risque de doublons ou de mauvaises attributions. La deuxième étape a été plus délicate, car il s'agissait d'importer les informations de tous les imprimeurs qui n'avaient pas d'identifiant, et donc pas de connexion avec l'autre base. Elle a nécessité plusieurs essais avant d'obtenir les bons résultats.

L'avantage du travail avec cette requête est que la récupération des informations a pu être faite automatiquement. Le désavantage est le script Python avait des défauts, qui ont dû ensuite être corrigés.

Une autre étape de correction manuelle a aussi été ajoutée, afin de rectifier ou compléter les adresses des imprimeurs à l'aide des informations biographiques ajoutées à partir d'IdRef. Un dernier nettoyage des données a été fait avec Dataiku afin de préparer le CSV pour son encodage automatique, notamment au niveau des formes des listes (autres formes de noms, identifiants), qui ont ainsi pu être traitées d'une même façon, sans cas particuliers.

Chapitre 8

Encodage de la base en XML-TEI

Le fichier CSV contenant les informations des imprimeurs une fois complet et corrigé, il était nécessaire de donner une nouvelle forme à cette base afin de pouvoir la mettre en valeur sur le site Web du projet. Il était aussi envisagé de créer des liens entre les imprimeurs et les textes qu'ils avaient imprimés. La solution qui a été trouvée était d'encoder la base de données en XML-TEI.

Deux raisons justifient ce choix : la première est que, tout comme pour les « repères », il était plus simple de tout encoder en TEI pour leur valorisation sur le site. La seconde raison est qu'avec la TEI, il était possible de créer une fiche individuelle et unique pour chacun des imprimeurs, que l'on pouvait enrichir des informations des mazarinades, et pour ce faire un ODD existait déjà.

8.1 L'ODD Imprinters

Cet ODD avait été créé en 2021 lors d'une collaboration entre le projet Antonomaz et le E-ditiones, projet sous la direction de Simon Gabay, qui tient son origine dans la création d'un catalogue des manuscrits de Madame de Sévigné¹, et a été rédigé par Alexandre Bartz et Claire Jahan².

L'objectif de cette collaboration, nommée « Imprinters », était la création d'une base de données en XML-TEI contenant les informations des imprimeurs du XVII^e siècle. L'intérêt de l'utilisation de la TEI dans ce contexte était de pouvoir exploiter les informations rassemblées sur les imprimeurs et de tenter d'attribuer des impressions anonymes grâce à un travail de comparaison du matériel typographique et des marques d'imprimeurs, en exploitant de nouveaux outils comme eScriptorium³ et la IIIF.

1. E-ditiones, *Carnet de recherche sur la philologie numérique et la littérature du XVII^e s.* URL : <https://editiones.hypotheses.org/> (visité le 21/08/2022)

2. Le dépôt GitHub de la collaboration : <https://github.com/Imprinters>, et le lien vers l'ODD : <https://github.com/Imprinters/ODD>

3. eScriptorium est un outil pour la transcription automatique des documents, et à terme de production des éditions numériques. Il permet notamment l'annotation des images. <https://escriptorium.fr/>

La création de la base des imprimeurs dans le cadre du projet Antonomaz diffère dans ses objectifs, le but n'étant pas ici de comparer les marques d'imprimeurs et le matériel typographique – du moins pour le moment – mais de rassembler autant d'informations que possible sur les imprimeurs et leurs impressions. Néanmoins, la forme de la base de données créée par « Imprinters » correspond à celle qui était souhaitée pour la base imprimeurs du projet. De plus, la réutilisation d'un ODD déjà publié permet de ne pas avoir à en créer un nouveau, ce qui est moins coûteux et évite de reproduire des outils déjà existants.

Il s'agissait donc alors d'adapter les choix d'encodage qui avaient déjà été faits pour « Imprinters » au cas particulier de la base des imprimeurs de la Fronde. La structure en elle-même ne devait pas changer, mais des éléments ont dû être adaptés pour correspondre aux données du projet.

8.2 Adaptation de l'encodage

8.2.1 La structure des « fiches imprimeurs »

Les documents encodés en XML-TEI ont souvent pour élément racine un `<TEI>`, qui « contient un seul document conforme à la TEI, qui comprend un en-tête TEI et un texte »⁴. Dans le cas de l'ODD d'« Imprinters », la construction des fiches est un peu plus complexe : on utilise l'élément `<teiCorpus>`, qui « contient la totalité d'un corpus encodé selon la TEI, comprenant un seul en-tête de corpus et un ou plusieurs éléments TEI dont chacun contient un seul en-tête textuel et un texte ».⁵

D'après l'ODD qui est utilisée pour la base, chaque `<teiCorpus>` correspond à la fiche d'un seul imprimeur. Dans un `<teiCorpus>`, l'en-tête de corpus, c'est-à-dire le `<teiHeader>`, contient toutes les informations concernant l'imprimeur lui-même (informations biographiques, nom, dates de naissance et de mort, etc.). Ce `<teiHeader>` est donc ensuite suivi « d'un ou plusieurs éléments TEI », qui correspondent chacun à une mazarinade, imprimée par l'imprimeur de la fiche.

En suivant la définition des *Guidelines*, ainsi que la construction logique construite par l'ODD d'« Imprinters », on obtient donc la structure suivante :

```
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0" xml:id="IL_FeugeRobert">
  <teiHeader>
    <!-- Informations concernant l'imprimeur -->
  </teiHeader>
</TEI>
```

4. Consortium TEI, `<TEI>`, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-TEI.html> (visité le 22/08/2022)

5. Id., `<teiCorpus>`, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-teiCorpus.html> (visité le 22/08/2022)

```

    <!-- Les informations d'une de ses mazarinades -->
  </TEI>
  <TEI>
    <!-- Les informations d'une autre de ses mazarinades -->
  </TEI>
</teiCorpus>

```

Pour chaque imprimeur, en suivant les règles de l'ODD d'« Imprinters », on doit ajouter un attribut `@xml:id`, qui prend pour valeur un identifiant unique attribué à chaque fiche d'imprimeur, qui permet d'y faire directement référence. D'après la règle définie par l'ODD, l'identifiant commence obligatoirement par la chaîne de caractère « IL_ ».

```

<elementSpec ident="teiCorpus" mode="change">
  <attList>
    <attDef ident="xml:id" mode="change" usage="req"/>
  </attList>
  <constraintSpec ident="xml:id" scheme="schematron">
    <constraint>
      <sch:rule context="tei:teiCorpus">
        <sch:assert test="./starts-with(@xml:id, 'IL')" role="error">Un
          @xml:id est nécessaire.</sch:assert>
        </sch:rule>
      </constraint>
    </constraintSpec>
  </elementSpec>

```

Le nommage des fiches a ensuite été adapté au cas de imprimeurs du projet : il a été décidé d'imposer la forme "IL_ + nom + prenom". Pour des raisons d'interopérabilité, il était préférable d'imposer de ne pas utiliser d'accents ou caractères particuliers (tels que « é », « ë » ou encore les apostrophes). La forme initialement prévue par « Imprinters » était de numéroter les imprimeurs de cette façon : `@xml:id="IL3"`⁶, mais il nous a semblé préférable de rendre les fichiers plus simple à identifier. En suivant cette même logique, les fichiers ont été nommés de la même façon que les fiches. En reprenant l'exemple ci-dessus, l'identifiant est `xml:id="IL_FeugeRobert"`, le nom du fichier est donc `IL_FeugeRobert.xml`.

6. Cet exemple est extrait d'un exemple créé dans le cadre d'« Imprinters ». De même que pour les exemples suivants, il provient de la fiche suivante : https://github.com/Imprinters/printersDatabase/blob/main/printers_files/CotinetArnould_IL3.xml. Pour les imprimeurs de la base d'Antonoz, une fiche complète a été jointe en annexe (voir l'annexe D), et la base complète se trouve dans les livrables du dépôt GitHub

8.2.2 Le <teiHeader>

Le <teiHeader> du <teiCorpus> contient donc les informations de l'imprimeur. D'après les informations de la base, on devrait donc y retrouver : le nom de l'imprimeur et ses variantes, ses dates de mort et de naissance, ses identifiants, ses éléments d'adresse, son genre, et ses informations biographiques.

On retrouve le nom de l'imprimeur de trois façons : la première dans le <titleStmt>, à l'intérieur du <fileDesc>, et les deux autres dans l'élément <person> (dans le <profileDesc>, à l'intérieur lequel on retrouve tous les éléments de description de l'imprimeur.

Le <titleStmt> sert ici à indiquer le nom de l'imprimeur en distinguant son prénom et son nom de famille dans deux éléments distincts `forename` (prénom) et `surname` (nom de famille). On indique aussi en attribut `@ref` l'identifiant de l'imprimeur, qui est celui qui était renseigné dans la base d'origine et correspond aussi à celui que l'on utilise dans l'encodage des mazarinades.

Exemple :

```
<titleStmt>
  <title type="standard" ref="isni:000000043524489X">
    <forename>Robert</forename>
    <surname>Feugé</surname>
  </title>
</titleStmt>
```

Les deux autres façons de présenter le nom de l'imprimeur dans l'élément <person> se font ainsi : la forme standard du nom ("*Nom, Prénom*") se trouve dans un élément <persName> (nom de personne), qui a pour attribut `@type="standard"`. Si l'on connaît des variantes à son nom, elle sont à indiquer dans un autre <persName> qui a pour attribut `@type="alternative"`. Cet élément peut être répété pour chacune des formes du nom.

```
<persName type="standard">Feugé, Robert</persName>
<persName type="alternative">Feugé II, Robert</persName>
```

A la suite des formes de nom à l'intérieur de l'élément <person>, on indique le genre de l'imprimeur dans un élément <sex>, sous la forme « M » ou « F ».

Exemple :

```
<sex>M</sex>
```

À la suite du genre de l'imprimeur, la date de naissance est renseignée dans un élément <birth>, à la fois en valeur textuelle à l'intérieur de la balise, et en valeur de son attribut `@when`. On procède de la même façon pour la date de son décès, mais avec un élément `death`. L'ODD d'« Imprinters » imposant l'utilisation des attributs `@when` et `@cert` (certainty) dans les deux cas mais ne prévoyant pas d'alternative pour les cas des dates

inconnues, il a été décidé d'attribuer les valeurs `@when="1600"` pour la date de naissance et `@when="1700"` pour la date de mort. L'attribut `@cert` prend alors la valeur « low ». Dans le cas d'une date de naissance ou de mort connue, il prend la valeur « high ».

Exemple d'une date de naissance connue et d'une date de mort inconnue :

```
<birth when="1628-02-20" cert="high"/>
<death when="1700" cert="low"/>
```

L'élément `<occupation>` a pour valeur dans toutes les fiches « Imprimeur-libraire », de la même façon que dans les exemples de la base d'« Imprinters ». Il n'apporte cependant pas d'information complémentaire, à moins de vérifier pour chacun des imprimeurs afin de savoir s'il est imprimeur, libraire, ou les deux : cela représente un temps de travail supplémentaire conséquent, et il pourrait être envisagé de ne pas conserver cet élément, au moins pour le moment.

C'est dans un élément `<residence>` que l'on trouve les éléments d'adresse. Dans un élément `<objectName>`, on indique l'enseigne de l'imprimeur, dans `<street>` la rue et les indications de lieu, dans `<settlement>` la ville (avec en attribut `@ref` son identifiant Geonames⁷) et le pays dans `<country>`. L'élément `<residence>` a, tout comme `<birth>` et `<death>`, un attribut `@cert` obligatoire d'après l'ODD : il a donc été décidé de lui attribuer la valeur « high » quand l'adresse contient suffisamment d'éléments pour être considérée comme identifiable (au moins une indication de rue et de ville), et « low » lorsqu'elle est trop incomplète. *Exemple :*

```
<residence cert="high">
  <address>
    <objectName>A l'image Saint-Sébastien</objectName>
    <street>proche le Puits-Certain</street>
    <settlement ref="geonames:2988507">Paris</settlement>
    <country>France</country>
  </address>
</residence>
```

Les éléments biographiques sont dans un élément `<listEvent>`. Il était prévu par l'ODD d'avoir plusieurs éléments en raisons de l'organisation des informations sur IdRef (comme nous l'avons vu avec la requête SPARQL, il a fallu récupérer les informations dans deux éléments différents, qui peuvent être répétés), c'est pourquoi on les place dans une liste. Chaque information est alors renseignée dans un élément `<label>` dans un `<event>`. En outre, la base de données du projet contient, en plus des informations issues d'IdRef,

7. *Geonames* est une base de données géographiques.

des notes ajoutées par les membres du projet. C'est pourquoi un attribut @source a été ajouté sur chaque <label>, afin de préciser si la source est « Idref » ou « Antonomaz ».

Exemple :

```
<listEvent>
  <event>
    <label type="bio" source="IdRef">Paris, 13 septembre 1644-1671?. A
      l'image Saint-Sébastien, près le Puits Certain</label>
  </event>
  <event>
    <label type="bio" source="IdRef">Imprimeur-libraire. Fils de Robert
      I Feugé (15..-1642). Gendre de Robert Mansion</label>
  </event>
</listEvent>
```

En ce qui concerne les différents identifiants, il a fallu adapter l'encodage : il était prévu par « Imprinters » d'ajouter, dans le <fileDesc>, d'intégrer dans un élément <sourceDesc> la liste des sources qui ont servi à créer la base. Chaque source aurait alors été indiquée dans un <bibl> avec un attribut @source avec pour valeur un lien externe vers la source. *Exemple de l'usage prévu par Imprinters :*

```
<sourceDesc>
  <listBibl>
    <bibl source="https://data.bnf.fr/fr/13527051/arnoul_cotinet"/>
      <bibl
        source="http://mazarinades.org/2011/04/liste-normalisee-des-imprimeurs-et-
      </listBibl>
</sourceDesc>
```

Cependant, il nous a semblé préférable de faire un autre usage de cette liste : les sources de la base étant expliquées dans la documentation du projet sur le site Web, la liste des <bibl> (<listBibl> pouvait alors être utilisée pour proposer la liste des identifiants de l'imprimeur. Chaque identifiant correspond alors à un élément <bibl>, dans lequel on en insère le lien en valeur de l'attribut @source.

Exemple de l'usage prévu par Antonomaz :

```
<sourceDesc>
  <listBibl>
    <bibl source="http://data.bnf.fr/ark:/12148/cb15915877m#foaf:Person"/>
    <bibl source="http://viaf.org/viaf/228063889"/>
    <bibl source="http://isni.org/isni/000000043524489X"/>
    <bibl source="http://www.idref.fr/069012245/id"/>
```

```

    </listBibl>
</sourceDesc>

```

8.2.3 Les <TEI>

Les éléments <TEI> qui doivent contenir les principales informations des mazarinades s'insèrent donc à la suite du <teiHeader>. D'après les *Guidelines*, il est obligatoire, même si l'on ne trouve pas de mazarinades pour tous les imprimeurs. On doit donc en insérer un qui ne contienne pas de valeur. Avec les règles imposées par l'ODD d'« Imprinters », cependant, des éléments et des attributs ne peuvent rester sans valeur. Un <TEI> sans information est alors créé. Il se construit ainsi :

```

<TEI corresp="none">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title/>
        <author ref="isni:0000"/>
      </titleStmt>
      <extent>
        <measure unit="pages" quantity="0"/>
      </extent>
      <publicationStmt>
        <publisher ref="https://github.com/Antonomaz">Antonomaz</publisher>
        <availability status="restricted" n="cc-by">
          <licence target="https://creativecommons.org/licenses/by/4.0"/>
        </availability>
      </publicationStmt>
      <sourceDesc>
        <msDesc>
          <msIdentifier>
            <repository>Sans lieu</repository>
            <idno source="None"/>
          </msIdentifier>
          <physDesc>
            <decoDesc>
              <decoNote/>
            </decoDesc>
          </physDesc>

```

```

        </msDesc>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <div/>
    </body>
  </text>
</TEI>

```

L'élément <TEI> doit, d'après l'ODD, avoir un attribut @corresp : celui-ci doit avoir pour valeur le nom du fichier de la mazarinade (dont se terminant par .xml), qui correspond aussi à son identifiant unique. De la même façon que pour les fiches imprimeurs, on le retrouve sous cette forme dans l'encodage des mazarinades :

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" xml:id="Moreau100_GALL">
```

Pour cette mazarinade, sa référence dans une fiche imprimeur se ferait donc de cette façon <TEI @corresp="Moreau100_GALL.xml">.

Seul le <teiHeader> est réellement utilisé dans le cas de la base imprimeurs d'Antonomaz : l'élément <text> reste vide dans tous les cas. En effet, il ne nous a pas semblé pertinent d'intégrer les textes des mazarinades dans les fiches imprimeurs. Cela serait redondant par rapport au travail effectué sur le corpus en lui-même et la mise à disposition des textes bruts avec les numérisations, et cela alourdirait considérablement les fiches imprimeurs sans y apporter d'information supplémentaire. On fait appel aux noms de fichiers des mazarinades dans l'attribut @corresp dans le but de créer un lien vers les notices des mazarinades sur le site Web, ce qui permettrait d'accéder rapidement aux textes sans importer toutes ces informations. Ces <TEI> permettent d'identifier les mazarinades, mais c'est dans leurs notices dédiées que l'on en trouve toutes les données.

Dans les informations que l'on importe, on trouve l'intitulé, l'auteur (si on le connaît), le nombre de pages et le nombre de marques d'imprimeurs, et le lieu de conservation de l'exemplaire présent dans le corpus du projet, ainsi que le lien vers sa numérisation.

Dans le <titleStmt>, on renseigne l'intitulé dans un élément <title>. L'auteur s'y trouve aussi, dans un élément <author>. De la même façon que pour les imprimeurs, on renseigne son identifiant dans un attribut @ref⁸. Si l'auteur n'a pas d'identifiant ou qu'il n'y a pas d'auteur, on reprend le même principe que pour les <TEI> sans mazarinade avec un @ref="isni:0000. L'ODD impose d'utiliser cet attribut et la chaîne de caractères « isni : »

8. Afin de retrouver facilement les auteurs et leurs identifiants, un fichier CSV avait été créé dans le cadre du stage à partir des informations encodées dans les mazarinades, à l'aide d'un script Python. Ce fichier et le script sont tous deux dans les livrables du dépôt Github du mémoire.

ou « viaf : » en valeur.

Les caractéristiques matérielles du document se trouvent dans deux éléments `<measure>`, contenu dans un élément `<extent>`. Les deux ont un attribut `@unit` et un attribut `quantity`. Le premier attribut a deux valeurs imposées par l'ODD, « pages » pour le nombre de pages, et « decoration » pour les marques d'imprimeurs. Les deux éléments `<measure>` correspondent chacun à l'une de ces deux valeurs. Dans l'attribut `@quantity`, on indique le nombre de pages ou de marques.

L'élément `<msDesc>` concerne l'exemplaire en lui-même. Créé pour l'encodage de manuscrits, son usage ici s'applique au livre imprimé. Dans le `<msIdentifier>`, on indique les éléments qui permettent de le localiser et le consulter : l'élément `<publisher>` contient des informations sur le lieu de conservation de l'exemplaire (« Ville, institution »), et dans l'attribut `@source` de l'élément `<idno>` on donne le lien vers sa numérisation.

Dans la logique de la base imprimeurs d'« Imprinters », ce `<msDesc>` devait contenir des images des ornements et des marques présentes dans l'ouvrage, à l'intérieur d'un élément `<decoDesc>`. Celui-ci doit donc être ajouté dans l'encodage, mais il n'est pas utilisé dans la base imprimeurs actuelle. Néanmoins, il se pourrait que les fiches imprimeurs soient enrichies de ce type d'informations à l'avenir, notamment grâce au travail effectué par Margaux Faure sur les numérisations des mazarinades et la IIIF.

Un élément `<TEI>` complet se présente donc ainsi :

```
<TEI corresp="Moreau3suppl58_GALL.xml">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Ministère (le) victorieux de l'envie, par M. du Faur, préd
          Roy.</title>
        <author ref="isni:0000000122809183">François Faure</author>
      </titleStmt>
      <extent>
        <measure unit="pages" quantity="95"/>
        <measure unit="decoration" quantity="1"/>
      </extent>
      <publicationStmt>
        <publisher ref="https://github.com/Antonomaz">Antonomaz</publishe
        <availability status="restricted" n="cc-by">
          <licence target="https://creativecommons.org/licenses/by/4.0"
        </availability>
      </publicationStmt>
      <sourceDesc>
        <msDesc>
```

```

<msIdentifier>
  <repository>Paris, Bibliothèque nationale de France</repo
  <idno source="https://gallica.bnf.fr/ark:/12148/bpt6k1520
</msIdentifier>
<physDesc>
  <decoDesc>
    <decoNote/>
  </decoDesc>
</physDesc>
</msDesc>
</sourceDesc>
</fileDesc>
</teiHeader>
<text>
  <body>
    <div/>
  </body>
</text>
</TEI>

```

L'ajout des mazarinades se fait automatiquement grâce au traitement de la base à l'aide d'un script Python, qui est décrit ci-dessous. Cependant, comme nous le verrons, cela ne se fait pas dans tous les cas. Un manuel d'encodage a donc été rédigé afin de documenter les choix qui ont été faits et la manière de procéder pour ajouter des informations dans les fiches. Cela peut s'avérer particulièrement utiles pour les mazarinades qui peuvent être liées à imprimeur. Il contient aussi une partie sur l'ajout d'informations et leur encodage pour les imprimeurs dans le `<teiHeader>`, dans les cas où de nouvelles informations peuvent compléter celles existantes - par exemple dans le cas où une notice d'autorité `IdRef` est trouvée pour un imprimeur dont on ne connaissait pas les identifiants - ou une nouvelle fiche, si l'imprimeur ne se trouvait pas dans la base de données. Dans ce dernier cas, on peut aussi ajouter ses informations dans le fichier CSV de la base, afin de créer automatiquement sa fiche encodée avec les informations de ses mazarinades.

8.3 Automatisation de l'encodage : Python et lxml

Un encodage automatique des 302 fiches imprimeurs permet un gain de temps considérable, et limite le risque d'erreur ou le problème que peuvent poser les différentes compréhensions d'encodage s'il est fait par plusieurs personnes. On obtient alors un ensemble de documents cohérent et uniforme.

Pour ce faire, un script Python est un moyen efficace et rapide. Une librairie Python existe par ailleurs pour traiter du XML - que ce soit pour la récupération d'informations des éléments ou la création d'éléments, d'attributs, voire de documents complets. Cette librairie, *lxml*⁹, a donc été utilisée pour encoder la totalité des fiches imprimeurs en un seul script¹⁰.

Encodage automatique des informations des imprimeurs

Le script¹¹ est divisé en plusieurs fonctions. Les premières, `get_maz_corresp(doc)`, `get_maz_title(doc)`, `get_maz_author(doc)`, etc., sont utiles pour l'étape suivante, c'est-à-dire l'encodage des mazarinades. L'encodage automatique des informations des imprimeurs qui se situent dans le `<teiHeader>` des fiches a été fait dans la fonction principale `def to_tei()`. Pour chaque imprimeur, on va chercher les informations dans les colonnes qui y correspondent dans le fichier CSV de la base. Avec *lxml*, on crée chaque élément, puis on y insère les valeurs. L'avantage de traiter automatiquement ces données est qu'il permet d'envisager tous les cas de figure, et de prévoir des solutions. Ainsi, par exemple, le problème des dates de naissance et de mort (connues ou inconnues) peut être géré en quelques lignes :

```
birth = ET.SubElement(person, 'birth') # On crée l'élément <birth>
birth_date = row[3] # On récupère la date de naissance dans une
                    # variable "birth_date"
if birth_date: # S'il y a bien une valeur:
    birth.set('when', birth_date) # on indique la date en valeur
                                  # d'un élément @ref
    birth.set('cert', 'high') # et on attribue la valeur "high"
                              # à l'attribut @cert
else: # Sinon, s'il n'y a pas de date de naissance:
    birth.set('when', '1600') # on attribue la valeur par
                              # défaut "1600" à @ref
    birth.set('cert', 'low') # et on donne la valeur "low"
                              # à l'attribut @cert
```

Les listes de valeurs ont pu poser quelques difficultés : il s'agit de créer, pour chaque item à l'intérieur d'une liste, un nouvel élément en TEI. C'est le cas notamment pour les listes d'identifiants, de variantes de noms et d'informations biographiques. La solution trouvée est la suivante :

9. *lxml - Processing XML and HTML with Python*, URL : <https://lxml.de/> (visité le 17/08/2022)

10. Ce script est consultables sur le dépôt Github du mémoire.

11. Le script complet est consultable dans le dossier « livrables » du dépôt GitHub du mémoire.

```

autres_ids = row[15] # On récupère les identifiants (sous forme
                    # de chaîne de caractères)
list_ids = list(autres_ids.split(",")) # On en fait une liste
                                       # en indiquant que le séparateur
                                       # est une virgule
if list_ids: # S'il y a une liste d'identifiants:
    for ident in range(len(list_ids)): # Pour chaque identifiant dans
                                       la liste:
        identifiant = str(list_ids[ident]) # on indique qu'il s'agit d'une
                                           # chaîne de caractères
        if identifiant: # s'il y en a bien un:
            bibl = ET.SubElement(listbibl, 'bibl') # On crée un
                                                    # élément <bibl>
            bibl.set('source', identifiant) # Et attribue à son attribut
                                             # @source la valeur
                                             # de l'identifiant
        else: # S'il n'y a pas d'identifiant:
            bibl = ET.SubElement(listbibl, 'bibl') # On crée tout de même un
                                                    # élément <bibl>
            bibl.set('source', 'none') # et on donne à son attribut @source
                                       # la valeur "none"

```

En plus de gérer automatiquement l'absence ou la présence des valeur et la création des éléments et attributs en conséquence, le script permet aussi de gérer le nommage des fichiers et l'attribution des identifiants aux fiches imprimeurs. Puisque les noms des imprimeurs étaient traités, cela impliquait un nettoyage des données complémentaires afin d'éviter des problèmes d'interopérabilité et d'encodage des caractères : par exemple la suppression de l'accentuation ou des espaces.

```

# on récupère le nom complet des imprimeurs "Nom + Prénom)
fullname2 = row[0] + row[1]
# on effectue un nettoyage de la chaîne de caractères afin
# de supprimer ou remplacer tous les caractères qui pourraient
# poser problème
fullname2 = re.sub(" \\(|\\)|'|'-|,|\\.\"", "", fullname2)
fullname2 = re.sub("é|è|ë|ê", "e", fullname2)
fullname2 = re.sub("É|È|Ë|Ê", "E", fullname2)
fullname2 = re.sub("ç|Ç", "c", fullname2)
fullname2 = re.sub("ô|Ô", "o", fullname2)
# on crée une variable qui permet de concaténer les chaînes

```

```

# de caractères "IL_" avec le nom complet nettoyé
xmlids = "IL_" + fullname2
# On crée l'attribut @xml:id sur l'élément <teiCorpus>, avec
# pour valeur le nom concaténé et nettoyé
root.set('xml:id', xmlids)
# on recrée la même variable - elle pourrait être fusionnée
# avec la précédente
name_file = 'IL_' + fullname2

# Écriture des fichiers .xml
tree = ET.ElementTree(root)
# on crée les fichiers à l'intérieur du dossier "imprimeurs_tei",
# et ces fichiers prennent le nom défini ci-dessus
tree.write("imprimeurs_tei/%s.xml" % name_file,
xml_declaration=True, encoding="utf-8")

```

Encodage automatique de leurs mazarinades

L'encodage des mazarinades a posé plus de difficultés : il s'agissait alors d'aller chercher les informations dans plus de 2 500 fichiers encodés en XML-TEI les informations nécessaires. Comme dit précédemment, il a été nécessaire de créer des fonctions supplémentaires afin de récupérer ces informations. C'est encore *lxml* que l'on utilise pour ce faire, car cette librairie permet aussi de lire et extraire les données des fichiers en XML. Le script fonctionne alors de cette manière : d'abord, pour chaque imprimeur, on crée une fiche. Puis, à l'intérieur de cette fiche, on va chercher dans les mazarinades celles qui ont été produites par le même imprimeur, et on insère leurs valeurs dans la fiche. *Exemple d'une fonction pour récupérer les liens des numérisations des mazarinades :*

```

# On définit ici la fonction. Le "doc", placé en paramètre de la fonction,
# a pour argument (pour valeur) les fichiers des mazarinades.
def get_maz_idno(doc):
    # on va chercher l'élément qui contient les informations
    # de l'imprimeur (<publisher>) dans les métadonnées des textes
    printers = doc.xpath("//tei:teiHeader//tei:sourceDesc/tei:biibl/tei:publisher",
namespaces=ns)
    # on crée un dictionnaire vide nommé "all_idno"
    all_idno = {}
    # pour chaque imprimeur dans tous les imprimeurs:
    for printer in printers:
        # on récupère son identifiant dans l'attribut @ref et on le place

```

```

# dans la variable "ident"
ident = printer.get("ref")
# si l'identifiant de l'imprimeur n'existe pas déjà dans
# le dictionnaire
if ident not in all_idno:
    # on l'ajoute en clé du dictionnaire, et on indique
    # que sa valeur sera sous forme de liste
    all_idno[ident] = []
    # on stocke les liens vers les numérisations des mazarinades dans la
    # variable "idnos"
    idnos = doc.xpath("//tei:teiHeader//
                      tei:sourceDesc/tei:bibl/tei:ref/@target",
                      namespaces=ns)
    # Pour chaque lien dans tous les liens:
    for idno in idnos:
        # On associe la valeur du lien à l'identifiant
        # de l'imprimeur
        all_idno[ident].append(idno)
# Le dictionnaire final prend alors la forme suivante :
# {"Identifiant de l'imprimeur" : [liste de liens vers les numérisations]}
return all_idno

```

Une fois ces fonctions déclarées, on les appelle pour la création de chaque élément dans la fonction principale. Les listes et dictionnaires dans lesquels les informations des mazarinades ont été stockées nécessitent alors un traitement particulier, qui a complexifié le script. Cela est particulièrement visible dans le cas des auteurs, car ces éléments avaient la particularité de pouvoir exister plusieurs fois (on pouvait avoir un, mais aussi deux auteurs ou plus pour un seul texte), et il fallait aussi récupérer l'identifiant de chaque auteur. *Fonction qui permet de récupérer les identifiants et les noms des auteurs :*

```

def get_maz_author(doc):
    # on va chercher l'élément qui contient les informations
    # de l'imprimeur (<publisher>) dans les métadonnées des textes
    printers = doc.xpath("//tei:teiHeader//tei:sourceDesc/tei:bibl/tei:publisher",
                          namespaces=ns)
    # On initie un dictionnaire vide appelé "all_authors"
    all_authors = {}
    # Pour chaque imprimeur dans les imprimeurs:
    for printer in printers:
        # On va chercher son identifiant dans l'attribut @ref

```

```

ident = printer.get("ref")
# Si l'identifiant n'est pas encore dans le dictionnaire:
if ident not in all_authors:
    # On l'ajoute en clé, et on lui donne pour valeur un dictionnaire vide
    all_authors[ident] = {}
    # On va chercher l'élément qui contient les informations de l'auteur
    # (<author>) dans les métadonnées des textes
    authors = doc.xpath("//tei:teiHeader//tei:sourceDesc/tei:bibl//tei:author",
                        namespaces=ns)
    # Pour chaque auteur dans les auteurs:
    for author in authors:
        # Si l'auteur a un attribut @ref (qui contient son identifiant)
        if author.xpath("./@ref", namespaces=ns):
            # alors on place la valeur de l'identifiant dans la
            # variable "author_ids"
            author_ids = author.xpath("./@ref", namespaces=ns)[0]
            # et son nom dans la variable "author_name"
            author_name = author.xpath("./text()", namespaces=ns)
            # Puis, dans le dictionnaire vide qui est la valeur de l'imprimeur
            # On ajoute l'identifiant de l'auteur en clé, et le nom de
            # l'auteur en valeur
            all_authors[ident][author_ids] = author_name
        # Si l'auteur n'a pas d'identifiant:
        else:
            # alors on procède de la même façon, en remplaçant l'identifiant
            # de l'auteur par la chaîne de caractères "id"
            author_name = author.xpath("./text()", namespaces=ns)
            all_authors[ident]["id"] = author_name

# le dictionnaire final (dictionnaire imbriqué, ou nested dict) se présente
# sous la forme suivante:
# {"Identifiant de l'imprimeur":
#   {"Identifiant de l'auteur": "nom de l'auteur"}
# }
return all_authors

```

La création de l'élément <author> et de l'attribut @ref pour chaque auteur nécessite alors de récupérer, d'abord, le dictionnaire imbriqué, puis la clé et la valeur de ce dictionnaire. Puis pour chaque auteur (il peut y en avoir plusieurs par texte), on lui crée un élément :

```

# Pour chaque clé et valeur dans le premier dictionnaire:

```

```

for authors_dict_key, authors_dict_value in get_maz_author(doc).items():
    # On récupère la valeur (c'est-à-dire le deuxième dictionnaire)
    authors_dict = authors_dict_value
    # Puis, dans le deuxième dictionnaire:
    # Pour chaque clé (l'identifiant de l'auteur) et sa valeur (le
    # nom de l'auteur:
    for author_id, author_name in authors_dict.items():
        # On crée un élément <author>
        author = ET.SubElement(titlestmt2, 'author')
        # Si l'identifiant de l'auteur correspond à la
        # chaîne de caractères "id":
        if author_id == "id":
            # c'est qu'il n'a pas d'identifiant, donc on crée
            # son attribut @ref avec la valeur neutre "isni:0000"
            author.set('ref', 'isni:0000')
            # Et on insère dans l'élément le nom de l'auteur
            author.text = ' '.join(author_name)
        # Sinon, si l'auteur a un identifiant:
        else:
            # On donne l'identifiant en valeur de l'attribut @ref:
            author.set('ref', str(author_id))
            # Et on place son nom à l'intérieur de l'élément
            author.text = ' '.join(author_name)

```

L'attribution des mazarinades à l'imprimeur repose donc entièrement sur son identifiant : s'il correspond à celui que l'on trouve dans la mazarinade, c'est qu'il en est l'imprimeur et on récupère les informations. Les <TEI> de chaque mazarinade peuvent alors être créés. Ce système est fiable - l'identifiant étant unique, il n'y a pas d'erreur d'attribution des mazarinades - mais pose un problème de taille : s'il fonctionne de façon efficace pour les imprimeurs dont on connaît l'identifiant, il est inutile pour ceux qui n'en ont pas.

En effet, si les imprimeurs n'ont pas d'identifiant, la recherche de leurs mazarinades ne peut pas fonctionner et il faut alors leur créer un élément <TEI> sans information. Ce cas est celui d'environ la moitié des imprimeurs sur les 302 fiches qui ont été créées. Cela peut s'expliquer par le fait qu'au moins une partie d'entre eux n'existe pas dans les référentiels, et l'on n'a donc pas pu leur trouver d'identifiant. Il est aussi possible que leur identifiant existe bien mais n'ait pas encore été ajouté dans la base.

Il y a aussi le cas des imprimeurs dont on a bien un identifiant, dont on ne connaît pas de mazarinade dans le corpus : ils ont pu les publier de façon anonyme, ou elles n'ont pas encore été récupérées et ajoutées au corpus. Des imprimeurs qui n'ont pas d'identifiant

ont aussi pu imprimer des textes de façon anonymes, et ainsi rendre leur association à celles-ci encore plus difficiles.

Pour le moment, 155 des 302 imprimeurs ont pu se voir attribuer des mazarinades. En prenant en compte les raisons de cet écart mentionnées ci-dessus, le nombre se justifie. Néanmoins, il est possible de compléter les lacunes de la base : la solution envisagée pour le moment serait de procéder manuellement à l'ajout des mazarinades manquantes.

Les *Guidelines* et l'ODD d'*Imprinters* imposent néanmoins l'ajout d'un élément <TEI> sans information aux imprimeurs qui n'ont pas de mazarinade associée. Pour cela, une autre fonction a été créée, à la suite de celle qui permet la création des fiches. Elle fonctionne selon la principe suivant : on vérifie d'abord s'il existe déjà un élément <TEI>, et si c'est le cas, on ne fait rien. En revanche, s'il n'en existe pas, on ajoute l'élément avec toutes les valeurs neutres.

Des mazarinades vers les imprimeurs

Les mazarinades et les imprimeurs sont donc maintenant liés - du moins pour une partie des imprimeurs - à partir des fiches des imprimeurs. Lorsque la base de données sera mise en ligne, il sera donc possible de consulter les mazarinades de chaque imprimeur et d'accéder à leurs informations à l'aide des liens qui seront créés grâce aux identifiants de leurs fichiers.

Une dernière étape semblait alors s'imposer : si des liens pouvaient être créés des imprimeurs vers les mazarinades, il devait être possible de faire la même chose dans le sens inverse, c'est-à-dire des mazarinades vers leurs imprimeurs. Il a donc été décidé d'écrire un autre script¹² afin de créer des attributs @xml:id dans les éléments <publisher> des mazarinades, en plus des attributs @ref qui contiennent leurs identifiants de type ISNI ou VIAF. Avec ce script, on insère les noms des fichiers des fiches imprimeurs (de type « "Il_NomPrenom.xml" ») dans ces @xml:ids. Ainsi, un lien pourra être créé pour amener directement sur la fiche de l'imprimeur.

Exemple d'un <publisher> d'une mazarinade auquel on a ajouté le lien vers la fiche de son imprimeur :

```
<publisher ref="isni:0000000110423830" xml:id="IL_LaVigneNicolasde.xml">
  <persName>
    <forename>Nicolas de</forename>
    <surname>La Vigne</surname>
  </persName>
</publisher>
```

C'est aussi là que le choix du nommage des fichiers prend son sens : il est plus simple de les retrouver et les ajouter aux mazarinades manuellement ensuite si les fichiers portent le nom

12. Le script complet est consultable dans le dossier « livrables » du dépôt GitHub du mémoire.

des imprimeurs. Car on retrouve là aussi la limite des identifiants : ce script fonctionnant à l'aide des identifiants identiques, tout comme l'ajout des mazarinades dans les fiches, il y a nécessairement des imprimeurs dont le nom de fichier ne sera pas ajouté aux mazarinades, et une partie du travail sera à faire à la main. *Extrait du script :*

```
# Si on a identifiant d'imprimeur (ISNI/VIAF) (dans les mazarinades):
```

```
if publisher_ref:
```

```
    # get_impr_id() est une fonction créée dans le script
```

```
    # pour récupérer dans les fiches imprimeurs : en clé, l'identifiant
```

```
    # de l'imprimeur, et en valeur le nom de fichier de sa fiche
```

```
    for key, value in get_impr_id().items():
```

```
        # Si les deux identifiants sont les mêmes:
```

```
        if str(publisher_ref) == key:
```

```
            # Alors on vérifie s'il existe déjà un attribut
```

```
            # @xml:id dans les <publisher> des mazarinades
```

```
            if doc.xpath("//tei:teiHeader//tei:sourceDesc/tei:bibl/tei:publisher/@xml
```

```
                # Dans ce premier cas, il y en a un : on ne fait rien
```

```
                """
```

```
        else:
```

```
            # Dans ce second cas il n'y en a pas: on crée alors un
```

```
            # attribut @xml:id qui prend en valeur le nom du fichier
```

```
            # de la fiche imprimeur
```

```
            publisher.set('xml:id', value)
```

```
            print(etree.tostring(publisher, pretty_print=True))
```

Conclusion

La base de données des imprimeurs est donc maintenant complètement encodée en XML-TEI et ses documents sont valides d'après l'ODD d'« Imprinters ». Elle est en cours de traitement pour être mise en ligne sur le site Web du projet¹³. Elle a cependant un certain nombre de limites qui ont été mentionnées ci-dessus : les imprimeurs sans identifiants ne peuvent être liés à des mazarinades, et seule une partie d'entre eux ont pu être associés à une notice d'autorité d'IdRef.

Dans ce dernier cas, il n'est guère possible d'améliorer les résultats : la requête est faite en se basant entièrement sur les identifiants des imprimeurs, et si l'on n'a pas pu récupérer leurs informations, c'est qu'ils n'ont pas de notice existante. Seul un enrichissement manuel peut pour le moment être envisagé.

Dans le premier cas, on pourrait tenter d'améliorer le script en ajoutant une possibilité : si l'imprimeur n'a pas d'identifiant, on pourrait le rechercher dans les métadonnées des mazarinades à l'aide de son nom et son prénom. Cette solution avait été écartée, car elle présente un certain nombre d'inconvénients : tout d'abord, les homonymes poseraient problème, puisque les mazarinades des uns pourraient être attribuée à d'autres. Ensuite, les façons d'orthographier les noms peuvent varier, et ont pu être modifiées dans la base de données même, lors de la récupération des informations des imprimeurs et le nettoyage manuel des données : on ne récupérerait alors pas certaines mazarinades, voire aucune, pour une partie des imprimeurs. Enfin, ces erreurs créées par l'ajout de cette fonction augmenteraient l'importance du travail à accorder à l'ajout manuel des mazarinades, puisqu'il s'y additionnerait la nécessité de relire et corriger les mazarinades ajoutées, et il resterait toujours des imprimeurs sans leurs mazarinades. Il est donc moins coûteux de ne pas ajouter cette fonction.

Une dernière remarque sur cette partie concerne le script de la requête SPARQL : celui-ci est très long et sa structure complexe, il est certainement possible de l'alléger et de le simplifier. Une amélioration à apporter pourrait être d'insérer, dans la mesure du possible, les nouvelles informations dans le premier CSV, ce qui éviterait de passer par plusieurs étapes de jointures des bases qui augmentent le risque d'erreurs.

En ce qui concerne les « repères », la question suivante s'était posée : doit-on conti-

13. On pourra bientôt consulter la base en suivant ce lien : <https://antonomaz.humanum.fr/exist/apps/Antonomaz/imprimeurs.html?collection=imprimeurs>

nuer à créer des règles et des contraintes qui la rendraient plus restrictive ? Le choix a été fait de ne pas poursuivre : on pouvait en effet l'envisager, de façon à avoir un encodage très strict qui n'autorise aucune variante d'utilisation selon les personnes. Cependant, on pourrait ajouter des règles et des contraintes à n'en plus finir, et malgré tout trouver toujours des cas particuliers d'utilisation qui n'avaient pas été envisagés. L'ODD et les choix d'encodage sont fonctionnels, et d'après l'utilisation qui a pu en être faite au cours du stage, l'utilisation correspond dans la plupart des cas à l'usage prévu. Le nombre de « repères » et de personnes participant à leur rédaction étant peu nombreux, les corrections éventuelles peuvent être faites sans être trop coûteuses en temps.

De plus, un ODD trop strict pourrait poser problème dans les cas où l'évolution de l'encodage est nécessaire : le cas des images en était un exemple, et le traitement de « repères » sur TEI Publisher pourrait amener à procéder à d'autres modifications de ce type¹⁴. Il est donc préférable de conserver un peu de souplesse dans l'encodage pour ne pas à avoir à en modifier trop de règles dans les cas d'un changement dans les choix déjà arrêtés.

14. Les « repères » peuvent déjà être consultés à cette adresse : <https://antonomaz.humanum.fr/exist/apps/Antonomaz/notices.html?collection=notices>

Annexe A

Manuel d'encodage des « repères »

Le manuel d'encodage, ici reproduit, est la copie exacte du manuel qui avait été rédigé alors que les « repères » étaient encore nommés « notices ». Le choix a été fait de ne pas faire la correction, afin de conserver les documents tels qu'ils ont été produits lors du stage et tels qu'ils sont encore utilisés pour le projet.

Manuel d'encodage des notices

Ce manuel explique comment encoder les notices par étape.

Pour avoir une vue d'ensemble, un exemple d'encodage complet et commenté se trouve dans ce même dossier (**Exemples**). Un autre exemple, sans commentaires, y a aussi été déposé pour servir de structure de base à l'encodage des nouvelles notices.

L'élément racine "TEI"

obligatoire

Le premier élément à modifier est le premier élément de l'encodage : le <TEI>, qui se trouve à la "racine" du fichier. Celui-ci a deux attributs, @type et @xml:id, qui doivent tout deux prendre des valeurs bien définies.

L'attribut @type propose une liste de valeurs qui nous permettent de préciser dans quelle catégorie la notice se situe, cela facilitera leur traitement sur le site ensuite. Celle-ci peut en effet concerner :

- des personnes
- des institutions
- des événements historiques
- des formes littéraires
- des formes du livre

L'attribut `@xml:id` doit avoir pour valeur le même nom que celui du fichier de la notice. On choisira alors un nom en rapport direct avec le sujet de la notice : pour une institution, le nom de l'institution ; pour une personne, le nom sous lequel on connaît le mieux cette personne ; pour une forme littéraire, le nom de cette forme... S'il y a besoin d'utiliser des espaces, on les remplacera par des tirets bas : " _ ".

Exemple :

```
<TEI xmlns="http://www.tei-c.org/ns/1.0" type="histoire"
  xml:id="journee_des_barricades">
```

```
<!-- On a ici une notice qui fait partie de la catégorie des
notices sur des événements historiques,
ce qui est indiqué dans l'attribut @type="notice_histoire";
elle est nommée "journee_des_barricades" dans son attribut @xml:id,
ce qui veut dire que son fichier
est nommé "journee_des_barricades.xml" -->
```

Le "teiHeader"

L'élément `<teiHeader>` comprend toutes les métadonnées des notices, il permet de comprendre l'ensemble du contexte de leur rédaction et leur place dans le projet.

Les éléments `<publicationStmt>`, `<sourceDesc>` et `<encodingDesc>` sont identiques dans toutes les notices et ne doivent pas être modifiés.

L'élément "titleStmt"

obligatoire

L'élément `<titleStmt>` est à modifier en deux endroits : dans les éléments `<title type="main">` et `<respStmt>`.

L'élément `<title type="main">` prend en contenu textuel de titre principal de la notice. Celui-ci doit correspondre exactement au `<head type="main">` que l'on retrouvera plus tard dans la rédaction de la notice

L'élément `<respStmt>` contient la liste des responsables de la chaîne de traitement des notices et du projet. Si le nom de la personne qui rédige la notice y figure déjà dans un élément `<persName>`, avec un attribut `@xml:id` qui contient ses initiales et un attribut `@role` qui renseigne sur son rôle dans le projet, il n'est pas nécessaire de faire des modifications ici (le numéro ORCID indiqué dans l'attribut `@ref` n'est à renseigner que si la personne en a déjà un et souhaite l'ajouter). En revanche, si son nom n'y figure pas, il est nécessaire de l'ajouter dans un `<persName>` à la suite de ceux déjà inscrits avec les attributs `@xml:id`

pour ses initiales et @role pour son rôle dans le projet, et éventuellement aussi l'attribut @ref pour le numéro ORCID.

Exemple :

```
<titleStmt>
  <title type="main">La fiscalité sous l'Ancien Régime</title>
  <title type="sub">Édition Antonomaz</title>
  <respStmt>
    <resp>Chaîne de traitement</resp>
    <persName xml:id="KA" ref="orcid:0000-0001-9518-1040"
      role="Project_manager">Abiven
      Karine</persName>
    <persName xml:id="GL" ref="orcid:0000-0002-4795-2362"
      role="Project_manager">Lejeune
      Gaël</persName>
    <persName xml:id="AB" ref="orcid:0000-0003-0850-8266"
      role="engineer">Bartz
      Alexandre</persName>
    <persName xml:id="ZC" ref="orcid:0000-0002-3327-3967"
      role="intern">Cappe
      Zoé</persName>
    <persName ref="orcid:0000-0001-5815-9506"
      role="intern">Faure Margaux</persName>
  </respStmt>
</titleStmt>
```

L'élément "editionStmt"

obligatoire

L'élément <editionStmt> sert à indiquer par qui et quand la notice a été rédigée.

Un premier élément <edition> contient un élément <date>. C'est à la fois dans son attribut @when et dans son contenu textuel qu'il faut indiquer la date de rédaction de la notice.

Un deuxième élément <respStmt> sert à nommer le responsable de la rédaction de la notice. L'élément <resp> et son contenu ("Rédaction de la notice") sont à laisser tels quels. L'élément à modifier ici est <name> : il prend en attribut @ref les initiales de la personne qui a rédigé la notice, telles qu'elles sont signalées dans l'attribut @xml:id de éléments <persName> du <titleStmt> expliqué ci-dessus, mais cette fois-ci avec un '#' devant les initiales. En contenu de l'élément, on indique le prénom suivi du nom de la même personne.

Exemple :

```
<editionStmt>
  <edition>
    <date when="2022-04-29">29 avril 2022</date>
  </edition>
  <respStmt>
    <resp>Notice rédigée par</resp>
    <name resp="#MF">Margaux Faure</name>
  </respStmt>
</editionStmt>
```

L'élément "abstract"

obligatoire

L'élément <abstract> contient un seul élément <p> (paragraphe) dans lequel le sujet de la notice doit être très brièvement présenté, en quelques mots seulement (pas plus d'une phrase). Cela apparaîtra sous les titres des notices pour les rendre plus explicites avant d'accéder à la notice complète.

Exemple :

```
<p>Institutions rendant la justice en dernier ressort, exerçant la grande police et
  enregistrant les lois.</p>
```

L'élément "revisionDesc"

obligatoire si modification ou relecture

L'élément <revisionDesc> contient un élément <change> qui sert à indiquer si la notice a été, ou non, relue ou modifiée. Par défaut son attribut @status a pour valeur "not_modified".

Une fois la notice déposée sur le dépôt, elle peut être relue ou être modifiée, dans ce cas la valeur de l'attribut @status devra être changée en "modified" ou "relue". Il faudra alors aussi préciser la date du changement avec un attribut @when, et qui est la personne responsable du changement avec l'attribut @who. Dans cet attribut, on indique les initiales de la personne après un '#', ce qui permet de faire un rappel de la valeur des attributs @xml:id des personnes renseignées dans le <titleStmt>.

Pour chaque modification ou relecture effectuée, on ajoutera un nouvel élément <change> à l'intérieur du <revisionDesc>.

Exemples :

```
<revisionDesc>
  <change status="modified" when="2022-05-04" who="#MF"/>
  <change status="relue" when="2022-05-09" who="#AB"/>
</revisionDesc>
```

Le "text"

L'élément `<text>` contient et structure le texte même de la notice.

L'élément "figure"

facultatif

L'élément `<figure>` sert à insérer des images et, si on le souhaite, à les décrire. Il s'insère en première place à l'intérieur de l'élément `<text>`, avant le `<body>`.

Il contient deux éléments :

- un élément `<graphic>` dans lequel on insère l'image en elle-même. On place le lien de l'image dans un attribut `@url`, et on indique la source d'où provient l'image dans un attribut `@source` (comme Gallica ou la Bibliothèque Mazarine). Cet attribut propose une liste des sources les plus récurrentes, mais cette liste n'est pas restrictive et il est possible d'y intégrer une autre source. Si cette autre source est citée dans plusieurs notices, il peut être intéressant de la signaler pour l'ajouter à la liste.

obligatoire

- un élément `<head>` permet de donner un titre à l'image (celui-ci sera inséré en légende de l'image avec le nom de la source). Une description de l'image peut aussi être ajoutée dans cet élément. *obligatoire*

Si la notice n'a pas d'image associée, on peut indiquer "none" en valeur des deux attributs de `<graphic>`, et laisser l'intitulé en balise auto-fermante : `<head/>`.

Exemple :

```
<figure>
  <graphic source="Gallica" url="lien de l'image"/>
  <head>Intitulé de l'image</head>
</figure>
```

L'élément "body"

obligatoire

Le body contient la notice en elle-même.

Il doit toujours commencer par un titre principal : `<head type="main">`, qui correspond exactement au `<title type="main">` au tout début de l'encodage. *obligatoire*

Ce titre principal peut être suivi d'un sous-titre : `<head type="sub">` *facultatif*

Pour la structure du texte, deux options sont possibles :

- une structure en paragraphes, utilisant au minimum un élément `<p>` (paragraphe)

La structure ressemble alors à ceci :

```
<body>
  <head type="main">La fiscalité sous l'Ancien Régime</head>
  <head type="sub">Présentation des impôts et taxes évoqués
    dans les mazarinades.</head>
  <p>Premier paragraphe</p>
  <p>Second paragraphe, etc.</p>
</body>
```

- une division du texte en plusieurs parties, chaque partie se trouvant alors dans un élément `<div>` (division). Chaque partie aura un titre `<head>` et au moins un paragraphe `<p>`.

La structure ressemble alors à ceci :

```
<body>
  <head type="main">La fiscalité sous l'Ancien Régime</head>
  <head type="sub">Présentation des impôts et taxes évoqués
    dans les mazarinades.</head>
  <div>
    <head>Titre de la première sous-partie</head>
    <p>Paragraphe de la sous-partie</p>
    <!-- possibilité d'ajouter d'autres paragraphes dans la sous-partie -->
  </div>
  <div>
    <head>Titre de la seconde sous-partie</head>
    <p>Paragraphe de la sous-partie</p>
    <!-- possibilité d'ajouter d'autres paragraphes dans la sous-partie -->
  </div>
</body>
```

Les paragraphes peuvent aussi contenir, si nécessaire, des éléments qui permettent d'insérer des citations (`<quote>`) et des listes (`<list><item></item></list>`).

L'élément "back"

facultatif

L'élément `<back>` peut venir après le `<body>` pour donner des informations complémentaires sur la notice, il n'est cependant pas obligatoire. On peut y trouver deux types d'éléments : les renvois vers d'autres notices ou des mazarinades qui sont liées à la notice, et qui peuvent intéresser le lecteur ; et/ou des références bibliographiques qui ont servi à la rédaction de la notice et qui peuvent intéresser le lecteur.

— la première partie : `<div type="renvois">` *facultatif*

Elle sert à créer des liens vers d'autres notices ou des mazarinades, sur le même principe que des mots-clés. Ces liens seront cliquables une fois sur le site.

Les renvois sont structurés sous forme de liste pour faciliter l'accès et la lisibilité. On trouve deux types de listes pour les deux types de renvois : les `<list type="notices">` et `<list type="mazarinades">`. *il n'est pas obligatoire de renseigner des informations dans les deux types de listes*

Chaque élément au sein d'une `<list>` est renseigné dans un élément `<item>`.

Pour lier un renvoi vers une notice ou une mazarinade, on utilise l'élément `<ref>` et on lui donne comme attribut `@target`. La valeur de cet attribut correspond au nom du fichier de la notice ou de la mazarinade vers laquelle on renvoie. Dans le cas des notices, on indique aussi dans un attribut `@type` le type de notice vers laquelle on renvoie (institution, histoire...).

Exemple :

```
<div type="renvois">
  <list type="notices">
    <item><ref type="institution" target="parlement_de_Paris.xml">Parlement
      de Paris</ref></item>
    <item><ref type="institution" target="parlements.xml">Parlement</ref></item>
  </list>
  <list type="mazarinades">
    <item>
      <ref target="Moreau2701-iii_MAZ.xml">Le Parlement burlesque
        de Paris</ref>
    </item>
  </list>
</div>
```

— la deuxième partie : `<div type="bibliography">` *facultatif, mais recommandé*

Cette partie sert à faire des références bibliographiques. Les éléments cités ont servi à rédiger la notice et/ou peuvent intéresser le lecteur pour aller plus loin sur le sujet. Le

but est de fournir quelques références et non une liste exhaustive de tous les ouvrages sur le sujet, il n'est pas nécessaire d'en renseigner beaucoup.

Chaque référence bibliographique est renseignée dans un élément `<bibl>`.

Dans cet élément, on peut donner un grand nombre d'informations sur les sources, le but est ici de donner les principales informations, et non de rentrer dans trop de détails, il est donc préférable de rendre l'encodage aussi simple que possible.

Les titres sont tous indiqués dans un élément `<title>`. Le type, ou niveau de titre, est indiqué dans un attribut `@level` qui a cinq valeurs possibles :

- a = articles, poèmes...
- m = monographie
- j = journal, revue, magazine
- s = série de publications ou collection
- u = unpublished (choses non publiées comme des thèses, des mémoires...)

Les guillemets ou indications comme de l'italique ne sont pas nécessaires, tout sera ajouté automatiquement ensuite sur le site.

Les auteurs sont renseignés dans un élément `<author>`. Chaque auteur se trouve dans un élément `<author>` qui lui est propre.

Les informations concernant le volume, les pages, etc., sont indiquées dans l'élément `<biblScope>`. Cet élément n'accepte pas de contenu textuel, toutes les informations sont indiquées dans ses attributs. On indique dans l'attribut `@unit` le type d'information (cet attribut en propose une liste prédéfinie : chapitre, partie, volume, page...). Les numéros sont indiqués dans l'attribut `@n`, sauf dans le cas où l'on souhaite indiquer une intervalle de pages : dans ce cas, on utilise les attributs `@from` ("de la page...") et `@to` ("à la page...").

Le lieu de publication est indiqué dans l'élément `<pubPlace>`.

L'éditeur est indiqué dans l'élément `<publisher>`.

La date de publication est indiquée dans l'élément `<date>`.

Les identifiants, tels que l'ISBN, sont indiqués dans l'élément `<idno>`, et son attribut `@type` propose une liste de types d'identifiants possibles, mais il est possible d'en indiquer d'autres qui ne figurent pas dans la liste. S'il n'y a pas d'identifiant à renseigner, on peut garder la balise en auto-fermante (`<idno/>`) pour marquer l'information manquante, ou ne pas l'ajouter.

S'il s'agit d'une ressource en ligne, on met le lien vers la ressource dans l'attribut `@target` de l'élément `<ptr>` (pointeur). Cet élément est une balise auto-fermante, qui n'accepte pas de contenu textuel.

Exemple :

```
<div type="bibliography">
  <bibl>
```

```
<author>Hubert Carrier</author>
<title level="m">La Presse de la Fronde (1648-1653): Les mazarinades</title>
<biblScope unit="volume" n="1"/>
<title level="m">La conquête de l'opinion</title>
<pubPlace>Genève</pubPlace>
<publisher>Droz</publisher>
<date>1989</date>
<idno type="ISBN"/>
</bibl>
</div>
```


Annexe B

Exemple de « repère »

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="../../../ODD/ODD_ANTONOMAZ_notices.rng" type="application/xml" schem
<?xml-model href="../../../ODD/ODD_ANTONOMAZ_notices.rng" type="application/xml" schem
<TEI xmlns="http://www.tei-c.org/ns/1.0" type="institution" xml:id="fiscalite">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title type="main">La fiscalité sous l'Ancien Régime</title>
        <title type="sub">Édition Antonomaz</title>
      <respStmt>
        <resp>Chaîne de traitement</resp>
        <persName xml:id="KA" ref="orcid:0000-0001-9518-1040"
          role="Project_manager">Abiven
          Karine</persName>
        <persName xml:id="GL" ref="orcid:0000-0002-4795-2362"
          role="Project_manager">Lejeune
          Gaël</persName>
        <persName xml:id="AB" ref="orcid:0000-0003-0850-8266"
          role="engineer">Bartz
          Alexandre</persName>
        <persName xml:id="ZC" ref="orcid:0000-0002-3327-3967"
          role="intern">Cappe
          Zoé</persName>
        <persName xml:id="MF" ref="orcid:0000-0001-5815-9506"
          role="intern">Faure
          Margaux</persName>
      </respStmt>
    </titleStmt>
```

```

<editionStmt>
  <edition>
    <date when="2022-05-02">2 mai 2022</date>
  </edition>
  <respStmt>
    <resp>Rédaction de la notice</resp>
    <name ref="#AB">Alexandre Bartz</name>
  </respStmt>
</editionStmt>

<publicationStmt>
  <publisher>Projet ANTONOMAZ / Sorbonne Université</publisher>
  <ref target="https://github.com/Antonomaz"/>
  <date type="file_creation" when="2022-04-21">21 avril 2022</date>
  <availability n="cc-by" status="restricted">
    <licence target="https://creativecommons.org/licenses/by/4.0"/>
  </availability>
</publicationStmt>

<sourceDesc>
  <p>Document électronique original</p>
</sourceDesc>
</fileDesc>
<encodingDesc>
  <projectDesc>
    <p>Le projet Antonomaz (« ANalyse auTOMatique et NumérisatiOn
      des MAZarinades ») vise à rassembler, visualiser et exploiter
      les textes des mazarinades, ensemble de libelles, pamphlets et
      autres textes imprimés en masse lors de la Fronde,
      entre 1648 et 1653.</p>
  </projectDesc>
  <editorialDecl>
    <p>Les notices permettent de présenter le contexte de production
      des mazarinades et leurs enjeux. Elles se limitent à la période
      de la Fronde pour les éléments biographiques.</p>
  </editorialDecl>
</encodingDesc>
<profileDesc>

```

```

<langUsage>
  <language ident="fra">Document en français</language>
</langUsage>
<abstract>
  <p>Cette notice traite des principales formes d'imposition
    sous l'Ancien Régime.</p>
</abstract>
</profileDesc>
<revisionDesc>
  <change status="relue" when="2022-05-10" who="#MF"/>
  <change status="modified" when="2022-05-10" who="#AB">Ajout
    d'informations sur les intendants.</change>
</revisionDesc>
</teiHeader>

<text>
  <figure>
    <graphic source="Bibliotheque_Mazarine"
      url="https://mazarinum.bibliotheque-mazarine.fr/i/?IIIF=/
      b8/42/d8/0a/b842d80a-6199-48db-9e07-73c35911c17a/iiif/751063002_
      MAZ014455_000000a_000009.tif/117.60000000000001,113.61191860465115,
      1940.4,988.844476744186/!462,235/0/default.jpg"/>
    <head>Extrait de la page de titre de <ref target="Moreau242_MAZ.xml">Arrest
      de la cour de parlement [...]</ref></head>
  </figure>

  <body>
    <head type="main">La fiscalité sous l'Ancien Régime</head>
    <head type="sub">Présentation des impôts et taxes évoqués dans les mazarinad
    <div>
      <head>Les aides</head>
      <p>À l'origine, au XVe siècle, les aides correspondent à toute forme
        d'imposition qui vise à "aider" le roi à faire face aux dépenses
        du royaume. Au XVIIe siècle, ce sont des taxes perçues sur la
        circulation et la consommation de certains produits et sont
        principalement collectées à l'entrée des villes. Elles sont
        particulièrement impopulaires et inégalitaires car elles font
        augmenter le prix des denrées et, à l'instar d'autres taxes,
        certains groupes sociaux (nobles, ecclésiastiques, une partie

```

des bourgeois) en sont exemptés. Elles connaissent une forte augmentation sous le règne de Louis XIII.</p>

</div>

<div>

<head>Les aisés</head>

<p>La taxe d'aisés est un prélèvement qui vise principalement les riches parisiens. Particielly d'Hémery tente d'y avoir recours en 1644 pour financer le coût de la guerre, mais devant les plaintes des bourgeois et des financiers, il décide de renoncer à cette idée.</p>

</div>

<div>

<head>La gabelle</head>

<p>La gabelle est un impôt sur le sel généralisé au début du XIVE siècle et relativement rentable puisqu'il rapporte environ 13 millions de livres en 1646. Consommé et utilisé en quantités importantes, le sel est alors un produit du quotidien que le roi taxe lors de sa vente. Le montant de cet impôt n'est pas le même dans tout le royaume puisqu'il varie en fonction du lieu d'achat du sel. Cet impôt est tout particulièrement détesté, les cris de "Vive le Roi sans gabelle" n'étaient pas rares lors des émeutes fiscales du XVIIe siècle.</p>

</div>

<div>

<head>Les rentes de l'Hôtel de Ville</head>

<p>Créées en 1522, les rentes de l'Hôtel de Ville correspondent à une forme d'emprunt public particulièrement rentable. Les Parisiens peuvent alors déposer leur argent à l'Hôtel de Ville de Paris et percevoir ensuite des intérêts sous forme de versements trimestriels. Cet argent peut ensuite être utilisé par le roi. Il paraît alors plus sûr de prêter à l'Hôtel de Ville qu'au roi.</p>

<p>Le rendement de ces rentes, relativement élevé à l'origine (8.33 \%), ne cesse de décroître sous les ministériats de Richelieu et Mazarin et cette baisse est telle qu'à la veille de la Fronde, certains craignent une annulation simple des rentes. Le versement des intérêts est alors irrégulier en raison de la guerre et de l'état général des finances du roi.</p>

<p>Les rentes de l'Hôtel de Ville sont fréquemment évoquées dans les mazarinades, aussi bien dans les pamphlets que dans les textes officiels. Il semble en effet que de nombreux Parisiens aient investi une partie

de leurs économies dans ce système. La cessation des paiements des rentes de l'Hôtel de Ville pousse ainsi les rentiers à se regrouper en comités, ce qui donne naissance à une nouvelle force politique que le Parlement cherche à contrôler, avec plus ou moins de succès. Le versement des rentes reste relativement irrégulier durant toute la Fronde et ce jusqu'à l'intervention de Colbert dans les années 1660.

</div>

<div>

<head>La taille</head>

<p>La taille est le principal impôt royal. Personnelle dans le Nord du royaume, la taille est levée sur les personnes et correspond à une forme d'impôt sur le revenu. Dans le Sud du royaume, elle prend la forme d'un impôt foncier et ce sont les terres qui sont taxées. On parle alors de taille réelle.</p>

<p>Le montant de la taille est fixé chaque année par le souverain et est ensuite réparti dans tout le royaume. Son principal bénéfice pour la monarchie est l'absence totale de droit de regard du Parlement de Paris sur cet impôt (contrairement à d'autres, qui doivent au préalable être enregistré par celui-ci). Cet impôt pouvait ainsi être épisodiquement augmenté, on parle alors de "crues" de la taille.</p>

<p>À partir de 1642/1643, ce sont les intendants, des agents placés sous l'autorité directe du roi, et non plus les bureaux des finances, qui sont chargés de répartir et lever cet impôt. Leur action se caractérise par un usage fréquent de la force et par la confiscation de biens pour obtenir les sommes dues au roi. Face au fort mécontentement des parlements, des trésoriers et de la population, les intendants sont suspendus par une déclaration royale de juillet 1648. Symbole fort de l'autorité monarchique, ils sont cependant rétablis dès 1653, une fois les troubles de la Fronde apaisés.</p>

<p>Enfin, cet impôt cesse d'être payé par une grande partie des paysans au printemps 1648.</p>

</div>

<div>

<head>Les traites</head>

<p>Les traites correspondent à une forme de droits de douane prélevés sur les marchandises à leur entrée et sortie du royaume ou de certaines provinces. Cette taxe, comme la plupart des autres

prélèvements, se caractérise par sa grande diversité et pouvait, en fonction de sa provenance et des lieux traversés, faire doubler le prix d'un produit.</p>

</div>

</body>

<back>

<div type="renvois">

<list type="mazarinades">

<item>

<ref target="Moreau242_MAZ.xml">Arrest de la cour de Parlement, pour empescher le divertissement des deniers des tailles, subsistances, fermes, aydes, gabelles & autres deniers de sa Majesté, ordonnez par precedens arrests de ladite Cour, estre portez en l'Hostel de ville de Paris. Du huictième febvrier 1649.</ref>

</item>

</list>

</div>

<div type="bibliography">

<bibl>

<author>Lucien Bély</author>

<title level="a">Aides</title>

<title level="m">Dictionnaire de l'Ancien Régime: royaume de France: XVIIIe siècle</title>

<pubPlace>Paris</pubPlace>

<publisher>Presses Universitaires de France</publisher>

<date>2010</date>

<biblScope unit="page" from="45" to="46"/>

<idno type="ISBN">978-2-13-058422-3</idno>

</bibl>

<bibl>

<author>Lucien Bély</author>

<title level="a">Aisés</title>

<title level="m">Dictionnaire de l'Ancien Régime: royaume de France: XVIIe-XVIIIe siècle</title>

<pubPlace>Paris</pubPlace>

<publisher>Presses Universitaires de France</publisher>

```
<date>2010</date>
<biblScope unit="page" n="46"/>
<idno type="ISBN">978-2-13-058422-3</idno>
</bibl>
<bibl>
  <author>Lucien Bély</author>
  <title level="a">Gabelle</title>
  <title level="m">Dictionnaire de l'Ancien Régime: royaume de France:
    XVIe-XVIIIe siècle</title>
  <pubPlace>Paris</pubPlace>
  <publisher>Presses Universitaires de France</publisher>
  <date>2010</date>
  <biblScope unit="page" from="581" to="583"/>
  <idno type="ISBN">978-2-13-058422-3</idno>
</bibl>
<bibl>
  <author>Lucien Bély</author>
  <title level="a">Taille</title>
  <title level="m">Dictionnaire de l'Ancien Régime: royaume de France:
    XVIe-XVIIIe siècle</title>
  <pubPlace>Paris</pubPlace>
  <publisher>Presses Universitaires de France</publisher>
  <date>2010</date>
  <biblScope unit="page" from="1200" to="1201"/>
  <idno type="ISBN">978-2-13-058422-3</idno>
</bibl>
<bibl>
  <author>Lucien Bély</author>
  <title level="a">Traites</title>
  <title level="m">Dictionnaire de l'Ancien Régime: royaume de France:
    XVIe-XVIIIe siècle</title>
  <pubPlace>Paris</pubPlace>
  <publisher>Presses Universitaires de France</publisher>
  <date>2010</date>
  <biblScope unit="page" from="1221" to="1222"/>
  <idno type="ISBN">978-2-13-058422-3</idno>
</bibl>
</div>
</back>
```

```
</text>  
</TEI>
```

Annexe C

Manuel d'encodage des imprimeurs

Manuel d'encodage des imprimeurs

Les fiches des imprimeurs sont construites au sein d'un élément englobant `<teiCorpus>`, dans lequel on insère à la fois les informations sur les imprimeurs et les mazarinades qu'ils ont imprimées (s'il y en a dans le corpus).

Ajout d'informations sur les imprimeurs

Les informations sur les imprimeurs sont renseignées dans le `<teiHeader>` du `<teiCorpus>`.

Selon les sources d'où proviennent les informations (IdRef, Antonomaz ou Moreau), celles-ci peuvent être plus ou moins fournies. Certains imprimeurs n'ont ainsi aucune information de renseignée, et leur fiche "vide" se présente alors sous cette forme :

```
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0" xml:id="IL_BenarLouis">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title type="standard" ref="isni:0000">
          <forename>Louis</forename>
          <surname>Benar</surname>
        </title>
      </titleStmt>
      <editionStmt>
        <edition/>
        <respStmt xml:id="ZC" ref="orcid:0000-0002-3327-3967">
          <name>Zoé Cappe</name>
          <resp>Créateur de la fiche</resp>
        </respStmt>
      </editionStmt>
    </fileDesc>
  </teiHeader>
</teiCorpus>
```

```

</editionStmt>
<publicationStmt>
  <publisher ref="https://github.com/Antonomaz"
    xml:id="Antonomaz">Antonomaz</publisher>
  <availability status="restricted" n="cc-by">
    <licence target="https://creativecommons.org/licenses/by/4.0"/>
  </availability>
</publicationStmt>
<sourceDesc>
  <listBibl>
    <bibl source="none"/>
  </listBibl>
</sourceDesc>
</fileDesc>
<profileDesc>
  <particDesc>
    <listPerson>
      <person>
        <persName type="standard">Benar, Louis</persName>
        <persName type="alternative"/>
        <sex>M</sex>
        <birth when="1600" cert="low"/>
        <death when="1700" cert="low"/>
        <occupation>Imprimeur-libraire</occupation>
        <residence cert="low">
          <address>
            <objectName/>
            <street/>
            <settlement ref="geonames:0000"/>
            <country>France</country>
          </address>
        </residence>
        <listEvent>
          <event>
            <label type="bio" source="IdRef"/>
          </event>
        </listEvent>
      </person>
    </listPerson>

```

```

        </particDesc>
    </profileDesc>
    <revisionDesc>
        <listChange>
            <change when="2022-07-10" who="#ZC">Création de la fiche</change>
        </listChange>
    </revisionDesc>
</teiHeader>
<!-- pour les mazarinades, voir la partie suivante -->
</teiCorpus>

```

Si l'on souhaite enrichir une fiche imprimeur, en premier lieu, il est préférable de vérifier que l'attribut `@xml:id` de l'élément `<teiCorpus>` contient bien en valeur "IL_" suivi du "Nom" et "Prénom" de l'imprimeur, donc sous cette forme "IL_NomPrenom" (par exemple : *IL_AlliotGervais*)

Puis :

fileDesc et titleStmt

Au sein de l'élément `<titleStmt>`, l'élément `<title>` a pour attribut `@ref`. Si l'imprimeur n'a pas d'identifiant de type ISNI ou VIAF, il prend par défaut la valeur *"isni :0000"*. Si l'identifiant de l'imprimeur peut être ajouté, il sera à indiqué sous la forme *"isni :numéro"*, *"viaf :numéro"* ou *"bnf :numéro"*. Dans le cas où l'imprimeur a plusieurs identifiants, on préférera indiquer l'ISNI, sinon le VIA, ou la BnF.

Dans l'élément `<title>`, si ces informations ne s'y trouvent pas, on indique le prénom de l'imprimeur dans un élément `<forename>` et son nom de famille dans un élément `<surname>`.

Exemple :

```

<titleStmt>
    <title type="standard" ref="isni:0000000121202363">
        <forename>Gervais</forename>
        <surname>Alliot</surname>
    </title>
</titleStmt>

```

fileDesc et sourceDesc

Dans le `<sourceDesc>` et son sous-élément `<listBibl>`, on indique dans un ou des éléments `<bibl>` le ou les identifiants de l'imprimeur, s'il en a. Chaque `<bibl>` correspond

à un identifiant, et celui-ci est à renseigner en valeur de son attribut `@source` sous forme de lien.

Exemple :

```
<sourceDesc>
  <listBibl>
    <bibl source="http://data.bnf.fr/ark:/12148/cb12229635p#foaf:Person"/>
    <bibl source="http://viaf.org/viaf/224751101"/>
    <bibl source="http://isni.org/isni/0000000362335140"/>
  </listBibl>
</sourceDesc>
```

profileDesc, particDesc, listPerson

Dans ces trois éléments on trouve l'élément `<person>`, dans lequel peuvent être renseignées toutes les informations complémentaires de l'imprimeur.

- le `<persName @type="standard">` doit contenir la forme normalisée du nom (Nom, Prénom)
- le `<persName @type="alternative">` peut contenir les autres formes du nom de l'imprimeur si elles existent. Il peut être répété autant de fois qu'il y a de formes du nom (on crée ainsi un nouveau `<persName>` de ce type pour chacune des formes du nom)

Exemple :

```
<persName type="standard">Belley, Jacques</persName>
<persName type="alternative">'Bellay, Jacques</persName>
<persName type="alternative">'Barlay, Jacques</persName>
<persName type="alternative">'Beley, Jacques</persName>
<persName type="alternative">'Bellé, Jacques</persName>
<persName type="alternative">'Berlay, Jacques'</persName>
```

- L'élément `<sex>` doit contenir le genre de l'imprimeur, qui doit être renseigné sous la forme *M* ou *F*.
- Les dates de naissance et de mort sont à indiquer respectivement dans des éléments `<birth>` et `<death>`, en tant que valeur de l'attribut `@when` et non en texte de l'élément (la balise doit rester auto-fermante). Si la ou les dates sont connues et indiquées en valeur de cet attribut, on changera alors aussi la valeur de l'attribut `@cert` en *high*. Si la ou les dates sont inconnues, on conservera la forme par défaut.

Exemple :

```
<sex>M</sex>
<birth when="1624" cert="high"/>
<death when="1655" cert="high"/>
```

- L'élément `<occupation>` contient par défaut la valeur "Imprimeur-libraire" et n'est pas à modifier.
- L'élément `<residence>` contient les éléments d'adresse de l'imprimeur. Si un élément d'adresse est connu, on change sa valeur en *high*. Dans le `<address>` du `<residence>`, on indique l'adresse principale de l'imprimeur. L'enseigne, si elle est connue, est à indiquer dans `<objectName>`. La rue et les indications d'emplacement, s'il y en a, sont à indiquer dans `<street>`. La ville, si elle est connue, est à indiquer dans `<settlement>`. Cet élément doit aussi prendre pour attribut `@ref` le geonames de la ville, sous la forme *geonames :numéro*. Le pays, par défaut la France, est à indiquer dans l'élément `<country>`.

Exemple :

```
<residence cert="high">
  <address>
    <objectName>Au pied de biche</objectName>
    <street>rue Saint-Jacques ; proche des Jacobins</street>
    <settlement ref="geonames:2988507">Paris</settlement>
    <country>France</country>
  </address>
</residence>
```

- Les informations sur la vie de l'imprimeur sont à renseigner dans l'élément `<listEvent>`. Les informations sont divisées selon leur source, par exemple *IdRef* ou *Antonomaz*. Pour chaque type d'informations, on crée un élément `<event>`. Dans celui-ci, on crée un élément `<label>` qui contiendra les informations. Celui-ci prend pour attributs `@type` avec pour valeur *bio* (pour indiquer qu'il s'agit d'éléments biographiques), et `@source` qui prend pour valeur la source des informations.

Exemple :

```
<listEvent>
  <event>
    <label type="bio" source="IdRef">'Imprimeur du roi à Toul
      (Meurthe-et-Moselle). Activité attestée en 1647.
      Associé avec Jean Laurent.'<</label>
  </event>
  <event>
```

```

    <label type="bio" source="Antonomaz">imprimeur du roi à Toul.</label>
  </event>
</listEvent>

```

Modifications

Les modifications effectuées sur la fiche sont à indiquer à deux endroits :

1. dans le `<fileDesc>`, dans son sous-élément `<respStmt>`. Les initiales de la personnes sont à indiquer en valeur de son attribut `@xml:id`, et son numéro ORCID, s'il existe, en valeur de l'attribut `@ref`. Son nom complet doit être indiqué dans le sous-élément `<name>` et son rôle dans le deuxième sous-élément `<resp>`.
2. dans les éléments `<revisionDesc>` et `<listChange>`, avec un sous-élément `<change>` pour chaque modification. Cet élément prend en attribut `@when` la date de modification (sous la forme AAAA-MM-JJ) et avec l'attribut `@who` qui prend en valeur un `"#"` devant les initiales de la personne.

Exemple :

```

<!-- [...] -->
<editionStmt>
  <edition/>
  <respStmt xml:id="ZC" ref="orcid:0000-0002-3327-3967">
    <name>Zoé Cappe</name>
    <resp>Créateur de la fiche</resp>
  </respStmt>
</editionStmt>
<!-- [...] -->
<revisionDesc>
  <listChange>
    <change when="2022-07-10" who="#ZC">Création de la fiche</change>
  </listChange>
</revisionDesc>

```

Ajout de mazarinades dans les fiches imprimeurs

Les mazarinades sont à ajouter des des éléments `<TEI>`, sous le `<teiHeader>` contenant les informations sur l'imprimeur, à l'intérieur du `<teiCorpus>`.

Tous les imprimeurs ont au moins un élément `<TEI>` existant, soit parce que des mazarinades leur sont déjà associées, soit parce qu'il n'y en a pas. Dans le second cas, le `<TEI>` est alors vide, et se présente sous la forme suivante :

```

<teiCorpus>
  <!-- [...] -->
  <TEI corresp="none">
    <teiHeader>
      <fileDesc>
        <titleStmt>
          <title/>
          <author ref="isni:0000"/>
        </titleStmt>
        <extent>
          <measure unit="pages" quantity="0"/>
        </extent>
        <publicationStmt>
          <publisher
            ref="https://github.com/Antonomaz">Antonomaz</publisher>
          <availability status="restricted" n="cc-by">
            <licence
              target="https://creativecommons.org/licenses/by/4.0"/>
          </availability>
        </publicationStmt>
        <sourceDesc>
          <msDesc>
            <msIdentifier>
              <repository>Sans lieu</repository>
              <idno source="None"/>
            </msIdentifier>
            <physDesc>
              <decoDesc>
                <decoNote/>
              </decoDesc>
            </physDesc>
          </msDesc>
        </sourceDesc>
      </fileDesc>
    </teiHeader>
    <text>
      <body>
        <div/>
      </body>
    </text>
  </TEI>

```

```

    </text>
  </TEI>
</teiCorpus>

```

Pour ajouter une nouvelle mazarinade à la fiche d'un imprimeur, on commence par indiquer le nom du fichier de la mazarinade dans l'attribut @corresp de l'élément <TEI>, au lieu du *none* qui est renseigné par défaut. (*par exemple* : <TEI corresp="Moreau1975_GBOOKS.xml">

Ensuite, seuls les éléments du <fileDesc> sont à compléter.

Le titleStmt

Dans le <titleStmt>, on indique à la fois le titre de la mazarinade et son ou ses auteurs.

Le titre est dans un élément <title>, et chaque auteur est à indiquer dans un élément <author> distinct, avec en attribut @ref son identifiant (ISNI ou VIAF), sous la même forme que pour les imprimeurs dans leur <title>. Si l'auteur n'a pas d'identifiant, on l'indique sous la forme @ref="isni:0000". Si l'auteur est inconnu, la balise <author> sera alors auto-fermante avec un @ref="0000".

Exemples :

```

<titleStmt>
  <title>Consultation et ordonnance des médecins de l'État pour la purgation de la
  France malade, par le sieur Du Teil.</title>
  <author ref="isni:0000000054900937">Jean Du Teil</author>
</titleStmt>

```

```

<titleStmt>
  <title>Prediction où se voit comme le roy Charles II. Roy de la Grand' Bretagne
  doit estre remis aux royaumes d'Angleterre, Escosse & Irlande apres la
  mort de son pere. Avec la conference du feu Roy & le docteur Henderson
  Escossois touchant le gouvernement de l'Eglise anglicane. Ensemble diverses
  pieces de quoy le contenu est en la page suivante. Le tout en suite du
  portrait royal.</title>
  <author ref="isni:0000000118675506">Charles II</author>
  <author ref="isni:0000000123209622">Charles I</author>
  <author ref="isni:0000">Alexander Henderson</author>
  <author ref="isni:0000000081702805">John Suckling</author>
  <author ref="isni:0000000062998962">Paul Grebner</author>
</titleStmt>

```

```

<titleStmt>
  <title>Justes (les) complaints des bourgeois de Paris, adressées à messieurs du

```

```
Parlement.</title>
<author ref="isni:0000"/>
</titleStmt>
```

L'extent

L'élément `<extent>` sert à donner des informations sur les particularités physiques des documents. Les informations qui nous intéressent ici sont le nombre de pages et la présence ou l'absence d'une marque d'imprimeur.

En sous-élément de `<extent>`, il faut donc deux éléments `<measure>` :

- le premier prend pour premier attribut `@unit="pages"`, et pour second attribut `@quantity`, avec comme valeur le nombre de pages de la mazarinade ;
- le deuxième prend pour premier attribut `@unit="decoration"`, et pour second attribut `@quantity`, avec pour valeur le nombre de marques d'imprimeurs dans la mazarinade (donc 0 s'il n'y en a pas).

Exemples :

```
<extent>
  <measure unit="pages" quantity="8"/>
  <measure unit="decoration" quantity="1"/>
</extent>
```

```
<extent>
  <measure unit="pages" quantity="4"/>
  <measure unit="decoration" quantity="0"/>
</extent>
```

sourceDesc, msDesc et msIdentifier

Dans le `<msIdentifier>`, deux éléments nous intéressent : le `<repository>` et le `<idno>`.

- `<repository>` doit contenir le lieu (la ville) et le nom de l'institution de conservation de la mazarinade dont on a la numérisation (*par exemple : Paris, Bibliothèque nationale de France*)
- `<idno>` est une balise auto-fermante, il contient dans son attribut `@source` le lien vers la numérisation de ladite mazarinade (*par exemple : source="https://books.google.fr/books?id=uktKAAAACAAJ"*).

Annexe D

Exemple d'une fiche imprimeur

Une partie des éléments <TEI> n'ont pas été reproduits ici en raison de leur nombre important qui n'apportait pas d'informations supplémentaires.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model
  href="https://raw.githubusercontent.com/Imprinters/ODD/main/schema/ODD_printers.
<?xml-model
  href="https://raw.githubusercontent.com/Imprinters/ODD/main/schema/ODD_printers.
  type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"?>
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0" xml:id="IL_FeugeRobert">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title type="standard" ref="isni:000000043524489X">
          <forename>Robert</forename>
          <surname>Feugé</surname>
        </title>
      </titleStmt>
      <editionStmt>
        <edition/>
        <respStmt xml:id="ZC" ref="orcid:0000-0002-3327-3967">
          <name>Zoé Cappe</name>
          <resp>Créateur de la fiche</resp>
        </respStmt>
      </editionStmt>
      <publicationStmt>
        <publisher ref="https://github.com/Antonomaz" xml:id="Antonomaz">
          >Antonomaz</publisher>
        </publicationStmt>
      </teiHeader>
    </fileDesc>
  </teiCorpus>
```

```

    <availability status="restricted" n="cc-by">
      <licence target="https://creativecommons.org/licenses/by/4.0"/>
    </availability>
  </publicationStmt>
  <sourceDesc>
    <listBibl>
      <bibl
        source="http://data.bnf.fr/ark:/12148/cb15915877m#foaf:Person
      <bibl source="http://viaf.org/viaf/228063889"/>
      <bibl source="http://isni.org/isni/000000043524489X"/>
      <bibl source="http://www.idref.fr/069012245/id"/>
    </listBibl>
  </sourceDesc>
</fileDesc>
<profileDesc>
  <particDesc>
    <listPerson>
      <person>
        <persName type="standard">Feugé, Robert</persName>
        <persName type="alternative">Feugé II, Robert</persName>
        <sex>M</sex>
        <birth when="1628-02-20" cert="high"/>
        <death when="1700" cert="low"/>
        <occupation>Imprimeur-libraire</occupation>
        <residence cert="high">
          <address>
            <objectName>A l'image Saint-Sébastien</objectName>
            <street>proche le Puits-Certain</street>
            <settlement ref="geonames:2988507">Paris</settlement>
            <country>France</country>
          </address>
        </residence>
      <listEvent>
        <event>
          <label type="bio" source="IdRef">Paris,
            13 septembre 1644-1671?. A l'image
            Saint-Sébastien, près le Puits Certain</label>
        </event>
      <event>

```

```

        <label type="bio" source="IdRef">Imprimeur-libraire.
            Fils de Robert I Feugé (15..-1642). Gendre de
            Robert Mansion</label>
    </event>
</listEvent>
</person>
</listPerson>
</particDesc>
</profileDesc>
<revisionDesc>
    <listChange>
        <change when="2022-07-10" who="#ZC">Création de la fiche</change>
    </listChange>
</revisionDesc>
</teiHeader>
<TEI corresp="Moreau3395_GBBOOKS.xml">
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title>Réponse de la reine d'Angleterre au prince de Galles,
                    son fils (Lettre véritable du prince de Galles,
                    etc.).</title>
                <author ref="viaf:52930938">Henriette-Marie,
                    reine d'Angleterre</author>
            </titleStmt>
            <extent>
                <measure unit="pages" quantity="7"/>
                <measure unit="decoration" quantity="0"/>
            </extent>
            <publicationStmt>
                <publisher
                    ref="https://github.com/Antonomaz">Antonomaz</publisher>
                <availability status="restricted" n="cc-by">
                    <licence
                        target="https://creativecommons.org/licenses/by/4.0"/>
                </availability>
            </publicationStmt>
            <sourceDesc>
                <msDesc>

```

```

    <msIdentifier>
      <repository>Vienne, Bibliothèque
        nationale d'Autriche</repository>
      <idno
        source="https://books.google.fr/books?id=-kJKAAAaAAJ
    </msIdentifier>
    <physDesc>
      <decoDesc>
        <decoNote/>
      </decoDesc>
    </physDesc>
  </msDesc>
</sourceDesc>
</fileDesc>
</teiHeader>
<text>
  <body>
    <div/>
  </body>
</text>
</TEI>
<TEI corresp="Moreau3305_GB00KS.xml">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Remontrance de la reine de Pologne à la reine de France
          touchant le déplaisir qu'elle a devoir combattre les
          Polonois contre les François.</title>
        <author ref="isni:0000000109064340">Louise-Marie
          de Gonzague</author>
      </titleStmt>
      <extent>
        <measure unit="pages" quantity="7"/>
        <measure unit="decoration" quantity="0"/>
      </extent>
      <publicationStmt>
        <publisher
          ref="https://github.com/Antonomaz">Antonomaz</publisher>
        <availability status="restricted" n="cc-by">

```

```

        <licence
            target="https://creativecommons.org/licenses/by/4.0"/>
        </availability>
    </publicationStmt>
    <sourceDesc>
        <msDesc>
            <msIdentifier>
                <repository>Lyon, Bibliothèque municipale</repository>
                <idno
                    source="https://books.google.fr/books?id=jPM8GH4zcSAC
            </msIdentifier>
            <physDesc>
                <decoDesc>
                    <decoNote/>
                </decoDesc>
            </physDesc>
        </msDesc>
    </sourceDesc>
</fileDesc>
</teiHeader>
<text>
    <body>
        <div/>
    </body>
</text>
</TEI>
<!-- [...] -->
</teiCorpus>

```


Bibliographie

Contexte et sources historiques

- BÉLY, LUCIEN, *La France moderne : 1498-1789*, PUF, 2e éd, dir. Joël Cornette, Paris, 2013.
- BERTRAND, MARION, *Les gazetiers frondeurs : les mazarinades et la presse d'information en 1652*, mémoire de master, Enssib, dir. Dominique Varry, Lyon, 2014.
- BIBLIOTHÈQUE MAZARINE, *Bibliographie des mazarinades*, URL : <https://mazarinades.bibliotheque-mazarine.fr/> (visité le 20/08/2022).
- CARRIER, HUBERT, *La presse de la Fronde (1648-1653) : les mazarinades. La conquête de l'opinion*. Droz, vol. I, Genève, 1989.
- *La presse de la Fronde (1648-1653) : les mazarinades. Les hommes du livre*, Droz, vol. II, Genève, 1991.
- DRÉVILLON, HERVÉ, *1629-1715 : les rois absolus*, Gallimard, Paris, 2021.
- ÉQUIPE RIM, *Projet Mazarinades*, URL : <http://mazarinades.org/> (visité le 20/08/2022).
- HAFFEMAYER, STÉPHANE, REBOLLAR, PATRICK et SORDET, YANN, “Introduction”, *Mazarinades, nouvelles approches*, XII (2016).
- JOUHAUD, CHRISTIAN, *Mazarinades : la Fronde des mots*, Aubier, Paris, 2009.
- MOREAU, CÉLESTIN, *Bibliographie des Mazarinades*, Renouard, 3 t., Paris, 1850.
- PERNOT, MICHEL, *La Fronde : 1648-1653*, Éditions Tallandier, Paris, 2019.
- PONCET, OLIVIER, *Mazarin : l'art de gouverner*, Perrin : Bibliothèque nationale de France, Paris, 2021.
- SORDET, YANN, “Mazarine et mazarinades : la plus grande bibliothèque d'Europe dans la tourmente de la Fronde”, *Dix-septième siècle*, n°284-3 (2019), p. 459, DOI : 10.3917/dss.193.0459.
- *Histoire du livre et de l'édition : production et circulation, formes et mutations*, Albin Michel, Paris, 2021.
- VELLET, CHRISTOPHE, “Les mazarinades à l'affiche ? Armand d'Artois et la collection de la Bibliothèque Mazarine”, *Mazarinades, nouvelles approches*-12 (2016), p. 51-67, URL : <https://revues.droz.org/index.php/HCL/article/view/2442> (visité le 15/08/2022).

WALSBY, MALCOLM, *L'imprimé en Europe occidentale, 1470-1680*, Presses universitaires de Rennes, Rennes, 2020.

Numérique et humanités numériques

ABES, *Documentation du triple store d'IdRef*, URL : <https://documentation.abes.fr/aideidrefdata/index.html> (visité le 19/08/2022).

— *Les notices d'autorité en RDF*, URL : <https://documentation.abes.fr/aideidrefrdf/index.html> (visité le 19/08/2022).

ABIVEN, KARINE, BARTZ, ALEXANDRE, LEJEUNE, GAËL et TANGUY, JEAN-BAPTISTE, “Vers une collection numérique des libelles de la Fronde, ou comment relier les mazarinades”, *Le Verger - bouquet XXIII* (, mai 2022).

ABIVEN, KARINE et LEJEUNE, GAËL, “Analyse automatique de documents anciens : tirer parti d'un corpus incomplet, hétérogène et bruité”, *Information Retrieval, Document and Semantic Web*, 19–1 (25 févr. 2019).

ABIVEN, KARINE, LEJEUNE, GAËL et TANGUY, JEAN-BAPTISTE, “Exploiter un corpus de données textuelles sans post-traitement : l'écriture burlesque de la Fronde”, *Humanités numériques*–4 (1^{er} déc. 2021), DOI : 10.4000/revuehn.2355.

BARTZ, ALEXANDRE et JAHAN, CLAIRE, *ODD - Imprinters*, URL : <https://github.com/Imprinters/ODD> (visité le 17/08/2022).

BAUMANN, SYD, “tei_customization : A TEI customization for writing TEI customizations”, *Proceedings of the Text Encoding Initiative Conference and Members Meeting* (, 2017), URL : https://hcmc.uvic.ca/tei2017/abstracts/t_110_bauman_teicustomization.html (visité le 22/08/2022).

BURNARD, LOU, “What is TEI Conformance, and Why Should You Care?”, 12 (2019), URL : <https://doi.org/10.4000/jtei.1777>.

BURNARD, LOU et BURGHART, MARJORIE, *Qu'est-ce que la Text Encoding Initiative ?*, Marseille, 2015, URL : <http://books.openedition.org/oep/1237> (visité le 17/08/2022).

BURNARD, LOU et RAHTZ, SEBASTIAN, “Relax NG with Son of ODD”, *Proceedings of Extreme Markup Languages 2000* (, 2016), URL : <https://ora.ox.ac.uk/objects/pubs:394056>.

CAVALIÉ, ÉTIENNE, *L'indexation matière en transition - De la réforme de Rameau à l'indexation automatique*, Paris, 2019.

CHIFFOLEAU, FLORIANE, “Publication of my digital edition – Working with TEI Publisher” (, 2020), URL : <https://digitalintellectuals.hypotheses.org/3912> (visité le 19/08/2022).

CONSORTIUM TEI, <TEI>, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-TEI.html> (visité le 22/08/2022).

- *<teiCorpus>*, URL : <https://tei-c.org/release/doc/tei-p5-doc/fr/html/ref-teiCorpus.html> (visité le 22/08/2022).
 - *Documentation Elements*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/TD.html> (visité le 22/08/2022).
 - *Elements Available in ALL TEI Documents*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/C0.html> (visité le 22/08/2022).
 - *Linking, Segmentation, and Alignment*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/SA.html> (visité le 22/08/2022).
 - *TEI P5 : Guidelines for Electronic Text Encoding and Interchange*, URL : <https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html> (visité le 17/08/2022).
 - *The TEI Header*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/HD.html> (visité le 22/08/2022).
 - *The TEI Infrastructure*, URL : <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/ST.html> (visité le 22/08/2022).
- CORBIÈRES, CAROLINE, *Du catalogue au fichier TEI, création d'un workflow pour encoder automatiquement en XML-TEI des catalogues d'exposition*, mémoire de master TNAH, dir. Thibault Clérie et Béatrice Joyeux-Brunel, Paris, 2020.
- CSV - *Lecture et écriture de fichiers CSV*, URL : <https://docs.python.org/fr/3/library/csv.html> (visité le 22/08/2022).
- CURÉ, OLIVIER et BLIN, GUILLAUME, *RDF Database Systems. Triples Storage and SPARQL Query Processing*, Amsterdam, 2014.
- E-DITIONES, *À propos*, URL : <https://editiones.hypotheses.org/a-propos> (visité le 18/08/2022).
- *Carnet de recherche sur la philologie numérique et la littérature du XVIIIe s.* URL : <https://editiones.hypotheses.org/> (visité le 21/08/2022).
- E-DITIONES, URL : <https://e-editiones.org/> (visité le 22/08/2022).
- E-DITIONES, *TEI Publisher - Application*, URL : <https://github.com/eeditiones/tei-publisher-app> (visité le 22/08/2022).
- FAURE, MARGAUX, *Rassembler et valoriser les mazarinades. Traiter les numérisations et les métadonnées d'une collection d'imprimés du XVIIIe siècle pour sa mise à disposition numérique*, mémoire de master « Technologies numériques appliquées à l'histoire », dir. Thibault Clérie, École nationale des chartes, 2022.
- GABARIK, RADOVAN, "Processing XML Text with Python and ElementTree - a Practical Experience", dans Bratislava, Slovakia, 2006.
- GABAY, SIMON, "E-editiones, 17th c. French sources", dans Neuchâtel, Switzerland, 2018, URL : <https://hal.archives-ouvertes.fr/hal-02388415> (visité le 19/08/2022).

- GABAY, SIMON, BARTZ, ALEXANDRE et DEGUIN, YOHANN, “CORPUS17 : a philological corpus for 17th c. French”, dans Hammamet, Tunisia, 2020, URL : <https://dx.doi.org/10.1145/3423603.3424002> (visité le 22/08/2022).
- GABAY, SIMON, CLÉRICE, THIBAUT et REUL, CHRISTIAN, *OCR17*, URL : <https://github.com/e-ditiones/OCR17/> (visité le 23/08/2022).
- GESLOT, JEAN-CHARLES et REBOLLEDO-DHUIN, VIERA, “Du livre imprimé au Web sémantique : le projet du Dictionnaire des éditeurs français du xixe siècle”, *Humanités numériques-2* (1^{er} juin 2020), DOI : 10.4000/revuehn.426.
- ICHIMARU, TADAKO, “Enjeux de la numérisation des mazarinades”, *Histoire et civilisation du livre-12* (2016), 12, p. 77-89.
- KHEMAKHEM, MOHAMED, ROMARY, LAURENT, GABAY, SIMON, BOHBOT, HERVÉ, FRONTINI, FRANCESCA, LUXARDO, GIANCARLO et ET AL., “Automatically Encoding Encyclopedic-like Resources in TEI”, dans Tokyo, Japan, 2018, URL : <https://hal.inria.fr/hal-01819505> (visité le 19/08/2022).
- LE PROVOST, ALINE, NICOLAS, YANN et MISTRAL, FRANÇOIS, “Linking to IdRef. A Read/Write Network of Authority Data for/by Humans and Machine”, dans Wrocław, Poland, 2017, URL : <https://hal.archives-ouvertes.fr/hal-02139256> (visité le 19/08/2022).
- lxml - Processing XML and HTML with Python*, URL : <https://lxml.de/> (visité le 17/08/2022).
- PESSAC, JEAN-BAPTISTE, “Introduction aux ontologies : RDFS, OWL, inférences”, dans En ligne, France, 2020, URL : <https://hal.archives-ouvertes.fr/hal-03199699> (visité le 19/08/2022).
- RELAX NG*, URL : <https://relaxng.org/> (visité le 29/08/2022).
- RICHTER, FRIEDERIKE et STEGMANN, BEEKE, “Text encoding using XML”, dans *Ch.2 of The Menota handbook : Guidelines for the electronic encoding of Medieval Nordic primary sources*, Version 3.0, Bergen : Medieval Nordic Text Archive, 2019, URL : https://www.menota.org/HB3_ch2.xml (visité le 22/08/2022).
- RONDEAU DU NOYER, LUCIE, *Encoder automatiquement des catalogues en XML/TEI, principes, évaluations et application à la revue des autographes de la librairie Charavay*, mémoire de master TNAH, dir. Thibault Clérice et Simon Gabay, Paris, 2019.
- SCHULTZ, ÉMILIE et BUSSONNIER, MATTHIAS, *Python pour les SHS : introduction à la programmation pour le traitement de données*, Rennes, 2021.
- SPARQL Endpoint interface to Python*, URL : <https://sparqlwrapper.readthedocs.io/en/latest/> (visité le 22/08/2022).
- TEI Publisher - Documentation*, URL : <https://teipublisher.com/exist/apps/teipublisher/doc/documentation.xml?odd=docbook.odd> (visité le 22/08/2022).
- VANHOUTTE, E., “An Introduction to the TEI and the TEI Consortium”, *Literary and Linguistic Computing*, 19-1 (1^{er} avr. 2004), p. 9-16, DOI : 10.1093/llc/19.1.9.

VIGLIANTI, RAFFAELE, “One Document Does-it-all (ODD) : a language for documentation, schema generation, and customization from the Text Encoding Initiative”, dans Washington, DC, DOI : 10.4242/BalisageVol24.Viglianti01.

Why Is Schematron Different ?, URL : https://schematron.com/home/whyisschematrondifferent_.html (visité le 19/08/2022).

WITTERN, C., CIULA, A. et TUOHY, C., “The making of TEI P5”, *Literary and Linguistic Computing*, 24-3 (1^{er} sept. 2009), p. 281-296, DOI : 10.1093/l1c/fqp017.

Table des matières

Résumé	i
Remerciements	iii
Introduction	v
I Contexte et projet	1
1 La période historique du projet	3
1.1 La Fronde	3
1.2 Les mazarinades	4
1.3 Les imprimeurs	5
2 Le projet Antonomaz	9
2.1 Réalisations techniques	9
2.2 Projets similaires	10
2.2.1 La Bibliographie des mazarinades	10
2.2.2 Le Projet Mazarinades	10
II Les « repères »	13
3 Présentation des « repères »	15
3.1 Typologie des documents	15
3.2 Un encodage en XML-TEI	16
3.3 Cahier des charges	18
4 Réalisation : ODD et choix d’encodage	21
4.1 Choix d’encodage des « repères »	21
4.1.1 Les métadonnées : le <code>teiHeader</code>	21
4.1.2 Les images	24
4.1.3 Le corps de texte	24

4.1.4	Liens et éléments bibliographiques : le back	26
4.2	Rédaction de l'ODD	28
5	Vers TEI Publisher : Présentation et premiers pas	33
III	Base imprimeurs	37
6	Éléments d'introduction à la base des imprimeurs	39
6.1	La constitution de la première base	39
6.2	Référentiels et identifiants pérennes : ISNI et VIAF	40
6.3	Cahier des charges	40
7	Enrichissement de la base	43
7.1	SPARQL et IdRef	43
7.2	Réalisation de l'intégration des données	45
8	Encodage de la base en XML-TEI	49
8.1	L'ODD Imprinters	49
8.2	Adaptation de l'encodage	50
8.2.1	La structure des « fiches imprimeurs »	50
8.2.2	Le <teiHeader>	52
8.2.3	Les <TEI>	55
8.3	Automatisation de l'encodage : Python et lxml	58
	Conclusion	67
A	Manuel d'encodage des « repères »	69
	Manuel d'encodage des notices	69
	L'élément racine "TEI"	69
	Le "teiHeader"	70
	Le "text"	73
B	Exemple de « repère »	79
C	Manuel d'encodage des imprimeurs	87
	Manuel d'encodage des imprimeurs	87
	Ajout d'informations sur les imprimeurs	87
	Ajout de mazarinades dans les fiches imprimeurs	92
D	Exemple d'une fiche imprimeur	97

Bibliographie	103
Contexte et sources historiques	103
Numérique et humanités numériques	104