From page to content

Which TEI representation for HTR output?

TEI Conference and Members' Meeting 2021 - 10/26/21

"Next-Gen TEI"

Hugo Scheithauer (Inria, École des chartes), Alix Chagué (Inria), Simon Gabay (Université de Genève), Laurent Romary (Inria), Juliette Janes (École des chartes), Claire Jahan (École des chartes)
Overview - from source documents to scholarly editions

Context: heterogeneous formats at the various stages of a digitisation workflow

- ALTO/PAGE XML vs. TEI

Objectives:

- Improve reusability of content at various stages
- Hence, easier combination of different software solutions

Results:

- Propagation of metadata elements
- Mapping of layout, transcription information
- Architecture for integrating the content in further editions
Documenting a transcription: metadata representation with TEI

TEI

(METS XML)

PAGE XML

<Metadata>
  <Creator>escriptorium</Creator>
  <Created>2021-10-07T07:46:39.64183+00:00</Created>
  <LastChange>2021-10-07T07:46:39.864229+00:00</LastChange>
</Metadata>

<fileDesc>
  <titleStmt>
    <Title>FRAN_0025_3056_L-0</Title>
    <respStmt>
      <resp>Transcribed with</resp>
      <name>escriptorium</name>
    </respStmt>
  </titleStmt>
  <publicationStmt>
    <p/>
  </publicationStmt>
  <sourceDesc>
    <p/>
  </sourceDesc>
</fileDesc>

<change when="2021-10-07T07:46:39.64183+00:00">Creation</change>
<change when="2021-10-07T07:46:39.864229+00:00">Last change</change>
Any metadata missing for documenting an automatic transcription?

→ The transcription model
→ Documenting automatic and manual post-processing, such as correction
→ Information regarding how the transcription was produced
Which TEI representation for the transcription itself?
What does the TEI have to offer for the representation of data resulting from HTR/OCR?

Beyond facsimiles, the `<sourceDoc>` element:

How are we using the <sourceDoc>?

Two key principles:

→ The <sourceDoc> must be the strict transposition of all automatic transcription output elements.

→ All elements in <body> are the user’s responsibility and contain their interpretation/edition of the transcription.
A few words about automatic text recognition terminology
A page layout can be segmented into one or several **text regions**, each with their own coordinates:

<table>
<thead>
<tr>
<th>No.</th>
<th>Noms. Prénoms et Domiciles des Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>2159</td>
<td>Jeanne, 25 rue de Montmartre, PARIS 18</td>
</tr>
<tr>
<td>2158</td>
<td>Marie, 47 avenue de la Bourdonnais, PARIS 14</td>
</tr>
<tr>
<td>2157</td>
<td>Louis, 12 boulevard Haussmann, PARIS 9</td>
</tr>
</tbody>
</table>

Text lines can be nested in a text region.
A line of text combines:

- A baseline (pink) or a topline: a line defined by at least 2 points
- A mask (blue): a polygon defined by at least 3 points
- A text node
Representation of a page with TEI using the `<sourceDoc>` element
The PAGE XML `<Page>` element, with basic metadata, for instance:

```xml
<Page imageFilename="FRAN_0025_3056_L-0.jpg" imageWidth="2094" imageHeight="4393"/>
```

becomes a `<sourceDoc>` element in TEI:

```xml
<sourceDoc>
  <graphic url="FRAN_0025_3056_L-0.jpg" width="2894px" height="4393px"/>
</sourceDoc>
```
PAGE XML and TEI structure of an image content:
text regions and baselines

```xml
<TextRegion id="eSc_textblock_aebbab00" custom="structure {type:col_1;}">
  <Coords points="421,615 421,2236 465,2211 465,2266 421,2269 425,2449 419,4148"

  <TextLine id="eSc_line_86b00a8e" >
    <Coords points="285,838 293,812 322,790 380,801 377,863 289,874"/>
    <Baseline points="289,841 389,845"/>
    <TextEquiv>
      <Unicode>198</Unicode>
    </TextEquiv>
  </TextLine>
  ...
</TextRegion>

<surface xml:id="eSc_textblock_aebbab00"
  type="structure_{type:col_1;}
  points="421,615 421,2236 465,2211 465,2266 421,2269 425,2449"

<zone xml:id="eSc_line_86b00a8e"
  type="mask"
  points="285,838 293,812 322,790 380,801 377,863 289,874">
  <path type="baseline" points="289,841 389,845"/>
  <line>198</line>
</zone>
...
```
All TEI elements together:
Linking interpreted content in the <text> elements with the various components available in <sourceDoc>

- processing coords and tree: create table + row + cell
- processing face/@ids: create relations between body and sourceDocs
- processing text nodes: adding date and @when-ISO, adding persName etc, correcting text

Simplification of the LEPIDEMO workflow
Keeping the link with IIIF image servers?
Conclusion

- All elements from PAGE XML and ALTO files can be mapped to a `<sourceDoc>` element
- Switching to TEI earlier in the pipeline simplifies the workflow
- We present a proof of concept and a series of rather stable specifications and call for feedback from the community

The next steps are:

- Testing an implementation in a software like eScriptorium (under scrutiny)
- Adapting tools like TEI Publisher so they can readily display sourceDoc contents