Which TEI representation for the output of automatic transcriptions and their metadata? An illustrated proposition

Hugo Scheithauer, Alix Chagué, Laurent Romary

Which TEI Representation for the Output of Automatic Transcriptions and Their Metadata? An Illustrated Proposition

Hugo Scheithauer[1], Alix Chagué[1, 2, 3], Laurent Romary[4]

[1] ALMAnaCH - Automatic Language Modelling and ANAlysis & Computational Humanities, Inria Paris
[2] UdeM - Université de Montréal
[3] EPHE - École pratique des hautes études
[4] Inria, Directorate for Scientific Information and Culture

Abstract

The recent and fast development of automatic transcription software is accompanied by a growing heterogeneity of formats to save their output. TEI P5 can help to simplify workflows and bring more coherence into digitization pipelines. We present a twofold modelization in TEI which brings essential information resulting from the transcription phase together with the editorial layers. The usefulness of this modelization is illustrated with several examples showing how such an approach can be leveraged at different stages of a digitization process.

Keywords: HTR; OCR; digital edition; metadata; layout; eScriptorium; TEI Publisher.

Introduction

The recent growth of computer power available for processing data with machine learning techniques—particularly for Deep Learning contexts—has led to the faster development of technologies such as optical character recognition (OCR), and recently handwritten text recognition (HTR). It is now possible to envision reading manuscripts from the twentieth century (Massot, Sforzini, and Ventresque 2019) as easily as those from the thirteenth century.[1] The main challenge for automatic transcription is the

[1] See, for example, the CREMMA (Consortium pour la reconnaissance d'écriture manuscrite des matériaux anciens) Lab post-doctorate work of Ariane Pinche (École nationale des chartes

variation of letter shapes through time and space, even within the same alphabetical system. However, thanks to the development of the appropriate software and tools with relatively accessible interfaces, the technology has now moved from computer vision labs to those in social sciences and the humanities, as well as to cultural institutions. Such software accelerates our capacity to create digital text which can then be used for research projects or for the digital editions of large cultural heritage collections. Meanwhile, new workflows are emerging for digitizing and editing texts. It has now become crucial to address the question of how to keep and organize all the information deriving from the automatic transcription process in a reliable and sustainable manner.

Expanding the Role of the TEI Guidelines in the Context of HTR

In the field of automatic text recognition, whether applied to printed documents or to manuscripts, two de facto standards are currently used to record the output of the recognition process: ALTO XML[2] and PAGE XML (Pletschacher and Antonacopoulos 2010).[3] These two XML formats are mainly intended to render the details of the layout of the original documents in combination with the textual elements that have been recognized. They thus follow a representation paradigm which is far from sufficient when the underlying objective is to produce complex digital editions. In contrast, TEI XML is specifically appropriate for digital editions because it enables one to simultaneously encode the intellectual organization of the document and also its physical characteristics. ALTO and PAGE also work at the image level, a document is then fragmented into several files, whereas TEI allows putting a whole document into a single XML file. For

(ENC) – Centre Jean Mabillon (CJM) – Inria), accessed February 12, 2022, https://cremmalab.hypotheses.org/.

[2] See the Analyzed Layout and Text Object (ALTO) 4.2 schema specifications: release announcement accessed April 23, 2024, https://www.loc.gov/standards/alto/v4/alto-4-2.xsd.

[3] See also the PRImA (Pattern Recognition & Image Analysis) Research Lab's PAGE XML GitHub repository, accessed April 23, 2024, https://github.com/PRImA-Research-Lab/PAGE-XML.

these reasons, it is common that once the transcription process is over, and the creation of the edition begins, editors move on to TEI XML and keep only the information necessary for the edition. This usually forces projects to manage an aggregate of heterogeneous formats in order to (a) keep the precise information resulting from the HTR process and (b) enrich the content in the form of a structured edition, while avoiding losing the connection between the two stages.

We would like to advocate here for a change of paradigm where the TEI Guidelines would be given a stronger and earlier role in these workflows, with the advantage of better organizing the articulation between the various transcription, edition, and enrichment levels. Thus, our question is the following: how far can we map the results of an automatic transcription process to TEI, and by which representational means? We also argue that either ALTO XML or PAGE XML can be used as an intermediary representation to generate a comprehensive TEI file containing all data produced by automatic text recognition software.

Initially motivated by reflections conducted within the framework of the LECTAUREP project (which we will present later), we tried to generalize our proposal to as many use cases as possible. Our experiment is based on the use of eScriptorium, a virtual research environment (VRE) which combines a web interface and the OCR/HTR engine Kraken (Stokes et al. 2021).[4] Another notable HTR engine used by the digital humanities community is Transkribus (Mühlberger et al. 2019). They both offer the option to export the output of the automatic transcription in formats such as plain text, ALTO XML, and

[4] See also Benjamin Kiessling's Kraken OCR system repository on GitHub, accessed March 23, 2022, https://github.com/mittagessen/kraken.

PAGE XML.[5] Transkribus has also implemented the ability to export to TEI XML[6] and other formats like .docx. Apart from ALTO and PAGE, all these formats generate representations that cannot be reinjected into the automatic transcription software since they lose a great deal of the segmentation and layout information. It is worth noting that ALTO and PAGE still suffer from a lack of normalization and compatibility. For instance, in 2020, ALTO files exported with Transkribus were not compatible with the eScriptorium import module because the former used ALTO 2 and the latter ALTO 4.[7] Users may also opt for the PAGE format, but this choice amplifies the potential for numerous incompatibilities across different formats and their respective versions. Moreover, PAGE and ALTO are not compatible with each other, and without a clear choice from the community, archive users in the future will have to deal with two different formats that need to be converted back and forth, if the conversion is eventually made possible. PAGE also appears to receive infrequent updates, with its most recent update dating back to 2019. In contrast, ALTO demonstrates ongoing evolution, with its latest update being as recent as April 2023 (4.4 version). Using TEI XML would allow end users to rely on one single format to standardize automatic text recognition output in the context of textual edition, and would then ease archiving such files. One of our objectives is to improve the reusability of contents generated through HTR and OCR at various stages of the workflow leading to a digital edition. Using TEI XML to encode

---

[5] The implementation of these two export formats entails issues that are due to the absence of a normalization strategy, which results in tedious transitions from one interpretation of the format to another (Transkribus' ALTO and eScriptorium's ALTO are not immediately compatible) and from one version to the next (Transkribus' shift from ALTO 2 to ALTO 4 in March 2021 broke many pipelines based on ALTO 2).

[6] The XSLT for this export can be found in the page2tei GitHub repository of Dario Kampkaspar, accessed March 23, 2022, https://github.com/dariok/page2tei.

[7] See Aspyre-GT, a python tool developed in 2020–21 to make Transkribus' ALTO 2 files compatible with eScriptorium. The transformation scenario was later extended to other sorts of ALTO files produced by software like LIMB: see the GitHub repository of Alix Chagué, accessed July 8, 2023, https://github.com/alix-tz/aspyre-gt.

pivot files would mean aggregating the raw transcription with its edited version into one single file. Ideally, this would allow users to go back and forth between these two states of the text while avoiding complex manipulations of the data. Such a pivot file entails making the TEI encoding compatible with the requirements of automatic transcription programs and their traditional formats, so that all information would be encoded in TEI XML and could then be easily converted back,[8] but it could also improve the propagation of the metadata throughout the pipeline.

New issues are also currently arising following the creation of large corpora produced with automatic transcription technologies, such as the need to reconstruct the logical structure of digitized text documents (Clérice 2023). Having access to layout information allows semi-automatic and automatic processing in order to do so. Layout analysis is at the interface of computer vision and information extraction, and can be done manually with segmentation software, or with machine learning models (Clérice 2023; Oliveira, Seguin, and Kaplan 2018). Converting annotations directly to TEI is not the scientific norm. To the best of our knowledge, only GROBID takes an image with a text layer as input and outputs a series of TEI XML files based on the inferences of different CRF models using layout token features (Khemakhem 2020).[9] Otherwise, layout encoding is usually done manually or semi-automatically using scripts that convert ALTO XML and PAGE XML to TEI, based on text line annotations and content.

Our approach has also been inspired by the TEI in Libraries initiative, which also envisioned bringing together automatic transcription and more structured TEI-based representations. Formed in 1999 as a TEI workgroup, TEI in Libraries[10] released its most

---

[8] With an XSLT stylesheet, or with a script, for instance.

[9] See, for example, the machine learning library GROBID, accessed March 23, 2022, https://grobid.readthedocs.io/en/latest/Principles/#layout-tokens-not-text.

[10] See the workgroup's wiki, accessed March 25, 2022, https://wiki.tei-c.org/index.php/Workgroup_to_revise_the_Best_Practices_for_TEI_in_Libraries, and the workgroup's GitHub repository, accessed March 25, 2022, https://github.com/kshawkin/Best-Practices-for-TEI-in-Libraries.

recent version in 2018. It recommended best practices for using TEI in the context of library text digitization projects.[11] A notable aspect of the initiative is that it proposed to keep OCR output in a TEI XML file, and established key concepts which paved the way for the direction we decided to take: TEI can "be suited to the goals of a preservation unit or mass digitization initiative," and it allows a text "to be a faithful representation of the appearance of the source document derived from OCR."[12] In this paper, we will go a step further by considering text as data, as argued by Burnard, Schöch, and Odebrecht (2021), and specifically address the challenges now posed by a greater use of OCR and HTR in a variety of new contexts, with a specific emphasis on modeling layout information in TEI documents.

OCR and HTR Terminology

Before entering into the details of our TEI modelization for the encoding of automatic transcription output, we would like to introduce the reader to a few aspects of the OCR and HTR terminology, which we will later map onto TEI concepts.

In most automatic transcription software components, an image can be divided into one or more **text regions** (also commonly called zones, or text zones). They are used to mark zones in a page that usually bear a semantic significance. Figure 1 represents a poem, "Le Pont Mirabeau," written by Guillaume Apollinaire, circa 1912, whose manuscript is held at the French National Library (BnF). The layout is fairly simple, consisting of a page number, a title, and stanzas. All of them are directly identified on the image as text regions. We should mention that there exist initiatives that intend to build ontologies to normalize the naming of the zones and their use, such as SegmOnto (Gabay et al. 2021).

---

[11] See Best Practices for TEI in Libraries: A Guide for Mass Digitization, Automated Workflows, and Promotion of Interoperability with XML Using the TEI, accessed March 23, 2022, https://tei-c.org/extra/teiinlibraries/.

[12] See TEI in Libraries, especially v. 4.0.0, edited by Kevin Hawkins, Michelle Dalmau, Elli Mylonas, and Syd Bauman, published September 2018, accessed March 23, 2022, https://tei-c.org/extra/teiinlibraries/4.0.0/bptl-driver.html.

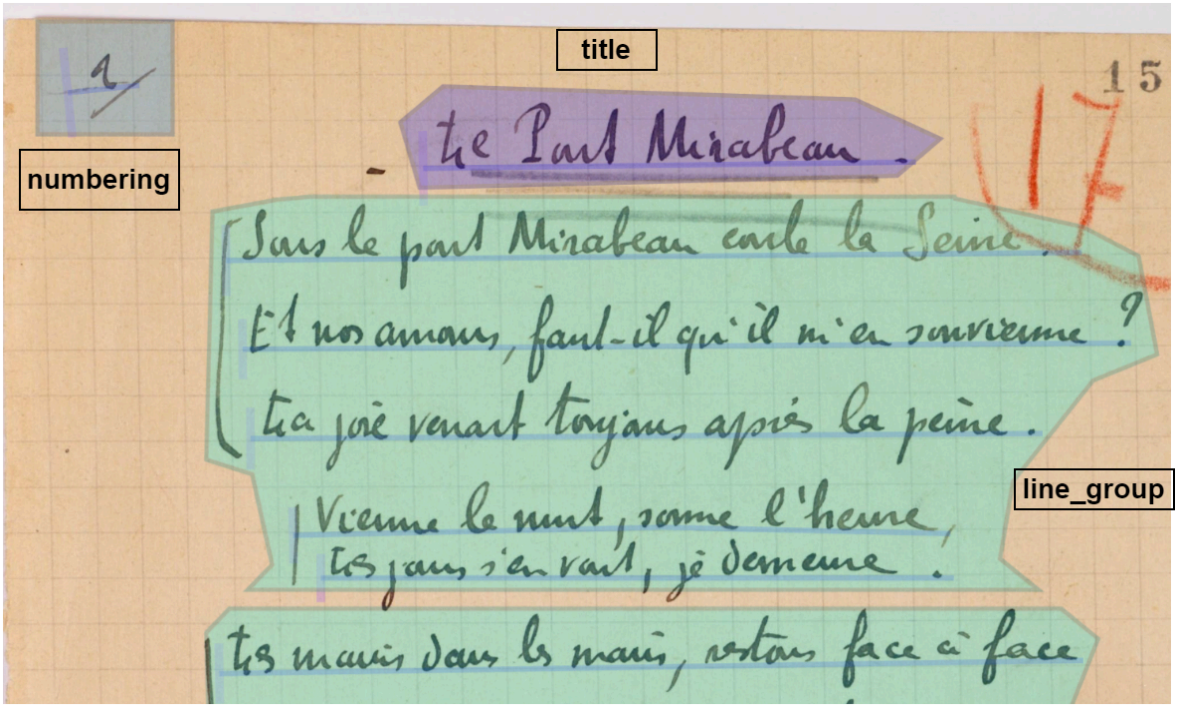

Figure 1. Layout annotation of a poem written by Guillaume Apollinaire, "Le Pont Mirabeau," ca. 1912, Bibliothèque nationale de France (BnF). https://gallica.bnf.fr/ark:/12148/btv1b525056707/f33/.

A zone normally contains one or more lines of text. The lines themselves are composed of three elements (see fig. 2): a **baseline** or **topline** defines a virtual line, passing through at least two points, on which the text is written or from which it is hanging; a **mask** is a polygon, defined by at least three points, which delimits the area of pixels containing the text of the line; and lastly, the **text** itself.

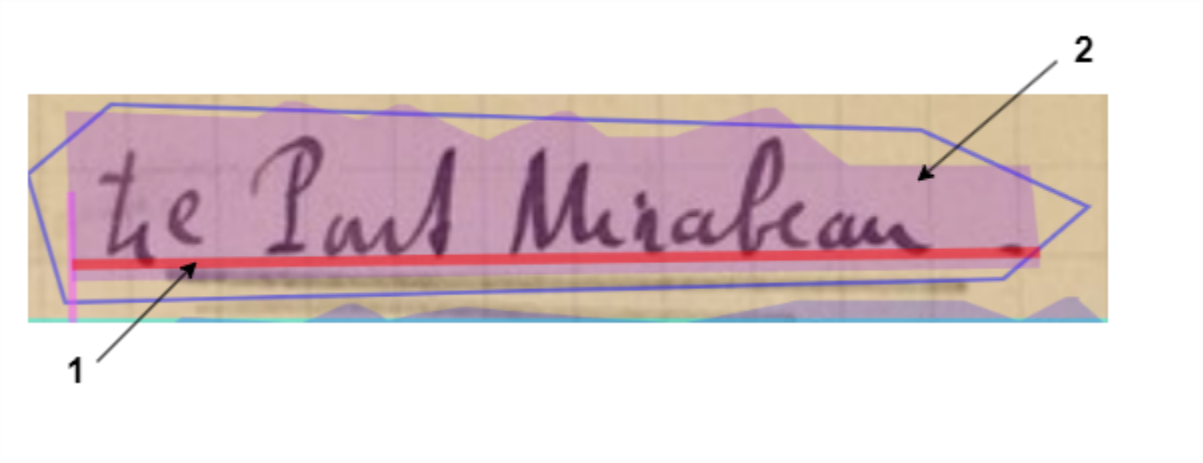

Figure 2. A text line corresponding to the poem's title, its baseline (red, 1), and mask (purple, 2). Guillaume Apollinaire, "Le Pont Mirabeau," ca. 1912, Bibliothèque nationale de France (BnF). https://gallica.bnf.fr/ark:/12148/btv1b525056707/f33/

Encoding Automatic Text Transcription in TEI

As of March 2022, the ALTO and PAGE XML formats used in eScriptorium to export the recognition output include only rudimentary metadata about the transcription itself. The resulting PAGE XML files, for instance, identify the creator (i.e., the application used: eScriptorium) and changes made to the transcription on the platform, such as the dates of creation and of the last modification.[13] Such information is quite straightforward to map onto **<teiHeader>** components, using **<respStmt>** for information about the users and the software and **<revisionDesc>** for temporal information. This can be complemented with another mapping, for information such as the title of the document or any identifier linked to the image, within **<titleStmt>** (see fig. 3).

[13] As of March 2022, the current version of eScriptorium treats the date of creation as the date of last modification. See the FAIR Principles website, accessed April 24, 2024, https://www.go-fair.org/fair-principles/.

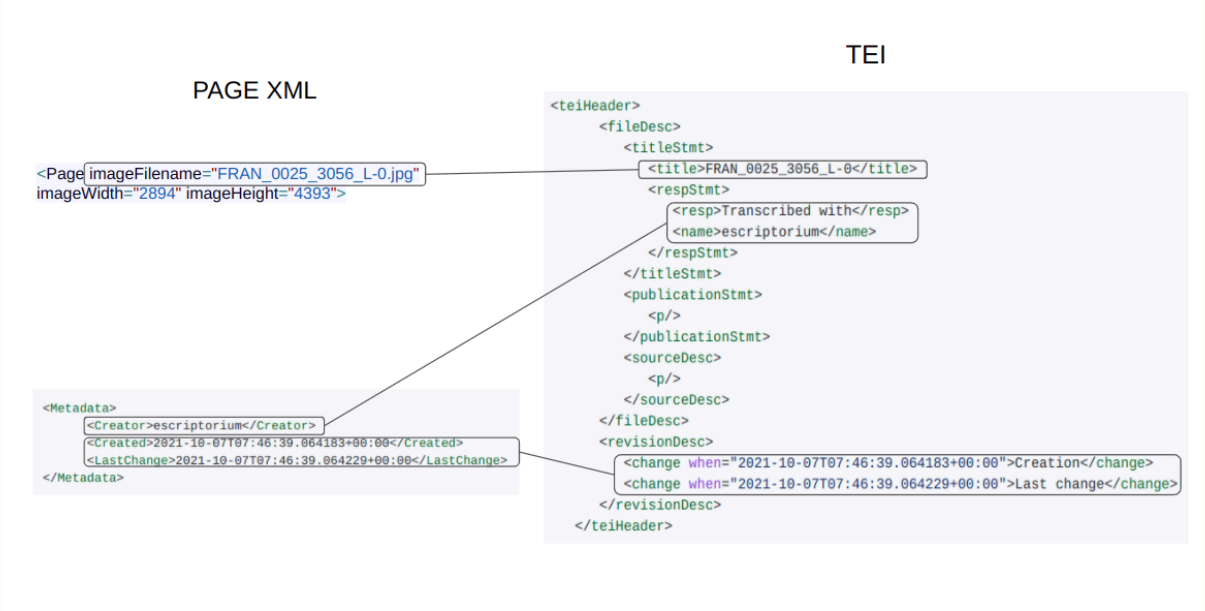

Figure 3. Documenting a transcription: metadata representation from ALTO XML to TEI. Created by the authors.

As of now, eScriptorium does not provide enough metadata for ensuring FAIRness[14]. However, with the continuous development of automatic transcription software, more complex and detailed metadata will become available in the future. The **<teiHeader>** already offers the needed components to fully document most of those we can already foresee. For instance, the metadata attached to the transcription models—such as the name of the model, who trained it, the OCR/HTR engine used and its version, or the list of codecs known by the model—could be given in the **<respStmt>**. The name of the software and of the individuals or organizations who modify the transcription afterwards could be indicated there as well.

Details on the transcription itself, such as the transcription guidelines, could be supplied in the **<editorialDecl>** (see example 4). This element already offers a set of sub-elements which can be used to describe regular decisions often made before training and applying a transcription model, such as hyphenation (**<hyphenation>**); normalization

[14] FAIRness refers to the principle of making data Findable, Accessible, Interoperable, and Reusable. These principles are designed to ensure that data is managed in a way that maximizes its potential for discovery and reuse.

(**<normalization>**), that is, how diacritics and other typographic elements were transcribed; and how punctuation (**<punctuation>**) is handled and transposed into Unicode characters. The output of OCR and HTR often needs to undergo manual or automatic correction to remove the remaining errors. Post-transcription corrections, whether done automatically, semi-automatically, or by hand, can be detailed in **<correction>** (Rigaud et al. 2019; Nguyen et al. 2019).[15]

```
<editorialDecl>
  <normalization>
    <p>In the training corpus, abbreviations were transcribed using
"^".</p>
  </normalization>
  <correction>
    <p>No post-transcription corrections.</p>
  </correction>
  <hyphenation eol="all">
    <p>All hyphenations were kept. They were transcribed with "-".</p>
  </hyphenation>
</editorialDecl>
```

Example 1. An example of **<editorialDecl>**. Created by the authors.

As for the representation of the raw transcription itself, we chose to keep a clear separation between it and the actual edited text. While the latter would go in the **<body>** element, all the raw output of the transcription is defined and contained in a **<sourceDoc>** element. As stated in the TEI P5 Guidelines, **<sourceDoc>** contains the

[15] See the TEI Guidelines (TEI Consortium 2022, Appendix C: Elements) for more documentation about these elements: **<normalization>**, https://tei-c.org/Vault/P5/4.5.0/doc/tei-p5-doc/en/html/ref-normalization.html; **<punctuation>,** https://tei-c.org/Vault/P5/4.5.0/doc/tei-p5-doc/en/html/ref-punctuation.html; and **<correction>,** https://tei-c.org/Vault/P5/4.5.0/doc/tei-p5-doc/en/html/ref-correction.html.

transcription or other representations of a single source document.[16] Several child elements are available: **<graphic>**, **<surface>**, and **<zone>**.

Our implementation of **<sourceDoc>** follows two key principles. First, **<sourceDoc>** must be the strict transposition of any output information resulting from the HTR or OCR process. For instance, since ALTO and PAGE XML exports include masks' coordinates, we need to reflect these in the TEI representation. Secondly, we want to keep **<body>** free from any HTR or OCR information, meaning that what we find there will be the sole responsibility of the editor. As a result, any interpretation of the output of the transcription will be contained in **<body>** and never inside **<sourceDoc>**. Distinguishing the raw transcription from the edition, understood as a stable transcription with information on its layout, guarantees a continuum between the original document and its final publication. We deliberately decided to double the textual content in order to have two distinct blocks: the initial output of the transcription, and its edition (see fig. 4).

In conjunction with our specification work to map the OCR/HTR output to TEI XML, we have implemented a full-fledged XSLT transform from PAGE XML to TEI which generates the content of **<sourceDoc>** as presented here.[17] Note that a similar transformation scenario could be created to go from ALTO to TEI. We managed to ensure that all elements available in a PAGE XML document could be retrieved during the transformation and transposed into the resulting TEI document.

In a PAGE XML document, a **<Page>** element represents the transcribed image. Basic metadata can be assigned to various attributes: for instance, **@imageFilename** will store the name given to the source image, and **@imageWidth** and **@imageHeight** give a set of *x* and *y* values in pixels, defining a two-dimensional space bearing text lines. With TEI, and nested inside **<sourceDoc>**, we use **<graphic>** for documenting such information,

[16] See TEI Consortium 2022, Appendix C: Elements, **<sourceDoc>**: https://tei-c.org/Vault/P5/4.5.0/doc/tei-p5-doc/en/html/ref-sourceDoc.html.

[17] The XSLT stylesheet is available in Alix Chagué and Hugo Scheithauer, "page2tei: An XSL Transformation to Transform PAGE XML into TEI XML" (GitHub repository), accessed March 24, 2022, https://github.com/TEI4HTR/page2tei.

with the **@url**, **@width,** and **@height** attributes. In addition, we give the image an identifier with **@xml:id** for linking it to its transcription later, allowing one to go back and forth between the raw state of the text generated via HTR and the edited version.

We then use **<surfaceGrp>** to represent the sum of all text regions and their associated text lines for a given image. The elements **<surfaceGrp>** and **<surface>** are linked using an **@facs** attribute on the former element, and **@xml:id** on the latter. For each text region, a **<surface>** element is created and nested within this parent element.

PAGE XML uses **<textRegion>** and **<Coords>** elements to document text regions. The former gives an identifier, and the latter gives its coordinates as pairs of points, which can be located on the image. In TEI, we distribute this information with three attributes inside **<surface>**: **@xml:id** gives an identifier; **@type** will be associated with values defined with an ontology in the transcription software to qualify the different text regions (for example "numbering", "title", and "line_group", as seen in our example in fig. 1). Finally, **@points** indicates the coordinates of the text region. This last attribute will be found in any other element bearing layout information and defining an area or a line.

Nested in **<surface>**, each text line associated with a text region is represented with **<zone>** and its child elements **<path>** and **<line>**. The **<zone>** element corresponds to the mask of the text line. Then, **<path>** and **<line>** are respectively equivalent to the baseline/topline and the text node (see fig. 6).

The **<body>** can then be encoded based on the content of the transcription, text lines, and ideally layout information, stored in the **<sourceDoc>**, as found in ALTO XML or PAGE XML.

Fully encoded TEI examples can be found on GitHub.[18]

---

[18] Two fully encoded TEI examples can be found in Chagué and Scheithauer, “page2tei: An XSL Transformation to Transform PAGE XML into TEI XML” (GitHub repository), accessed March 29, 2022: https://github.com/TEI4HTR/page2tei/blob/main/ressources/FRAN_0025_0227_L-0.tei.xml and https://github.com/TEI4HTR/page2tei/blob/main/tei/32_c42c1_default.tei.xml. The first file

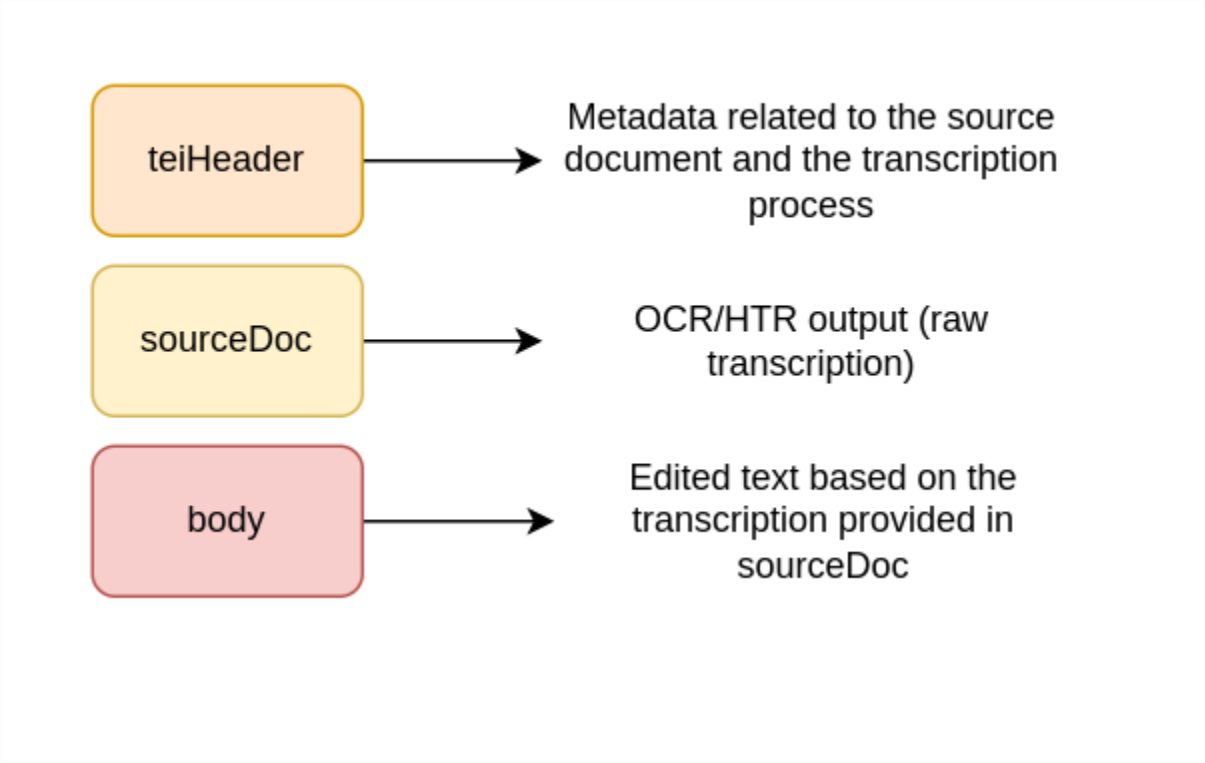

Figure 4. Components of **<sourceDoc>**-based encoding. Created by the authors.

```
<sourceDoc>
<graphic xml:id="f33" url="ark:/12148/btv1b525056707/f33/" width="2312px"
height="3469px"/>
  <surfaceGrp facs="#f33">
  <!-- ... -->
     <surface xml:id="eSc_textblock_2ed880a5" type="title" points="1594,196
1744,265 1668,330 827,351 793,236 868,173">
        <zone xml:id="eSc_line_08a9f120" type="mask" points="833,317
827,179 998,185 1024,170 1027,170 1027,170 1027,170 1030,170 1030,170
1033,170 1033,170 1033,170 1035,170 1090,187 1125,173 1125,173 1128,173
1128,173 1128,173 1131,173 1131,173 1131,173 1134,173 1134,173 1189,199
1206,205 1218,199 1270,179 1270,179 1273,179 1273,179 1273,179 1276,179
1276,179 1278,179 1278,179 1278,179 1281,179 1313,202 1371,202 1452,176
1455,176 1455,176 1455,176 1458,176 1458,176 1461,176 1461,176 1461,176
1464,176 1464,176 1513,216 1530,228 1692,228 1701,306 1701,320 833,332">
          <path type="baseline" points="833,317 1701,306"/>
```

results from an automatic transcription, and the second was manually transcribed.

```
            <line>Le Pont Mirabeau.</line>
          </zone>
        </surface>
    <!-- ... -->
    </surfaceGrp>
</sourceDoc>
```

Example 2. An example of the use of **<sourceDoc>**. Created by the authors.

The Example of LEPIDEMO: TEI as a Pivot File

As stated earlier, we do not envision **<sourceDoc>** as the place where the transcription is interpreted, nor where the creation of the edition takes place. However, the text provided in **<sourceDoc>** will often be the foundation for later editions and interpretations which will be rendered in **<body>**. The LECTAUREP project provided a use case to demonstrate how a TEI file built according to our modelization can usefully serve both the raw transcription and its edition.

LECTAUREP was a project jointly led by Inria (ALMAnaCH team) and the Archives nationales de France (DMC team) between 2018 and 2021 (Chagué and Rostaing 2021). It aimed at facilitating the exploration of directories listing minutes and deeds recorded by Parisian notaries between the beginning of the nineteenth century and the mid-twentieth century. Once the transcriptions were acquired with eScriptorium, we wanted to detect named entities before publishing the final enriched transcriptions. However, owing to the complex layout used in the documents, we realized that it would be difficult to automatically parse and analyze hundreds of pages without structuring the output of the transcription. Such structuration would help us target specific text regions where named entities could be found. As illustrated in figure 5, each page contains a table consisting of a header and several columns (usually seven) dividing the information into different categories. With a fully encoded table in TEI, it would theoretically be possible to choose a specific column and then tag elements according to the column's nature. For

example, we could tag every type of act in the third and fourth column with simple rules because the information is rather simple, consisting only of tokens specifically describing the type of act. In our use case, named entity recognition (NER) would then be performed semi-automatically, without the need to train a model for every column, and automatically with an NER model when information is syntactically more complex (e.g., the fifth column).

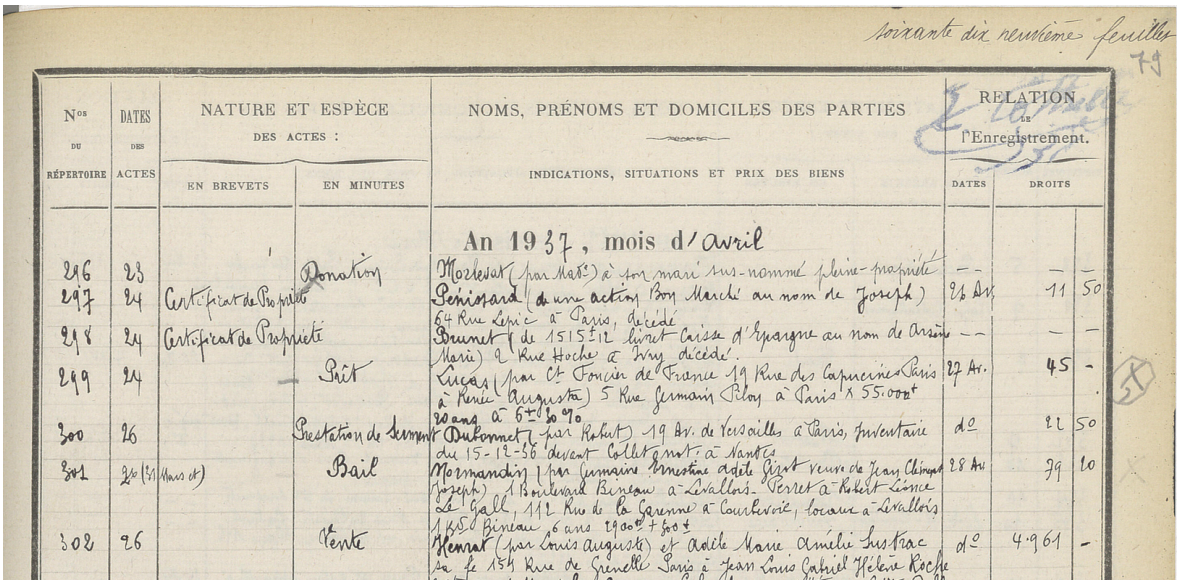

Figure 5. A notary directory dated April 1937 (inventory number: FRAN_0025_4648_L-1). From left to right, the header indicates: repertory number; date; the type of deed; the persons who signed it and a summary; when it was paid; and the amount paid to the notary.

We built a demonstration pipeline called LEPIDEMO (for LECTAUREP Pipeline Demonstration; see fig. 6) during which images are first loaded into eScriptorium, where the layout is annotated and the documents transcribed, before being exported as PAGE

XML files.[19] The PAGE XML files were converted into TEI with an XSLT stylesheet.[20] All these files could then be aggregated into a single file representing the whole document. As mentioned before, the extracted information populated **<teiHeader>** and **<sourceDoc>** only. Then, we processed coordinates available in **<sourceDoc>** using a Python script in order to reconstruct each page's logical structure. This helped us create the content of **<body>**: mainly a **<table>** element with child elements such as **<row>** and **<cell>** encoding its content. We also created relations between the **<sourceDoc>** and the **<body>**, linking the lines of texts by means of a pair of **@xml:id** attributes in the former, and a **@facs** attribute in the latter. Finally, we were able to normalize dates given that one line specifies the year and month applicable to the whole page while a column gives only the date of the day applicable to the minute (see fig. 7). We were also able to narrow down the scope of the search for named entities in the text. Corrections and the annotation of named entities would be appended to the content of **<body>**.

[19] See Alix Chagué and Hugo Scheithauer, *LEPIDEMO: A Pipeline Demonstrator for LECTAUREP to Go from eScriptorium to TEI-Publisher*, v. 1.0, 2021, accessed March 28, 2022, https://github.com/lectaurep/lepidemo, https://doi.org/10.5072/zenodo.977657. Restructured files can be found in the output folder, accessed March 29, 2022, https://github.com/lectaurep/lepidemo/tree/master/data/output.

[20] We also developed a python script that allows users to download TEI files from eScriptorium. See , accessed March 28, 2022, https://github.com/lectaurep/TEI-From-eScriptorium. We hope to implement a TEI export directly in eScriptorium in the future.

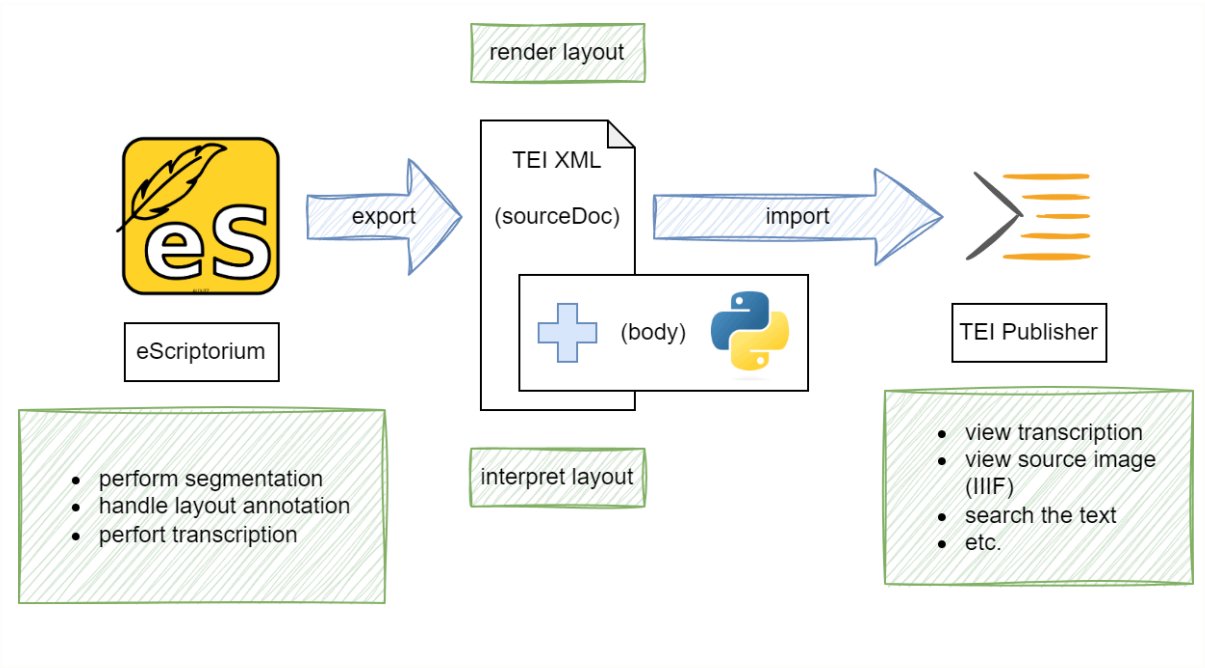

Figure 6. Simplification of the LEPIDEMO workflow. Created by the authors.

With the example of LEPIDEMO, we show the importance of distinguishing the raw output of OCR/HTR from the edition phase where the coordinates are interpreted and the text annotated. Without this interpretation of **<sourceDoc>**, we would not be able to make sense of the different resulting text lines. However, the content of **<body>** is irrelevant to OCR/HTR program because in order to become readable the information no longer follows the physical structure rendered on the image. Keeping both **<sourceDoc>** and **<body>** in the same document enables one to show a physical structure compatible with OCR/HTR program (to train a new transcription or segmentation model, for example) while still being able to move on to more detailed edition steps.

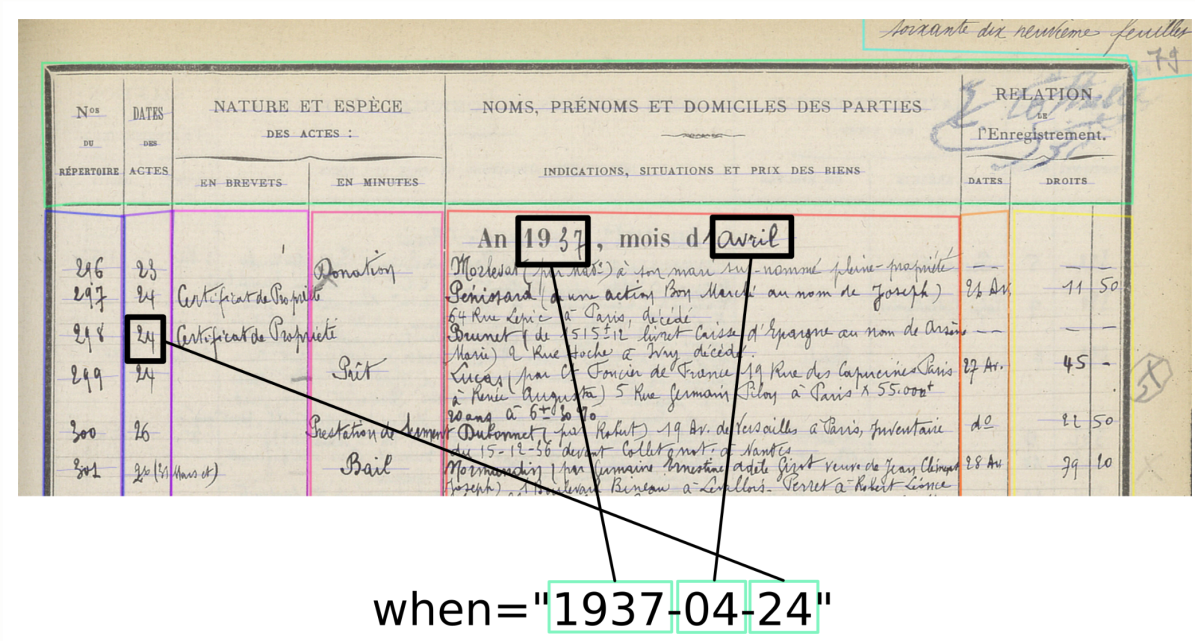

Figure 7. Building an ISO-compliant representation of the date based on the elements present on the page (excerpt from FRAN_0025_4648_L-1.jpg). Created by the authors.

Using Layout Information When Publishing TEI

Having access to layout information inside the final encoded text also allows for further usage during the publication phase.

With TEI Publisher (e-editiones, n.d.), we propose a solution to display the resulting TEI file where the transcription output is associated with an image facsimile. TEI Publisher is an open-source publication application powered by eXist-db. It offers a fully customizable interface, and uses ODDs and web templates to visualize a corpus of files encoded with TEI. Furthermore, it allows faceted search and corpora exploration.[21]

We used the modelization presented above to propose a visualization of the resulting TEI files that provides four side-by-side views, as illustrated in figure 8. The first view is a IIIF (International Image Interoperability Framework) viewer displaying a facsimile: a IIIF identifier referring to the image is stored with a **@url** attribute inside a **<graphic>**

[21] See the TEI Publisher website, accessed March 28, 2022, https://teipublisher.com/index.html.

element in **<sourceDoc>**. Second, a diplomatic representation of the document is based on the content of the **<body>** element. Third, a flat representation of the text provided in **<sourceDoc>** is displayed region by region, allowing users to see the original image's segmentation. Finally, an imitative transcription of the text is rendered using SVG, an XML-based vector image format for two-dimensional graphics. This transcription is based on the information contained in <sourceDoc>, including the size of the canvas as well as the segments' coordinates and their associated text nodes.[22]

Whole documents would then be displayed using an interface such as TEI Publisher with layout based on the information stored in the **<sourceDoc>**. However, as TEI files become bigger, hardware issues arise. Splitting such a TEI file on the **@xml:id** specified in **<graphic>** and nested in the **<sourceDoc>** could allow bypassing hardware limitations.

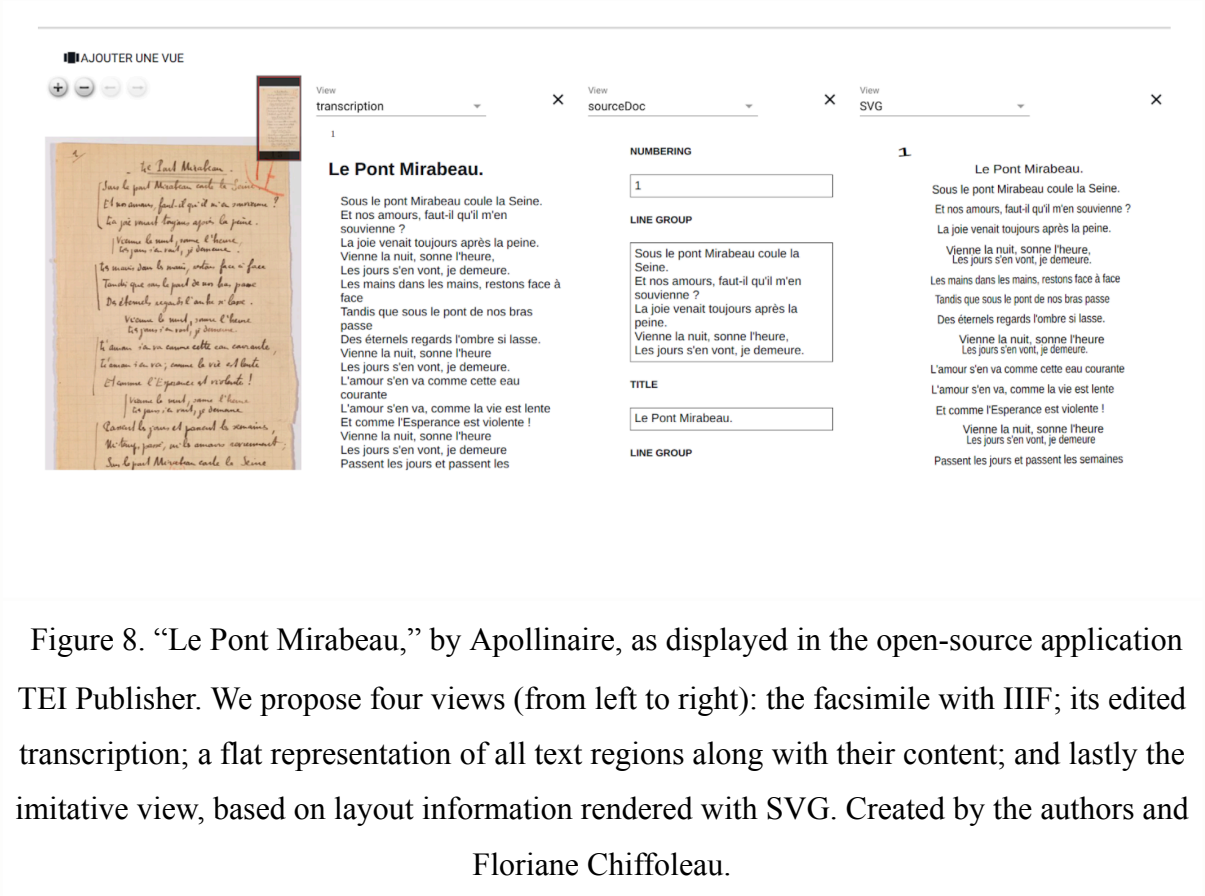

Figure 8. "Le Pont Mirabeau," by Apollinaire, as displayed in the open-source application TEI Publisher. We propose four views (from left to right): the facsimile with IIIF; its edited transcription; a flat representation of all text regions along with their content; and lastly the imitative view, based on layout information rendered with SVG. Created by the authors and Floriane Chiffoleau.

[22] See the Scalable Vector Graphics (SVG) 2 W3C Candidate Recommendation, October 4, 2018, accessed March 28, 2022, at https://www.w3.org/TR/SVG2/.

The SVG visualization provides a way to display a transcription alongside its original layout. This gives the community an opportunity to publish text with a complex layout, as seen in figure 8, where the meaning of the text is extremely tied to the layout.

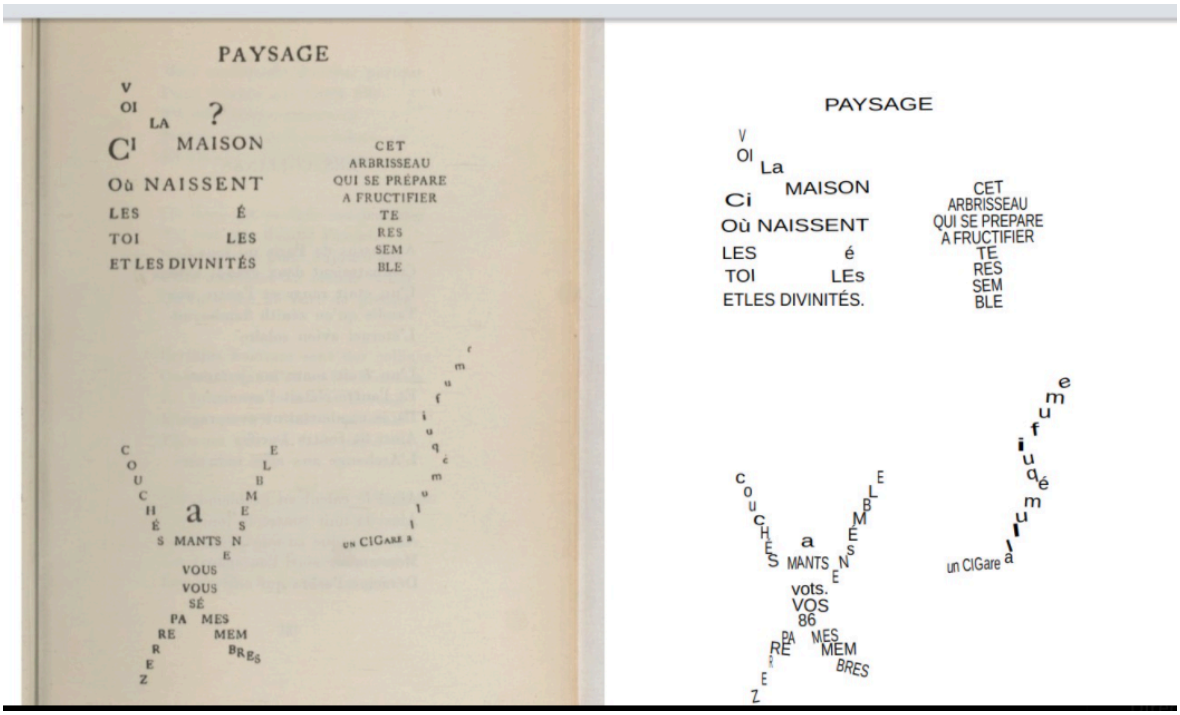

Figure 9. A visualization in TEI Publisher of a facsimile of calligrams written by Guillaume Apollinaire. Guillaume Apollinaire, *Calligrammes*, edited by Roger de La Fresnaye (Lausanne: n.p., 1952), https://gallica.bnf.fr/ark:/12148/bpt6k9775732c/f31. Created by the authors. Note: The transcription was acquired with an HTR model and was not corrected, hence some characters appearing as lowercase although they should be uppercase.

Conclusion

We have argued that all elements resulting from an OCR or an HTR process can be mapped to a **<sourceDoc>** element as defined by the TEI Guidelines. With a bipartite organization revolving around the articulation of **<sourceDoc>** and **<body>**, we can

leave the output of the transcription intact and render any interpretation or correction of the text (otherwise incompatible with transcription software) inside **<body>**, with the appropriate elements. By adopting this architecture, we propose that digitization pipelines switch earlier to TEI and move away from relying on ALTO or PAGE as an archival format, with the objective of simplifying the workflow and better documenting the transcription phase and its later edition, either for keeping track of this process, or for further usage, such as described in our examples. This is particularly crucial for archival purposes, as it allows reliance on a single well-maintained format. OCR and HTR are both still evolving, with HTR doing so even more rapidly, in terms of formats as much as software. Our modelization is intended as a first step toward a better stabilization of the landscape using TEI. New use cases and approaches will certainly arise; we therefore welcome feedback and are open to collaborations.

Acknowledgments

We would like to mention the essential contribution of our colleague Floriane Chiffoleau on the elaboration of the visualization prototype with TEI Publisher. We also wish to thank Simon Gabay, Juliette Janes, Claire Jahan, and Ariane Pinche for their insightful feedback.

Bibliography

Burnard, Lou, Christof Schöch, and Carolin Odebrecht. 2021. “In search of Comity: TEI for Distant Reading.” *Journal of the Text Encoding Initiative* 14. doi:10.4000/jtei.3500.

Clérice, Thibault. 2023. “You Actually Look Twice At It (YALTAi): Using an Object Detection Approach Instead of Region Segmentation within the Kraken Engine.” In “Historical Documents and Automatic Text Recognition,” special issue, *Journal of Data Mining and Digital Humanities*. Published December 26, 2023. https://doi.org/10.46298/jdmdh.9806.

Chagué, Alix, and Aurélia Rostaing. 2021. “LECTAUREP: Paris Notary Record Books Automated Reading.” Paper presented at the conference Fantastic Futures 2021 / Futures Fantastiques, Paris, France, December 8–10, 2021. https://hal.inria.fr/hal-03479258.

e-editiones. n.d. “TEI Publisher: Documentation.” N.p.: TEI Publisher. Accessed January 18, 2024. https://teipublisher.com/exist/apps/tei-publisher/doc/documentation.xml?odd=docbook.odd.

Gabay, Simon, Jean-Baptiste Camps, Ariane Pinche, and Claire Jahan. 2021. “SegmOnto: Common Vocabulary and Practices for Analysing the Layout of Manuscripts (and More).” Paper presented at the 1st International Workshop on Computational Paleography (IWCP@ICDAR 2021), Lausanne, Switzerland, September 7, 2021. https://hal.archives-ouvertes.fr/hal-03336528.

Khemakhem, Mohamed. 2020. “Standard-Based Lexical Models for Automatically Structured Dictionaries.” PhD diss., Université Paris Cité. https://theses.hal.science/tel-03274454v2.

Massot, Marie-Laure, Arianna Sforzini, and Vincent Ventresque. 2019. “Transcribing Foucault’s Handwriting with Transkribus.” In “Digit_Hum Workshop” (Atelier Digit_Hum), special issue, *Journal of Data Mining and Digital Humanities*. Published March 5, 2019. https://doi.org/10.46298/jdmdh.5043.

Mühlberger, Guenter, Louise Seaward, Melissa Terras, Sofia Ares Oliveira, Vicente Bosch, Maximilian Bryan, Sebastian Colutto, et al. 2019. “Transforming Scholarship in the Archives through Handwritten Text Recognition: Transkribus as a Case Study.” *Journal of Documentation* 75, no. 5: 954–76. https://doi.org/10.1108/JD-07-2018-0114.

Nguyen, Thi-Tuyet-Hai, Adam Jatowt, Mickaël Coustaty, Nhu-Van Nguyen, and Antoine Doucet. 2019. “Deep Statistical Analysis of OCR Errors for Effective Post-OCR Processing.”

In *Proceedings: 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, edited by Maria Bonn, Dan Wu, Stephen J. Downie, and Alain Martaus, 29–38. Los Alamitos, CA: IEEE Computer Society. https://doi.org/10.1109/jcdl.2019.00015.

Oliveira, Sofia Ares, Benoit Seguin, and Frederic Kaplan. 2018. "dhSegment: A Generic Deep-Learning Approach for Document Segmentation." In *Proceedings: 2018 16th International Conference on Frontiers in Handwriting Recognition; ICFHR 2018*, 7–12. Los Alamitos, CA: IEEE Computer Society. doi:10.1109/ICFHR-2018.2018.00011.

Pletschacher, Stefan, and Apostolos Antonacopoulos. 2010. "The PAGE (Page Analysis and Ground-Truth Elements) Format Framework." In *Proceedings: 2010 20th International Conference on Pattern Recognition; ICPR 2010*, 257–60. Los Alamitos, CA: IEEE Computer Society. https://doi.org/10.1109/ICPR.2010.72.

Rigaud, Christophe, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2019. "ICDAR 2019 Competition on Post-OCR Text Correction." In *Proceedings: 15th IAPR International Conference on Document Analysis and Recognition; ICDAR 2019*, 1588–93. Los Alamitos, CA: IEEE Computer Society. https://hal.archives-ouvertes.fr/hal-02304334; https://doi.org/10.1109/ICDAR.2019.00255.

Stokes, Peter A., Benjamin Kiessling, Daniel Stökl Ben Ezra, Robin Tissot, and El Hassane Gargem. 2021. "The EScriptorium VRE for Manuscript Cultures." In "Ancient Manuscripts and Virtual Research Environments," edited by Claire Clivaz and Garrick V. Allen, special issue, *Classics@* 18. https://classics-at.chs.harvard.edu/classics18-stokes-kiessling-stokl-ben-ezra-tissot-gargem/.

TEI Consortium. 2022. TEI P5: Guidelines for Electronic Text Encoding and Interchange. Version 4.5.0. Last updated October 25. N.p.: TEI Consortium.

https://tei-c.org/Vault/P5/4.5.0/doc/tei-p5-doc/en/html/.

Biographies

Hugo Scheithauer, Research & Development Engineer, ALMAnaCH (Inria), Paris:

hugo.scheithauer@inria.fr.

PhD student in the Inria ALMAnaCH team, **Hugo Scheithauer** holds a master's degree in art history and in *technologies numériques appliquées à l'histoire* (digital technologies applied to history) at the École nationale des chartes. He works on document layout analysis, automatic text recognition, and information extraction for historical documents.

Alix Chagué, PhD Candidate in Digital Humanities, ALMAnaCH (Inria), Paris, and CRIHN (Université de Montréal), Montreal: alix.chague@inria.fr.

Alix Chagué is a PhD candidate in digital humanities, focusing on the resources and methods for applying HTR technologies in the various fields of the humanities. As a co-founder of the HTR-United initiative (https:/htr-united.github.io), she advocates for the construction by the community of homogenous practices within an open science framework. She was also a coordinator for the master's degree program in *documentation et humanités numériques* (documentation and digital humanities) at the Ecole du Louvre in Paris.

Laurent Romary, Director for Scientific Information and Culture, Inria, Paris:

laurent.romary@inria.fr

Laurent Romary is a research director and the director for scientific information and culture at Inria, Paris. He has conducted multiple research projects in computer science, linguistics, and digital humanities and developed a strong interest in standardization and open data. He also identified the need to implement shared infrastructures for open science.