

ÉCOLE NATIONALE DES CHARTES

Josselin Morvan

licencié en droit

licencié en histoire de l'art

Modélisation XML-TEI, XSLT et publication web

La correspondance d'Armand Horel, un marin durant la Grande Guerre

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

2015

Résumé

Ce mémoire a été réalisé à la suite d'un stage de trois mois et demi effectué à la bibliothèque de documentation internationale contemporaine (BDIC). Cet établissement spécialisé dans l'histoire du ^{xx}^e et ^{xxi}^e siècles, est impliqué depuis sa création dans le développement de nouveaux champs de recherches. À la fois bibliothèque, centre d'archives et musée, elle ne cherche pas seulement à mettre en lumière des documents inexploités, mais à impulser une nouvelle réflexion quant à leur exploitation, afin de faire émerger de nouvelles problématiques. La BDIC a depuis quelques années intégré l'aspect numérique dans sa démarche, en s'engageant aux côtés de la Maison de l'archéologie et de l'ethnologie (MAE), dans le Labex, *les passés dans le présent* qui s'« attache plus spécifiquement à comprendre les médiations de l'histoire à l'ère du numérique, les politiques de la mémoire et les appropriations sociales du passé en amont et en aval des politiques patrimoniales ».

« Bibliothèque de et pour la recherche », la BDIC est soucieuse d'accompagner ses usagers dans leurs prospections, en leur offrant notamment des outils qu'ils pourront mettre à profit dans leurs travaux. Le contexte de commémoration du centenaire de la Grande Guerre et l'intérêt grandissant du public pour les ressources liées au conflit, étaient propices au développement d'un modèle d'édition électronique en XML-TEI dédié aux correspondances. Cette modélisation a été réalisée avec comme objectifs principaux, de respecter les « sources » au maximum lors de l'encodage, de proposer une version normalisée du texte et de permettre un double accès chrono-géographique aux documents. Ce modèle XML-TEI est accompagné d'une solution de publication web, mise en place à partir de feuilles de transformation XSL de deux types ; XML vers HTML pour la publication des lettres et XML vers XML pour une utilisation comme base de données de certaines informations. La correspondance d'Armand Horel, conservée à la BDIC, a permis de mettre en œuvre ces solutions d'édition et de publication électroniques. Le développement d'un site internet spécifique à ce fonds a nécessité l'écriture de feuilles de style CSS et de scripts JavaScript liés à la mise en forme et à l'ergonomie des pages web.

Mots-clés : XML ; TEI ; XSLT ; édition électronique ; publication électronique ; correspondance ; géolocalisation.

Informations bibliographiques : Josselin Morvan, *Modélisation XML-TEI, XSLT et publication web : La correspondance d'Armand Horel, un marin durant la Grande Guerre*, mémoire de master « Technologies numériques appliquées à l'histoire », dir. [Frédérique Joannic-Seta], École nationale des chartes, 2015.

Avertissement : ce mémoire est accompagné d'un appendice numérique consultable à l'adresse suivante : github.com/sardinecan/TEIcorrespondances_

iv

horel

Remerciements

LA RÉALISATION DE CE MÉMOIRE a été possible grâce au concours de plusieurs personnes à qui je souhaiterais témoigner ma reconnaissance.

Tout d'abord, je voudrais adresser toute ma gratitude à ma directrice de stage, Frédérique Joannic-Seta, directrice adjointe de la BDIC, pour avoir rendu ce stage possible et pour m'avoir encouragé à formuler des propositions tout au long de cette entreprise.

J'aimerais également remercier Jean-Baptiste Camps, tuteur pédagogique et responsable du master « technologies numériques appliquées à l'histoire » pour son enseignement et ses conseils avisés à propos du langage XML.

Je me dois également de remercier Emmanuel Chateau qui, le premier, m'a apporté son soutien et orienté vers les humanités numériques. Puis, que Camille Desenclos, Patrick Latour et Lionel Maurel acceptent ici ma gratitude pour leur soutien et leurs précieux conseils.

Enfin, que mes parents et amis acceptent ici mes remerciements pour leur support moral, intellectuel et leur écoute tout au long de ma démarche. Sans oublier Sara à qui je dois tant, pour sa confiance, sa patience et son soutien inestimable.

Introduction

La bibliothèque de documentation internationale contemporaine (BDIC) est un établissement spécialisé dans l'histoire contemporaine et les relations internationales des ^{xx}e et ^{xxi}e siècle. À la fois bibliothèque, centre d'archives et musée, elle a été créée en janvier 1918 à l'initiative d'un couple d'industriels, les époux Leblanc. Ce couple décida de récolter dès le début de la guerre, tous les documents qui leur permettraient de comprendre les origines du conflit mondial. Elle se définit depuis toujours comme une « bibliothèque de l'événement »¹, dont la vocation est de collecter tous les matériaux pouvant servir à l'interprétation et l'écriture de l'histoire contemporaine.

« Un laboratoire de et pour la recherche »²

Depuis sa création, la bibliothèque de documentation internationale contemporaine s'est vue attribuer la double mission d'être à la fois « une œuvre d'éducation populaire — et de diffusion des résultats de la recherche — » et un « laboratoire d'histoire fondé sur les méthodes de l'érudition scientifique »³. Elle a su apporter un regard neuf sur les fondements des sources de l'histoire. Elle a intégré dans ses collections, des documents écrits et iconographiques qui étaient encore ignorés par les chercheurs. Cette réflexion des historiens révèle la cohabitation dans ses fonds, de sources primaires et secondaires.

Acteur de la recherche, la BDIC a pris conscience des enjeux du numérique en s'engageant notamment dans le Labex *les passés dans le présent* dont elle est un des piliers avec la Maison de l'archéologie et de l'ethnologie (MAE). Ce laboratoire

1. Jean-Luc Doménach, Jaap Kloosterman, Jean-Louis Crémieux-Brilhac, Hermann Schäfer, Thérèse Blondet-Bish, Laure Barbizet-Namer, Aldo Battaglia, Florence Poncé, Anne-Marie Blanchenay, Catherine Penin, *et al.*, « Réflexion sur les activités de la BDIC », *Matériaux pour l'histoire de notre temps*, 49 (1998), p. 10–33, URL : http://www.persee.fr/doc/mat_0769-3206_1998_num_49_1_410678.

2. Valérie Tesnière, « Un laboratoire de et pour la recherche », *Journal de la BDIC*, 38 (avr. 2015), p. 1, URL : http://www.bdic.fr/images/stories/BDIC/journal/BDIC_Journal_38_web.pdf.

3. *Recherche en histoire contemporaine*, BDIC, URL : <http://bdic.fr/recherche-en-histoire-contemporaine> (visité le 26/08/2015).

d'excellence s'attache « à comprendre les médiations de l'histoire à l'ère du numérique, les politiques de la mémoire et les appropriations sociales du passé en amont et en aval des politiques patrimoniales »⁴. Pôle associé de la Bibliothèque nationale de France, elle mène également une importante politique de numérisation de ses collections patrimoniales.

Un projet d'édition électronique de correspondances

Attachée à sa mission d'accompagnement des chercheurs, la BDIC est soucieuse de leur proposer des outils qu'ils pourront mettre à profit dans leur travaux. Avec plus de soixante mille documents numérisés, sa bibliothèque numérique est l'une des plus importantes de l'enseignement supérieur⁵. Ce type d'infrastructure représente un véritable bond en avant pour la recherche, particulièrement lorsqu'à l'instar de la BDIC, les institutions patrimoniales décident d'offrir leurs documents sous licence ouverte.

Poursuivant cette démarche, la bibliothèque de documentation internationale contemporaine a souhaité se doter d'un modèle d'édition en XML-TEI dédié aux correspondances de poilus. Les ensembles épistolaires sont des sources importantes, demeurées paradoxalement, peu ou mal mises en valeur. Si elles sont susceptibles d'intéresser un grand nombre de chercheurs dans des domaines variés — histoire, philologie, biographie, linguistique etc. — il est dommage de constater qu'elles sont encore trop souvent reléguées à un rôle documentaire de second plan et non exploitées comme les sources primaires qu'elles représentent⁶. En effet, ces matériaux ne sont pas toujours des plus aisés à manipuler ; ensembles hétérogènes,

They are produced by (at least) two people, writing in different locations, and contain a wealth of information of various kinds.

[i]ls sont produits par (au moins) deux personnes, communiquant depuis des lieux différents, et recelant une richesse d'information de différentes sortes⁷.

Une édition de correspondances bien menée offre au plus grand nombre, des matériaux fiables, replacés dans leur contexte et qui pourront être facilement réex-

4. *Les passés dans le présent*, URL : <http://passes-present.eu/> (visité le 27/08/2015).

5. Bibliothèque numérique Argonnaute : <http://argonnaute.u-paris10.fr/>

6. Edward Vanhoutte et Ron Van den Branden, « Describing, transcribing, encoding, and editing modern correspondence material : a textbase approach », *Literary and Linguistic Computing*, 24 (4 janv. 2009), p. 77–98, URL : <http://llc.oxfordjournals.org/content/24/1/77> (visité le 27/08/2015).

7. Camille Desenclos, « Early modern correspondence : A new challenge for digital editions », dans Elena Pierazzo et Matthew Driscoll, *Digital Scholarly Editing : Theory, Practice and Future Perspectives*, OpenEdition Books, 2015.

ploités⁸. C'est là le principal enjeu auquel devait répondre ce stage ; mettre en place une modélisation XML-TEI simple, permettant à des personnes non spécialistes des nouvelles technologies — mais souhaitant toutefois les intégrer à leurs travaux —, d'encoder un corpus de correspondances de manière à ce qu'il soit valorisable, dans leur approche, mais aussi réutilisable par d'autres, s'ils décident de le partager.

Le projet a été élaboré autour de l'édition électronique de la correspondance d'Armand Horel⁹. Il a impliqué trois phases de développement : l'analyse du corpus afin de réaliser un modèle d'édition, la transcription et l'encodage d'une partie des lettres, et enfin, la réalisation d'une interface de publication électronique.

8. Christine Nougaret et Élisabeth Parinet, *L'édition critique des textes contemporains (XIXe-XXIe siècle)*, 2015.

9. Bibliothèque de documentation internationale contemporaine ; F Δ 1854/20, Don de Madame la Comtesse de la Forest, [disponible en ligne : <http://argonnaute.u-paris10.fr/resource/a011403267952LEa5xJ>].

Bibliographie

- BURGHART (Marjorie) et DUFOURNAUD (Nicole), *Édition électronique de sources : XML et Text Encoding Initiative (TEI) à l'École*, La lettre de l'École des hautes études en sciences sociale, 2011, URL : <http://lettre.ehess.fr/1530> (visité le 27/08/2015).
- BUZZETTI (Dino), « Digital Representation and the Text Model », *New Literary History* 33 (2002), URL : https://muse.jhu.edu/login?auth=0&type=summary&url=/journals/new_literary_history/v033/33.1buzzetti.html (visité le 26/08/2015).
- CHATEAU (Emmanuel), *L'édition des cours, Le modèle XML-TEI*, Cours de Desgodets, 2013, URL : <http://www.desgodets.net/edition-des-cours/model#editionModelBodyFr02> (visité le 31/08/2015).
- COCHET (François), « L'armée d'Orient, des expériences combattantes loin de Verdun », *Cahiers de la Méditerranée*, 81 (15 déc. 2010), p. 91–103, URL : <http://cdlm.revues.org/5498?lang=en> (visité le 13/07/2015).
- COOMBS (James H.), RENEAR (Allen H.) et DEROSE (Steven J.), « Markup Systems and the Future of Scholarly Text Processing », *Communications of the ACM* 30, (1987), p. 933–947, URL : <http://xml.coverpages.org/coombs.html> (visité le 30/08/2015).
- D'ANDURAIN (Julie), « Le général Gouraud, chef du corps expéditionnaire des Dardanelles en 1915 », *Revue historique des armées*, 258 (15 mars 2010), p. 46–56, URL : <http://rha.revues.org/6921> (visité le 13/07/2015).
- DEBUSSCHERE (Joke) et VAN DEN BRANDEN (Ron), *Stretching the TEI for the transcription of corpora of modern correspondence*. URL : <http://ctb.kantl.be/seminar/01-02/seminar-debusschere.htm> (visité le 27/08/2015).
- DESENCLOS (Camille), *Édition de correspondances*, URL : <http://developpements.enc.sorbonne.fr/diple/schema/correspondance> (visité le 28/05/2015).
- *L'ambassade extraordinaire des duc d'angoulême, comte de béthune et abbé de préaux vers les princes et potentats de l'empire (1620-1621)*, mémoire de master 2, Paris : École nationale des chartes 2009, 467 p.

- DESENCLOS (Camille), « Early modern correspondence : A new challenge for digital editions », dans Elena Pierazzo et Matthew Driscoll, *Digital Scholarly Editing : Theory, Practice and Future Perspectives*, OpenEdition Books, 2015.
- DOMÉNACH (Jean-Luc), KLOOSTERMAN (Jaap), CRÉMIEUX-BRILHAC (Jean-Louis), SCHÄFER (Hermann), BLONDET-BISH (Thérèse), BARBIZET-NAMER (Laure), BATTAGLIA (Aldo), PONCÉ (Florence), BLANCHENAY (Anne-Marie), PENIN (Catherine), *et al.*, « Réflexion sur les activités de la BDIC », *Matériaux pour l'histoire de notre temps*, 49 (1998), p. 10–33, URL : http://www.persee.fr/doc/mat_0769-3206_1998_num_49_1_410678.
- HAROLD (Elliott Rusty) et MEANS (W. Scott), *XML en concentré : manuel de référence*, trad. par Philippe Ensarguet et Frédéric Laurent, Paris, France, 2005.
- IDE (Nancy) et VÉRONIS (Jean), « Une application de la TEI aux industries de la Langue : le Corpus Encoding Standard », *Cahiers GUTenberg*, 24 (juin 1996), URL : <http://sites.univ-provence.fr/veronis/pdf/1996gut-corpus.pdf> (visité le 01/09/2015).
- La Bulgarie dans les collections de l'ECPAD*, Mission Centenaire 14-18, URL : <http://centenaire.org/fr/bulgarie/la-bulgarie-dans-les-collections-de-lecpad> (visité le 14/07/2015).
- Les passés dans le présent*, URL : <http://passes-present.eu/> (visité le 27/08/2015).
- NOUGARET (Christine) et PARINET (Élisabeth), *L'édition critique des textes contemporains (XIXe-XXIe siècle)*, 2015.
- POUPEAU (Gautier), « L'édition critique des sources historiques : du support papier au Web », dans *rencontres de la communauté francophone de la TEI*, Nancy, 2005, URL : <http://www.lespetitescases.net/node/1088> (visité le 03/09/2015).
- Projet corpus 14*, URL : <http://corpus14.ortolang.fr> (visité le 06/11/2015).
- Recherche en histoire contemporaine*, BDIC, URL : <http://bdic.fr/recherche-en-histoire-contemporaine> (visité le 26/08/2015).
- Repères historiques*, BDIC, URL : <http://bdic.fr/la-bdic/sous-menu-1> (visité le 26/08/2015).
- ROCHE (Jean-Michel), *Dictionnaire des bâtiments de la flotte de guerre française de Colbert à nos jours*, Toulon, France, 2005, 591 p.
- TESNIÈRE (Valérie), « Bibliothèque et recherche », *Journal de la BDIC*, 38 (avr. 2015), p. 3, URL : http://www.bdic.fr/images/stories/BDIC/journal/BDIC_Journal_38_web.pdf.
- « Un laboratoire de et pour la recherche », *Journal de la BDIC*, 38 (avr. 2015), p. 1, URL : http://www.bdic.fr/images/stories/BDIC/journal/BDIC_Journal_38_web.pdf.

- VANHOUTTE (Edward) et BRANDEN (Ron Van den), « Describing, transcribing, encoding, and editing modern correspondence material : a textbase approach », *Literary and Linguistic Computing*, 24 (4 janv. 2009), p. 77–98, URL : <http://llc.oxfordjournals.org/content/24/1/77> (visité le 27/08/2015).

Première partie

Éditer une correspondance de poilus

Chapitre 1

Mettre en valeur des sources

La vocation de ce stage ne résidait pas tant dans l'édition de la correspondance d'Armand Horel, Quartier-maitre canonnier pendant la Grande Guerre, que d'offrir à la bibliothèque de documentation internationale contemporaine un modèle d'édition électronique. Celui-ci est destiné à la diffusion auprès d'utilisateurs souhaitant intégrer le numérique à leur démarche. Toutefois, une modélisation nécessite de procéder à une analyse fine des matériaux afin de proposer des solutions cohérentes au regard des objectifs et principes éditoriaux retenus pour un tel projet. Une édition électronique est donc avant tout une édition. Il est important d'avoir cet élément en tête car si la mise en œuvre peut être différente — ne serait-ce que par le support d'édition proposé —, la démarche doit rester la même. Éditer, demande du temps et des moyens ; afin de rationaliser ces ressources, il est nécessaire de bien s'interroger sur les motivations de l'édition car elles guideront les différentes orientations du projet.

1.1 L'édition de correspondances

1.1.1 Délimitation d'un corpus

Pour un tel projet, la première question qui vient à l'esprit est de déterminer ce que l'on va éditer. Dans le cadre de ce stage, si le type des documents ciblés a été défini assez rapidement, il restait encore à qualifier un fonds qui puisse être l'objet de ce travail. Outre les intérêts historiques et techniques que nous développerons plus en détail par la suite, il a été retenu la correspondance d'Armand Horel, avec ses cent cinquante cartes postales, qui était suffisamment consistante pour donner lieu à un premier projet d'édition. Comme nous l'avons vu plus haut, éditer est chronophage et demande des ressources. Cela nécessite de bien fixer à l'avance les limites de l'ensemble à traiter sous peine de ne pouvoir mener à bien l'entreprise. Sur ce point, le numérique offre sans doute une plus grande souplesse dans la ges-

tion du projet, dans la mesure où le versement des textes édités peut se répartir plus facilement dans le temps ; une édition papier aurait peut être contraint au choix d’une édition partielle. Sur cette question l’objectif du stage n’était pas d’encoder l’intégralité de la correspondance mais environ trente pour cent du corpus. Cela semble peu mais, outre la modélisation, il fallait également développer une interface de consultation en ligne, permettant de mettre en valeur les apports de l’édition numérique ainsi que la documentation attenante au modèle d’encodage. Au final, ce sont près de cent deux cartes postales et enveloppes qui ont été traitées.

Pour une édition de correspondances, les problématiques liées à la taille du corpus ne sont pas les seules à émerger. En effet, les ensembles épistolaires sont issus d’un processus d’échange entre — au moins — deux personnes communiquant sur des sujets variés depuis des lieux et à des dates différents. Une correspondance est un phénomène qui se développe et évolue dans le temps. Elle peut également être suivie ou discontinue et impliquer plusieurs personnes. De plus, elle peut nous arriver lacunaire. Tous ces paramètres sont à prendre en compte dans la constitution du corpus, afin de construire un propos cohérent et quelque soit la manière dont ils s’imposent à nous, ils doivent être justifiés¹. Enfin, le choix de ces documents doit être réfléchi et non le fruit du hasard ou par dépit, constituant ainsi un ensemble cohérent, avec l’intention de fournir au chercheur un outil fiable.

1.1.2 Mettre en valeur des sources

Pourquoi édite-t-on ? cette question doit être conservée à l’esprit durant tout le développement du projet. Il ne s’agit pas encore de cibler un type d’utilisateur ou quelle forme d’édition, nous souhaitons mettre en place. Il est ici question des objectifs auxquels toute édition scientifique doit répondre. En plus de diffuser ses textes auprès du plus large public possible, un éditeur scientifique doit également proposer des sources fiables et réutilisables pour la recherche². Il doit aussi s’efforcer de rendre compréhensible le texte édité en le replaçant dans son contexte afin de reconstituer le propos des auteurs. Ainsi il cherchera par exemple à rassembler des recueils éparpillés, ou à combler les manques ; il est bien sûr, invraisemblable de restaurer un vide, mais il est souvent envisageable de créer des liens entre différents documents afin de compenser cette perte^{3 4}. Enfin, l’éditeur joue aussi un rôle dans la valorisation des sources. C’est intrinsèquement lié à son objectif de

1. C. Desenclos, « Early modern correspondence : A new challenge for digital editions ».

2. C. Nougaret et É. Parinet, *L’édition critique des textes contemporains (XIXe-XXIe siècle)*, p. 19.

3. C. Desenclos, « Early modern correspondence : A new challenge for digital editions ».

4. C. Nougaret et É. Parinet, *L’édition critique des textes contemporains (XIXe-XXIe siècle)*, p. 19.

diffusion, qui doit favoriser l'accessibilité du public à ces matériaux, que l'éditeur participe à la mise en valeur de ces documents soit, en proposant des documents inédits, encore ignorés des chercheurs ou en participant à la protection des sources. Ces dernières une fois éditées et réceptibles, pouvant être retirées de la communication⁵.

Sur ce point de l'accès aux sources, le numérique joue indéniablement un grand rôle dans la mesure où toute édition publiée sur le web est abordable partout et par tous. Il faut toutefois bien faire la distinction entre la publication de fac-similé, comme le proposent certaines bibliothèques — telle la BDIC — et l'édition de textes ; toutes deux répondent à des impératifs relativement différents. Malgré tout, ces démarches doivent être encouragées.

1.2 Quel type d'édition

1.2.1 Définir une édition

Qu'elle soit électronique ou papier, une édition constitue immanquablement une vue réduite de la source originale et en conséquence, n'est pas neutre. Ce phénomène s'explique par le ciblage d'un public particulier et l'adoption de certains objectifs. Cela implique des choix éditoriaux et révèle qu'aucune édition n'a pour l'instant permis de satisfaire à l'ensemble des champs de la recherche⁶. Si nous prenons l'exemple de ce stage, l'édition a été pensée pour servir de support aux historiens. Sans entrer dans les détails que nous développerons dans la section suivante, ce choix a entraîné des interventions dans le texte afin d'en faciliter la lecture, comme la correction des fautes d'orthographe ou la restitution d'une ponctuation inexistante. Il est certain que notre entreprise intéressera moins les linguistes que le projet *corpus 14* développé par le laboratoire Praxiling de l'université Paul-Valéry Montpellier 3. Ce projet d'encodage de lettre de poilus

privilégie les écrits peu-lettrés, encore peu exploités par les historiens de la Grande Guerre. Ces documents, mis à disposition par les Archives départementales de l'Ain, de l'Hérault et par les familles qui en sont les dépositaires, fourniront aux linguistes des ressources nouvelles pour l'histoire de l'écrit peu-lettré⁷.

Le fait que nous ayons proposé des fac-similés aux côtés du texte édité n'apporte pas une solution satisfaisante. Que ce soit sur format papier ou format électro-

5. *Ibid.*

6. E. Vanhoutte et R. V. d. Branden, « Describing, transcribing, encoding, and editing modern correspondence material ».

7. *Projet corpus 14*, URL : <http://corpus14.ortolang.fr> (visité le 06/11/2015).

nique, les reproductions photographiques ne relèvent pas de la même intention que l'édition du texte. Les représentations fac-similaires apportent indéniablement une forme d'authenticité, mais elles contraignent toujours le lecteur à un exercice de transcription et sur format numérique, elles n'autorisent pas l'utilisation d'outils de recherche en plein texte par exemple.

1.2.2 Les principes éditoriaux retenus

L'édition scientifique requiert également l'observation de certains principes. Ces prescriptions n'en sont pas pour autant des normes figées, et répondent à un objectif de lisibilité offrant aux lecteurs un instrument compréhensible. Ces règles restent toutefois souples ; devant la variété des documents pouvant être édités et les objectifs suivis, c'est toujours à l'éditeur que revient le choix final. Ces conventions doivent toutefois être précisées avant le texte afin de prévenir le lecteur des opérations qui ont été menées sur celui-ci.

Concernant les textes contemporains, ces conventions ont été récemment réunies dans un ouvrage publié par l'École nationale des chartes et qui a servi de guide pour l'édition de la correspondance d'Armand Horel⁸.

Transcription

La transcription est une étape importante dans laquelle un grand nombre d'éléments sont à prendre en compte. En fonction des choix retenus cette phase peut orienter l'édition. Concernant la correspondance d'Armand Horel, il a été convenu de traiter les documents comme des textes historiques. Si l'orthographe est un témoignage de la condition de l'auteur, elle peut gêner la lecture lorsqu'elle est trop déficiente. La correction des écrits a donc été un parti pris en vue d'en faciliter la lecture et donc l'accès à l'information. Toutefois, la *Text Encoding Initiative* permet des alternatives intéressantes autorisant un double encodage du texte original et de sa forme normalisée. Les choix éditoriaux sont alors le fruit de l'application d'une feuille de style qu'il est possible de modifier. C'est la méthode que nous avons utilisée.

En tout état de cause, l'orthographe a donc été corrigée, et deux cas de figures ont été identifiés. Nous avons pu constater la récurrence de certaines fautes d'orthographe ou d'accord de conjugaison. Par exemple, Armand Horel confond très souvent le participe passé et l'infinitif des verbes du premier groupe. Ces corrections ont donc été appliquées mais pas indiquées dans la transcription, faute de quoi la lecture serait apparue plus difficile. Les fautes plus éparses et ne gênant pas la compréhension sont signalées avec la mention « *(sic)* ».

8. C. Nougaret et É. Parinet, *L'édition critique des textes contemporains (XIXe-XXIe siècle)*.

D'autre part, la graphie de l'auteur a été normalisée, les nombres ont été transcrit en toutes lettres, à l'exception des quelque cas particuliers prescrits dans le manuel⁹. Les abréviations ont été résolues sans faire mention des lettres restituées, les omissions rétablies ou les manques sont signalés par l'utilisation de crochets carrés, enfin, les mots agglomérés ont été séparés.

La ponctuation aussi a été modernisée dans la mesure où elle était quasiment inexistante ou très mal employée. Nous nous sommes donc également appliqués à restituer les minuscules ou les majuscules lorsque cela s'avérait nécessaire. Les mots biffés ou répétés ont été retirés du texte. Enfin lorsqu'une incertitude subsiste quant à la transcription du mot, cela est signifié par l'utilisation de la mention suivante : « (?) »

Précédant le texte de la lettre, l'auteur, le lieu, le support et la date d'envoi sont rappelés ainsi que le destinataire. Les adresses, lorsqu'elles étaient présentes, ont été, elles aussi, ajoutées avant le texte, de même que la transcription des enveloppes.

Mise en forme

La mise en forme du document est une partie importante du travail d'édition. elle comporte deux aspects : la présentation matérielle et l'organisation générale de l'édition¹⁰.

La présentation matérielle ne doit pas être négligée car elle permet au lecteur d'identifier formellement et à première vue, le type de document qui est présenté. Cela permet éventuellement de révéler des manques comme une signature ou une date par exemple et qui peuvent modifier la compréhension du texte. Concernant la correspondance d'Armand Horel, il s'agissait essentiellement de faire ressortir les mentions de dates et de lieux, les formules de salutation et de courtoisie ainsi que la signature. Celles ci justifient l'identification d'une correspondance en étant précisément séparées du corps du document.

Les mentions de date et de lieu sont représentées en amont du texte et justifiées à droite, suivent ensuite les salutations qui sont centrées sur la page et enfin le texte. Les formules de salutation sont restituées après le contenu en étant justifiées à gauche alors que la signature est justifiée à droite. Le *postscriptum* est placé après la signature et précédé de la mention "[PS]".

La correspondance d'Armand Horel est rédigée sur des cartes postales, nous souhaitons que cela soit mis en évidence dans notre mise en forme. Ainsi chaque carte correspond à un paragraphe de texte. Si les mentions pré-imprimées n'ont pas

9. *Ibid.*, p. 63.

10. *Ibid.*, p. 73.

été reprises, il est précisé, avec les informations d'identification placées précédemment au texte — expéditeur, destinataire, date et lieu d'envoi —, qu'il est question de cartes postales. Les enveloppes lorsqu'elles étaient présentes ont été placées préalablement au texte dans un menu déroulant. Ces éléments révèlent souvent une mine d'informations, comme le délai nécessaire pour l'expédition d'une carte lorsque les cachets de la poste sont présents. Enfin le fonds ayant été numérisé, nous avons fait le choix de les éditer aux côtés de leurs fac-similés.

Concernant l'organisation du contenu, les éditions électroniques autorisent la multiplication des accès dans le texte. Il était en revanche important de ne pas revenir sur les acquis de l'édition papier, à savoir : proposer une lecture linéaire et chronologique des correspondances. Ainsi pour chaque lettre, des liens renvoient vers les correspondances précédentes et suivantes. Lorsque les cartes n'étaient pas datées clairement mais qu'il semblait possible de la situer, avec plus ou moins de certitude, elles ont été replacées dans un calendrier des plus probables. Dans les cas d'une datation impossible, elles ont été positionnées à la fin.

Deux « tables » interactives ont également été développées. La première est une frise chronologique. Les différentes missives y sont regroupées par site. Cela permet de se représenter plus facilement les temps de stationnement du marin. La seconde prend la forme d'une carte où les différentes lettres sont rassemblées par lieu. La cartographie est une aide précieuse dans le cadre d'une édition de correspondances de guerre. Elle suscite une meilleure perception des enjeux stratégiques et militaires dans l'évolution du conflit.

Enfin un dernier accès est proposé en fonction des expéditeurs. L'ensemble des lettres de ce fonds ont été adressées à un seul destinataire, la Comtesse de la Forêt. En revanche, nous avons deux expéditeurs : Armand Horel et son frère Louis qui a rédigé une lettre. Cette particularité justifie cette modalité supplémentaire.

Annotation et recherche historique

Une annotation historique a été réalisée afin d'identifier certains événements dont il est fait mention ou allusion dans le texte. Dans la mesure du possible, les personnes, les circonstances ou lieux ont été expliqués ou re-situés dans un contexte historique plus général. Afin de ne pas surcharger les notes de bas de pages, certaines notes plus générales s'affichent lors d'un clic sur le terme identifié. Cela permet de ne pas répéter certaines informations sur plusieurs lettres sans pour autant casser le rythme de lecture, en allant chercher les informations en bas de page.

Dans un souci d'utilité pour le public et l'historien, des recherches ont ainsi été entreprises. Il serait tentant de croire qu'une édition reviendrait simplement à

éditer un texte. Or, replacer les lettres dans leur contexte historique et obtenir des renseignements sur les protagonistes sont nécessaires à la compréhension du corpus. Cette tâche est d'autant plus importante lorsqu'elle concerne un anonyme. Ici, nous avons effectué deux types de recherches ; d'une part concernant la biographie des protagonistes et d'autre part concernant les faits historiques.

En premier lieu, nous nous sommes intéressés à l'expéditeur des lettres : Armand Horel. Parce qu'il n'était pas connu, les démarches pour apprendre à le connaître se sont avérées délicates ; il était donc requis de croiser les sources. Notre premier réflexe a été de rechercher des informations sur les bases de données disponibles — dont le site institutionnel « Mémoire des Hommes » —, concernant les conflits mondiaux et la première guerre mondiale. Malheureusement, son nom n'y était pas répertorié. Alors, nous nous sommes tournés vers un autre protagoniste mainte fois mentionné par Armand dans ses lettres : son frère Louis Horel. Soldat décédé au front, nous savions désormais que ce frère était né près de Lorient en Bretagne. En retrouvant son état civil, nous pouvions obtenir également ceux des parents de Louis et Armand Horel. Enfin, l'utilisation des tables décennales et des registres militaires nous ont permis de retrouver la fiche militaire d'Armand Horel. En second lieu, il est apparu essentiel de réaliser des recherches sur les faits historiques mentionnés par Armand Horel tout au long de sa correspondance avec la Comtesse. Ces informations ont été récoltées grâce à des ouvrages historiques concernant les conflits mondiaux et la Grande Guerre. Enfin, nous avons entrepris des recherches sur les différents bâtiments de la flotte française, mentionnés par le marin tout au long de ses lettres. Pour ce faire, les bases de données en ligne en lien avec le forum « pages 14/18 », qui fait l'objet d'une étude dans le cadre du projet "le devenir du patrimoine numérisé en ligne", se sont révélés être précieux pour notre démarche, sans compter sur l'utilité des divers ouvrages et dictionnaires.

Indexation

Enfin un index/glossaire a été constitué, sa fonction de glossaire est utilisée pour les notes d'identification, l'index quant à lui est purement un index, il identifie, les personnes, lieux ou navires mais ne contient pas de descriptions "biographiques" des entrées. Afin de repérer les informations le plus rapidement les noms de personnes sont en petite capitale, les noms de lieux en italique et enfin les navires entre guillemets.

Les entrées se signalent au nom de famille, ou au prénom pour les souverains, les découpages administratifs pour les lieux géographiques ont été respectés. Lorsque plusieurs formes coexistent, des renvois ont été opérés vers les formes acceptées.

Chapitre 2

Une édition électronique XML-TEI

Nous l'avons vu, cette édition est destinée à être publiée en ligne. Si les éditions électroniques ne sont pas encore très répandues au regard des éditions papier, la multiplication des projets — *corpus 14*, *l'édition des guides de Paris* etc. — démontre un attrait de plus en plus important pour cette forme de publication. Indéniablement, les éditions électroniques ne manquent pas d'atouts. Accessibles en ligne en permanence, elles accordent également une plus grande diffusion du travail de l'éditeur.

L'édition scientifique souffre généralement d'un manque de visibilité mais demeure pourtant essentielle. Elle fournit régulièrement de la matière première aux études des chercheurs. Si les coûts inhérents à une publication web sont équivalents, ils peuvent toutefois être rationalisés¹. Assurément, la technologie exige un investissement plus important pour l'acquisition d'un matériel informatique adéquat, toutefois, les frais de production sont souvent négligeables. Les fichiers produits peuvent être exploités et ré-exploités sous de nombreuses formes, y compris pour une publication papier.

L'édition électronique permet également une plus grande souplesse d'usage. Pour les ensembles importants d'édition, elle permet des mises en ligne successives, au fur et à mesure de l'avancement des travaux, là où une édition papier n'aurait pu exister ou alors dans une version fragmentaire.

La publication web fournit également une ergonomie et un confort d'utilisation incompatible avec une version papier. Elle permet de multiplier les points d'accès aux documents, facilite la recherche dans le texte et favorise les lectures non-linéaires — sur ce point il faut toutefois faire attention de ne pas revenir sur les acquis des éditions papiers qui apportent une grande cohérence au texte et que l'édition électronique doit également garantir. Elle repose en grande partie sur

1. Gautier Poupeau, « L'édition critique des sources historiques : du support papier au Web », dans *rencontres de la communauté francophone de la TEI*, Nancy, 2005, URL : <http://www.lespetitescases.net/node/1088> (visité le 03/09/2015).

l'utilisation des liens hypertextes qui permettent d'associer directement les lettres entre elles mais également aux différentes tables, index et glossaires, voire à des ressources extérieures. En facilitant l'accès à l'information, l'édition électronique permet également d'alléger les notes en les rationalisant par l'usage de liens hypertextes. Elle présente également une autre expérience de lecture. En proposant comme nous l'avons vu, l'affichage de certaines annotations dans une info-bulle au survol des termes explicités, elle limite ainsi les allers-retours en bas de page obligeant le lecteur à interrompre sa lecture. Les entrées d'index ou de glossaire peuvent par exemple, être directement identifiées dans le texte par cette méthode.

Avec la rigueur que toute édition impose, tous ces paramètres impliquent qu'une édition électronique doit être pensée comme telle, et non, comme la numérisation d'une édition papier. La volonté de ce stage était donc l'élaboration d'un modèle d'édition électronique permettant une publication web. Les pages internet reposant, *a minima*, sur le langage HTML (*Hypertext Markup Language*), toute édition en ligne nécessite une phase d'encodage.

2.1 Le recours à un balisage descriptif

Il est admis que les langages de balisage sont des solutions adaptées à l'édition et à l'enrichissement de l'information textuelle. On en discerne principalement trois sortes : les langages présentationnels, procéduraux et descriptifs². Ils se caractérisent par l'utilisation de balises servant à délimiter une portion de texte au sein d'un flux de caractères. Même s'ils présentent des différences, les langages procéduraux et présentationnels, sont essentiellement centrés sur la mise en forme du texte et ne conçoivent pas de rendre compte du contenu ou de la structure d'un document. Les langages de balisage descriptifs reposent quant à eux sur une structure arborescente et hiérarchique. Ils offrent en outre, de nombreux avantages pour l'édition de textes :

- Ils permettent de témoigner de la structure et de la sémantique d'un document sans tenir compte de sa représentation.
- L'établissement — ou la modification — d'un mode de représentation est géré de manière indépendante.
- Ils fournissent de meilleures garanties en matière de portabilité et d'interopérabilité, tout en autorisant différents modes de sortie à partir d'un seul fichier.

2. James H. Coombs, Allen H. Renear et Steven J. DeRose, « Markup Systems and the Future of Scholarly Text Processing », *Communications of the ACM* 30, (1987), p. 933–947, URL : <http://xml.coverpages.org/coombs.html> (visité le 30/08/2015).

2.2 Le dialecte XML-TEI

La technologie XML (*eXtensible Markup Language*) s'est rapidement imposée comme le système de formatage de données le plus adapté au développement de notre projet. Il s'agit d'un « métalangage à balises pour les documents textuels »³, reposant sur une structure arborescente et hiérarchique. Ce point est particulièrement intéressant car cela met en œuvre un langage informatique facilement compréhensible par l'homme mais aussi par la machine⁴. Développé par le W3C (*World Wide Web Consortium*), il est résolument tourné vers la publication internet⁵. Il s'est imposé comme un standard pour le formatage de données pérennes ; XML assure aux documents numériques un haut niveau de portabilité et d'interopérabilité. Son utilisation a notamment été recommandée par la direction générale de modernisation de l'État dans son *référentiel général d'interopérabilité* publié en 2009.

Toutefois, XML n'offre pas une solution prête à l'emploi ; « XML est un métalangage à balises. Cela signifie qu'il n'a pas de jeu de balises prédéfini et d'éléments qui sont toujours supposés fonctionner pour tout le monde et pour n'importe quel sujet »⁶. Ce langage formule en réalité, des règles que nous devrions retrouver dans un document bien formé — qui respecte des règles syntaxiques strictes — et valide — conforme à un schéma défini. L'association d'un schéma à un document XML est facultative. Toutefois, pour des raisons d'interopérabilité ou de travail en groupe, il est nécessaire de se mettre d'accord sur l'utilisation des balises en ayant recours à une DTD (*Document Type Definition*) ou un schéma, si l'on désire des règles plus contraignantes. De plus, XML propose un cadre de travail pertinent disposant d'outils performants, qui autorisent la création de feuilles de style, de feuilles de transformation, ou encore qui tolèrent de combiner plusieurs documents entre eux.

Dans le champ des humanités numériques, la *Text Encoding Initiative* (TEI) est le standard international qui s'est imposé pour l'édition des sources primaires. Apparue en 1987, elle offre un cadre de travail documenté tout en favorisant différentes approches sur le texte ; analytique, linguistique, historique, littéraire ou encore morpho-syntaxique⁷. La TEI autorise un double niveau d'encodage du texte, relatif à sa structure et sa sémantique. Elle permet également d'atteindre un niveau de description (granularité) particulièrement important. Bénéficiant du cadre

3. Elliotte Rusty Harold et W. Scott Means, *XML en concentré : manuel de référence*, trad. par Philippe Ensarguet et Frédéric Laurent, Paris, France, 2005, p. 3.

4. *Ibid.*, p. 97.

5. Marjorie Burghart et Nicole Dufournaud, *Édition électronique de sources : XML et Text Encoding Initiative (TEI) à l'École*, La lettre de l'École des hautes études en sciences sociale, 2011, URL : <http://lettre.ehess.fr/1530> (visité le 27/08/2015).

6. E. R. Harold et W. S. Means, *XML en concentré*, p. 4.

7. M. Burghart et N. Dufournaud, *Édition électronique de sources : XML et Text Encoding Initiative (TEI) à l'École*.

de travail offert par XML, nous nous sommes donc spontanément tournés vers ce dialecte pour la conduite de notre projet.

2.3 La nécessité de personnaliser le schéma TEI

La *Text Encoding Initiative* a été pensée pour autoriser l’encodage de tout type de texte dans des domaines aussi variés que la publication électronique, l’analyse historique et littéraire ou encore la recherche documentaire. De fait, avec plus de cinq cent éléments proposés, la TEI offre un ensemble de balises à la fois trop riche et trop généraliste pour une application donnée. Conscients de ces difficultés, ses concepteurs l’ont envisagée selon un principe modulaire, offrant un cadre de travail à l’encodeur⁸. Ce cadre de travail, composé d’un vocabulaire documenté et organisé selon un ensemble de recommandations, les *Guidelines*, permet à l’encodeur de venir puiser dans les différentes briques, les ressources dont il aura besoin pour le développement de son projet.

Si cette modularité assure la flexibilité et l’adaptabilité du modèle TEI, elle implique en contre-partie que l’encodeur ait bien conscience du résultat prévisible engendré par ses choix⁹. C’est plus généralement, la question des objectifs de l’édition qui est posée ici. Chaque texte à éditer impose de personnaliser le schéma TEI, réclamant un traitement particulier qu’il convient de conjuguer avec les exigences du projet.

La première exigence de personnalisation vient de l’obligation de standardiser les pratiques d’encodage. La réalisation d’une modélisation pouvant être utilisée au sein de projets collectifs, contraint à la fixation des règles d’encodage rigoureuses afin que les différents membres du projet aient une utilisation uniforme du balisage. Il s’agit donc, dans un premier temps de sélectionner une portion représentative du corpus à encoder et d’opérer des allers-retours avec les *Guidelines* afin d’identifier les solutions proposées par la TEI et permettant de répondre aux spécificités du sujet.

Cet effort de standardisation des pratiques d’encodage présente un autre intérêt. « Un document XML est, mais il ne fait rien »¹⁰. En effet, avant même d’être compréhensible par la machine, un document XML — et par extension, un fichier TEI — est avant tout un document textuel. Pour être utilisé dans le cadre d’une

8. Nancy Ide et Jean Véronis, « Une application de la TEI aux industries de la Langue : le Corpus Encoding Standard », *Cahiers GUTenberg*, 24 (juin 1996), URL : <http://sites.univ-provence.fr/veronis/pdf/1996gut-corpus.pdf> (visité le 01/09/2015).

9. Emmanuel Chateau, *L’édition des cours, Le modèle XML-TEI*, Cours de Desgodets, 2013, URL : <http://www.desgodets.net/edition-des-cours/model#editionModelBodyFr02> (visité le 31/08/2015).

10. E. R. Harold et W. S. Means, *XML en concentré*, p. 5.

publication web par exemple, le fichier XML doit être soumis à une instruction de traitement, ou de transformation, afin d'être rendu lisible. Si cette tâche s'opère généralement de manière automatisée, par l'intermédiaire d'une feuille de style, elle nécessite en revanche que les fichiers qui y sont soumis, soient conçus sur le même modèle.

Une édition électronique change quelque peu le rapport du lecteur au texte. Le média numérique aide à la recherche d'informations, multiplie les points d'accès au texte ou encore facilite la lecture et la compréhension en permettant par exemple l'affichage de certaines données au survol, évitant ainsi des allers-retours vers les index ou glossaires d'une édition papier. Nonobstant, il faut garder à l'esprit que ces outils offerts par le numérique ne doivent représenter que des aides supplémentaires qui ne doivent pas se substituer aux acquis d'une édition papier, à savoir : offrir une lecture linéaire et compréhensible du texte. Une édition doit avant toute chose, produire un texte cohérent, utilisable par tous et pour des finalités différentes¹¹. C'est l'objectif premier auquel nous avons tenté de répondre au travers de cette modélisation XML-TEI, dédiée à l'édition de correspondance de poilus.

11. C. Desenclos, « Early modern correspondence : A new challenge for digital editions ».

Deuxième partie

Un modèle d'édition XML-TEI

Chapitre 3

Rappels généraux et notions de corpus électronique

3.1 La structure générale du schéma TEI

Afin que chacun puisse se représenter les concepts et notions mis en œuvre dans cette partie, un rappel sur la structure du schéma TEI semble indispensable. Certains éléments étant obligatoires à tout document TEI, cette section nous permettra de les identifier et de comprendre leurs rôles.

La technologie XML impose à la TEI une structure hiérarchique reposant sur l'imbrication de différentes données. Ces éléments constituent la structure de base d'un fichier XML et par extension d'un fichier TEI. Le premier que l'on retrouve est l'élément racine ; il est unique et contient l'ensemble des sous-éléments. Il s'agit généralement de l'élément `<TEI>` — ou `<teiCorpus>` si l'on réalise un fichier maître —. Le reste du document est ensuite partagé entre l'élément `<teiHeader>` et l'élément `<text>`.

```
<tei>
  <teiHeader>...</teiHeader>
  <text>...</text>
</tei>
```

Le premier correspond à l'entête du document et rassemble les métadonnées permettant son identification et se subdivise ainsi :

```
<teiHeader>
  <fileDesc>(obligatoire) description bibliographique
  <titleStmt>
    (obligatoire) mentions de titre
  </titleStmt>
```

```

<publicationStmt>
  (obligatoire) mentions de publication
</publicationStmt>
<sourceDesc>
  (obligatoire) consigne toutes les informations sur la source originale
</sourceDesc>
</fileDesc>
<encodingDesc>description des normes d'encodage</encodingDesc>
<profileDesc>
  description des aspects non bibliographiques
</profileDesc>
<revisionDesc>
  historique des révisions du fichier numérique
</revisionDesc>
</teiHeader>

```

Le second, <text>, contient l'ensemble du texte édité il peut se subdiviser en quatre parties.

```

<text>
  (obligatoire)
  <group>
    <!-- l'élément <group> permet de rassembler plusieurs élément <text> -->
    <text>...</text>
    <text>
      <front>
        Texte préliminaire contenant tout ce qui est au début du document, avant le corps du
        texte.
      </front>
      <body>
        Corps du texte, contient la totalité du corps d'un seul texte simple, à
        l'exclusion de toute partie pré-liminaire.
      </body>
      <back>
        Texte annexe, contient tout supplément placé après la partie principale d'un texte :
        annexe, etc.
      </back>
    </text>
  </group>
</text>

```

3.2 Création d'un corpus électronique

Lorsque l'on procède à une édition électronique de correspondances le premier enjeu auquel nous sommes confrontés, réside dans l'encodage d'une multitude de petits textes indépendants mais pour autant liés les uns aux autres. À la différence d'une œuvre littéraire, ces textes ne peuvent pas être encodés à la suite comme différents chapitres. En effet, à des fins d'ergonomie, la publication web nécessite que chaque lettre possède sa propre page de consultation. De la même manière que nous avons élaboré un corpus de documents pour cette édition, il est nécessaire de réfléchir à un corpus électronique organisé pour les éditer. Le schéma mis en place par la *Text Encoding Initiative* n'a pas été élaboré pour cet exercice mais des solutions peuvent tout de même être envisagées grâce notamment aux outils offerts par le cadre de travail XML.

Quand nous avons débuté l'encodage de la correspondance d'Armand Horel, nous avons été confrontés à cette difficulté. Le premier réflexe, à partir du principe que chaque lettre doit disposer de sa page HTML pour sa publication, a été de créer un fichier XML-TEI par lettre. Nonobstant, certaines informations nécessitaient d'être reprises dans chaque document ; les index notamment dont les informations contenues dans les différentes entrées allaient être réutilisées pour générer des notes d'identification. Procéder ainsi, expose déjà à une complication dans la mesure où, si on ajoute ou modifie une entrée, il faut le faire dans chaque fichier. Cela peut se concevoir lorsque l'on doit gérer quelques documents mais cette organisation peut rapidement devenir ingérable lorsque l'on doit traiter une correspondance comprenant parfois plus d'une centaine d'items.

Une première solution est proposée par la *Text Encoding Initiative* avec la création d'un fichier *teiCorpus* qui permet de rassembler, au sein d'un seul fichier, un ensemble de documents TEI bien distincts et pouvant partager un ensemble de métadonnées communes.

```
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <!--[en-tête du corpus pouvant contenir les différentes métadonnées communes]-->
  </teiHeader>
  <TEI>
    <teiHeader>
      <!--[en-tête de la première lettre]-->
    </teiHeader>
    <text>
      <!--[première lettre du corpus]-->
    </text>
  </TEI>
  <TEI>
```

```

<teiHeader>
  <!--[en-tête de la deuxième lettre]-->
</teiHeader>
<text>
  <!--[deuxième lettre du corpus]-->
</text>
</TEI>
</teiCorpus>

```

Cette proposition de la TEI permet de fait une gestion plus aisée des fichiers dans la mesure où il n'y a plus qu'un seul document à gérer. Toutefois, cela pose encore un problème pour la création des différentes pages d'un site internet dédié à l'édition. Nous ne pouvons pas créer par le biais d'une seule feuille de transformation XSLT, les différentes pages web correspondantes aux différentes lettres.

Une solution a donc été trouvée avec l'environnement de travail XML qui dispose d'outils puissants. Le standard Xinclude autorise notamment la combinaison de différents documents XML entre eux. En effet, il permet de cibler certaines parties identifiées d'un document afin de l'intégrer au sein d'un autre fichier. Nous avons donc procédé de la manière suivante ; un fichier `teiCorpus` a été créé contenant chaque lettre au sein d'un élément `<TEI>`. Pour chaque lettre, un autre document XML a été créé et complété par le biais de différents appels `Xinclude`. Ces derniers ont permis par la suite la création des différentes pages web en leur appliquant les feuilles de transformation XSLT. Si cela semble complexe au premier abord, cela s'avère en réalité extrêmement commode et pertinent dans la gestion du projet, dans la mesure où une fois que les éléments nécessaires à chaque lettre ont été identifiés, il n'y a plus qu'un seul document à modifier, le fichier maître contenant le `teiCorpus`. Cela s'avère intéressant également pour la création d'un schéma TEI spécifique, car ce document rassemble l'ensemble des pratiques d'encodage.

```

<!--Voici un exemple d'utilisation d'Xinclude-->
<!--L'espace de nom Xinclude doit être déclaré avec la mention suivante-->
<text xmlns:xi="http://www.w3.org/2001/XInclude">
  <!--l'attribut @href cible le fichier contenant la partie de document à intégrer-->
  <xi:include href="../master/Master.xml" xpointer="body_imbro_23-09-1915" parse="xml"/>
  <!--l'attribut @xpointer cible l'@xml:id du morceau de code à inclure-->
  <!--l'attribut @parse indique le type de données-->
</text>

```

Concernant les liens, cela permet généralement de les simplifier car ils deviennent des liens internes au document. Toutefois, il faut faire attention ; dans une édition de correspondances comme nous l'avons vu, les différentes lettres doivent

être publiées sur leur propre page. Cela nécessite dans la gestion des liens qui deviendront des liens hypertextes une attention toute particulière car ils doivent être gérés en tenant compte du fichier XML servant à la sortie et non du teiCorpus.

```
<ref target="../Toulon/F_delta_1854_20_1_Toulon_8-08-1915.xml">
<!-- Ici par exemple ce chemin décrit par l'attribut @target tient compte de l'emplacement des
      fichiers de sortie -->
  <persName type="sentBy">Armand Horel</persName>
  <persName type="deliveredTo">Madame la Comtesse L. de la Forêt</persName>
  <placeName>Toulon</placeName>
  <date when="1915-08-08">08 août 1915</date>
</ref>
```


Chapitre 4

La personnalisation XML-TEI

Comme nous l'avons vu, le schéma TEI nécessite d'être contraint afin de standardiser les pratiques d'encodage. Cette personnalisation doit être en mesure de répondre à l'orientation scientifique de l'édition mais aussi aux principes éditoriaux retenus. Cette étape implique dans un premier temps une analyse poussée des matériaux à éditer, typologies, caractéristiques générales, spécificités etc. afin dans un second temps de rechercher dans les *TEI Guidelines* des solutions permettant de retranscrire ces caractéristiques de manière compatible avec la finalité de l'édition.

Concernant notre projet, l'édition de la correspondance d'Armand Horel, devait surtout permettre l'élaboration d'un modèle d'édition électronique réutilisable par la BDIC mais également par ses usagers qui en formuleraient la demande. Il s'agissait donc de formuler des principes d'encodage qui soient transposables à d'autres corpus de lettres tout en tenant compte des particularités de ce fonds. En effet, la correspondance d'Armand Horel est constituée principalement de cartes postales. Comme nous l'avons vu, cette édition procède d'une lecture historique des textes ; et il n'a pas été envisagé une étude du support pré-imprimé, toutefois les cartes postales présentent une particularité face aux lettres plus « classiques », elles sont à la fois objet et support d'écriture. Il s'agissait donc, au sein de notre modélisation, de tenir compte de ces caractéristiques tout en assurant la compatibilité avec d'autres ensembles épistolaires. Malgré quelques singularités que nous développerons par la suite, les cartes postales sont avant tout, des correspondances, de fait, elles partagent donc aussi beaucoup de propriétés avec les autres types de lettres. Elles ont été écrites par un auteur, qui est aussi l'expéditeur, en un lieu et une date déterminés, enfin elles comportent un message adressé à un ou plusieurs destinataire. Ce constat détermine les modalités essentielles qui constituent plus ou moins toute missive.

Le respect des sources ainsi que la traçabilité des informations sont des élé-

ments essentiels pour en vérifier la pertinence et favorisent ainsi la transmission du savoir, sans oublier l'héritage de connaissances qui a été légué. Il était fondamental de respecter une certaine exhaustivité des éléments lors de notre encodage. Sachant qu'une édition constitue de fait, une vision réduite de l'original, il était donc capital de conserver le plus possible d'informations (disposition de certains segments de texte, orthographe, ponctuation etc). Il fallait néanmoins offrir, pour l'édition électronique, une version normalisée du texte, du point de vue de sa mise en forme mais également de sa graphie.

Pour ce chapitre, relatif aux techniques d'encodage retenues, il ne s'agit nullement de reprendre le manuel d'encodage de ce projet et ne se substitue pas à ce dernier ou aux *TEI Guidelines*. Il présente les solutions techniques testées et validées qui ont soulevé des interrogations lors de l'avancement du projet.

Par convention les éléments XML seront à partir de maintenant représentés entre chevrons (<monElement>) les attributs seront précédés d'un arobase et en italique (@*monAttribut*), enfin les valeurs de ces derniers seront placées entre guillemet (@*monAttribut*="maValeur").

4.1 Identification des lettres et métadonnées

Sur une édition papier, la structure conférée aux lettres permet un repérage rapide des diverses informations d'identification de la lettre : expéditeur, destinataire, lieu d'envoi ou encore la date.

Ces informations sont essentielles dans une édition électronique et leur encodage représente un premier enjeu. Ces données concourent à l'identification des lettres et par extension, apportent une preuve de leur existence au sein de la correspondance. En nous fournissant le nom de l'expéditeur, du destinataire, l'objet ainsi que des mentions de dates et de lieux, elles permettent véritablement des distinguer chaque lettre les unes des autres et participent à les replacer dans leur contexte. Ce sont d'ailleurs ces éléments qui permettent d'alimenter les instruments de recherche dans ce type édition. Ces données doivent donc être traitées avec attention. Jusque récemment la *Text Encoding Initiative* ne proposait pas de solution spécifique pour ces informations.

Dans le cadre du projet DALF (*Digital Archives and Letters in Flanders*) développé par le *Center for Scholarly Editing and Document Studies*, une DTD avait été réalisée à partir de la TEI et proposait la création d'un ensemble d'éléments au sein de la balise <sourceDesc> déjà existante et autorisant le balisage de tous ces renseignements .

<sourceDesc>

```

<!-- Informations pour l'identification des lettres -->
<letIdentifier>
  <country>France</country>
  <settlement>Nanterre</settlement>
  <repository>BDIC</repository>
  <idno>F delta 1854/20</idno>
</letIdentifier>
<letHeading>
  <!-- description bibliographique de la lettre -->
  <author attested="yes">Armand Horel</author>
  <addressee attested="yes">La comtesse de la Forest</addresse>
  <placeLet attested="yes">Salonique</placeLet>
  <dateLet>1915-12-27</dateLet>
</letHeading>
<letContents>
  <!-- Objet de la lettre -->
  <p>Guerre Sous marine</p>
</letContents>
</sourceDesc>

```

Cette réflexion aboutit à la création en 2013 d'un groupe de discussion au sein de la TEI afin de trouver des solutions pour l'encodage de ces informations spécifiques aux correspondances. Juste avant le début de ce stage un élément <correspDesc> a été intégré au schéma TEI. Reprenant une partie des idées de la DTD du projet DALF, elle rationalise surtout les éléments créés, au nombre de trois : L'élément <correspDesc> placé dans le <profileDesc> et deux sous-éléments <correspAction> et <correspContext>

L'élément <correspDesc> « contient les informations des actions relatives à un acte de correspondance ». La balise <correspAction> contient quant à elle « une description structurée du lieu, de la personne et de la date et relative aux actes d'envoi, de réception, ou de transmission du courrier ». Enfin, l'élément <correspContext> « fourni les références des lettres suivantes et précédentes de celle décrite ». Cet ajout dans le schéma de la TEI est très intéressant car s'il ne répond pas à toutes les problématiques liées à l'encodage des correspondances, il permet de fournir une identification précise de chaque missive et de les replacer dans leur contexte propre. C'est là tout le sens de ce module. L'élément <correspDesc> ne fournit pas tant une description bibliographique de la lettre qu'une recontextualisation, car ce sont bien les circonstances de sa production qui sont décrites et non ses acteurs — même si elle y participe. Cette proposition d'encodage avait l'avantage d'être conforme à la TEI, et cela validait naturellement notre choix.

```

<correspDesc xml:id="context_toulon_08-08-1915">

```

```

<!--Les <correspAction> décrivent le rôle des protagonistes par l'utilisation de l'attribut
  @type-->
<correspAction type="sent">
  <persName type="sentBy" ref="#Horel">Armand Horel</persName>
  <placeName ref="#Toulon">Toulon</placeName>
  <date when="1915-08-08">08 août 1915</date>
</correspAction>
<correspAction type="received">
  <persName type="deliveredTo" ref="#Comtesse_Forest">Madame la Comtesse L. de la
    Forêt</persName>
</correspAction>
<correspContext>
<!--les éléments <ref> permettront avec un traitement XSLT des attributs @target, de créer
  des liens hypertextes vers les correspondances suivante -->
<ref type="previous" target="Salonique_15-05-1915.xml">
  <persName type="sentBy" ref="#Horel">Armand Horel</persName>
  <persName type="deliveredTo" ref="#Comtesse_Forest">Madame la Comtesse L. de la
    Forêt</persName>
  <placeName ref="#Salonique">Salonique</placeName>
  <date when="1915-05-15">15 mai 1915</date>
</ref>
<ref type="next" target="Toulon_29-08-1915.xml">
  <persName type="sentBy" ref="#Horel">Armand Horel</persName>
  <persName type="deliveredTo" ref="#Comtesse_Forest">Madame la Comtesse L. de la
    Forêt</persName>
  <placeName ref="#Toulon">Toulon</placeName>
  <date when="1915-08-29">29 août 1915</date>
</ref>
</correspContext>
</correspDesc>

```

Si l'on analyse plus finement cette suggestion d'encodage on observe dans un premier temps, que ce sont les rôles et actions de chacun des protagonistes qui sont décrits et précisés par l'utilisation de l'attribut *@type* de l'élément *<correspAction>* qui reprend l'envoi et la réception. Aucun élément spécifique à l'encodage de l'expéditeur ou du destinataire n'étant prévu, il faut donc se tourner vers l'attribut *@type* des éléments *<persName>*, plus génériques, pour circonstancier l'état de chacun des interlocuteurs.

Dans un deuxième temps la balise *<correspContext>* permet de recontextualiser cet envoi au sein de la correspondance en créant un lien vers la lettre précédente et la suivante. Dans la mesure où aucun élément spécifique n'a été créé, nous avons utilisé l'élément *<ref>* précisé par les attributs *@type* de valeurs "next" ou "previous" afin d'identifier les correspondances suivantes et précédentes et l'attribut *@target* afin de lier les différentes missives ensembles. Un simple élément

<ptr/> (pointeur) aurait pu être utilisé en remplacement de l'élément <ref>. Toutefois le pointeur est une balise qui ne peut contenir de texte là où l'élément <ref> nous a permis d'identifier plus précisément la correspondance suivante, en précisant de la même manière l'expéditeur, le destinataire, le lieu d'envoi ou encore la date.

Si l'on revient aux propositions formulées par le projet DALF, on s'aperçoit que le schéma TEI ne répond que partiellement aux attentes. En effet, rien n'est prévu pour les références bibliographiques des lettres de même que pour l'objet des correspondances.

Pour le premier point, les éléments <correspAction> n'y répondent qu'en partie mais ne comprennent pas de solution pour localiser ces documents. Une solution consiste à regarder du côté de la balise <msIdentifier> contenu dans l'élément <msDesc>. Cette balise « contient les informations requises pour identifier le manuscrit en cours de description ».

```
<sourceDesc>
  <msDesc>
    <msIdentifier xml:id="msIdentifier">
      <country>France</country>
      <settlement>Nanterre</settlement>
      <repository>BDIC</repository>
      <idno>F delta 1854/20</idno>
    </msIdentifier>
  </msDesc>
</sourceDesc>
```

Toutefois, cette version que nous avons retenue pour la modélisation de notre projet, n'offre qu'une réponse imparfaite car ne sont concernés ici que les documents manuscrits. Si les lettres rédigées manuellement représentent l'immense majorité des ensembles épistolaires de la Grande Guerre, il faut garder à l'esprit que certains ne le sont pas.

Concernant l'objet, aucune solution spécifique n'a encore été prévue par la TEI. Sur ce point nous avons repris la proposition de Camille Desenclos formulée dans sa modélisation dédiée à l'édition de correspondances modernes. Elle suggère de placer ces informations dans l'élément <front>, au sein d'une balise <argument> qui peut contenir selon les *Guidelines*, « [une] liste formelle ou description en texte libre des sujets traités dans une subdivision d'un texte ».

```
<front>
  <argument>
    <p>Guerre sous-marine</p>
```

```
</argument>
</front>
```

L'analyse de cette section consacrée aux métadonnées est intéressante car au-delà de la définition et de l'utilisation des différentes balises elle nous informe sur le mode d'évolution du schéma de la *Text Encoding Initiative*. Son développement repose en partie sur des propositions formulées dans le cadre de projets d'édition et d'un besoin des usagers, toutefois elle prend soin de ne pas multiplier les balises lorsque d'autres, déjà intégrées dans le schéma, peuvent s'adapter à des besoins plus spécifiques.

Plus largement se pose la question de la conformité au schéma TEI. La création d'élément, si elle permet de répondre à des besoins spécifiques et ponctuels, représente également un frein à l'interopérabilité du projet. Elle peut également être considérée comme une solution temporaire, le temps que le schéma TEI évolue¹.

4.2 Le balisage du texte

4.2.1 Le balisage logique

Il est question ici de la macro-structure des documents originaux. Cette partie de l'encodage a soulevé de nombreuses controverses lors de l'édition de la correspondance d'Armand Horel. En effet, les cartes postales sont des matériaux très structurés présentant certaines analogies avec des lettres classiques mais aussi de vraies disparités.

Une des premières particularités se traduit par le fait que les courriers étaient souvent composées de plusieurs cartes. Il a donc été décidé de ne pas les encoder à la pièce mais de les réunir au sein d'un élément <div> comportant un attribut @type de valeur "letter" afin d'opérer une réunion logique des matériaux. Ensuite s'est posée la question du balisage des différentes parties que l'on retrouve traditionnellement sur une carte postale avec notamment le recto comportant l'image ainsi que le verso subdivisé entre le message et la partie réservée à l'adresse du destinataire.

```
<text>
  <div type="letter">
    <div type="carte" n="1">
      <div type="recto">
        <figure>
```

1. Ead., « Early modern correspondence : A new challenge for digital editions ».

```

    <graphic url="../../FACS/BDIC_FD_1854_20_1_027.jpg"/>
    <figDesc>
      [la description de l'image]
    </figDesc>
  </figure>
</div>
<div type="verso">
  <div type="message">
    [ici le texte d'une des lettres]
  </div>
  <div type="envoi">
    [ici les informations relatives à l'envoi, adresse du destinataire, cachet de la poste etc.]
  </div>
</div>
</div>
</text>

```

Cet encodage a l'avantage de parfaitement décrire la structure d'une carte postale mais il présente également des inconvénients. Tout d'abord, il est très lourd, d'autant plus lorsqu'une lettre comporte plusieurs cartes, ce qui oblige à multiplier les éléments <div>. D'autre part il ne fallait pas perdre de vue l'objectif du stage qui était, de proposer une modélisation pour l'édition de correspondances, or les cartes postales ne représentent pas l'essentiel des ensembles épistolaires détenus par la Bibliothèque de Documentation Internationale Contemporaine. Enfin un modèle d'encodage s'il peut comporter des points délicats doit aussi se caractériser par une facilité d'emploi, particulièrement lorsqu'il est destiné à une diffusion vers des non-spécialistes.

Nous avons donc gardé le rassemblement "logique" de l'élément <div @type="letter">. Au sein de celui ci, nous avons pu traiter chaque face de chaque carte comme une page. Comme nous souhaitions que notre modélisation soit adaptable, le plus simple consistait à considérer les cartes postales comme de simples correspondances. Ainsi les passages du recto au verso ont été encodés des éléments <pb/>. Pour les changements de carte, cette balise est complétée de l'attribut @type de valeur "postcardBreak". Concernant la position de l'adresse du destinataire dans la mesure où elle est toujours sur la colonne de droite, le schéma TEI propose l'élément <cb/> pour le changement de colonne qui peut être utilisé à cet effet.

```

<text>
  <div type="letter">
    <pb n="1"/>
    <figure>

```

```

<graphic url="../../../FACS/BDIC_FD_1854_20_1_027.jpg"/>
<figDesc>
  [la description de l'image de la carte N 1]
</figDesc>
</figure>
<pb n="2"/>
<p>
  [ici le texte de ma première carte]
<cb/>
<address>ici l'adresse du destinataire</address>
<p>
<pb n="3" type="postcardBreak"/>
<figure>
  <graphic url="../../../FACS/BDIC_FD_1854_20_1_028.jpg"/>
  <figDesc>
    [la description de l'image de la carte N 2]
  </figDesc>
</figure>
<pb/>
<p>[ici le texte de ma seconde carte]<p>
</div>
</text>

```

Cette solution possède l'avantage d'être plus facile à mettre en œuvre et peut s'adapter à une édition de lettre plus classique, le fait de placer les paragraphes (<p>) au même niveau que les changements de page (<pb/>) permet de marquer la matérialité des cartes. En revanche nous perdons de fait, un niveau de description en cas de présence de plusieurs paragraphes sur une carte postale. Toutefois, nous n'avons pas de document présentant cette particularité. Ce choix d'encodage peut sembler singulier. Il trouve son explication dans la nature même d'une carte postale. C'est à la fois un support d'écriture mais également un objet autonome. Les cartes postales ne se plient pas comme les différents feuillets d'une lettre afin de les réunir pour les placer dans une enveloppe ou tout simplement pour les conserver. Finalement les cartes postales ne sont reliées entre elles que par le message qu'elles délivrent et conservent donc une certaine individualité. De plus, en cas de mauvaise association de cartes de notre fait, ce système garantit une imbrication limitée autorisant des corrections plus légères.

Pour les lettres plus classiques et dans la mesure où il n'y a pas de nécessité de faire ressortir la matérialité des différents feuillets, les paragraphes du texte seront respectés et les éventuels changements de pages y seront intégrés.

```

<div type="letter">
  <p>

```

```
[ici le texte de mon premier feuillet]
<pb/>
[ici le texte de mon second feuillet]
</p>
</div>
```

4.2.2 Balisage physique

Cette sous-section est relative à la mise en forme de la source. S'il n'a pas été envisagé de produire une version fac-similaire, comme nous l'avons déjà évoqué, nous souhaitons un encodage respectueux des sources. Les changements de ligne, de page, ou de colonnes ont donc été encodés avec les éléments TEI idoines à savoir respectivement <lb/>, <pb/> et <cb/>. Si ce balisage physique ne pose pas de problème certains points particuliers ont soulevé des interrogations.

Césure des mots en fin de ligne

Deux éventualités se sont présentées à nous lors de l'encodage. Le premier est le cas classique de la césure matérialisée dans le texte qui a été dans un premier temps encodé en utilisant l'attribut @rend="hyphen" de l'élément <lb/>, sans reprendre la marque de la césure afin de faciliter une sortie HTML ne respectant pas les passages à la ligne. Il est en effet plus commode de rajouter la marque de la césure, si l'on désire un affichage diplomatique, lorsque sa localisation est identifiée que de la supprimer. Toutefois cette solution n'était pas totalement satisfaisante dans la mesure où certains mots ont été coupés sans que la césure ne soit signifiée. Or si l'on peut penser que l'élément <lb/> suffit il était toutefois intéressant de baliser cette pratique de manière à la faire ressortir. Ainsi lorsque la césure est marquée, nous avons ajouté à l'élément <lb/> l'attribut @type="marked". Dans le cas contraire seul l'attribut @rend à été utilisé.

```
<!--Ici la césure était matérialisée dans la source-->
<lb/>la figure, qui est ce que nous
aper<lb rend="hyphen" type="marked"/>çevons qui planait au dessus de nous

<!--Ici la césure n'a pas été matérialisée dans la source-->
<lb/>la figure, qui est ce que nous
aper<lb rend="hyphen"/>çevons qui planait au dessus de nous
```

Mise en valeur

Afin de rendre compte de la particularité physique de certains mots ou segments de phrase (centrage, soulignement, mise en exposant, etc.) nous avons utilisé l'attribut *@rend* qui peut être appliqué par tout élément TEI ; les valeurs diffèrent en fonction du cas de figure, "center"(centré), "underline"(souligné), "overline"(surligné), "super"(exposant) etc. Ainsi, lorsque l'élément était déjà balisé, l'attribut *@rend* a simplement été ajouté. Sinon nous avons utilisé l'élément `<hi>` de la TEI qui « distingue un mot ou une expression comme graphiquement distincte du texte environnant, sans en donner la raison » et l'attribut *@rend* de la même manière.

```
<opener>
  <dateline>
    <placeName>Imbros</placeName>,
    <date>le 23 7<hi rend="super">bre</hi> septembre 1915</date>
  </dateline>
  <salute>
    <persName rend="center">Comtesse</persName>,
  </salute>
</opener>
```

Cet élément `<hi>` n'a en revanche pas été utilisé pour la normalisation des majuscules ou des minuscules comme c'est parfois l'usage. Nous avons préféré dans ce cas l'élément `<choice>` complété par les balises `<reg>` et `<orig>` pour les versions modernisées et originales. De même l'attribut *@rend* doit être compris comme décrivant la représentation de la source et non la manière dont on souhaite mettre en forme ces éléments pour la publication.

Les ajouts en marge

C'est un cas particulier, que nous avons rencontré à de multiples reprises lors de l'encodage de la correspondance d'Armand Horel. Les cartes postales sont des supports relativement petits. Lorsque ses messages étaient trop longs et qu'il était arrivé au bout de sa seconde ou troisième carte sans avoir la place de saluer son destinataire ou de signer sa correspondance, il ajoutait ces mentions en marge des cartes précédentes, à la verticale ou à l'envers. Afin de signifier ces éléments qui correspondent exclusivement à des signatures, formules de politesse et postscriptum, les éléments TEI dédiés à ces portions de texte n'ont pu être utilisés car, si nous avons bien opéré des regroupements logiques pour l'encodage des lettres, les cartes n'en n'ont pas moins été traitées à la pièce et le schéma TEI impose des

règles dans l'ordre d'utilisation de certains éléments. Par exemple, la balise <closer> qui permet l'encodage des formules de politesse ou de la signature ne peut être placée avant un paragraphe (<p>).

Nous avons trouvé une solution par l'utilisation des balises <seg> et <add>. La première peut « [contenir] une unité de texte quelconque de niveau 'segment' ». En utilisant l'attribut @type de valeur "closer" ou "postscript" nous avons pu qualifier les informations ajoutées. Les éléments <add> ont ensuite été utilisés pour baliser chaque ajout identifié. L'élément <add> a la particularité de pouvoir être complété par l'attribut @place qui localise l'ajout sur le document. Les valeurs "margin-right", "margin-left", "margin-top" et "margin-bottom" pour indiquer les marges droites, gauches, supérieures et inférieures ont été employées. Comme nous l'avons vu, l'attribut @rend permet de préciser l'orientation du texte ("vertical" ou "rotate") si besoin. Enfin, concernant la signature l'élément <add> peut être typé, nous lui avons donc attribué la valeur "signed".

```
<seg type="closer">
  <add place="margin-top" rend="rotate">
    <lb/>Je suis en parfaite santé et souhaite que vous soyez de
    <lb/>même à l'arrivée de mes deux cartes,
  </add>
  <add place="margin-bottom" rend="rotate">
    un petit ami qui ne vous oublie
    <lb/>pas,
  </add>
  <add type="signed" place="margin-right" rend="vertical">
    <persName>A Horel</persName>
  </add>
</seg>
```

4.2.3 Balisage diplomatique

Le texte d'une carte postale présente fréquemment une structure similaire aux missives plus classiques dans laquelle on retrouve une mention de date et de lieu, une formule de salutation, puis les paragraphes de texte, les formules de politesse ou de courtoisie, la signature et éventuellement un postscriptum. Sur ces points la *Text Encoding Initiative* propose des solutions particulièrement bien adaptées aux correspondances contemporaines par les éléments <opener>, <closer> et <postscript>².

Nous avons pu constater certains cas où cette structure ne ressortait pas, non pas que les salutations ou que les signatures n'y étaient pas, mais plus simplement

2. *TEI guidelines*

parce que le verso de la carte a été rédigé comme un paragraphe afin d'utiliser toute la place possible et ne laisser aucun vide. Il était toutefois nécessaire de faire ressortir ces différents éléments lors de l'encodage car ils favoriseront la mise en forme du texte pour la publication.

<opener> (formule de début) regroupe la date, la mention de responsabilité, la formule de politesse et d'autres expressions de ce type dans un groupe préliminaire au début d'une division, en particulier au commencement d'une lettre.

De par sa définition, cet élément est adapté aux formules se trouvant avant le texte de la lettre. Si elle peut contenir une liste importante de balises, deux, nous intéressent plus particulièrement, <dateline> et <salute>. le premier est défini comme « [contenant] une brève description des lieux, date, heure, etc. concernant la production d'une lettre, d'un article de journal ou d'un autre texte qui, placé au début ou à la fin, lui est associé comme en-tête ou annonce de fin ». Il sera naturellement complété par l'utilisation de ses sous-éléments <placeName> et <date>. La balise <salute> quant à elle, est dédiée aux formules de politesse qui précèdent un avant propos, ou qui concluent un texte ou une lettre. Elle a été également utilisée pour les apostrophes.

```
<opener>
  <dateline>
    Bord <placeName>Salonique</placeName>, <date>le 15 Avril 1916</date>
  </dateline>
  <salute>
    <lb/><persName>Comtesse</persName>,
  </salute>
</opener>
```

La *Text Encoding Initiative* ne prévoit en revanche pas d'élément propre pour l'encodage des formules de politesse ou de courtoisie que l'on retrouve généralement à la fin des missives. Elles doivent être balisées avec le même élément <salute>. Ce dernier ne peut toutefois pas être complété par l'attribut @type et la seule distinction concernant le segment de texte encodé se fera donc en fonction de l'élément parent, à savoir s'il s'agit d'un <opener> ou d'un <closer>.

L'utilisation de l'élément <postscript> a en revanche soulevé plus d'interrogation. Le schéma définit cet élément comme « [contenant] un post-scriptum, par exemple au bas d'une lettre ». Si cette définition est précise quant à ce dont il s'agit, son usage est très contraint ou plutôt contraignant. En effet, dans plusieurs correspondances l'auteur termine sa lettre, signe et ajoute un petit postscriptum. Ce cas de figure ne pose pas de problème et peut être encodé comme suit :

```

<p>[...]</p>
<closer>
  <salute>
    Rien autre chose à vous dire pour l'instant, un petit ami qui vous envoie ces meilleurs
    souvenirs de <placeName>Salonique</placeName>.
  </salute>
  <signed><persName>A Horel</persName></signed>
</closer>
<postscript n="1">
  <p>Je crois que cette maudite <placeName>Grèce</placeName> ne m'inspire guère confiance,
  c'est une neutralité hypocrite.
  </p>
</postscript>

```

En revanche la suite de la lettre est plus intéressante, car avant de terminer, il reçoit une réponse de son destinataire et décide d'ajouter un nouveau développement relativement long prenant presque la forme d'une nouvelle missive. Or la TEI est particulièrement contraignante puisque dès lors que nous avons utilisé une fois un élément `<closer>` et/ou `<postscript>` il n'est plus possible d'utiliser l'élément paragraphe `<p>` librement. Cela oblige à l'utilisation, d'un nouvel élément `<postscript>`. Une solution alternative serait de ne pas utiliser d'éléments `<closer>` ou `<postscript>` pour la première partie mais d'utiliser une balise `<seg>` précisée par l'attribut `@type`. Toutefois cette solution n'est pas, non plus, totalement satisfaisante car elle force à l'utilisation d'éléments moins spécifiques alors que d'autres sont adaptés.

La TEI semble toutefois palier à ce problème en donnant la possibilité à son élément `postscript` de la prendre la forme d'une lettre, commençant par une balise `<opener>` comportant son développement dans un élément `<p>` et terminant par un autre élément `<closer>`. C'est cette solution que nous avons retenue.

Enfin un dernier élément a attiré notre attention et trouvé sa solution dans une personnalisation plus poussée du schéma TEI. Il s'agit de la question des adresses. Il est important dans une édition de correspondance de savoir à qui correspondent les adresses. Or la TEI ne permet de typer l'élément `<address>` dédié à ce type d'information. Une solution serait de placer cet élément `<address>` dans un élément `<seg>` afin de lui associer un attribut `@type="address"` et un attribut `@subtype` indiquant s'il s'agit de l'expéditeur ou du destinataire. Si cette solution permet d'aller un peu plus loin dans la description des éléments elle correspond encore à une solution incomplète car aucun attribut `@place` permettant de situer l'adresse ne peut être inclus. Nous avons donc décidé d'ajouter deux attributs à l'élément `<address>`, inexistants pour l'heure dans le schéma TEI : `@type` afin d'identifier

l'expéditeur ou le destinataire et *@place* pour la place qu'occupe cette information à savoir si elle est placée sur une enveloppe dans le texte, sur l'espace dédié dans une carte postale etc.

Il faut se rappeler qu'une telle personnalisation nuit à l'interopérabilité de la modélisation et de l'encodage effectué. Toutefois cela répond à un besoin auquel le schéma TEI ne permet pas encore de faire face. La TEI évolue avec ce genre de proposition, l'ajout récent de l'élément `<correspDesc>` en témoigne, toutefois, cela peut prendre du temps et reste hypothétique mais cela peut tout de même constituer une solution d'attente.

```
<address type="deliveredTo" place="postcard">
  <persName>Comtesse L. de la Forest</persName>
  <addrLine>Bonne-sur-Menoge</addrLine>
  <addrLine>Haute-Savoie</addrLine>
  <addrLine>France</addrLine>
</address>
```

4.3 Normalisation du texte

Lors de la transcription du texte, l'encodeur dispose de plusieurs possibilités quant à la gestion d'une orthographe fautive qui, faut-il le rappeler, est le reflet de la condition de l'auteur. Le transcritteur peut tout d'abord décider de moderniser cette orthographe "au fil de l'eau", lors de l'encodage du texte, perdant ainsi toute trace de la graphie originale. Pour notre édition nous avons cherché à perdre le moins d'informations possible et le schéma TEI propose une solution adaptée. En effet, il est possible de proposer une correction orthographique du texte tout en maintenant la forme fautive ou ancienne de l'auteur. L'élément `<choice>` autorise de regrouper des balisages alternatifs (au moins deux) pour certains segments du texte. Il est également possible d'affiner sa correction en fonction des cas de figure. Il est parfois intéressant de distinguer les fautes d'orthographe récurrentes ou systématiques, de celles plus isolées. Dans la première éventualité, il peut s'agir par exemple d'un auteur qui confond systématiquement l'infinitif des verbes du premier groupe et leur participe passé ou encore qui utilise le verbe avoir à la place de la préposition "à". Nous avons établie que ce type de fautes devait être encodé avec les éléments `<orig>` pour la graphie originale, et `<reg>` pour la forme corrigée. Concernant les fautes plus éparses, sur le même principe nous utiliserons alors les éléments `<sic>` pour l'orthographe fautive et `<corr>` pour la correction. cette même solution de la balise `<choice>` peut être employée pour le rétablissement de la ponctuation, des abréviations etc. (*voir annexe A manuel d'encodage*). Cette solution est

très souple car elle permet avec l'application d'une feuille de style différente de produire plusieurs éditions à partir du même document. Les principes éditoriaux peuvent alors intervenir au dernier moment toutefois l'encodage est un peu plus lourd.

4.4 Réalisation d'une table des correspondances

Il nous fallait également trouver un moyen de réaliser une table des correspondances poursuivant un triple objectif, permettre un classement des correspondances en fonction du lieu d'envoi, de leur date et de leur expéditeur ou destinataire — Dans la mesure où toutes les lettres étaient adressées à la Comtesse de la Forest nous avons opté pour les expéditeurs. Une édition de correspondances doit contenir de telles tables afin de permettre un accès rapide aux différentes missives. Traduire ces tables par le biais de la TEI a été l'un des enjeux majeurs de ce stage. En effet, il faut rassembler toutes les informations permettant l'identification de chaque carte (expéditeur, destinataire, date et lieu d'envoi). Si cela ne crée pas de difficultés de cibler toutes ces informations avec une feuille de transformation XSLT pour la création d'un unique fichier de sortie, cela devient en revanche un obstacle pour l'édition d'une correspondance qui demande un traitement individualisé de chaque texte. Xinclude ne permet pas d'interroger un fichier de la même manière qu'XSLT. Il nous a donc fallu trouver un élément pouvant être ciblé par Xinclude et intégré dans nos différents fichiers XML dédiés à la transformation HTML. Nous avons d'abord pensé à l'utilisation d'un élément `<list>` afin d'y rassembler les informations propres à chaque correspondance ainsi que les différents liens. Toutefois, cette solution n'est pas assez souple ; en effet, elle permet de créer une table chronologique mais plus difficilement de rassembler les tables en fonction d'un lieu ou d'une personne (expéditeur ou destinataire).

Nous sommes donc repartis de l'intégration récente dans le schéma TEI de l'élément `<correspDesc>` dédié à la description des actions relatives à un acte de correspondance. Sur ce point, nous l'avons utilisé d'une manière non directement prévue par la TEI, en y rassemblant chaque acte de correspondance. Nous avons fait en sorte de réunir au sein de ce `<correspDesc>` tous les éléments permettant de créer une table chronologique, une table géographique et une table des expéditeurs (nous pouvions également créer une table des correspondants, mais dans la mesure où toutes les lettres étaient adressées à la même personne, cela ne semblait pas pertinent mais reste néanmoins faisable). Comme il s'agit d'une table globale nous avons placé cette balise `<correspDesc>` disposant d'un attribut `@type="sommaire"` au sein de l'élément `<teiHeader>` relatif au corpus. L'élément `<correspAction>` a été utilisé pour définir les différents rôles joués par les protagonistes. Ce sont bien

les rôles qui doivent être décrit et non les protagonistes, Ainsi, si nous avons un des interlocuteurs qui a envoyé et reçu des lettres, cela sera encodé dans deux éléments <correspAction> différents et signifié par un attribut @type d'une valeur "sent" ou "received" suivant qu'il soit qualifié d'expéditeur ou de destinataire et d'un attribut @subtype="global". Au sein de cet élément <correspAction>, le nom de la personne (destinataire ou expéditeur) ont été encodés avec un élément <persName> précisé par un attribut @xml:id. Les lieux et date sont respectivement encodés dans des éléments <placeName> et <date>, les différentes dates, correspondant à une session d'envoi sur un lieu donné sont identifiées par un @xml:id.

```
<correspAction type="sent" subtype="global">
<!--C'est le rôle d'expéditeur d'Armand Horel qui est décrit ici-->
  <persName ref="#Horel" xml:id="sent_Horel">Armand Horel</persName>

  <!--Chaque lieu où le protagoniste a envoyé ses cartes et les sessions d'envoi sont rappelés et
    identifiées par un xml:id.-->
  <placeName ref="#Salonique">Salonique</placeName>
  <date corresp="#Salonique" when="1915-05-15" xml:id="envoi_salonique_1"/>
  <date corresp="#Salonique" from="1915-11-16" to="1916-11-04"
    xml:id="envoi_salonique_2"/>

  <placeName ref="#Corfou">Corfou</placeName>
  <date corresp="#Corfou" when="1916-11-04" xml:id="envoi_corfou_1"/>

  <placeName ref="#Mytilene">Mytilene</placeName>
  <date corresp="#Mytilene" when="1915-10-06" xml:id="envoi_mytilene_1"/>

  <placeName ref="#Imbros">Imbros</placeName>
  <date corresp="#Imbros" when="1915-09-23" xml:id="envoi_imbros_1"/>

  <placeName ref="#Moudros">Moudros</placeName>
  <date corresp="#Moudros" when="1915-09-03" xml:id="envoi_moudros_1"/>

  <placeName ref="#Toulon">Toulon</placeName>
  <date corresp="#Toulon" from="1915-08-08" to="1915-08-29" xml:id="envoi_toulon_1"/>

  <placeName ref="#Port_Iero">Port Iero</placeName>
  <date corresp="#Port_Iero" from="1915-10-09" to="1915-11-10" xml:id="envoi_portiero_1"/>
</correspAction>
```

Dans un deuxième temps un élément <correspContext> a été créé par lieu d'envoi et comprenant une liste de sous-éléments <ref> comportant un @xml:id (qui nous serviront pour l'index), un attribut @type="context" et enfin attribut @corresp relié aux différentes dates listées dans les éléments <correspAction> (cela nous

servira à établir une table chronologique des correspondances organisée par lieu). Chaque élément <ref> doit être complété par les informations d'identification de chaque cartes, l'expéditeur doit être encodé avec la balise <persName> précisée par l'attribut @type="sentBy" et un attribut @corresp lié à l'@xml:id lié à un des <correspAction>. Le même traitement est réalisé pour le destinataire. Enfin, on retrouve la date et le lieu, qui ne pose pas de problème quant à leur encodage.

Ce système permet d'inclure l'ensemble du correspDesc dans tous les fichiers de sortie destinés à être transformés en page HTML. Il permet par les différents liens qui sont opérés entre les dates, les personnes et les lieux d'envoi, de réaliser des Tables en fonction de n'importe laquelle des informations d'identification et de les inclure dans un menu déroulant accessible depuis toutes les pages du site.

```
<correspContext xml:id="context_corfou">
<!-- Les éléments <ref> sont liés par leur attribut @corresp aux éléments <date> contenus dans
un <correspAction> -->
<ref xml:id="Corfou_04-11-1916" type="context"
target="../Corfou/F_delta_1854_20_1_Corfou_04-11-1916.xml" corresp="#envoi_corfou_1">
<!-- Le même principe est retenu pour les personnes -->
<persName type="sentBy" ref="#Horel" corresp="#sent_Horel">Armand Horel</persName>
<persName type="deliveredTo" ref="#Comtesse_Forest" corresp="#delivered_Comtesse">la
Comtesse L. de la Forêt</persName>
<placeName ref="#Corfou">Corfou</placeName>
<date when="1916-11-04" cert="high">04 novembre 1916</date>
</ref>
</correspContext>

<correspContext xml:id="context_mytilene">
<ref xml:id="Mytilene_06-10-1915" type="context"
target="../Mytilene/F_delta_1854_20_1_Mytilene_06-10-1915.xml"
corresp="#envoi_mytilene_1">
<persName type="sentBy" ref="#Horel" corresp="#sent_Horel">Armand Horel</persName>
<persName type="deliveredTo" ref="#Comtesse_Forest" corresp="#delivered_Comtesse">la
Comtesse L. de la Forêt</persName>
<placeName ref="#Mytilene">Mytilene</placeName>
<date when="1915-10-06" cert="high">06 octobre 1915</date>
</ref>
</correspContext>
```

Pour créer une table des correspondants par exemple, il suffit alors, avec le processeur XSLT, de sélectionner tous les éléments <persName> contenus dans une balise <correspAction> comportant un attribut @type="received", puis de récupérer dans une variable l'@xml:id attribué à cette personne puis de le comparer avec l'attribut @corresp des balises <persName> comportant un attribut

@type="deliveredTo" inclus dans les éléments <ref> des différentes balises <correspContext>. le même principe peut être retenu pour les dates ou les lieux.

Cette utilisation n'est pas directement prévue par la TEI et constitue, de notre part, une interprétation large de ces éléments. Toutefois ce système a fait ses preuves et il est relativement souple tout en permettant de créer des tables en fonction de n'importe laquelle des informations d'identification et de les regrouper par lieu ou date.

4.5 La création d'un index

L'index est un instrument utile pour le chercheur, très complexe à mettre en œuvre mais néanmoins obligatoire pour toute édition scientifique. Sa mise en place nécessite d'y réfléchir afin de savoir ce que l'on veut en faire. Sur ce point l'édition électronique et l'encodage TEI encourage le développement d'un Index/glossaire, afin de rassembler l'identification et la définition de certains noms de lieu ou de personne. Dans le cadre de l'édition de la correspondance d'Armand Horel, trois types d'entrées ont été retenus ; les noms de personnes, les noms de lieux et les bâtiments de flottes militaires. L'objectif de cet index était de fournir les informations d'identification générale des différentes entrées, permettre le renvoi vers les différentes cartes et enfin être en mesure d'apporter ces informations pour des notes dans le texte lors d'un clic sur les différents termes (*voir Annexe B Figure B.1.*).

Dans la mesure où cet index intéressait l'ensemble de notre corpus, il a été créé comme un document TEI autonome intégré dans notre *teiCorpus*.

```
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <!--[en-tête du corpus pouvant contenir les différentes métadonnées communes]-->
  </teiHeader>
  <TEI>
    [les différents index/glossaires]
  </TEI>
  <TEI>
    [Première carte]
  </TEI>
  <TEI>
    [...]
  </TEI>
</teiCorpus>
```

Cet index est donc composé de trois éléments de même niveau <list>, <listPlace>, et <listPerson>, contenant chacun une liste d'éléments <item>, <place> et

<person> tous dotés d'un attribut `@xml:id` qui nous permettra de lier les différentes entrées du texte avec ces vedettes.

Pour la création de nos fichiers de sortie, ces trois index ont été intégrés par le biais d'Xinclude afin, lors de la transformation, de permettre de récupérer des informations pour des notes. Ils ont été intégrés dans l'élément <back>.

4.5.1 Indexation des entrées du texte

Lier les entrées du texte avec les vedettes de l'index n'était pas complexe dans la mesure où ces dernières disposent toutes d'un attribut `@xml:id`, il suffisait de le cibler avec l'attribut `@ref` des entrées du texte. Toutefois, nous souhaitions que l'index renvoie directement vers la première occurrence d'une lettre donnée. Pour ce faire il fallait donner à cette circonstance un identifiant `@xml:id` qui sera à son tour lié à l'index, qui servirait également d'ancre lors de l'application de la feuille de transformation XSLT.

```
<dateline>
  <!-- Comme nous allons le voir l'attribut @xml:id sera utilisé par les liens retour de l'index afin
        de cibler précisément cette première occurrence -->
  Bord <placeName ref="#Salonique" xml:id="Salonique_21041916">Salonique</placeName>,
  <date when="1916-04-21">le 21. 4. 1916</date>
</dateline>
```

D'une manière générale les entrées du texte, lorsqu'elles sont bien identifiées, — c'est à dire lors qu'elles sont nommées dans le texte — ne sont pas difficiles à indexer, les personnes, lieux ou navire, sont logiquement balisés avec les balises idoines : <persName>, <placeName>, et <name> — pour cette dernière l'utilisation d'un attribut `@type="navire"` est requis —. En revanche cela se complique lorsque ces personnes ou lieux ne sont que suggérés ou tout du moins nommés de manière indirecte avec des formules comme, "mon frère", ou "Mme la Présidente"; si la lecture permet d'identifier ces personnes comme étant Louis Horel, ou Henriette Poincaré, elles doivent faire partie des entrées d'index. Toutefois, l'élément <persName> ne peut être utilisé car il « contient un nom propre ou une expression nominale se référant à une personne, pouvant inclure tout ou partie de ses prénoms, noms de famille, titres honorifiques, noms ajoutés, ». Nous avons donc utilisé la balise <rs> qui « contient un nom générique ou une chaîne permettant de s'y référer » que nous avons précisée avec l'emploi d'un attribut `@type="person"`.

```
<p>
  au sujet de <rs type="person" ref="#Horel_Louis" xml:id="Horel_Louis_16021916">mon
  frère</rs>,</p>
```

```
<persName ref="#Horel" xml:id="Horel_16021916">Armand Horel</persName>,
</p>
```

4.5.2 L'index des noms de personnes

Comme nous l'avons vu, cet index est constitué d'une liste d'éléments `<person>` tous identifiés par un attribut `@xml:id`. Chaque balise `<person>` peut contenir les sous-éléments suivants : `<persName>` pour les patronymes, `<state>` pour qualifier l'état d'une personne, `<birth>` et `<death>` pour ses dates de naissance et de mort, `<listEvent>` contenant des faits marquants de la vie de cette personne et enfin `<link>` permettant d'opérer les liens vers les différentes lettres.

Nous avons fait le choix de bien structurer les différentes entrées, en distinguant notamment les prénoms, noms de famille, titres etc. au sein de l'élément `<persName>` ou les différentes données biographiques avec l'élément `<listEvent>`. Cet encodage est certes lourd mais il permet de réaliser des entrées d'index très précises, pouvant être réutilisées pour d'autres bases. Le traitement des ces informations étant complexe, il est important que l'ordre de ces patronymes respectent les conventions requises pour l'établissement des vedettes d'index. Hors cas particuliers, les entrées se font au nom de famille (`<surname>`) suivi du prénom (`<forename>`) et du titre (`<rolename>`), par exemple :

```
<person xml:id="Putnik">
  <persName>
    <surname>Putnik</surname>
    <forename>Radomir</forename>
  </persName>
  <state>
    <p>Voïvode (Maréchal)</p>
  </state>
  <birth when="1847"/>
  <death when="1917"/>
  <link type="cit" corresp="#Salonique_07-12-1915" target="#Putnik_07121915"/>
</person>
```

En revanche, l'indexation des noms de souverains se fait à leur prénom.

```
<person xml:id="Pierre_Serbie">
  <persName type="nobility">
    <!-- Sur ce type d'entrée il est important de bien structurer l'information. Cela facilite
         grandement le traitement de sortie -->
```

```

    <forename>Pierre</forename>
    <genName>Ier</genName>
    <nameLink>de</nameLink>
    <placeName>Serbie</placeName>
  </persName>
  <state>
    <p><roleName type="nobility">Roi de Serbie</roleName></p>
  </state>
  <birth when="1844"/>
  <death when="1921"/>
  <listEvent>
    <event>
      <p>Saint-cyrien de la promotion Plevna (1862–1864)</date></p>
    </event>
    <event>
      <p>Engagé volontaire dans la légion étrangère en 1870 pour la défense de la France</p>
    </event>
    <event><p>Monte sur le trône en 1903</p></event>
    <event>
      <p>Malade en 1914, il se résout à transmettre la régence et le commandement militaire à
        sont fils, Alexandre Ier.</p>
    </event>
  </listEvent>
  <link type="cit" corresp="#Salonique_03-01-1916" target="#Pierre_Serbie_03011916"/>
</person>

```

Lorsqu'une personne est connue sous différents patronymes, l'entrée doit s'effectuer sous sa forme la plus connue. Nous avons donc mis en place un système de renvoi. Ainsi, l'élément <persName> doit comporter un attribut @type de valeur "renvoi" et un attribut @corresp renvoyant vers l'identifiant @xml:id de sa forme acceptée.

```

<person xml:id="Comtesse_Forest">
  <persName type="renvoi" corresp="#Silve_Claude">
    <surname>La Forest Divonne</surname>
    <forename>Philomène</forename>
    <roleName type="nobility">Comtesse</roleName>
    <nameLink>de</nameLink>
  </persName>
</person>
<person xml:id="Silve_Claude">
  <persName>
    <surname>Silve</surname>
    <forename>Claude</forename>
  </persName>
  <birth when="1887"/>

```

```

    <death when="1978"/>
    [...]
  </person>

```

4.5.3 Index des noms de lieux

La réalisation de cet index a soulevé d'autres problèmes spécifiques. En effet si sa structure globale est basée sur le même principe et ne pose pas de problème particulier, il est important que ses entrées soient localisées de la manière la plus précise possible en partant « du général au particulier »³. Ces vedettes nécessitent donc d'être replacées dans leur contexte géographique, en tenant compte des différents découpages administratifs (*Annexe B Figure B.2.*). Si cela ne pose pas de difficultés lors de la saisie, il en est autrement lors de la phase de traitement et de mise en forme de ces informations car cela oblige quasiment à créer une exception de traitement par entrée. Afin d'alléger ce traitement il a été décidé de structurer les informations directement dans l'index.

```

<place xml:id="Moudros">
  <placeName type="etranger">Moudros</placeName>
  <location>
    <country>Grèce</country>
    <district type="peripherie">Egée–Septentrionale</district><!-- région -->
    <!-- Les entités administratives sont directement rappelées dans l'information textuelle,
         plutôt que par l'attribut @type de l'élément <district> -->
    <district type="district">dist. Lemnos</district><!-- district régional -->
    <district type="deme">Lemnos</district><!-- communauté -->
  </location>
  <event>
    <p>Le 30 octobre 1918, l'Empire ottoman y signe avec le Royaume–uni un armistice mettant
      fin aux combats entre la triple Entente et la Turquie.</p>
  </event>
  <link type="cit" corresp="#Toulon_29–08–1915" target="#Moudros_29081915"/>
  <link type="cit" corresp="#Moudros_03–09–1915" target="#Moudros_03091915"/>
</place>
<place xml:id="Imbros">
  <placeName type="etranger">Imbros</placeName>
  <placeName type="today">Gökçeada</placeName></placeName>
  <location>
    <country>Turquie</country>
    <district type="region">Marmara</district><!-- région -->
    <district type="province">prov. Çanakkale</district><!-- district régional -->
  </location>

```

3. C. Nougaret et É. Parinet, *L'édition critique des textes contemporains (XIXe-XXIe siècle)*, p. 138.

```
<district type="district">dist. Gökçeada</district><!-- communauté -->  
</location>  
<link type="cit" corresp="#Imbros_23-09-1915" target="#Imbros_23091915"/>  
</place>
```


Chapitre 5

Personnalisation TEI

L'ensemble des besoins ayant été déterminés et les solutions décelées et adaptées pour leur encodage, il est nécessaire ensuite, afin de garantir un balisage homogène, de contraindre le schéma TEI. Nous avons réalisé un fichier ODD (*One Document Does It All*) qui rassemble dans un seul document XML, le plan de la personnalisation et toute la documentation attenante. Il permettra ensuite de générer un schéma RelaxNG ainsi que toute la documentation au format HTML, PDF ou ODT.

Le schéma TEI doit être contraint, d'une part pour standardiser les pratiques d'encodage et favoriser les échanges mais surtout pour permettre le traitement automatisé des fichiers par la suite. Comme nous l'avons déjà précisé, « un fichier XML est, mais il ne fait rien »¹. Il requiert d'être transformé afin de pouvoir être interprété par un programme ou un navigateur web. Cette transformation ne peut s'opérer de manière automatique qu'en présence de fichiers construits sur le même modèle. Un schéma procure également, à l'encodeur, une assistance pour autant qu'il l'utilise conjointement avec un logiciel comme *Oxygen*, il pourra valider son travail au regard du schéma en lui fournissant une part de la documentation nécessaire.

5.1 Création d'un fichier ODD

Il existe trois possibilités pour obtenir un fichier ODD : la rédaction manuelle, l'utilisation d'une feuille de transformation XSL (*Odd By Example*) et enfin l'exploitation du logiciel Roma. L'écriture manuelle, si elle est intéressante, n'en demeure pas moins un long et fastidieux exercice. Dans la mesure où nous avons réalisé un *teiCorpus* associant l'ensemble des pratiques, l'application d'une feuille de style est un vecteur rapide pour la création de ce document. En revanche elle n'offre que

1. E. R. Harold et W. S. Means, *XML en concentré*.

peu de souplesse d'utilisation. Pour exemple, si certaines classes d'attributs n'ont jamais été utilisées, la feuille de style en bannit l'utilisation purement et simplement. De plus, un inconvénient dans la transformation du format ODD vers le Schéma RelaxNG, ayant été repéré avant le début de ce stage, elle n'a pas été utilisée. Le logiciel Roma, accessible en ligne, permet de sélectionner, par une interface graphique, les différents modules, éléments et attributs souhaités pour notre personnalisation. C'est donc cette solution qui a été privilégiée. En revanche, la prise en main se révèle fastidieuse notamment s'agissant des personnalisations les plus poussées pour lesquelles, nous avons nettement préféré la méthode de rédaction manuelle.

5.1.1 Utilisation de Roma

Roma est un logiciel en ligne mis à disposition par le consortium TEI qui favorise les personnalisations de la TEI². Il permet de générer des schémas ainsi que des fichiers ODD à partir d'une personnalisation. Elle offre plusieurs choix pour contraindre le schéma TEI :

- Créer son schéma en ajoutant des éléments
- Créer son schéma en retirant des éléments
- Créer son schéma à partir d'une liste des modèles déjà existant.
- Enfin elle offre la possibilité de modifier son propre schéma.

Nous avons commencé la création par l'ajout d'éléments. Ce choix propose de démarrer sa personnalisation à partir des quatre modules obligatoires de la *Text Encoding Initiative* : TEI, core, textstructure, et header. Il est ensuite possible d'ajouter des modules en fonction des éléments qui seront nécessaires, à notre projet (voir *Annexe B Figure B.6.*). Sur la vingtaine de modules disponibles, seul, onze nous intéressaient. Cela augurait déjà des erreurs possibles lors de l'encodage des documents si certains n'avaient pas été retirés. Il est ensuite possible de modifier, la liste des balises disponibles dans chaque module (voir *Annexe B Figure B.7.*).

Enfin, il convient de faire de même avec les attributs. Sur ce point deux possibilités sont proposées. Soit la modification s'effectue par élément, ce qui peut s'avérer très fastidieux, soit, si certaines *class* ne sont jamais utilisées, de les exclure d'une manière globale.

Il est également possible de créer des éléments en leur attribuant certaines *class* d'attributs mais cette fonctionnalité n'a pas été testée.

Une fois tous les choix effectués il est possible de créer un fichier ODD en cliquant sur l'onglet « enregistrer », ou de générer un schéma avec l'onglet idoine.

2. <http://www.tei-c.org/Roma/>

5.1.2 Personnalisation manuelle du schéma

Une fois le Fichier ODD généré, il contient le bagage nécessaire à la production d'un schéma ainsi que sa documentation TEI. Toutefois il est également possible de le modifier manuellement sans repasser par Roma dont l'interface est parfois envahissante. Un fichier ODD est un fichier XML.

La personnalisation du schéma TEI est définie au sein d'une balise `<sche-
maSpec>`.

Elle se subdivise ensuite en une liste d'éléments `<moduleRef>`, `<classSpec>` et `<elementSpec>`.

Les éléments `<moduleRef>` servent à déclarer les modules intégrés à la personnalisation et identifiés par l'attribut `@key`. Un second attribut, `@except` liste ensuite les éléments exclus du module. Si l'on désire en ajouter un, il suffit de le retirer de cette liste.

```
<moduleRef key="textstructure" except="byline div1 div2 div3 div4 div5 div6 div7 docAuthor  
docDate docEdition docImprint docTitle epigraph floatingText imprimatur titlePage titlePart  
trailer"/>
```

La balise `<classSpec>` est dédiée aux *class* d'attributs. L'attribut `@ident` permet de nommer la *class*, l'attribut `@module` identifie le module auquel la *class* se rapporte, et enfin l'attribut `@mode` indique les modifications qui sont apportées ; les valeurs peuvent être "add", "change", "delete", "replace".

Cette balise peut contenir un sous élément, `<attList>` comprenant l'ensemble des attributs ciblé par la *class*.

```
<classSpec ident="att.tableDecoration" module="tei" mode="change" type="atts">  
  <attList>  
    <!--Les attributs sont identifiés avec l'identifiant @ident -->  
    <attDef ident="cols" mode="delete"/>  
    <!--L'attribut @mode indique la modification apportées à l'attribut ciblé -->  
    <attDef ident="role" mode="delete"/>  
    <attDef ident="rows" mode="change">  
      <desc>  
        <!--Il est également possible de modifier les description des attributs-->  
      </desc>  
      <valList mode="add" type="closed">  
        <!--Il est également possible de proposer des valeurs d'attribut prédéfinies. @type  
          permet de préciser si cette liste est ouverte ("semi") ou close ("closed")-->  
        <valItem ident="nom_de_la_valeur"/>  
      </valList>  
    </attDef>  
  </attList>
```

```
</classSpec>
```

Il est possible de modifier les attributs relatifs à un élément particulier. C'est le rôle de la balise `<elementSpec>`. Elle s'utilise de la même manière que les deux éléments précédents. L'exemple suivant est relatif à l'ajout de deux attributs (`@type` et `@place` pour l'élément TEI `<address>`).

```
<elementSpec ident="address" module="core" mode="change">
<!--On cible l'élément address contenu dans le module "core" et on indique qu'il va être modifié
("change") -->
<attList>
<!--On définit le nom de l'attribut et on indique qu'il s'agit d'un ajout ("add")-->
<attDef ident="type" mode="add">
<desc>Dans le cas de correspondance cet attribut permet de préciser l'origine de cette
adresse.</desc>
<!--avec l'attribut @type="closed" on indique que la liste des valeurs acceptées est
limitée.-->
<valList type="closed">
<valItem ident="addressee"/>
<valItem ident="sender"/>
<valItem ident="other"/>
</valList>
</attDef>
<attDef ident="place" mode="add">
<desc>Précise la localisation de l'adresse.</desc>
<valList type="semi">
<valItem ident="letter"/>
<valItem ident="back"/>
<valItem ident="postcard"/>
<valItem ident="envelope"/>
</valList>
</attDef>
</attList>
</elementSpec>
```

Cette personnalisation manuelle nous a permis d'intégrer, dans notre schéma, les valeurs d'attribut acceptées ou suggérées pour chaque élément de notre modélisation.

5.1.3 Rédaction d'un manuel d'encodage

Afin d'aider l'encodeur dans son travail, il est intéressant de lui fournir un manuel d'encodage contenant des explications sur l'utilisation de certaines balises et des exemples d'application.

Ce guide d'encodage peut être intégré dans un fichier ODD ainsi ce dernier comportera toute la documentation nécessaire à l'utilisation du schéma qu'il contient.

Cette documentation ne pose pas de problème particulier dans sa rédaction. Elle doit être placée dans l'élément <body> du fichier ODD après la balise <schemaSpec> que nous avons vue plus haut. Les niveaux de chapitre sont donnés par l'imbrication d'éléments <div1>, <div2>, <div3>, <div4> etc. Le titre de chacune de ces divisions est donné par un élément <head> alors que le texte est placé logiquement dans une balise de paragraphe <p>.

```
<div1>
  <head>Manuel d'encodage en XML–TEI</head>
  <div2>
    <head>Pr&#xE9;ambule</head>
    <p>
      Ce manuel constitue une aide pour le chercheur lors de son travail d'encodage. Il présente
      des explications sur certains points techniques mais il n'a pas vocation à
      remplacer la documentation offerte par la <ref
        target="http://www.tei-c.org/index.xml">Text Encoding Initiative</ref>.
    </p>
    [...]
  </div2>
</div1>
```

Des éléments de mise en forme sont prévus. Ceux ci seront par exemple, placés dans des balises <gi>, les attributs dans une balise <att>, et les valeurs de ces derniers dans une balise <val> les éléments <hi> et <emph> permettent de mettre du texte en gras ou en italique.

Des liens peuvent être opérés soit vers des ancres placées dans le fichier ODD soit vers des liens extérieurs. Il convient dans ce cas, d'utiliser la balise <ref>. Il est également possible de citer des parties du code XML avec la balise <egXML> et enfin d'appeler automatiquement des définitions de la Documentation TEI avec les balises <specDesc> contenu dans un élément <specList>.

```
<p>
  Cet élément, relatif à l'encodage, précise les principes éditoriaux qui ont permis la transcription
  du texte. Il peut contenir les sous-éléments suivants :
  <specList>
    <!-- Chaque balise <specDesc> appelle automatique la définition des éléments ciblés par
      l'attribut @key -->
    <specDesc key="projectDesc"/>
    <specDesc key="editorialDecl"/>
    <specDesc key="correction"/>
    <specDesc key="hyphenation"/>
```

```

    <specDesc key="normalization"/>
    <specDesc key="punctuation"/>
  </specList>

[...]
<!-- cette balise permet de citer des exemples de code XML -->
<egXML xmlns="http://www.tei-c.org/ns/Examples">
  <opener>
    <dateline>
      <placeName>Imbros</placeName>, <date when="1915-09-23">Le 23 septembre
        1915</date>
    </dateline>
    <salute><persName>Comtesse</persName>, </salute>
  </opener>
</egXML>

</p>

```

Enfin n'oublions pas, comme nous l'avons vu plus haut, que les définitions des différents éléments et attributs du schéma peuvent être également personnalisées. Dans la mesure où les logiciels comme Oxygen proposent de l'auto-complétion, cela procure une aide précieuse et accessible au moment de l'encodage.

Troisième partie

La publication web

Il était important de développer une interface de consultation web, afin que la correspondance puisse être publiée et serve à illustrer les possibilités offertes par la TEI. La durée du stage était relativement courte pour mener à bien la réalisation de l'ensemble des opérations de modélisation, de transcription, d'encodage et de publication web. Ainsi, nous avons décidé de nous limiter à la première moitié de la correspondance d'Armand Horel, dans le but de s'attacher au développement d'une interface de consultation web. Compte tenu de la période durant laquelle le stage a eu lieu, il n'a pas été possible de mettre en ligne le site réalisé. Toutefois, il devrait être publié après relecture par la bibliothèque de documentation internationale contemporaine.

Par la forme et les fonctionnalités qu'elle devait offrir, l'interface de consultation web a soulevé un certain nombre d'interrogations. En effet, dès lors que nous éliminons les solutions basées sur le schéma TEI-lite, les outils autorisant la publication web de documents TEI ne sont pas nombreux. Une autre contrainte résidait en la possession de deux sites internet par la BDIC : l'un institutionnel et le suivant dédié à sa bibliothèque numérique. Il était donc exclus d'utiliser un CMS (*Content Management System*) pour cette édition, dans la mesure où il s'agit de structures relativement lourdes, réclamant d'être maintenues régulièrement, en particulier pour un corpus très limité — cette solution serait peut être à reconsidérer si ce projet d'édition prend de l'ampleur. En outre, compte tenu des délais qui nous étaient impartis, la réalisation d'une base de données fournissant des instruments de recherche avancés, utilisant les langages MySQL et php, ne pouvait pas être mise en place.

Enfin, il est possible d'inclure des fragments de pages HTML dans d'autres pages web ; cette solution a retenu notre attention. Compte tenu que le site de la BDIC est actuellement en pleine refonte, cette procédure n'a pas pu être mise en place. De ce fait, notre choix s'est porté sur la création d'une interface de consultation web "statique", c'est à dire uniquement composée de pages HTML. Afin de ne pas créer de rupture, l'interface devait reprendre les codes esthétiques du prochain site web de la Bibliothèque, sans pour autant le copier complètement. Ce choix n'offrant que des possibilités de recherche limitées — il n'est pas possible d'opérer des recherches dans le texte, ou d'interroger une base de données par exemple —, il nous fallait trouver un moyen de multiplier les points d'accès dans le texte, tout en garantissant une ergonomie conviviale et performante.

Chapitre 6

Transformation XML vers HTML

La première tâche qui nous incombait était de transformer nos fichiers XML en page HTML. Pour se faire, nous pouvions ici aussi nous appuyer sur le cadre de travail proposé par XML. En effet, le langage de programmation XSLT (*eXtensible Stylesheet Language Transformations*), permet de transformer un document XML vers d'autres formats, comme un fichier PDF, Docx, et surtout HTML, langage de base des pages web. Ce langage repose en partie sur des expressions Xpath (une syntaxe non XML) qui permettent de localiser des nœuds dans un arbre XML. En d'autres termes, cette syntaxe permet de cibler des balises au sein d'un document XML. L'objectif de ce chapitre ne consistera pas à détailler l'ensemble des feuilles de transformation que nous avons réalisé, mais plutôt de faire un panorama de leur utilité dans la tenue du projet.

6.1 Principes de base

6.1.1 Application de principes éditoriaux

En ce sens, nous avons donc utilisé XSLT pour transformer nos fichiers XML en page HTML. Comme évoqué en amont, nous avons tenté de respecter les sources le plus possible lors de leur encodage, tout en gardant comme objectif de proposer une lecture régularisée des différentes missives. Par un jeu de balises (<choice>), la TEI permet notamment d'encoder côte à côte la version fautive d'un terme présent dans le texte original, et une version corrigée, proposée par l'éditeur ou l'encodeur. Il en est de même pour la ponctuation ou la régularisation des majuscules et minuscules qui peuvent également être encodées, tout en gardant la trace des emplois fautifs.

Nos choix éditoriaux ne sont donc que le choix de l'application de notre feuille de transformation XSLT. Il serait d'ailleurs possible de créer une feuille de style respectant la graphie originale et d'exclure les corrections. Ajoutons que notre vo-

lonté est de proposer les fichiers sources librement, afin que chacun puisse éventuellement créer son propre format de sortie.

```
<!-- Cette fonction sélectionne tous les éléments <choice> contenus (à n'importe quel niveau,
      dans l'élément <text>-->
<xsl:template match="tei:text//tei:choice">
<!-- On demande ici à ne choisir en sortie que les éléments <reg>, <corr> ou <expan> qui
      correspondent aux éléments normalisés-->
  <xsl:apply-templates select="tei:reg | tei:corr | tei:expan"/>
</xsl:template>

<!-- Cette fois-ci, la fonction sélectionne toujours tous les éléments <choice> contenus dans
      l'élément <text>-->
<xsl:template match="tei:text//tei:choice">
<!-- En revanche, on demande ici de n'appliquer que les versions originales et d'exclure les
      formes corrigées-->
  <xsl:apply-templates select="tei:orig | tei:sic | tei:abbr"/>
</xsl:template>
```

De même il est possible d'exclure d'emblée de notre texte de sortie certains éléments ; par exemple, il arrive que l'auteur double certains termes. Donc, pour faciliter la lecture, il convient de retirer ces mots du texte. XSLT permet également de le faire.

```
<xsl:template match="//tei:surplus | //tei:pc[@type='surplus']"/>
<!-- cette fonction refuse par exemple d'appliquer un traitement à
      tous les éléments <surplus> et <pc> disposant d'un attribut @type de valeur surplus-->
```

6.1.2 Mise en page

Dans un second temps, la feuille de style nous a permis de mettre en page les correspondances. En effet, XSLT permet d'inclure dans une feuille de style des balises HTML. Le langage HTML est un dialecte de balisage présentationnel, utilisé pour mettre en forme un texte pour sa publication web. Une partie du balisage structurel de la TEI est le même que celui utilisé en HTML. Nous pouvons prendre l'exemple de balise paragraphe (<p>) ou division (<div>) qui se font écho. Il est donc très facile avec XSLT de structurer le texte en vue de sa publication web.

```
<!-- Cette fonction-->
<xsl:template match="tei:div[@type='letter']//tei:p">
  <p>
    <xsl:apply-templates/>
```

```
</p>
</xsl:template>
```

XSLT permet d'aller encore plus loin dans la mise en forme des textes, en étant capable par des fonctions, d'opérer un traitement directement sur une portion de texte déterminée. Par exemple, afin de normaliser les noms de personnes, il convient que la première lettre soit en majuscule et le reste en minuscules. XSLT permet d'automatiser ce traitement sans forcément devoir le corriger dans le texte.

```
<!-- Cette fonction cible tous les éléments <forename> contenus dans une division identifiée
      comme une lettre. -->
<xsl:template match="tei:div[@type='letter']/tei:forename">
  <!-- Une variable est créée avec la valeur de l'élément <forename> -->
  <xsl:variable name="upfirst">
    <xsl:value-of select="."/>
  </xsl:variable>
  <!-- La fonction substring permet de cibler les caractères dans un segment de texte (ici la
        première lettre de notre variable d'une part et toutes les lettres après la première d'autre
        part). Les fonctions upper-case et lower-case permettent de passer ces segments en
        majuscule et minuscule. Enfin, la fonction concat agrège le tout. -->
  <xsl:value-of select="concat(upper-case(substring($upfirst, 1, 1)),
    lower-case(substring($upfirst, 2)))"/>
</xsl:template>
```

6.2 Fonctions avancées

Si le schéma que nous avons réalisé permet de contraindre l'encodage, en revanche, il est toujours possible que des cas particuliers se présentent ou que certaines informations soient traitées en série. Une feuille de transformation XSLT peut prévoir ces différentes éventualités afin de réellement automatiser le traitement des fichiers de sortie.

6.2.1 Les boucles

Certaines données nécessitent un traitement en série, sans pour autant que l'on en soit en mesure de les dénombrer. Afin de traiter ce cas de figure il faut utiliser ce que l'on nomme une boucle. Cela permet d'appliquer une même fonction à un ensemble de données identifiées. Pour la création de l'index par exemple, qui se présente sous la forme d'une liste, il fallait sélectionner chaque entrée et les

placer chacune entre des éléments de liste HTML. Cette exercice est assurée par l'instruction `<xsl:for-each>`

```
<xsl:template match="//tei:div[@xml:id='index']">
  <ul>
    <!-- Pour chaque élément <item>, <person> ou <place> cette instruction crée un balise <li> et
         y applique le contenu des trois éléments cités -->
    <xsl:for-each select="tei:list/tei:item | tei:listPerson/tei:person | tei:listPlace/tei:place">
      <!-- Cette instruction <xsl:sort> permet de classer les entrées par ordre alphabétique -->
      <xsl:sort select="tei:label | tei:persName/tei:surname |
        tei:persName[@type='nobility']/tei:forename | tei:placeName" order="ascending"
        case-order="upper-first"/>
      <xsl:sort select="tei:persName/tei:forename" order="ascending" case-order="upper-first"/>
      <li class="entree">
        <xsl:attribute name="id" select="@xml:id"/>
        <xsl:apply-templates select="tei:label[@type='navire'] | tei:persName | tei:placeName"/>
      </li>
    </for-each>
  </ul>
</xsl:template>
```

6.2.2 Les conditions

Afin d'organiser nos données comme nous le souhaitions, il était nécessaire d'opérer des distinctions en fonction des cas de figure. Le balisage doit nous aider en cela. Dans la mesure où il décrit le document, il est possible d'identifier les circonstances relevant d'une même logique. Une fois ces particularités encodées, il devient normalement possible de les cibler avec une expression Xpath et donc de leur appliquer un traitement personnalisé en XSLT. Afin d'appliquer ces traitements nous avons utilisé des éléments de traitement conditionnel ; `<xsl:if>` et `<xsl:choose>`. Ces éléments permettent notamment de palier à des incertitudes dans le traitement des données et limitent ainsi les risques d'erreur. Par exemple nous souhaitions éditer les lettres d'Armand Horel aux cotés de leurs facsimilés disponibles sur la bibliothèque numérique de la BDIC. Les liens vers les facsimilés ont été encodés en TEI avec l'attribut `@facs` des éléments `<pb/>`. Toutefois nous tenions à ce que la face imagée de la carte soit accompagnée de sa description. La seule solution pour effectuer cette instruction était de différencier les éléments `<pb/>` placés juste avant une balise `<figure>` contenant la description de l'image et de leur appliquer ainsi un traitement particulier.

```
<xsl:template match="tei:div[@type='letter']/tei:pb | tei:div[@type='enveloppe']/tei:pb"
  mode="affichage">
  <!-- On cible ici les élément <pb/> et on place dans deux variables le lien photographique et le
       lien vers le viewer de la bibliothèque numérique -->
```

```

<xsl:variable name="source">
  <xsl:value-of select=" substring-before(@facs,' ')" />
</xsl:variable>
<xsl:variable name="lien">
  <xsl:value-of select=" substring-after(@facs,' ')" />
</xsl:variable>
<!--comme nous l'avons vu précédemment nous appliquons les mêmes instructions à tous les é
lément ciblés-->
<xsl:for-each select=".">
  <xsl:variable name="image" select="@facs"/>
  <li>
    <a href="{ $lien }">
      
    </a>
    <xsl:choose>
      <!--On place ici une condition : si l'élément <pb/> est suivi directement par un élément
      <figure> alors on récupère la description-->
      <xsl:when test="following-sibling::[1][self::tei:figure]"><div
        class="orbit-caption"><xsl:for-each select="following-sibling : [1][self::tei:figure]"
        >
          <xsl:value-of select=".">
          </xsl:value-of>
        </xsl:for-each>
      </div>
      </xsl:when>
      <!--Dans le cas contraire, il ne se passe rien-->
      <xsl:otherwise/>
    </xsl:choose>
  </li>
</xsl:for-each>
</xsl:template>

```

6.2.3 Les modes

Les modes nous ont été d'une aide précieuse dans l'élaboration de nos fichiers de sortie afin de formater différemment certaines informations et les faire apparaître où nous le souhaitions, dans les documents de sortie. Par exemple nous avons vu qu'Armand Horel, lorsqu'il n'avait plus de place pour signer sa correspondance sur la dernière carte de sa lettre, ajoutait celle-ci dans la marge de l'une des précédentes cartes. Afin d'offrir une plus grand confort de lecture, il était important de replacer des informations en bonne place pour la sortie HTML. Le mode nous y a aidé. Il s'agit d'un attribut de l'élément `<xsl:template>` qui permet de créer des « règles modèles alternatives ».

```
<xsl:template match="tei:text//tei:seg[@type='closer']"/>

<!--L'attribut mode permet de créer plusieurs règles pour cibler un même éléments et les
appeler (ou non) àun endroit précis dans le texte de sortie-->

<xsl:template match="tei:text//tei:seg[@type='closer']" mode="ordreLecture">
  <p>
    <xsl:apply-templates select="tei:add[not(@type='signed')']"/>
  </p>
  <xsl:if test="tei:add[@type='signed']">
    <p class="text-right">
      <xsl:apply-templates select="tei:add[@type='signed']/tei:persName"/>
    </p>
  </xsl:if>
</xsl:template>
```

Chapitre 7

L'utilisation d'un framework CSS

Afin de faciliter le développement d'une interface web et dans la mesure où l'utilisation d'un CMS était à proscrire, nous nous sommes tournés vers l'utilisation d'un *framework* CSS pour assurer le développement d'une interface web.

Les *frameworks* sont des cadres de développement web, généralement composés d'une ou plusieurs bibliothèques CSS et JavaScript. Ces outils sont particulièrement intéressants et utiles car ils assurent la compatibilité entre les différents navigateurs, facilitent le positionnement CSS et enfin permettent la mise en place d'un « responsive design », design s'adaptant en fonction des périphériques de lecture.

Il existe une multitude de *frameworks* disponibles librement sur le web, le plus connu d'entre eux étant certainement Twitter-Bootstrap. Pour ce projet c'est toutefois le *framework* Foundation qui a été retenu¹. Plusieurs raisons ont guidé ce choix ; tout d'abord il s'agit d'un cadre de développement *open source*, déployé depuis 2011, il bénéficie d'une communauté d'utilisateurs relativement importante. Sa prise en main est aisée et la documentation est extrêmement bien réalisée, tout y est expliqué et illustré clairement. Enfin il inclut de nombreuses extensions JavaScript particulièrement intéressantes. De plus, il est possible de personnaliser les fonctionnalités proposées par Foundation avant le téléchargement par le biais d'une interface graphique en ligne.

Foundation nous a permis de mettre en place un design adaptatif afin que le site puisse être lisible à partir des smartphones ou tablettes tactiles. Il a également été utilisé pour mettre en place les *sliders*, c'est à dire les diaporamas pour l'affichage des fac-similés. En outre, Il a facilité la mise en place des notes s'affichant lors d'un clic sur les entrées d'index en bleu.

1. [Foundation Zurb](#)

Chapitre 8

Intégration d'outils interactifs

8.1 Cartographie

L'une des particularités des lettres de la correspondance d'Armand Horel réside dans la grande disparité des lieux d'envoi. Il semblait donc essentiel de pouvoir suivre sur une carte ses différents déplacements et de localiser précisément ses stationnements militaires (*voir Annexe B. Figure B.3.*). L'aspect cartographique ne doit pas être négligé, surtout dans le cadre d'une édition à visée historique car l'observation d'une carte voire d'un planisphère peut nous faciliter la compréhension des événements ou des stratégies déployées par les belligérants. Pour cette raison, nous souhaitons intégrer une carte interactive, afin de visualiser le parcours du marin dans le développement du conflit et ainsi pouvoir accéder aux lettres en fonction de différents lieux (*voir Annexe B Figure B.4.*).

L'intégration d'une cartographie sur un site web repose sur des bibliothèques JavaScript dont les trois principales sont Google Maps, Open Layers et LeafLet. Notre choix s'est orienté sur cette dernière s'agissant d'un projet *open source*, relativement bien documenté et reposant sur des cartes de qualité réalisées par OpenStreetMap. De plus tout est réalisé en ligne sans recours à aucun téléchargement.

8.1.1 Intégration d'une carte

L'intégration d'une cartographie sur une page web avec leafLet est relativement simple. Cet outil interactif offre un réel avantage à l'édition électronique .

Au sein de la page HTML où nous souhaitons insérer la carte, il faut inclure dans le <header> les liens vers la feuille de style CSS relative à leafLet ainsi que la bibliothèque JavaScript. Nous prendrons soin également d'intégrer un lien vers un fichier JavaScript qui nous servira à la gestion des informations relatives à la carte dans un fichier externe à la page HTML que nous avons nommée cartographie.js.

```
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
```

```
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
<script src="cartographie.js"></script>
```

Il convient ensuite de placer l'élément `<div>` qui permettra d'accueillir la carte au sein de la page web. Il faut le placer dans le `<body>` de notre page internet.

```
<body onload="InitialiserCarte();">
  <div id="map"></div>
</body>
<!--notons que le body doit être précisé avec l'attribut onload afin de d'initialiser la carte lors du
chargement-->
```

Il convient ensuite de compléter le fichier "cartographie.js" avec le code suivant :

```
function InitialiserCarte() {
// cette variable place la carte dans la balise
<div> comportant l'@id 'map'
//la fonction setView centre la carte sur les
coordonnées géographiques suivante
    var map = L.map('map').setView
    ([41.508577297439324, 15.64453125], 5);

    //fait le lien vers la tuile pour le fond de carte.
    var tuile = 'http://{s}.tile.osm.org/{z}/{x}/{y}.png';

    var attrib='Map data © <a href="http://openstreetmap.org">
OpenStreetMap</a> contributors';

    var osm = L.tileLayer(tuile, {
        minZoom: 5,
        maxZoom: 17,
        attribution: attrib
    });

    osm.addTo(map);

$(document).ready(function() {
    InitialiserCarte();
});
```

8.1.2 Récupération des informations par une feuille de style XSLT

Afin de permettre l'accès aux lettres par une carte, il était nécessaire d'y incorporer des *marker* ouvrant des fenêtres *popup*, le morceau de code suivant doit être intégré dans le fichier *geographie.js*.

```
// il est possible de mettre autant de marker que l'on souhaite, bien prendre soin en revanche de
// leur attribuer un nom de variable propre.

//Toulon
var marker = L.marker([43.1166667, 5.933333333333334]).addTo(map);

//corfou
var marker3 = L.marker([39.5980556, 19.852222222222224]).addTo(map);
```

Une fois que ces *marker* sont placés il est possible de de leur attacher des *popup* contenant les liens vers les différentes lettres.

```
marker3.bindPopup('<a href="Corfou_04-11-1916.html">
04 novembre 1916</a>',
{maxWidth: 300, minWidth: 250,
maxHeight: 260, autoPan: true,
closeButton: true, autoPanPadding: [5, 5]});
```

Ces liens s'opèrent à partir de code HTML. Une feuille de style a été réalisée pour générer une liste de liens vers toutes les lettres et organisées par lieu. Le code généré doit ensuite être entré manuellement, dans ce code JavaScript¹. Nous n'avons pas réussi à créer ce code directement à partir de XSLT dans la mesure où chaque caractère compte, ainsi que les espaces, pour que le code soit opérant.

Notons enfin que nous avons contraint la taille des fenêtre *popup* afin qu'un ascenseur (*scrollbar*) soit présent, permettant de gérer plus facilement les grands ensembles de cartes.

8.2 Intégration d'une frise chronologique

Pour donner un accès chrono-géographique aux lettres, nous avons intégré une frise chronologique permettant de visualiser les différents temps de stationnement du soldat (*voir Annexe B Figure B.5.*). C'est la frise développée par le Simile

1. /XSL/liens_geographie.xsl

Project que nous avons utilisée. Afin d'être alimentée, elle nécessite que les données à afficher soient encodées en XML. En ce sens, une feuille XSLT a été réalisée pour opérer la transformation des données TEI en un autre fichier XML compatible avec la frise². La difficulté a été d'y intégrer les liens dans la mesure où les balises devaient être encodées avec les entités représentant les chevrons.

2. Voir le fichier : /XSL/Date.XSL.

Conclusion

Au terme de ce mémoire pour le diplôme de master « technologies numériques appliquées à l'histoire », il convient d'effectuer un bilan de cette expérience professionnelle. Ce stage s'est déroulé entre les mois d'avril et de juillet au sein de la bibliothèque de documentation internationale contemporaine. La vocation de ce stage ne résidait pas tant dans l'édition de la correspondance d'Armand Horel, Quartier-maitre canonnier pendant la Grande Guerre, que d'offrir à la BDIC un modèle d'édition électronique. De surcroît, nous avons pu grâce à ce stage mettre en pratique des connaissances acquises tout au long de notre année d'étude.

Sensible à la valorisation du patrimoine par les technologies numériques et à l'histoire de la Grande Guerre, nous avons particulièrement apprécié travailler à partir de sources de première main. Ce fonds conséquents de cartes postales se prêtait volontiers à notre entreprise et s'est également révélé riche d'informations. Tout d'abord, d'un point de vue iconographique et documentaire ; le recto de ces cartes est précieux pour quiconque recherche des vues citadines ou rurales de l'époque et des scènes de genre, parfois teintées de folklore. Puis, la correspondance entre le marin et la Comtesse sur le verso offre des renseignements concernant la vie quotidienne de ce sous-officier de la Marine Nationale en temps de guerre, loin de son pays et de sa famille, le ressenti de celui-ci face aux ennemis et au danger, ses inquiétudes et ses craintes, renforçant par là même l'empathie du chercheur à son encontre. Outre ces informations, la correspondance d'Armand Horel permet également au public d'être à l'avant poste des événements qui se déroulaient sous les yeux du marin et qui allaient marquer l'histoire il y a tout juste cent ans.

L'intérêt principal de ce stage résidait dans le traitement de toutes les tâches qui nous incombaient, de la modélisation jusqu'à l'encodage, la publication de ce fonds de correspondances et de notre travail. Mais, parce que ce projet était ambitieux, la difficulté principale résidait dans la création d'un modèle général à partir d'une correspondance. Ainsi, ce mémoire est là pour présenter les différentes étapes de cette entreprise et accompagner le lecteur dans la compréhension de notre démarche. De plus, la documentation s'y référant aidera le chercheur souhaitant intégrer le numérique à son projet de recherche ou de réaliser une édition numérique.

Lors de ce stage, nous avons dû faire face aux difficultés liées à l'édition. Ces correspondances étant manuscrites mais aussi historiques, il était nécessaire de faciliter la lecture en corrigeant l'orthographe, sans toutefois modifier le vocabulaire utilisé par l'auteur. Parfois, Armand Horel effectuait des erreurs d'orthographe et de conjugaison, ou employait mal la ponctuation. Respecter scrupuleusement la graphie des mots n'aurait pas servi le chercheur, car la lecture aurait été déli-

cate. En procédant à une édition électronique de correspondances le premier enjeu auquel nous avons été confrontés, résidait dans l'encodage d'une multitude de petits textes tous différents, mais néanmoins liés entre eux. Nous avons donc cherché à personnaliser le balisage XML-TEI. Pour ce faire, un fichier `teiCorpus` a été créé contenant chaque lettre au sein d'un élément `<TEI>`. Pour chaque lettre, un autre document XML a été créé et complété par le biais de différents appels `Xinclude`. Nous devons également garder en mémoire que l'édition de la correspondance d'Armand Horel devait surtout permettre l'élaboration d'un modèle d'édition électronique réutilisable par la BDIC mais également par ses usagers qui en formuleraient la demande. Il s'agissait donc de formuler des principes d'encodage qui soient transposables à d'autres corpus de lettres tout en tenant compte des particularités de ce fonds.

Par ailleurs, malgré ce que l'on pourrait en penser, trois mois de stage se sont avérés trop court pour parvenir à gérer l'ensemble des phases de développement de ce projet d'édition, sans compter le fait que tout le fonds de la correspondance d'Armand Horel n'a pu être édité. Il eut été envisageable avec un délai supplémentaire d'enrichir la cartographie, en délimitant les frontières terrestres et maritimes des pays à l'époque où Armand Horel rédige ses cartes postales par exemple. Ce développement serait un atout pour cette édition, offrant au chercheur un véritable outil documentaire. Ensuite, compte tenu du fait que le site internet de la bibliothèque de documentation internationale contemporaine est en pleine refonte, nous n'avons pas pu finaliser cette entreprise en la mettant en ligne.

Pour conclure, l'édition scientifique est un travail complexe qui nécessite de bien cerner ses objectifs sans forcément chercher à répondre à tous les besoins de tous les chercheurs. C'est un exercice qui demande un investissement personnel important durant lequel on développe un lien particulier avec le texte, c'est également cela qui en fait une expérience particulièrement enrichissante et formatrice.

Annexes

Annexe A

Manuel d'encodage

Schéma TEI pour l'édition de correspondances

Josselin Morvan

2015-11-06

XML : un langage de balisage

XML est un langage qui s'appuie sur l'utilisation de balises.

Les balises ont un rôle syntaxique, permettant de délimiter et de localiser une séquence dans une chaîne de caractères.

- Exemple : Ici j'ai du texte en **gras** et ici en *italique*

- Principe : Ici j'ai du texte en **/début_Gras gras /fin_Gras** et ici en */début_Italique italique /fin_Italique*

XML permet de distinguer la définition sémantique des informations et leur mise en forme. Par exemple dans un index, il est d'usage d'utiliser l'italique pour distinguer les noms de lieux des noms de personnes. Il existe donc deux solutions pour un encodage XML :

- Prise en compte de l'apparence :

```
<italique>Beaune</italique>,  
ch.-l. dép. Côte-d'or.  
<!--Beaune est en italique-->
```

- Prise en compte du sémantisme de l'information :

```
<lieu>Beaune</lieu>, ch.-l.  
dép. Côte-d'or.  
<!--Beaune est en italique parce que c'est un lieu-->
```

Un langage à balises structuré

Un document XML respecte une structure hiérarchique formant une arborescence. Ce qui définie peut être le mieux cette structure ce sont les *Matriochkas*.

Cette structure implique que les éléments ne peuvent pas s'entrecroiser :

```
Bien formé :  
<donateur>  
  <personne>Louise</personne> et <personne>Henri  
  Leblanc</personne>  
</donateur>
```

Mal formé :

```
<lb/>
```

```
<donateur><personne>Louise et <personne></personne>Henri Leblanc</donateur></personne>
```

Avantages et inconvénients d'un langage sémantique et structuré

Avantages :

- La structure sémantique et hiérarchique est compréhensible par les machines.
- Le contenu est distinct de sa représentation. Cette dernière peut être gérée à part et est configurable à volonté.
- L'utilisation d'un texte comme base de données.

Inconvénients :

Structure parfois lourde

Un langage de balisage eXtensible ?

XML est un métalangage : il permet de créer des langages avec leur propre grammaire (ex : EAD - TEI - XHTML).

XML ne propose donc pas un jeu d'éléments prédéfinis, mais un ensemble de règles sur ce que doit être un document bien formé et valide.

Composition d'un document XML

- Les éléments : structures de base d'un document XML. Ils sont délimités par des balises de début et des balises de fin. Tout ce qui se trouve entre ces deux balises représente le contenu, ex :

```
<personne>Henri
```

```
IV</personne>
```

```
<!--Notons que les balises sont délimitées par des chevrons "< et >"-->
```

- Les attributs : ce sont des paires "nom-valeur" associées à la balise de début d'un élément. Ils apportent généralement une information complémentaire pour l'encodeur ou sur le contenu de l'élément, ex : <element attribut="valeur">mon contenu</element>
- Des entités : substituts pour des séquences d'information.

```
<bonbon>tic &amp;
```

```
tac</bonbon>
```

```
<!-- L'esperluette (&) est un caractère  
réservé-->
```

- Des instructions de traitements : applications extérieures.
- Des commentaires :

```
<!--Les commentaires sont des notes dans le code-->
```

- Des sections CDATA : contenu textuel non parsé
- PCDATA : contenu textuel balisé.

Organisation des fichiers

Encodage de correspondances et XML-TEI

Lorsque l'on procède à une édition électronique de correspondances, le premier enjeu auquel nous sommes confrontés est celui d'avoir à encoder une multitude de petits textes indépendants

mais pour autant liés les uns aux autres. À la différence d'une œuvre littéraire, ces textes ne peuvent pas être encodés à la suite comme différents chapitres. En effet, à des fins d'ergonomie, la publication web nécessite que chaque lettre possède sa propre page de consultation. Il est nécessaire de réfléchir à un corpus électronique organisé. Le schéma mis en place par la Text Encoding Initiative n'a pas été élaborée pour cet exercice mais des solutions peuvent tout de même être envisagées grâce notamment aux outils offerts par le cadre de travail XML. Une première solution est proposée par la *Text Encoding Initiative* avec la création d'un fichier *teiCorpus* qui permet de rassembler, au sein d'un seul fichier, un ensemble de documents TEI bien distincts et pouvant partager un ensemble de métadonnées communes.

```
<teiCorpus xmlns="http://www.tei-c.org/ns/1.0" xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <!--[en-tête du corpus pouvant contenir les différentes métadonnées communes]-->
  </teiHeader>
  <TEI>
    <teiHeader>
      <!--[en-tête de la première lettre]-->
    </teiHeader>
    <text>
      <!--[première lettre du corpus]-->
    </text>
  </TEI>
  <TEI>
    <teiHeader>
      <!--[en-tête de la deuxième lettre]-->
    </teiHeader>
    <text>
      <!--[deuxième lettre du corpus]-->
    </text>
  </TEI>
</teiCorpus>
```

Cette proposition de la TEI permet de faire une gestion plus aisée des fichiers dans la mesure où il n'y a plus qu'un seul document à gérer. Toutefois, cela pose encore un problème pour la création des différentes pages d'un site internet dédié à l'édition. Nous ne pouvons pas créer par le biais d'une seule feuille de transformation XSLT, les différentes pages web correspondantes aux différentes lettres. Une solution a donc été trouvée avec l'environnement de travail XML qui dispose d'outils puissants. Le standard Xinclude autorise notamment la combinaison de différents documents XML entre eux. En effet, il permet de cibler certaines parties identifiées d'un document afin de l'intégrer au sein d'un autre fichier. Nous avons donc procédé de la manière suivante ; un fichier *teiCorpus* a été créé contenant chaque lettre au sein d'un élément `<TEI>`. Pour chaque lettre, un autre document XML a été créé et complété par le biais de différents appels `Xinclude`. Ces derniers ont permis par la suite la création des différentes pages web en leur appliquant les feuilles de transformation XSLT. Si cela semble complexe au premier abord, cela s'avère en réalité extrêmement commode et pertinent dans la gestion du projet, dans la mesure où une fois que les éléments nécessaires à chaque lettre ont été identifiés, il n'y a plus qu'un seul document à modifier, le fichier maître contenant le *teiCorpus*. Cela s'avère intéressant également pour la création d'un schéma TEI spécifique, car ce document rassemble l'ensemble des pratiques d'encodage.

```
<!--Voici un exemple d'utilisation d'Xinclude-->
<!--L'espace de nom Xinclude doit être déclaré avec la mention suivante--><text>
<!--l'attribut @href cible le fichier contenant la partie de document à intégrer--><xi:include href="../master/Master.xml" xpointer="body_imbro_23-09-1915"
parse="xml"/>
<!--l'attribut @xpointer cible l'@xml:id du morceau de code à inclure-->
```

```
<!--l'attribut @parse indique le type de données-->
</text>
```

Concernant les liens, cela permet généralement de les simplifier car ils deviennent des liens internes au document. Toutefois, il faut faire attention ; dans une édition de correspondances comme nous l'avons vu, les différentes lettres doivent être publiées sur leur propre page. Cela nécessite dans la gestion des liens qui deviendront des liens hypertextes une attention toute particulière, car ils doivent être gérés en tenant compte du fichier XML servant à la sortie et non du `teiCorpus`.

NOTE RECTIFICATIVE : XSLT 2.0 propose la fonction *result-document* permettant de générer plusieurs fichiers à partir d'une feuille de transformation.

Manuel d'encodage en XML-TEI

Préambule

Ce manuel constitue une aide pour le chercheur lors de son travail d'encodage. Il présente des explications sur certains points techniques mais il n'a pas vocation à remplacer la documentation offerte par la Text Encoding Initiative.

Cette aide a été rédigée en s'appuyant en partie sur les manuels d'encodage suivants : schéma pour l'édition de correspondance de l'École nationale des chartes, l'édition des cours de Desgodets et le manuel d'encodage TEI - Renaissance et temps modernes.

Au sein de ce manuel, les éléments sont représentés entre des chevrons ("`<`" et "`>`"). Des liens renvoient vers la documentation TEI fournie en annexe.

Enfin, par conventions les attributs seront représentés en italique et de la manière suivante : *@attribut* ; leur valeur sera précisée entre guillemets : "valeur".

L'en-tête TEI : `<teiHeader>`

Tout texte encodé en TEI comporte un élément `<teiHeader>`. Il s'agit de l'en-tête du document contenant les informations, métadonnées, permettant son identification.

Il se subdivise ainsi :

```
<teiHeader>
  <fileDesc>(obligatoire) description bibliographique</fileDesc>
  <encodingDesc> description des normes
    d'encodage</encodingDesc>
  <profileDesc> description des aspects non
    bibliographiques</profileDesc>
</teiHeader>
```

Notons tout de même que tous les éléments du `<teiHeader>` n'ont pas vocation à être remplis. Nous nous bornerons donc à la description des éléments principaux présentant un intérêt général pour l'édition de correspondances.

Élément de description bibliographique : `<fileDesc>`

Déclaration du titre : `<titleStmt>`

Cet élément est obligatoire. On y trouve les éléments suivants :

`<title>` : (répétable) Il désigne le/les titre(s) donné(s) à l'édition électronique.

`<respStmt>` : (statement of responsibility), cet élément se réfère aux mentions de responsabilités concernant l'édition électronique, les sous-éléments `<resp>` et `<name>` peuvent par exemple renvoyer vers la description du rôle d'un des responsables de l'édition et à son nom.

`<author>` : nom de l'auteur

`<editor>` : mentions de responsabilités secondaires

Exemple de `<titleStmt>` :

```

<titleStmt xml:id="titleStmt">
  <title n="1">Don de Madame <persName ref="#Comtesse_Forest">la Comtesse de la
  Forest</persName>
</title>
  <title n="2">Souvenir de soldats</title>
  <title n="3">Cartes postales illustrées</title>
<respStmt>
  <resp>Transcription et encodage XML/TEI : </resp>
  <name xml:id="J_M">Josselin Morvan (étudiant à
    l'École des chartes)</name>
</respStmt>
<respStmt>
  <resp>Sous la direction de </resp>
  <name xml:id="F_J_S">Frédérique Joannic-Seta
    (directrice adjointe de la
    <orgName>Bibliothèque de Documentation
    Internationale Contemporaine</orgName>)</name>
</respStmt>
</titleStmt>

```

Mentions d'édition : <editionStmt>

La balise *edition statement* : (optionnelle) renvoie aux mentions d'édition. Le sous-élément <edition> permet de décrire les spécificités de l'édition d'un texte.

Mentions de publication : <publicationStmt>

L'élément <publicationStmt> (obligatoire) regroupe les informations relatives à la publication et à la diffusion d'un texte :

<authority> (responsable de la publication.) donne le nom de la personne ou de l'organisme responsable de la publication d'un fichier électronique, autre qu'un éditeur ou un distributeur.

<address> contient une adresse postale ou d'un autre type, par exemple l'adresse d'un éditeur, d'un organisme ou d'une personne.

<publisher> (éditeur) donne le nom de l'organisme responsable de la publication ou de la distribution d'un élément de la bibliographie.

<availability> (disponibilité) renseigne sur la disponibilité du texte, par exemple sur toutes restrictions quant à son usage ou sa diffusion, son copyright, etc.

<noteStmt>

L'élément <noteStmt> (optionnel) permet d'ajouter des notes sur l'édition.

Description de la source : <sourceDesc>

L'élément <sourceDesc> (obligatoire) consigne toutes les informations sur la source originale. Elles peuvent être détaillées à l'intérieur d'une simple balise <p>, ou plus détaillées avec l'utilisation d'un élément <msDesc>.

Manuscript description : <msDesc> À l'origine, cette balise est dédiée à la description des manuscrits anciens. Il est admis que son utilisation peut être étendue à d'autres types de texte. Cet élément peut contenir les balises de description suivantes :

Informations d'identification de la source : <msIdentifier>

Cette balise rassemble les éléments d'identification de la source comme le lieu de conservation, son identifiant *etc.*

exemple de <msIdentifier> :

```

<msIdentifier xml:id="msIdentifier">
  <country>France</country>

```

```
<settlement>Nanterre</settlement>
<repository>Bibliothèque
  de Documentation Internationale
  Contemporaine</repository>
<idno>F delta
  1854/20</idno>
</msIdentifier>
```

<msContent>

Cet élément est relatif au contenu intellectuel du texte édité. Il peut aussi être décrit à l'aide d'un élément <p>.

Description physique du document : <physDesc>

Cette balise peut contenir :

<objectDesc> (description d'objet) contient la description des composants matériels de l'objet en cours de traitement

Histoire : <history>

Cet élément permet de retracer l'historique du document.

Description de l'encodage : <encodingDesc>

Cet élément relatif à l'encodage, précise les principes éditoriaux qui ont permis la transcription du texte. Il peut contenir les sous-éléments suivants :

<projectDesc> (description du projet) décrit en détail le but ou l'objectif visé dans l'encodage d'un fichier électronique, ainsi que toute autre information pertinente sur la manière dont il a été construit ou recueilli.

<editorialDecl> (déclaration des pratiques éditoriales) donne des précisions sur les pratiques et les principes éditoriaux appliqués au cours de l'encodage du texte.

<correction> (règles de correction) établit comment et dans quelles circonstances des corrections ont été apportées au texte.

<hyphenation> (césurage) résume la façon dont les traits d'union sécants de fin de ligne d'un texte source ont été traités dans sa version encodée.

<normalization> (normalisation) indique l'extension de la normalisation ou de la régularisation effectuée sur le texte source dans sa conversion vers sa forme électronique.

<punctuation> specifies editorial practice adopted with respect to punctuation marks in the original.

<profileDesc> Informations non bibliographiques

Cette partie concerne les informations non bibliographiques de description d'un document.

<langUsage> L'élément <langUsage> associé à <language> précise la ou les langues utilisées. L'identification de la langue doit être précisée par l'attribut @ident auquel est rattaché le code de la langue (format ISO 639-3)

```
<langUsage xml:id="langUsage">
  <language ident="fr">Français</language>
</langUsage>
```

Description d'un acte de correspondance : <correspDesc> Cet élément contient la description des actions relatives à un acte de correspondance. Il est complété par les sous-éléments:

<correspAction> contains a structured description of the place, the name of a person/organization and the date related to the sending/receiving of a message or any other action related to the correspondence

<correspContext> (correspondence context) provides references to preceding or following correspondence related to this piece of correspondence

<correspAction>

Cet élément peut contenir le nom de l'expéditeur ou du destinataire (**<persName>** ou **<orgName>**), le lieu d'expédition ou de réception (**<placeName>**), et la mention de date (**<date>**) tous relatifs à un acte d'envoi ou de réception d'une correspondance. l'attribut *type* doit venir préciser l'élément **<correspAction>** Afin d'identifier si c'est l'acte de expédition (sent) ou de réception (received) qui est décrit.

L'expéditeur et le destinataire seront identifiés par la valeur de l'attribut *type* du sous-élément **<persName>**, *sentBy* pour l'expéditeur et *deliveredTo* pour le destinataire.

exemple :

```
<correspAction type="sent">
  <persName ref="#Horel" type="sentBy">Armand Horel</persName>
  <placeName ref="#Toulon">Toulon</placeName>
  <date when="1915-08-08">08 août 1915</date>
</correspAction>
<correspAction type="received">
  <persName ref="#Comtesse_Forest"
    type="deliveredTo">Madame la Comtesse L. de la Forêt</persName>
</correspAction>
```

<correspContext>

Cette balise permet de replacer chronologiquement une correspondance dans un échange épistolaire et de lier les envois entre eux. La solution retenue est: la correspondance suivante et la précédente seront identifiées par l'élément **<ref>**, qui "définit une référence vers un autre emplacement", et l'attribut *type* de valeur *previous* pour la précédente et *next* pour la suivante. L'attribut *target* prendra la valeur du chemin d'accès vers les fichiers concernés. À la différence de l'élément **<ptr>** (pointeur), la référence peut être "complétée par un texte ou un commentaire". Dans le cas d'une correspondance, nous les complèterons avec les éléments **<persName>** (*type* de valeur *sentBy* ou *deliveredTo* afin d'identifier expéditeur et destinataire), **<placeName>** et **<date>**.

exemple

```
<correspContext>
  <ref target="../../Moudros/F_delta_1854_20_1_Moudros_03-09-1915.xml"
    type="previous">
    <persName ref="#Horel" type="sentBy">Armand
      Horel</persName>
    <persName ref="#Comtesse_Forest"
      type="deliveredTo">Madame la Comtesse L. de la
        Forêt</persName>
    <placeName ref="#Moudros">Moudros</placeName>
    <date when="1915-09-03">03
      septembre 1915</date>
  </ref>
  <ref target="../../Mytilene/F_delta_1854_20_1_Mytilene_06-10-1915.xml"
    type="next">
```



```
<persName ref="#Horel" type="sentBy">Armand
  Horel</persName>
<persName ref="#Comtesse_Forest"
  type="deliveredTo">Madame la Comtesse L. de la
  Forêt</persName>
<placeName ref="#Mytilene">Mytilene</placeName>
<date when="1915-10-06">06
  octobre 1915</date>
</ref>
</correspContext>
```

<correspDesc> et <teiCorpus>

Il est important, dans une édition de correspondance, d'établir une table des courriers envoyés. Si cette dernière est le plus souvent chronologique, elle peut aussi être organisée en fonction des expéditeurs et destinataires ou encore selon le lieu d'envoi. Un des points importants de ce projet étant la géolocalisation des courriers, il convient de réaliser, au sein du fichier maître (ou <teiCorpus>) une liste des toutes les lettres envoyées et organisée selon les lieux d'envoi.

Une balise <correspDesc> disposant d'un attribut *type*=sommaire doit être placée au sein de l'élément <teiHeader> relatif au corpus. L'élément <correspAction> doit être utilisé pour définir les différents rôles joués par les protagonistes. Ce sont bien les rôles qui doivent être décrits et non les protagonistes. Ainsi, si nous avons un des interlocuteurs qui a envoyé et reçu des lettres, cela sera encodé dans deux éléments <correspAction> différents et signifié par un attribut *type* d'une valeur sent ou received suivant qu'il soit qualifié d'expéditeur ou de destinataire et d'un attribut *subtype*=global. Au sein de cet élément <correspAction>, le nom de la personne (destinataire ou expéditeur) doivent être encodés avec un élément <persName> précisé par un attribut *xml:id*. Les lieux et dates sont respectivement encodés dans des éléments <placeName> et <date>, les différentes dates, correspondant à une session d'envoi sur un lieu donné sont identifiées par un identifiant *xml:id*.

```
<correspAction subtype="global" type="sent">
<!-- C'est le rôle d'expéditeur d'Armand Horel qui est décrit ici -->
  <persName ref="#Horel" xml:id="sent_Horel">Armand Horel</persName>
<!--Chaque lieu où le protagoniste a envoyé ses cartes et les sessions d'envoi sont
rappelés et identifiés par un @xml:id. -->
  <placeName ref="#Salonique">Salonique</placeName>
  <date corresp="#Salonique"
    when="1915-05-15" xml:id="envoi_salonique_1"/>
  <date corresp="#Salonique"
    from="1915-11-16" to="1916-11-04"
    xml:id="envoi_salonique_2"/>
  <placeName ref="#Corfou">Corfou</placeName>
  <date corresp="#Corfou" when="1916-11-04"
    xml:id="envoi_corfou_1"/>
  <placeName ref="#Mytilene">Mytilene</placeName>
  <date corresp="#Mytilene"
    when="1915-10-06" xml:id="envoi_mytilene_1"/>
  <placeName ref="#Imbros">Imbros</placeName>
  <date corresp="#Imbros" when="1915-09-23"
    xml:id="envoi_imbros_1"/>
  <placeName ref="#Moudros">Moudros</placeName>
  <date corresp="#Moudros" when="1915-09-03"
    xml:id="envoi_moudros_1"/>
  <placeName ref="#Toulon">Toulon</placeName>
  <date corresp="#Toulon" from="1915-08-08"
    to="1915-08-29" xml:id="envoi_toulon_1"/>
  <placeName ref="#Port_Iero">Port Iero</placeName>
  <date corresp="#Port_Iero"
```

```
from="1915-10-09" to="1915-11-10" xml:id="envoi_portiero_1"/>
</correspAction>
```

Dans un deuxième temps, un élément `<correspContext>` doit être créé par lieu d'envoi et comprendre une liste de sous-éléments `<ref>` comportant un *xml:id* (qui nous serviront pour l'index), un attribut *type=context* et enfin un attribut *corresp* relié au différentes dates listées dans les éléments `<correspAction>` (cela nous servira à établir une table chronologique des correspondances organisée par lieu). Chaque élément `<ref>` doit être complété par les informations d'identification de chaque carte, l'expéditeur doit être encodé avec la balise `<persName>` précisée par l'attribut *type=sentBy* et un attribut *corresp* lié à l'*xml:id* lié à un des `<correspAction>`. Le même traitement est réalisé pour le destinataire. Enfin, on retrouve la date et le lieu, qui ne posent pas de problème quant à leur encodage. Ce système permet d'inclure l'ensemble du `correspDesc` dans tous les fichiers de sortie destinés à être transformés en page HTML. Il permet par les différents liens qui sont opérés entre les dates, les personnes et les lieux d'envoi, de réaliser des tables en fonction de n'importe laquelle des informations d'identification et de les inclure dans un menu déroulant accessible depuis toutes les pages du site.

```
<correspContext xml:id="context_corfou">
<!--Les éléments <ref> sont liés par leur attribut @corresp aux éléments <date>
contenus dans un <correspAction>-->
  <ref corresp="#envoi_corfou_1"
    target="../Corfou/F_delta_1854_20_1_Corfou_04-11-1916.xml" type="context" xml:id="Corfou_04-11-1916">
    <!--Le même principe est retenu pour les personnes-->
    <persName corresp="#sent_Horel"
      ref="#Horel" type="sentBy">Armand Horel</persName>
    <persName corresp="#delivered_Comtesse"
      ref="#Comtesse_Forest" type="deliveredTo">la Comtesse L. de la Forêt</persName>
    <placeName ref="#Corfou">Corfou</placeName>
    <date cert="high" when="1916-11-04">04 novembre 1916</date>
  </ref>
</correspContext>
<correspContext xml:id="context_mytilene">
  <ref corresp="#envoi_mytilene_1"
    target="../Mytilene/F_delta_1854_20_1_Mytilene_06-10-1915.xml" type="context"
    xml:id="Mytilene_06-10-1915">
    <persName corresp="#sent_Horel"
      ref="#Horel" type="sentBy">Armand Horel</persName>
    <persName corresp="#delivered_Comtesse"
      ref="#Comtesse_Forest" type="deliveredTo">la Comtesse L. de la Forêt</persName>
    <placeName ref="#Mytilene">Mytilene</placeName>
    <date cert="high" when="1915-10-06">06 octobre 1915</date>
  </ref>
</correspContext>
```

Le corps de la dépêche : `<text>`

L'élément `<text>` contient l'ensemble du texte édité. Il peut être subdivisé en quatre sous-parties :

- `<front>` (texte préliminaire) contient tout ce qui est au début du document, avant le corps du texte : page de titre, dédicaces, préfaces, etc.
- `<group>` (groupe) contient un ensemble de textes distincts (ou des groupes de textes de ce type), considérés comme formant une unité, par exemple pour présenter les œuvres complètes d'un auteur, une suite d'essais en prose, etc.
- `<body>` (corps du texte) contient la totalité du corps d'un seul texte simple, à l'exclusion de toute partie pré- ou post-liminaire.

<back> (texte annexe) contient tout supplément placé après la partie principale d'un texte : appendice, etc.

front

Cet élément est optionnel. Il peut contenir la page de titre, les dédicaces, une préface ou encore l'introduction. La page de titre pourra être encodée avec l'élément **<titlePage>**.

Pour le reste, aucune balise spécifique n'a été prévue, l'introduction pourra donc être encodée au sein d'une **<div>** portant l'attribut *type* de valeur introduction.

Un attribut *xml:id* peut lui être fourni.

exemple :

```
<front>
  <div type="introduction"
    xml:id="introduction_generale">
    <p> La Correspondance d'<persName ref="#Horel">Armand
      Horel</persName> regroupe les cartes postales et
      lettres qu'il a envoyées à sa marraine de
      guerre, la <persName ref="#Comtesse_Forest">Comtesse de
      la Forest</persName>, plus connue sous le nom de
      <persName ref="#Silve_Claude">Claude
      Silve</persName> ; écrivain et prix
      fémina 1935. Ce fonds, issu d'un don, est
      composé de cent cinquante-quatre cartes postales,
      huit lettres, dont une de Louis Horel, le frère
      d'Armand et d'un programme de théâtre
      (« Achi Baba and the forty
      thieves »). Il est conservée sous la
      côte F Δ 1854/20 par la bibliothèque de
      documentation internationale contemporaine (BDIC).
      L'intégralité de ce fonds a été
      numérisé, il est disponible, sous licence
      libre, sur la bibliothèque numérique de la
      BDIC, l'Argonnote. </p>
    <p> Mon second paragraphe [...] </p>
    <p> Dans l'élément <gi>front</gi> sont également encodés les
      sujets des cartes au sein d'un élément <gi>argument</gi>.

  </div>
</front>
```

<back>

Cet élément est dévolu aux textes annexes. Il contient tous les suppléments placés après la partie principale d'un texte. Nous y placerons l'index ainsi que les notes comme décrit par la suite.

Balisateur du corps du document.

Introduction

L'encodage en XML-TEI ne permet pas un affichage direct en HTML; il faut pour cela passer par une feuille de style XSLT.

Il est nécessaire de garder à l'esprit que "toute édition constitue une diminution de la réalité" car elle est le résultat de choix opérés par l'éditeur.

Le modèle d'encodage proposé par la BDIC a été développé de manière à respecter au maximum la source. Les interventions opérées lors de l'encodage correspondent à des données qui sont ajoutées et non supprimées. Produire une version normalisée sera donc le simple résultat de l'application d'une feuille de style spécifique et non celui de choix éditoriaux irréversibles traités lors de la phase d'encodage.

Le Balisage physique

Cette section est relative à la structure physique du document originale. S'agissant de correspondances manuscrites, celles-ci peuvent être rédigées sur plusieurs feuillets délimités par des sauts de page etc. De même, au sein d'une page les lignes de texte peuvent être identifiées par des passages à la ligne; il peut aussi exister des changements de colonne dans le cas des cartes postales par exemple. Enfin, le texte peut également être mis en valeur grâce à d'autres manières (souligné, surligné, écrit en italique etc.)

Le saut de page : <pb> Le changement de feuillet ou le passage du recto au verso doit être encodé par l'utilisation de l'élément <pb>. Il peut être complété par l'attribut *n* afin de numéroté les pages. Dans ce cas, il faut également penser à en placer un avant le premier feuillet, afin que la numérotation soit complète. Dans le cas d'un corpus numérisé, les références aux fac-similés peuvent être faites par l'attribut *facs*. Enfin, si l'encodage porte sur des cartes postales, il est intéressant lorsque la correspondance tient sur plusieurs cartes de typer ce changement avec l'attribut *type* en lui donnant la valeur "postcardBreak"

exemple:

```
<pb facs="http://argonnaute.u-paris10.fr/medias/customer_3/archives/lettres_soldats_jpg/BDIC_FD_1
http://argonnaute.u-paris10.fr/ark:/naan/86a66lceae "
  n="3" type="postcardBreak"/>
<figure>
  <graphic url="../../../FACS/BDIC_FD_1854_20_1_179.jpg"/>
  <figDesc>
    <placeName ref="#Toulon">TOULON.</placeName> - Entrée de la nouvelle
    Caserne de Grignan </figDesc>
  </figure>
<pb facs="http://argonnaute.u-paris10.fr/medias/customer_3/archives/lettres_soldats_jpg/BDIC_FD_1
http://argonnaute.u-paris10.fr/ark:/naan/86a66lceae "
  n="4"/>
<p>
  <lb/>
  <add n="3" rend="left" type="numbering">
    <pc>.</pc>3<pc>.</pc>
  </add>
  <lb/>J'ai reçu
  une lettre de mon <rs ref="#Horel_Louis" type="person"
    xml:id="Horel_Louis_23091915">frère</rs>
  datée [...]
</p>
```

Les sauts de ligne <lb> Par convention, l'élément <lb> marque le début d'une nouvelle ligne ; il est donc préférable de le placer avant le texte. Les lignes peuvent être numérotées avec l'attribut *n* de la même manière que l'élément <pb>.

Coupure des mots La coupure d'un mot sera aussi matérialisée avec l'élément <lb> mais, cette fois-ci, précisé par l'attribut *rend* de valeur "hyphen" et l'attribut *type* de valeur "marked" lorsque la coupure est matérialisée par un tiret dans le texte. Dans le cas contraire, il convient d'utiliser la seule balise <lb>.

Afin de faciliter le développement d'un affichage ne tenant pas compte des changements de ligne, il conviendra de ne pas marquer le retour à la ligne ou un espace à la place de la coupure.

exemple :

```
<p> [...] <lb/>la figure, qui est ce que nous aper<lb rend="hyphen"/>cevons qui
planait au dessus de
nous ? [...] </p>
```

Lorsque que deux mots sont représentés agglutinés, ils seront encodés avec l'élément `<w>` complété de l'attribut *rend* de valeur "aggl".

exemple :

```
<w rend="aggl">plus</w>
<w>tard</w>
```

Les changements de colonnes : `<cb>`

(saut de colonne) marque le début d'une nouvelle colonne de texte sur une page multi-colonne. Cette balise est particulièrement intéressante dans le cas de cartes postales. Elle s'utilise de la même manière que les deux précédentes.

La mise en valeur : `<hi>`

(mis en évidence) distingue un mot ou une expression comme graphiquement distincte du texte environnant, sans en donner la raison. L'utilisation de cette balise doit être précisée par l'attribut *rend*. Cet élément et son attribut n'interprètent pas le texte contenu, ils précisent simplement son apparence physique. Dans le cadre d'une correspondance manuscrite, il peut s'agir d'une portion de texte soulignée, centrée, ou écrite en lettres capitales. Les valeurs de l'attribut *rend* seront donc, "underline", "center", "sc". Ces informations sont ensuite traitées avec la feuille de style XSLT.

Notons que tous les éléments peuvent être complétés par l'attribut *rend* et qu'il n'est donc pas toujours nécessaire de passer par l'élément `<hi>`.

Segmentation ou balisage logique

Le corps du document doit être balisé par l'élément `<body>`. Toutefois, chaque type de document (correspondance, poème, manuscrit ancien etc.) dispose d'une structure qui lui est plus ou moins propre et qui peut être répartie dans des divisions logiques. Une correspondance répond généralement d'une "structure diplomatique à la fois simple et flexible" : mention de date, apostrophe, paragraphes de texte, formule de courtoisie et signature. Afin d'uniformiser la sortie et de faciliter la lecture, il convient d'identifier ces différentes divisions.

La division : `<div>` Cette balise correspond à une subdivision du corps du texte édité. Cet élément doit être caractérisé par l'attribut *type* afin d'identifier au mieux son contenu. Dans le cadre de l'édition des correspondances, nous avons retenu deux valeurs pour l'attribut *type* qui sont "letter" et "enveloppe".

exemple :

```
<body>
  <div type="letter"> [Ici le corps de ma
    dépêche] </div>
  <div type="enveloppe"> [Ici
    la transcription des mentions présentes sur
    l'enveloppe] </div>
</body>
```

Formule d'ouverture : `<opener>` L'élément `<opener>` contient une formule d'ouverture. Pour une correspondance, il s'agit généralement des mentions de date et de lieu (`<dateline>`) ainsi que de la formule de politesse ou l'apostrophe (`<salute>`).

exemple :

```
<opener>
  <dateline>
    <placeName>Imbros</placeName>,
    <date when="1915-09-23">Le 23 septembre
      1915</date>
  </dateline>
  <salute>
    <persName>Comtesse</persName>,
  </salute>
</opener>
```

paragraphe : **<p>** La balise **<p>** représente un paragraphe. C'est à l'intérieur de cet élément que l'essentiel du texte de la correspondance sera placé. Chaque paragraphe dans le texte original doit être placé dans un élément **<p>**. Il peut être numéroté avec l'attribut *n* et s'il présente un alinéa, il sera indiqué par l'attribut *rend* de valeur indent.

Dans le cadre d'une correspondance tenant sur plusieurs cartes postales, et afin de rendre compte de cette matérialité, c'est le texte de chaque carte qui sera placé à l'intérieur d'un paragraphe.

Pour rappel, dans le cadre des cartes postales, l'élément **<pb>** devra être précisé par l'attribut *type="postcardBreak"*.

exemple :

```
<p>
  <lb/>Un bonjour de <placeName ref="#Port_Iero">Port
  <lb/>Iéro</placeName> d'un petit ami qui
  <lb/>ne vous oubliera jamais.
</p>
```

Formule de politesse/fermeture : **<closer>** **<closer>** permet d'encoder les formules de fin d'une correspondance. De la même manière que pour l'**<opener>**, on peut y trouver : une formule de courtoisie (**<salute>**), une signature (**<signed>**), une mention de date (**<dateline>**) ou encore une adresse (**<address>**).

Cet élément peut se répéter.

Postscriptum : **<postscript>** **<postscript>** trouve sa place soit avant soit après l'élément **<closer>**. Il peut reprendre une structure similaire au corps de la dépêche.

exemple :

```
<postscript>
  <p>
    <lb/>J'ai été content de
    ce courrier <lb/>car j'ai reçu de bonnes
    nouvelles <lb/>de la maison.
  <choice>
    <orig>m</orig>
    <reg>M</reg>
  </choice>es deux
  petites nièces <lb/>sont en excellente
  santé <pc type="supplied">;</pc>
  <lb/>elles sont hors de danger <lb/>de cette
  mauvaise <lb/>rougeole qui a
  <choice>
    <orig>regnée</orig>
    <reg>régné</reg>
  </choice>
  <lb/>chez nous pendant
```

```
<lb/>
<choice>
  <orig>l</orig>
  <reg>un</reg>
</choice>
mois environ. </p>
</postscript>
```

Le balisage éditorial

Dans la mesure où l'édition porte sur des documents qui n'ont jamais été édités, ce balisage fait essentiellement référence aux interventions ajoutées dans le texte par les encodeurs ceci dans le but d'en améliorer la compréhension et la lisibilité.

L'orthographe Lors de la transcription du texte, l'encodeur dispose de plusieurs possibilités quant à la gestion d'une orthographe fautive qui, faut-il le rappeler, est le reflet de la condition de l'auteur. Le transcripteur peut tout d'abord décider de moderniser cette orthographe "au fil de l'eau", lors de l'encodage du texte, perdant ainsi toute trace de la graphie originale. À l'inverse, il peut décider de ne rien corriger, ce qui peut poser problème lorsque l'orthographe est particulièrement déficiente. Une alternative est permise par la TEI. Il est possible de proposer une correction orthographique du texte tout en maintenant la forme fautive ou ancienne de l'auteur.

L'élément **<choice>** autorise de regrouper des balisages alternatifs (au moins deux) pour certains segments du texte.

Les fautes d'orthographe

Il est possible d'affiner sa correction en fonction des cas de figure. Il est parfois intéressant de distinguer les fautes d'orthographe récurrentes ou systématiques, de celles plus isolées. Dans la première éventualité, il peut s'agir par exemple d'un auteur qui confond systématiquement l'infinitif des verbes du premier groupe et leur participe passé ou encore qui utilise le verbe avoir à la place la préposition "à". Il conviendra d'utiliser dans ce cas les éléments **<orig>** pour la graphie originale, et **<reg>** pour la forme corrigée.

Concernant les fautes plus éparses, sur le même principe nous utiliserons alors les éléments **<sic>** pour l'orthographe fautive et **<corr>** pour la correction.

```
<p>
  <lb/>petites
nouvelles de guerre. Nous <lb/>sommes
<choice>
  <orig>appareiller</orig>
  <reg>appareillés</reg>
</choice>
pour l'île <lb/>
  <placeName ref="#Imbros">d'Imbros</placeName>
</p>
```

Abréviations

Dans la mesure du possible, les abréviations doivent être développées, à l'exception des plus courantes et des sigles. Il en va de même pour les noms de personnes s'ils peuvent être identifiés. Il convient d'utiliser la balise **<choice>** complétée par les éléments **<abbr>**, contenant l'abréviation, et **<expan>** pour la version développée. Il n'est en revanche pas nécessaire de distinguer les lettres ajoutées pour la résolution de l'abréviation.

Les nombres

S'il est décidé de moderniser l'orthographe, il convient de faire de même pour les nombres. Les chiffres doivent être écrits en toutes lettres à quelques exceptions près:

En chiffres arabes : les dates, énumérations, degrés, numéros d'ordre etc.

En chiffres romains : les siècles, les numéros dynastiques ou les régimes politiques, les régions militaires etc.

Les balises `<choice>`, `<reg>` et `<orig>` pourront être utilisées.

La ponctuation

Lorsque la ponctuation gêne la compréhension, parce qu'elle est fautive ou tout simplement absente, il convient de la corriger.

Toute marque de ponctuation sera encodée avec l'élément `<pc>`.

L'établissement d'une ponctuation inexistante devra être signifiée par l'attribut *type=supplied*.

Les éléments superflus seront aussi précisés par l'attribut *type* mais de valeur surplus.

Enfin, en cas de mauvaise utilisation d'une marque de ponctuation, il convient d'utiliser les éléments `<reg>` et `<orig>`.

Rétablissement des majuscules et des minuscules

Le rétablissement de la ponctuation nécessite bien souvent de rétablir l'utilisation des majuscules et des minuscules. Il en est de même dans le texte; en cas de correction, il faut veiller à rétablir les majuscules pour les débuts de phrase et les noms propres, et les minuscules pour les noms communs, à l'exception de noms abstraits ou lorsque l'auteur leur donne un sens ou une valeur particulière. Cet opération requiert l'utilisation des balises.

exemple : `<reg>` et `<orig>`

```
<lb/>car
j'ai reçu de bonnes nouvelles <lb/>de la
maison
<pc>.</pc>
<choice>
  <orig>m</orig>
  <reg>M</reg>
</choice>es
deux petites nièces <lb/>sont en excellente
santé
<pc type="supplied">,</pc>
```

L'État du texte

Il s'agit ici de rendre compte de l'état physique d'un texte, en signalant les parties abîmées, incomplètes, illisibles, ajoutées etc.

Lecture incertaine

Il convient de signifier cette incertitude par l'élément `<unclear>`, dont la raison sera précisée dans la mesure du possible par l'attribut *reason* (exemple de valeur, illegible (illisible), damage (dommage) etc).

Les manques

Toute partie manquante doit être encodée par l'élément `<gap>` qui peut être complété par le même attribut.

Les suppressions

L'élément `` identifie les parties qui ont été supprimées volontairement, soit par l'auteur (une rature par exemple), soit par un tiers. Il peut être précisé avec l'attribut *rend* et la valeur

overline lorsqu'il est simplement barré ou censored lorsque le texte a été censuré. L'attribut *hand* permet d'identifier la main responsable de cette suppression lorsqu'il ne s'agit pas de celle de l'auteur et qu'elle peut être identifiée.

Les ajouts

Les ajouts de l'auteur ou d'une autre personne seront encodés à la place où elles se situent dans le texte avec la balise `<add>`. Elle sera précisée par les attributs *place*, *rend* et *hand*. Lorsqu'il s'agit d'une signature ou d'une formule de politesse ajoutée en marge par manque de place, il convient de distinguer plusieurs cas de figure.

L'ajout est par exemple en marge mais à la fin du texte, il faut alors utiliser la balise `<closer>` complétée de la balise `<add>`

```
<closer>
  <salute>
    <add place="margin-right" rend="vertical">un petit ami qui vous envoie
      ses</add>
    <add place="margin-left" rend="vertical">meilleurs          souvenirs          de
  <placeName ref="#Salonique">Salonique</placeName>
    </add>
  </salute>
</closer>
```

La signature ou la formule de politesse sont rédigées sur un précédent feuillet. Il faut alors attribuer à l'élément `<rs>` l'attribut *type* avec les valeur `closer` et sera complété ainsi :

```
<rs type="closer">
  <add place="margin-right" rend="vertical"
    type="signed">
    <persName ref="#Horel"
      xml:id="Horel_27041916">ArmandHorel</persName>
    </add>
  <add place="margin-left" rend="vertical"> Bonne
    santé à tous et bonne <lb/>chance
  </add>
</rs>
```

Restitution

Lorsqu'il apparaît qu'un mot a été accidentellement oublié par l'auteur, il peut être restitué avec l'élément `<supplied>` et la raison identifiée par l'attribut *reason=omitted*. Les mots composés ne présentant pas leur tiret de liaison seront encodés comme suit :

```
sous<supplied reason="omitted">-</supplied>marin
```

Les notes

Les notes que l'on souhaite ajouter dans le texte doivent être encodées comme suit : au sein de l'élément `<back>`, il convient de créer une division `<div>` comportant l'attribut *type=note*. Nous y placerons autant d'élément `<note>` que nécessaire. Ils devront comporter un *xml:id* ainsi qu'un attribut de numérotation *n*. Dans le texte, à la place où doit apparaître la note, nous utiliserons un élément `<ref>` de *type=note* lié à la `<note>` correspondante par l'attribut *target*. Cette balise ne doit pas comporter de contenu. L'attribut *n* doit être de la même valeur que celui de la `<note>`.

```
<body>
  <p>
    <lb/>
```

```

<placeName ref="#Imbros">d'Imbros</placeName>
<ref n="1"
  target="#F_delta_1854_20_1_Imbros_23-09-1915_note1" type="note"/> qui se trouve
juste à
<lb/>l'embouchure du <placeName ref="#Golf_Saros"
  xml:id="Golf_Saros_23091915">golfe de
  <choice>
    <orig>Sarros</orig>
    <reg>Saros</reg>
  </choice>
</placeName>
<pc>.</pc>
</p>
</body>
<back>
  <div type="note">
    <note n="1"
      xml:id="F_delta_1854_20_1_Imbros_23-09-1915_note1">
      d'<placeName ref="#Imbros">Imbros</placeName> est restée, à L'île
      l'issue des guerres balkaniques, sous domination
      Ottomane. </note>
    </div>
  </back>

```

La balisage des noms et dates

Ce balisage permet d'identifier les noms propres et les dates afin de pouvoir les réutiliser ensuite avec un index ou un glossaire.

Noms de personnes : <persName>

Cet élément "contient un nom propre ou une expression nominale se référant à une personne. Il peut inclure tout ou partie de ses prénoms, noms de famille, titres honorifiques, noms ajoutés, etc.". Afin de le lier à l'index ou au glossaire, cet élément doit comporter l'attribut *ref* renvoyant à l'*xml:id* de l'entrée concernée ainsi que, pour la première occurrence de la lettre, un *xml:id* permettant de le localiser lors de la sortie HTML.

exemple :

```

<persName ref="#Horel"
  xml:id="Horel_a27-10_b3-11">Armand
Horel</persName>

```

L'élément <persName> doit être utilisé lorsque le nom de la personne est clairement identifié. Dans les autres cas, il convient de recourir à l'élément <rs> plus générique; il doit être précisé avec l'attribut *type*="person" et lié à l'index de la même manière.

```

<rs ref="#Horel_Louis" type="person"
  xml:id="Horel_Louis__09101915">mon
malheureux frère</rs>

```

Noms de lieux : <placeName>

Cet élément s'utilise de la même manière que le précédent.

```

<placeName ref="#Port_Iero"
  xml:id="Port_Iero_a27-10_b3-11">Port
Iéro</placeName>

```

Autres noms propres : **<name>**

L'élément **<name>** est plus générique que les deux balises précédentes. Il permettra d'identifier tous les autres noms propres qui ne correspondent pas à une personne ou à un lieu. Il sera précisé par un attribut *type*.

```
<name ref="#Shamrock_II" type="navire"
xml:id="Shamrock_II_27041916">Shamrock II</name>
```

Mention de date : **<date>**

L'élément **<date>** permet d'encoder les dates de n'importe quelle forme. Des attributs permettent de préciser et d'uniformiser la valeur. L'attribut *when* permet de standardiser la date sous la forme AAAA, ou, si la date est plus précise, AAAA-MM-JJ.

Les attributs *notBefore/notAfter* sont utilisés lorsque la date est imprécise.

L'attribut *cert* permet de préciser le degré de certitude ; les valeurs sont high(forte), medium(moyenne) et low(basse).

L'attribut *type=unknow* doit être utilisé lorsque la date est inconnue.

L'encodage des figures : **<figure>**

Lorsqu'une illustration est présente, il est intéressant de la signaler par l'utilisation de l'élément **<figure>**. Lorsqu'un facsimilé numérique est disponible, l'élément **<graphic>** aidé de l'attribut *url* permet de lier l'image au texte. Enfin l'élément **<figDesc>** permet de donner une description de cette image.

Dans le cas des cartes postales, la face illustrée sera encodée avec cet élément.

```
<figure>
<graphic url="../../../FACS/BDIC_FD_1854_20_1_181.jpg"/>
<figDesc>
  <placeName ref="#Toulon"
    xml:id="Toulon_23091915">TOULON.</placeName> - Le Casino
</figDesc>
</figure>
```

Index et Glossaire

De manière à ne pas surcharger l'apparat critique, les notes de bas de pages peuvent être limitées par la constitution d'un index-glossaire. Toutes les références nominales (ou tout du moins la première occurrence de chaque lettre) doivent être reliées à l'index afin d'être identifiées. Trois index peuvent être réalisés si besoin est ; un index nominum (**<listPerson>**), index des noms de lieux, (**<listPlace>**), et un index/glossaire (**<list>** *type=autre_index*). Chaque sous-éléments de ces listes (**<person>**, **<place>**, **<item>**) devront être identifiés par un attribut *xml:id*, afin de s'y référer lors de l'encodage du texte.

Ces index peuvent être placés dans le **<back>** et intégrés par l'intermédiaire d'**Xinclude**.

Index nominum : **<listPerson>**

L'élément **<listPerson>** permet la constitution d'un index des noms de personnes. Il est composé d'un ensemble de sous-éléments **<person>**, complété à son tour par des balises **<persName>**, **<birth>** et **<death>** pour les dates de naissance et de mort. Les éléments **<event>** et **<state>** rassembleront les informations concernant des événements ou un état qui les qualifie.

La réalisation d'un index structuré étant relativement complexe, il convient de remplir les entrées de manière standardisée et de respecter toujours le même ordre en fonction des cas de figure.

Enfin, les citations seront avec les éléments **<link>**

```

<person xml:id="Frederic_Guillaume">
  <persName type="nobility">
    <forename>Frédéric-Guillaume</forename>
    <nameLink>de</nameLink>
    <placeName>Prusse</placeName>
  </persName>
  <state>
    <p>
      <roleName type="nobility">Kronprinz</roleName>
    </p>
  </state>
  <birth when="1882"/>
  <death when="1951"/>
  <listEvent>
    <event>
      <p>Il Comande la V<hi rend="sup">e</hi> armée lors de la bataille
        de la Marne.</p>
    </event>
    <event>
      <p>En 1916 il
        mène les attaques contre
        Verdun.</p>
    </event>
    <event>
      <p>Comme son père il
        abdique en novembre
        1918.</p>
    </event>
  </listEvent>
  <link corresp="#Salonique_25-01-1916"
    target="#Frederic_Guillaume_25011916" type="cit"/>
</person>

```

Notons que l'attribut *target* de l'élément `<link>` renvoie à la citation dans le corps de la dépêche. L'attribut *corresp* quant à lui, cible la lettre concernée dans le `<correspDesc>`

Index des noms de lieu : `<listPlace>`

`<listPlace>` doit être complété sur le même principe que l'élément `<listPerson>`. L'index des noms de lieux doit rendre compte des spécificités locales (pays pour les lieux étrangers, découpage administratif etc.) L'élément `<placeName>` sera typé avec les valeurs *type=fr* pour les lieux français et *type=etranger* pour les autres. Ces entrées peuvent aussi être précisées avec l'emploi des éléments `<event>` et `<state>`.

```

<place xml:id="Gallipoli">
  <placeName corresp="#Canakkale"
    type="renvoi">Gallipoli</placeName>
</place>
<place xml:id="Canakkale">
  <placeName type="etranger">Çanakkale</placeName>
  <location>
    <country>Turquie</country>
    <district type="region">Marmara</district>
  <!-- région -->
    <district type="province">prov.
      Çanakkale</district>
  <!-- district régional -->
    <district type="district">dist.
      Marmara</district>
  <!-- communauté -->
</location>
  <link corresp="#Toulon_08-08-1915"

```

```
target="#Gallipoli_08081915" type="cit"/>
</place>
<place xml:id="Sanary">
  <placeName type="fr">Sanary-sur-Mer</placeName>
  <location>
    <country>France</country>
    <district type="departement">Var</district>
    <district type="arrondissement">arr.
      Toulon</district>
  </location>
  <link corresp="#Port_Iero_10-11-1915"
    target="#Sanary_10111915" type="cit"/>
  <link corresp="#Salonique_23-11-1915"
    target="#Sanary_23111915" type="cit"/>
  <link corresp="#Salonique_07-12-1915"
    target="#Sanary_07121915" type="cit"/>
  <link corresp="#Salonique_17-12-1915"
    target="#Sanary_17121915" type="cit"/>
  <link corresp="#Salonique_20-12-1915"
    target="#Sanary_20121915" type="cit"/>
  <link corresp="#Salonique_27-12-1915"
    target="#Sanary_27121915" type="cit"/>
  <link corresp="#Salonique_07-01-1916"
    target="#Sanary_07011916" type="cit"/>
  <link corresp="#Salonique_19-01-1916"
    target="#Sanary_19011916" type="cit"/>
  <link corresp="#Salonique_31-01-1916"
    target="#Sanary_31011916" type="cit"/>
  <link corresp="#Salonique_05-02-1916"
    target="#Sanary_05021916" type="cit"/>
  <link corresp="#Salonique_21-02-1916"
    target="#Sanary_21021916" type="cit"/>
  <link corresp="#Salonique_10-03-1916"
    target="#Sanary_10031916" type="cit"/>
  <link corresp="#Salonique_16-03-1916"
    target="#Sanary_16031916" type="cit"/>
  <link corresp="#Salonique_22-03-1916"
    target="#Sanary_22031916" type="cit"/>
</place>
```

Autre index : <list>

L'élément <list> permet de réaliser des entrées d'index ou de glossaire à partir de termes qui ne peuvent être placés dans les <listPerson> et <listPlace>. Il s'utilise de la même façon que les deux précédents.

```
<item xml:id="Lavoisier">
  <label type="navire">
    <name type="navire">Lavoisier</name>
    <rs>le</rs>
  </label>
  <desc>croiseur de 2e
    classe<date from="1897" to="1920"/>
  </desc>
  <listEvent>
    <event>
      <p>
        <date when="1915-12-30">1915 (décembre) </date>
        Envoyé à <placeName ref="#Port_Saïd">Port-Saïd</placeName>
      </p>
    </event>
  </listEvent>
</item>
```

```

    <p>
      <date when="1916-02">1916 (février)</date>
      Affecté au secteur de
      Méditerranée
      occidentale</p>
    </event>
    <event>
      <p>
        <date when="1916-09">1916 (septembre)</date> Envoyé
        au <placeName ref="#Maroc">Maroc</placeName>
      </p>
    </event>
    <event>
      <p>
        <date when="1919">1919</date> Stationnement en
        Syrie</p>
      </event>
    </listEvent>
    <link corresp="#Port_Iero_10-11-1915"
      target="#lavoisier_10111915" type="cit"/>
  </item>
  <item xml:id="Boeuf">
    <label type="navire">
      <name type="navire">Boeuf</name>
      <rs>le</rs>
    </label>
    <desc>remorqueur<date from="1909" to="1942"/>
    </desc>
    <link corresp="#Salonique_24-01-1916"
      target="#Boeuf_24011916" type="cit"/>
  </item>

```

Contents

Annexe B

Images



FIGURE B.1 – Exemple de note en survol.

<i>Bulgarie.</i> Citations : Port Iero, 09 oct 1915 — Salonique, 16 nov 1915
BUSSIÈRE (les), Proches de Madame la Comtesse de la Forest et d'Armand Horel . Citations : Salonique, 17 mar 1916 — Salonique, 09 avr 1916 — Salonique, 27 avr 1916 — Corfou, 04 nov 1916
<i>Çanakkale</i> , Turquie, Marmara, prov. Çanakkale, dist. Marmara. Citations : Toulon, 08 août 1915
« Charlemagne » (le), cuirassé type Charlemagne, (1897-1920). Citations : Salonique, 16 nov 1915
CONSTANTIN IER DE GRÈCE, 1868-1923, Roi des Hellènes. Citations : Salonique, 16 nov 1915 — Salonique, 23 nov 1915
<i>Corfou</i> , Grèce, Îles Ioniennes, dist. Corfou. Citations : Salonique, 17 mar 1916 — Salonique, 28 mar 1916 — Salonique, 21 avr 1916 — Corfou, 04 nov 1916
<i>Corinthe</i> , Grèce, Péloponnèse, dist. Corinthe, dème de Corinthe. Citations : Salonique, 25 jan 1916

FIGURE B.2 – Exemple d'une partie de l'index.

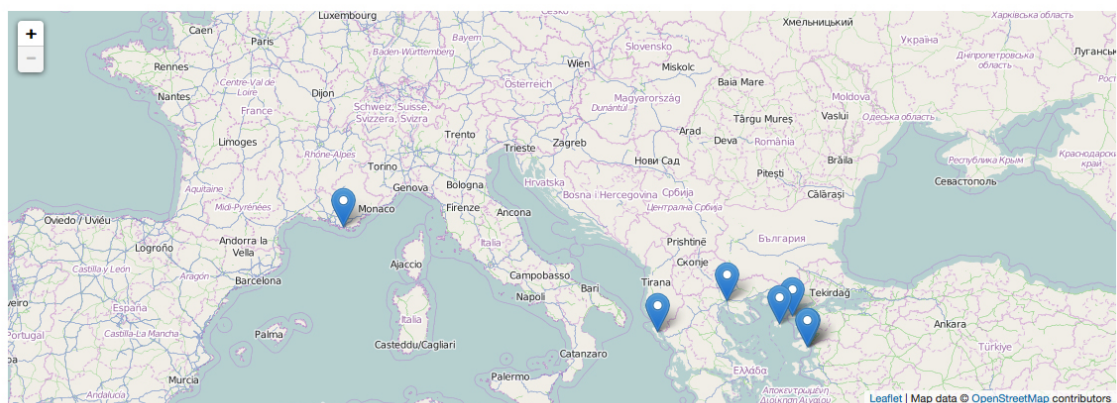


FIGURE B.3 – Cartographie.

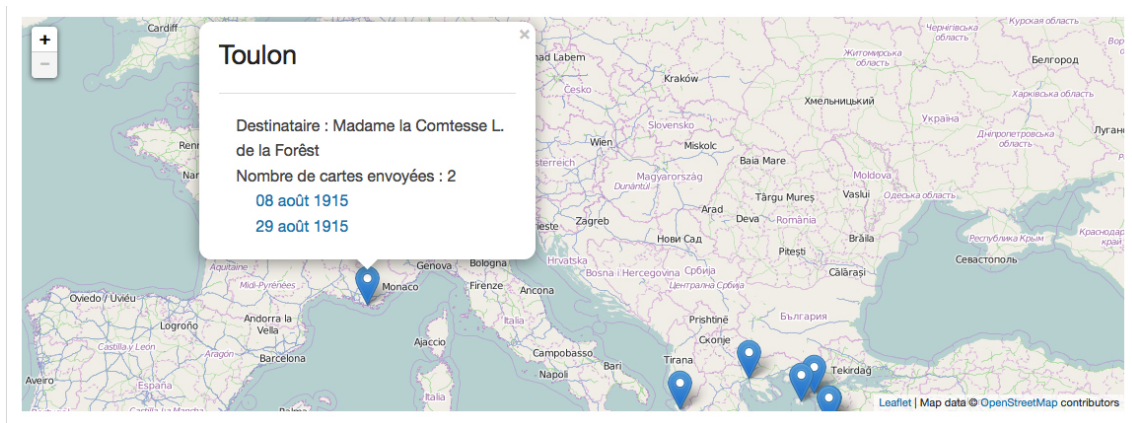


FIGURE B.4 – Cartographie.

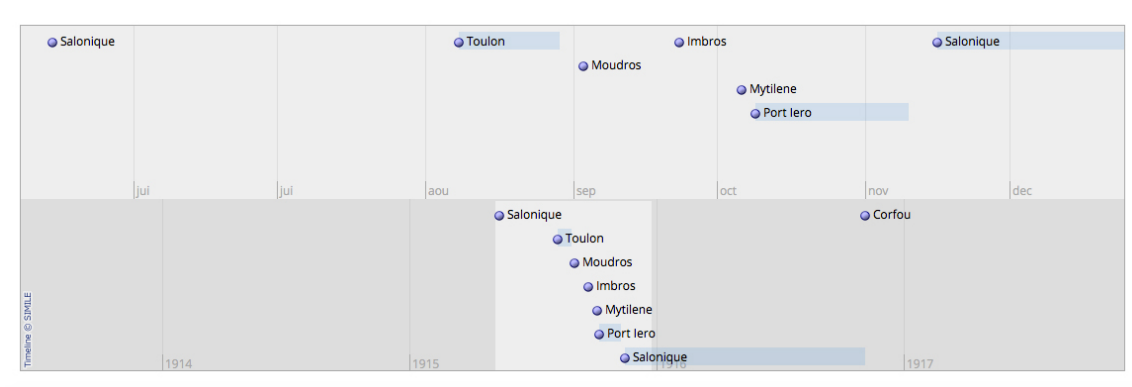



FIGURE B.5 – Frise chronologique.

Nouvelle	Personnaliser	Langage	Modules	Ajouter des éléments	Modifier les classes	Schéma	Documentation	Enregistrer	Contrôleur de validité
----------	---------------	---------	---------	----------------------	----------------------	--------	---------------	-------------	------------------------

Liste des modules TEI				
	Titre du module	Description brève	Modification	
ajouter	analysis	❓ Mécanismes analytiques simples	modifié	
ajouter	certainty	❓ Degré de certitude et responsabilité		
ajouter	core	❓ Éléments disponibles pour tous les documents TEI	modifié	
ajouter	corpus	❓ Corpus linguistiques		
ajouter	dictionaries	❓ Dictionnaires		
ajouter	drama	❓ Théâtre		
ajouter	figures	❓ Tableaux, formules et graphiques		
ajouter	gaiji	❓ Représentation des caractères et des glyphes non standard		
ajouter	header	❓ En-tête TEI	modifié	
ajouter	iso-fs	❓ Structures de traits		
ajouter	linking	❓ Liens, segmentation et alignement	modifié	
ajouter	msdescription	❓ Description de manuscrits		
ajouter	namesdates	❓ Noms, dates, personnes et lieux	modifié	

Liste des Modules sélectionnés	
supprimer	core
	tei
supprimer	header
supprimer	textstructure
supprimer	transcr
supprimer	namesdates
supprimer	msdescription
supprimer	figures
supprimer	analysis
supprimer	corpus
supprimer	linking

FIGURE B.6 – Roma 2



Roma: Production des validateurs TEI

Vous travaillez actuellement sur **Schéma TEI poi**

Modification d'un module

Nouvelle	Personnaliser	Langage	Modules	Ajouter des éléments	Modifier les classes	Schéma	Documentation	Enregistrer	Contrôleur de validité
----------	---------------	---------	---------	----------------------	----------------------	--------	---------------	-------------	------------------------

[back](#)

Liste des éléments compris dans ce module namesdates					
	Inclure	Exclure	Nom	Description	Attributs
addName	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="addName"/>	❓ (nom additionnel) contient une composante de nom additionnelle, comme un surnom, une épithète, un alias ou toute autre expression descriptive utilisée dans un nom de personne.	Changer les attributs
affiliation	<input type="radio"/>	<input checked="" type="radio"/>	<input type="text" value="affiliation"/>	❓ (affiliation) contient une description non formalisée portant sur l'affiliation présente ou passée d'une personne à une organisation, par exemple un employeur ou un sponsor.	Changer les attributs
age	<input type="radio"/>	<input checked="" type="radio"/>	<input type="text" value="age"/>	❓ (âge) précise l'âge d'une personne	Changer les attributs
birth	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="birth"/>	❓ (naissance) contient des informations sur la naissance d'une personne, comme la date et le lieu	Changer les attributs
bloc	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="bloc"/>	❓ (bloc) contient le nom d'une unité géo-politique composée d'au moins deux états ou pays	Changer les attributs
climate	<input type="radio"/>	<input checked="" type="radio"/>	<input type="text" value="climate"/>	❓ (climat) contient des informations sur le climat physique d'un lieu.	Changer les attributs
country	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="country"/>	❓ (pays) contient le nom d'une unité géo-politique, comme	Changer les attributs

FIGURE B.7 – Roma 1

Table des figures

B.1 Exemple de note en survol.	97
B.2 Exemple d'une partie de l'index.	97
B.3 Cartographie.	98
B.4 Cartographie.	98
B.5 Frise chronologique.	98
B.6 Roma 2	99
B.7 Roma 1	99

Table des matières

Résumé	iii
Remerciements	v
Introduction	vii
 I Éditer une correspondance de poilus	 1
1 Mettre en valeur des sources	3
1.1 L'édition de correspondances	3
1.1.1 Délimitation d'un corpus	3
1.1.2 Mettre en valeur des sources	4
1.2 Quel type d'édition	5
1.2.1 Définir une édition	5
1.2.2 Les principes éditoriaux retenus	6
 2 Une édition électronique XML-TEI	 11
2.1 Le recours à un balisage descriptif	12
2.2 Le dialecte XML-TEI	13
2.3 La nécessité de personnaliser le schéma TEI	14
 II Un modèle d'édition XML-TEI	 17
 3 Rappels généraux et notions de corpus électronique	 19
3.1 La structure générale du schéma TEI	19
3.2 Création d'un corpus électronique	21
 4 La personnalisation XML-TEI	 25
4.1 Identification des lettres et métadonnées	26
4.2 Le balisage du texte	30
4.2.1 Le balisage logique	30

4.2.2	Balisage physique	33
4.2.3	Balisage diplomatique	35
4.3	Normalisation du texte	38
4.4	Réalisation d'une table des correspondances	39
4.5	La création d'un index	42
4.5.1	Indexation des entrées du texte	43
4.5.2	L'index des noms de personnes	44
4.5.3	Index des noms de lieux	46
5	Personnalisation TEI	49
5.1	Création d'un fichier ODD	49
5.1.1	Utilisation de Roma	50
5.1.2	Personnalisation manuelle du schéma	51
5.1.3	Rédaction d'un manuel d'encodage	52
III	La publication web	55
6	Transformation XML vers HTML	59
6.1	Principes de base	59
6.1.1	Application de principes éditoriaux	59
6.1.2	Mise en page	60
6.2	Fonctions avancées	61
6.2.1	Les boucles	61
6.2.2	Les conditions	62
6.2.3	Les modes	63
7	L'utilisation d'un framework CSS	65
8	Intégration d'outils interactifs	67
8.1	Cartographie	67
8.1.1	Intégration d'une carte	67
8.1.2	Récupération des informations par une feuille de style XSLT	69
8.2	Intégration d'une frise chronologique	69
	Conclusion	71
	Annexes	75
A	Manuel d'encodage	75

<i>TABLE DES MATIÈRES</i>	105
B Images	97
Table des figures	101
Table des matières	103