Using MS Word as TEI-Editor? Experience with Two Letter-Editions

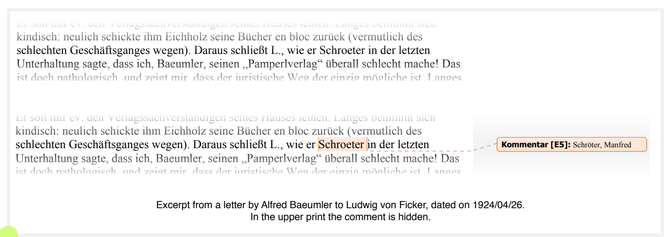
A presentation at the workshop "Datenmodellierung in digitalen Briefeditionen und ihre interpretatorische Leistung. Ontologien, Textgenetik und Visualisierungsstrategien", Berlin, May 2014, by Joseph Wang, University Innsbruck

Converting DOCX to TEI: How to (Ab-)Use MS-Word for Encoding Letters in TEI

Abstract: TEI without knowing XML

Encoding letters with the TEI vocabulary[1] is a very good way for producing machine-readable data in humanities. But the writing XML codes is very expensive: The learning curve is steep, and especially people who are already familiar with modern word processors like MS-Word do not want to leave their familiar area. On the other side, the benefits using TEI to create an electronic edition are self-explaining.

In two projects "The family correspondence of Ferdinand I." (Commission for Modern Austrian History / University Salzburg) and "Ludwig von Ficker as Cultural Agent" (Research Institute Brenner-Archives / University Innsbruck) a method is being tested. By using the comment function of MS-Word editors can also annotate words, and by a subsequent conversion of the DOCX-Files to XML/TEI these annotations are converted to TEI elements encoding semantic information.



Excerpt from a letter by Alfred Baeumler to Ludwig von Ficker, dated on 1924/04/26.
In the upper print the comment is hidden.

Prerequisites

In order to be able to apply the DOCX to TEI conversion, following prerequisites must be met:

A. The files must be saved in docx-format.

B. Some metadata must be put in places which are easy to process. E.g. information on sender, recipient and dating must always be written in the same manner, so the metadata can be extracted with pattern matching algorithms.

C. The comments should be written following some rule to enable automatic processing. E.g.: There should be a list of normalized names for <persName>, and another list for <placeName>. Or every comment on person (which should be transformed into <persName>) should start with "P."; and the name of a place should start with "O.".

D. Comments should not overlap.

E. In order to have a clean TEI markup, the word file itself should also be clean. Having lots of styling templates does not help, especially mixing paragraph styles with character styles can often create TEI output with wrong @rend attributes.

Why MS-Word?

Materials are already in DOC or DOCX-format.
For both Ferdinand and Ficker projects the transcription of the letters has begun long before people have thought about digital humanities and TEI. Having a printed book as result in their mind project staff have transcribed the text in MS-Word, and the WYSIWYG feature is a very good argument for using word processors. OpenOffice and its forks would be good alternatives, though.
Another reason is that the technician is familiar with MS-Word, especially with .docx-format, already.

Writing XML codes has a steep learning curve.
When confronted with XML codes, project staff, especially when they have been working with MS-Word for a long time, are unwilling to change their tools. Albeit all agree on the necessity of encoding texts in a markup language, the learning curve is very steep and people are not comfortable with XML editors such as Oxygen.

The comment function of MS-Word is very useful.
Since one can understand "tagging" as annotating a text, the commenting function of MS-Word is very useful. Project staff already know this function well. They can switch views to show or hide the comments according to their needs (e.g. when auditing transcriptions, the comments are hidden to ease the work).
Each comment in MS-Word has a starting and an ending marker, they have similar function as XML-tags.

Customizing XSLT-Stylesheet

The TEI Consortium provides us a stylesheet capabale of converting DOCX-files to TEI, written by Sebastian Rahtz.[2] In order to convert the comments into <rs> (and subsequently to e.g. <persName> and <placeName>) one needs to modify the default stylesheet.

1. The elements <w:commentRangeStart/> and <w:commentRangeEnd/> need to be transformed into <anchor/>s.
2. The <anchor/>s need to be converted to <rs>s with @n pointing to the corresponding <note>s.
3. Based on rules provided by [Prerequisite B] <teiHeader> and <facsimile> are populated with metadata.
4. Based on rules provided by [Prerequisite C] some of the <rs>s are transformed to <persName> and <placeName>, or any other elements.

Main templates for converting <anchor/> marked comments to <rs>s

```
<xsl:template match="text()|*" mode="elimateCommentStartEnd">
  <xsl:variable name="prevUlbkComment" select="preceding-sibling::tei:anchor[1]"/>
  <xsl:variable name="nextUlbkComment" select="following-sibling::tei:anchor[1]"/>
  <xsl:choose>
    <xsl:when test="$prevUlbkComment/@commentId =
                    $nextUlbkComment/@commentId">
      <!-- DELETED -->
    </xsl:when>
    <xsl:otherwise>
      <xsl:copy>
        <xsl:apply-templates select="@*" mode="elimateCommentStartEnd"/>
        <xsl:apply-templates select="text()|*" mode="elimateCommentStartEnd"/>
      </xsl:copy>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match="tei:anchor[@type='end']"
    mode="elimateCommentStartEnd">
  <xsl:variable name="commentId" select="@commentId"/>
  <xsl:variable name="prevUlbkComment"
    select="preceding-sibling::tei:anchor[1]"/>
  <xsl:variable name="canConvert">
    <xsl:if test="$prevUlbkComment/@type='start' and
      $prevUlbkComment/@commentId=$commentId">yes</xsl:if>
  </xsl:variable>
  <xsl:choose>
    <xsl:when test="$canConvert='yes'">
      <!-- Do nothing, this will be processed with another template. -->
    </xsl:when>
    <xsl:otherwise>
      <xsl:copy-of select="*"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

Convert "Comment Milestones" to <anchor/>

```
<xsl:template match="w:commentRangeStart">
  <xsl:element name="anchor">
    <xsl:attribute name="n" select="@w:id"/>
    <xsl:attribute name="type">start</xsl:attribute>
    <xsl:attribute name="subtype">rs</xsl:attribute>
  </xsl:element>
</xsl:template>

<xsl:template match="w:commentRangeEnd">
  <xsl:element name="anchor">
    <xsl:attribute name="n" select="@w:id"/>
    <xsl:attribute name="type">end</xsl:attribute>
    <xsl:attribute name="subtype">rs</xsl:attribute>
  </xsl:element>
</xsl:template>
```


Excerpt from TEI-file. The function "format and indent" of Oxygen has been applied.

Little Helper: Copy from X to Y

Please note that three parameters need to be passed on: the from node, the to node, the parent node.

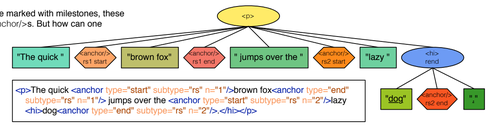
```
<xsl:template name="copyWithin">
  <xsl:param name="parentNode"/>
  <xsl:param name="fromNode"/>
  <xsl:param name="toNode"/>
  <xsl:variable name="startPos"
    select="count($fromNode/preceding-sibling::node())"/>
  <xsl:variable name="endPos"
    select="count($toNode/preceding-sibling::node())+1"/>
  <xsl:choose>
    <xsl:when test="$fromNode/parent::*!=$parentNode) or
      ($toNode/parent::*!=$parentNode)">
      <!-- Cannot copy, nodes do not have the same parent. -->
    </xsl:when>
    <xsl:when test="$startPos &gt; $endPos">
      <!-- Cannot copy, order of nodes is wrong. -->
    </xsl:when>
    <xsl:otherwise>
      <xsl:for-each select="$parentNode/node()">
        <xsl:if test="position() &gt; $startPos
          and position() &lt; $endPos">
          <xsl:copy>
            <xsl:apply-templates select="@*" mode="copy"/>
            <xsl:apply-templates mode="copy"/>
          </xsl:copy>
        </xsl:if>
      </xsl:for-each>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template match="@*|*|text()|processing-instruction()|comment()" mode="copy">
  <xsl:copy>
    <xsl:apply-templates select="@*" mode="copy"/>
    <xsl:apply-templates mode="copy"/>
  </xsl:copy>
</xsl:template>
```

Problem:

Comments in MS Word are marked with milestones, these can be transformed to <anchor/>s. But how can one transform <anchor/>s to <rs>s, especially if there are other overlapping elements, e.g. <hi>s which should be doubled in order to keep the xml wellformed?

Convert <anchor/>s to <rs>s



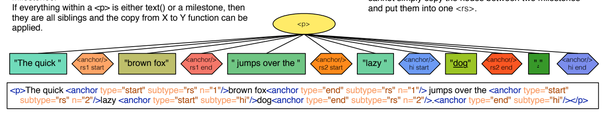
The quick <anchor type="start" subtype="rs" n="1"/>brown fox<anchor type="end" subtype="rs" n="1"/> jumps over the <anchor type="start" subtype="rs" n="2"/>lazy <hi>dog<anchor type="end" subtype="rs" n="2"/></hi>

Start: Milestones (<anchor/>) mark the boundaries of <rs/>s

After transforming the initial docx with the customized Docx2Tei-Stylesheet the boundaries of a comment is marked using milestones, while the highlighted passages are marked up with <hi>s.
As one can see, not all comment-milestones have the same parent, i.e. they are not siblings. And therefore one cannot simply copy the nodes between two milestones and put them into one <rs>.

Step 1: convert all elements to milestones and texts.

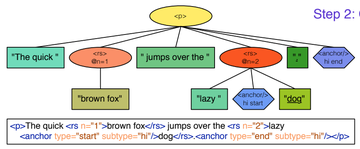
The first step is to make everything marked up using milestones.
If everything within a is either text() or a milestone, then they are all siblings and the copy from X to Y function can be applied.



The quick <anchor type="start" subtype="rs" n="1"/>brown fox<anchor type="end" subtype="rs" n="1"/> jumps over the <anchor type="start" subtype="rs" n="2"/>lazy <anchor type="start" subtype="hi"/>dog<anchor type="end" subtype="rs" n="2"/>.<anchor type="end" subtype="hi"/>

Step 2: Convert the comment milestones to <rs>

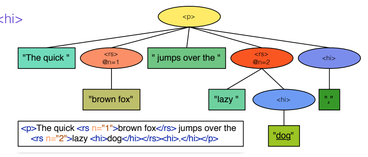
Since all nodes are now siblings within the we can safely transform the nodes like this:
A. Copy a node if it is not between the beginning and the end comment milestone.
B. With the starting milestone: make a <rs> and copy the whole siblings between the beginning and the ending comment milestone as children of <rs>.
C. Otherwise just do nothing.



The quick <rs n="1">brown fox</rs> jumps over the <rs n="2">lazy
<anchor type="start" subtype="hi"/>dog</rs>.<anchor type="end" subtype="hi"/>

Step 3: convert the milestones back to <hi>

Now copy the result of step 2, but test for each text() whether it is between two milestones marking up the boundaries of a <hi> or not. If it is, make a <hi> and copy the text into it, if it is not, just copy it.
By doing so we have doubled the <hi>s to the child- and "grandchild-"axes of .



The quick <rs n="1">brown fox</rs> jumps over the <rs n="2">lazy <hi>dog</hi></rs><hi>.</hi>

Project: Ferdinand I.

Ferdinand I as Emperor[5]

Editionproject: Family Correspondence of Ferdinands I.

Project leader Univ. Prof. Dr. Christopher F. Laferl, christopher.laferl@sbg.ac.at
Project institution Fachbereich Romanistik, University Salzburg, Erzabt-Klotz-Str. 1, AT - 5020 Salzburg
Project staff Bernadette Hofinger, Harald Kufner, Judith Moser-Kroiss, Nicola Tschugmell
Project website http://www.uni-salzburg.at/index.php?id=62915

Ferdinand I.

*1503 in Alcalá de Henares, †1564 in Vienna.

Ferdinand, son of Joanna of Castile ("The Mad") and Philip I of Castile ("The Handsome"), was Archduke from Austria (since 1521), King of Bohemia, Croatia and Hungary (since 1526/27), King of the Romans (since 1931) and Holy Roman Emperor (since 1558).

His older brother, Charles V. become Holy Roman Emperor in 1519. By request of Charles their younger sister, Mary of Hungary, ruled over the Netherlands since 1531.

Family letters

Although *prima facie* family letters do not seem to be of political relevance, due to the fact that the politicians at that time are all related to each other, the family correspondence of the Habsburg family becomes one of the most important sources for historical research of the 16th century. And, luckily, many letters are still preserved.

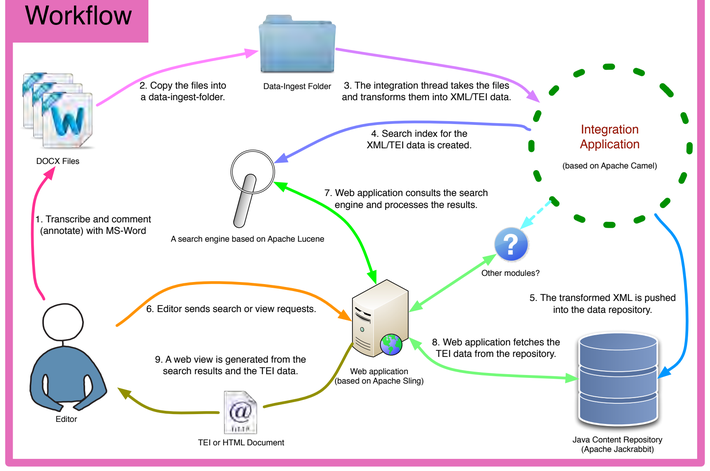
Most letters are exchanged between Ferdinand I., Charles V., and Mary. Most letters deal with political matter, but they also talk about health and finance.

An important aspect of the correspondence is the multitude of languages. French, German, Spanish and Latin are used. Most letters are written in French, the mother tongue of Charles. The choice of language depends seemingly on both the subject and the language skill of their secretaries. E.g. when the siblings talk about Germany, they use German, when they talk about Spain, they use Spanish.

History of Letter Editions

The family correspondence of Ferdinand I. has been in focus of historians for a very long time. At the end of the 19th century already, the newly founded Commission for Modern Austrian History has started a project of editing these materials. The first volume appeared 1912, with letters from 1514 to 1526. The next volume (1527-1530), consisted of two books, goes on print in 1937 (resp. 1938). We have to wait for another 35 years for the next volume to be published. The third volume (1531-32) consists of three separate book and appeared 1973/77 and 1984.
The latest volume appear 2000 and deals with letters from 1533 to 1535.[4]

Workflow

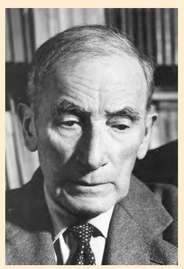


Pros and Cons

PRO	CON
Editors can stick to a WYSIWYG editor and a familiar word processor.	Project staff need more time to grasp the "text-modelling view" instead of the "text-printing view".
Since people already know the editor well, the transcription process is very efficient.	The efficiency of transcription is bought with the inability to encode elegantly.
With a control view editors have a quick feedback on the quality of the data conversion.	Some conversion errors (e.g. xml validation errors) are invisible in the control view.
Error of transcription can be easily detected and corrected.	If an error is detected, staff will still try to correct the DOCX file instead of the TEI-data.
This line is here solely to fill the space, so the pros and the cons are seemingly equal.	These projects rely heavily on xslt stylesheets; ane the stylesheets are difficult to maintain.
Project staff don't have the feeling of loosing control over the edition, albeit many processing steps are done automatically.	Instead of TEI guidelines staff come to learn special rules of commenting and annotating with MS Word.
Creating TEI data without knowing XML.	Creating TEI data without knowing XML.

Project: Ficker

Ludwig von Ficker



Photograph of Ludwig von Ficker's, ca. 1960

*1880 in Munich, †1967 in Innsbruck.
1910: Together with Carl Dallago he founded *Der Brenner,* a journal for art and culture, and the publishing house Brenner-Verlag.
1914: Ludwig Wittgenstein gave Ficker 100.000 Crowns and asked him to donate them to artists and writers in need.
1915-1918: Ficker was called to the First World War. After he came back, Brenner-Verlag was incorporated into the publishing house Verlag Wagner.
From 1926 Der Brenner starts to deal more and more with central subjects of the Catholic Church, several topics of the Second Vatican Council were already discussed here.
1940: The Reichsschrifttumskammer (one of the divisions within the Reich Chamber for Culture) banned *Der Brenner* as "corruptive and undesirable literature".
After the Second World War Ficker established contacts with several philosophers and writers, among others: Christine Busta, Ingeborg Bachmann, Theodor W. Adorno and Martin Heidegger.[3]

Ficker's Correspondence

As the head of a publishing house, Ficker has corresponded with more than 2000 persons and institutions. Today, more than 16.000 letters either from or to him are kept throughout the archives in the world. One can call his correspondence as his second major life achievement, as *Der Brenner* being the first.
These letters document a major part of the history of German literature.

Georg Trakl

Carl Dallago

Ludwig Wittgenstein

Else Lasker-Schüler

The Project *Ficker as Cultural Agent*

The project Ludwig von Ficker as Cultural Agent was founded by a grant of the Austrian Science Fund (FWF, P24283-G23) with its technical part financed by the Local Government of Vorarlberg. While many of the correspondence of Ficker's are transcribed since the 1970ies, the project also aims at scrutinizing the impact Ficker had on the German cultural life.

Project leader Ao. Univ.-Prof. Mag. Dr. Eberhard Sauermann, Eberhard.Sauermann@uibk.ac.at
Project institution Research Institute Brenner-Archives, University Innsbruck, Josef-Hirn-Str. 5-7, 10. Stock, AT - 6020 Innsbruck
Project staff Ingrid Führhapter, Markus Ender, David Franzoi, Joseph Wang with help of many others.
Project website http://www.uibk.ac.at/brenner-archiv/projekte/flickeralskulturvermittler/

References
[1] See: TEI Consortium, eds. Guidelines for Electronic Text Encoding and Interchange. Online resource: http://www.tei-c.org/P5/ (2014-04-29).
[2] The XSLT stylesheets of the TEI Consortium is maintained by Sebastian Rahtz and is accessible through their github repository: https://github.com/TEIC/Stylesheets (2014-04-29).
[3] See: Article "Ficker, Ludwig von" in Lexikon Literatur in Tirol. Online resource: http://orawww.uibk.ac.at/apex/uprod/f?p=tll:2:0::NO::P2_ID:147 (2014-04-29).
[4] See: Laferl, Christopher F.: "Die Familienkorrespondenz Ferdinands I.", online resource: http://www.uni-salzburg.at/fileadmin/multimedia/Romanistik/documents/Ferdinand_I_Familienkorrespondenz_Kurzbeschreibung_2013.pdf (2014-04-29)
[5] From: Dominicus Custos: Atrium Heroicum Caesarum, Pars I, 169. Picture taken from Wikipedia, online resource: http://commons.wikimedia.org/wiki/File:Ferdinand_I_HRR_MATEO.jpg (2014-04-29).

Joseph Wang
joseph.wang@uibk.ac.at
Institute for Historical Research and European Ethnology
University Innsbruck
Innrain 52d
AT-6020 Innsbruck