

Aufgabe 08: Kodieren von räumlich-prozessualen Aspekten

Christof Schöch

DARIAH-DE

Lizenz: Creative Commons Attribution 4.0 (CC-BY).

Das vorliegende Dokument wurde für DARIAH-DE neu erstellt. Das vorliegende Dokument beruht teilweise auf den folgenden Materialien: James Cummings, Renée Baalen, Ylva Berglund-Prytz: "An Introduction to XML and the Text Encoding Initiative", Digital.Humanities@Oxford Summer School 2012, University of Oxford, 2012, <http://digital.humanities.ox.ac.uk/dhoxss/2012/>, CC-BY-SA.

Kodieren von räumlich-prozessualen Aspekten

1. Lernziele

Nach dieser Übung sollten Sie folgendes gelernt haben:

- * Räumliche Aspekte von Manuskripten mit TEI kodieren
- * Den zeitlichen Verlauf des Schreibprozesses kodieren
- * Räumliche und textuelle Informationen zusammenführen

2. Überblick

In dieser Übung werden wir die räumliche Verteilung der Schrift auf einer Manuskript bzw. Tapuskript- Seite kodieren, diese Bereiche mit dem darin enthaltenen Text verbinden, und beides zusammenführen. Diese Übung verwendet die neuen Elemente aus dem Kapitel 11 "Transcription of Primary Sources", insbesondere `surface`, `zone` und `line`, sowie `change`. Für diese Übung benötigen wir Oxygen sowie den Bildeditor Inkscape.

3. Eine Bilddatei mit Inkscape importieren

Ein Dateiformat für Bilder, das mit TEI besonders gut harmoniert, ist SVG (Scalable Vector Graphics), denn es ist in XML kodiert! Die meisten Bilddateien liegen aber nicht als SVG vor, sodass wir zunächst die Datei importieren und transformieren müssen.

- * Öffnen Sie Inkscape.
- * Importieren Sie die Bilddatei `Durrenmatt.jpg` (im Ordner Übung 06): `File > Import`, Datei auswählen
- * Wählen Sie `link` als Import-Option
- * Setzen Sie die Dokumentgröße mit der Bildgröße gleich: `File > Document Properties`, dort klicken Sie auf `resize page to content` und dann auf den Button `Resize page to drawing or selection`; schließen Sie diesen Dialog.
- * Setzen Sie den Pfad zu absoluten Koordinaten: `File > Inkscape Preferences`, Bereich `SVG Output`: entfernen Sie das Häkchen bei `Allow relative coordinates`. Schließen Sie den Dialog.

4. Bereiche unterscheiden und markieren, Koordinaten notieren

Auf dem digitalen Faksimile sehen Sie eine Reihe von Dingen: einen mit der Schreibmaschine geschriebenen Text, einige handschriftliche Ergänzungen, eine (oder zwei?) Zeichnungen, und vielleicht noch andere Dinge. Wir möchten gerne vorschlagen, vier Bereiche zu unterscheiden:

- * 1. den Blocktext in Schreibmaschinenschrift mit Ergänzungen
- * 2. die handschriftlichen Ergänzungen oder Korrekturen am rechten Rand
- * 3. die handschriftlichen Zusätze unterhalb des Blocktextes
- * 4. Die Zeichnung (als eine Einheit) in der unteren Hälfte der Seite

Diese Bereiche sollten Sie nun mit Inkscape markieren und die Koordinaten in ein TEI-Dokument einfügen. Ziel ist es, in dem TEI-Dokument präzise auf die verschiedenen Bereiche der Bilddatei verweisen zu können.

- * Zoomen Sie etwas näher an das Bild heran, um genauer arbeiten zu können Menü: View > Zoom In
- * Wählen Sie am linken Rand das Werkzeug "Draw rectangle" aus (es ist ein... Rechteck- Symbol)
- * Gegebenenfalls müssen Sie die Einstellungen für die Füllung des Rechtecks ändern: Klicken Sie dafür am unteren Bildrand am linken Rand der Farbpalette auf das "Kreuz"-Symbol; dann wird das Rechteck nicht gefüllt und Sie können den Inhalt weiterhin sehen.
- * Zeichnen Sie ein Rechteck knapp um den maschinengeschriebenen Textblock.

5. Koordinaten in ein TEI-Dokument einfügen

Jetzt geht es darum, die Koordinaten dieses Textblocks in ein TEI-Dokument einzufügen. Öffnen Sie dafür die Datei `Durrenmatt-Start.xml` mit Oxygen. Wie Sie sehen, handelt es sich um ein TEI-Dokument, das neben einem Header mit einigen wenigen Informationen nur das Element `sourceDoc` beinhaltet. Wir werden nun diese `sourceDoc` verwenden, um die Informationen zum Bild und zu den Bereichen im Bild festzuhalten. In diesem Dokument ist die minimal notwendige Struktur eines `sourceDoc` bereits eingetragen, sodass das Dokument valide ist. Außerdem sind die Koordinaten der Bildfläche als Ganzes bereits eingetragen, als vier Attribute des Elements `surface`, die die linke obere und die rechte untere Ecke definieren.

- * `@ulx` definiert den X-Wert der oberen linken Ecke (0px horizontale Entfernung vom Bildrand)
- * `@uly` definiert den Y-Wert der oberen linken Ecke (0px vertikale Entfernung vom Bildrand)
- * `@lrx` definiert die Breite des Bildes (800px)
- * `@lry` definiert die Höhe des Bildes (1220px)

Außerdem hat das Element `surface` eine `@xml:id` mit dem Wert `surface-1`, damit wir es später eindeutig identifizieren können. Jetzt können wir die erste Zone

eintragen.

- * Fügen Sie nach dem Element `graphic` in einer neuen Zeile das Element `zone` ein.
- * Auch dieser ersten Zone sollten wir eine `@xml:id` geben, sagen wir `zone-1`.
- * Außerdem können wir nun dieser ersten Zone die Koordinaten geben, die wir in Inkscape feststellen können:
- * Gehen Sie zu Inkscape, und wechseln Sie in die XML-Editor Ansicht, im Menü: `File > XML Editor`: der unterste Eintrag in der Liste entspricht dem Rechteck (evtl. müssen Sie in der Baumstruktur links auf die kleinen Dreiecke klicken, um die Baumstruktur ganz zu öffnen).
- * Sie können dort die Koordinaten erkennen: `x=110, y=140, height=230, width=550`. Diese Koordinaten brauchen wir für unser TEI-Dokument.
- * Geben Sie also der `<zone>` im TEI-Dokument die Werte, die Sie in Inkscape sehen:
 - * Obere linke Ecke: `@ulx` (X-Wert eintragen) und `@uly` (Y-Wert eintragen)
 - * Untere rechte Ecke: `@lrx` (die Breite eintragen) und `@lry` (die Höhe eintragen)

Wollen Sie sicher gehen, dass Sie alles richtig gemacht haben? Fragen Sie ihren Nachbarn, oder schauen Sie ganz am Ende dieses Dokuments nach.

6. Transkription hinzufügen

Natürlich wollen wir auch dokumentieren, welcher Text zu dem bisher kodierten Bereich auf der Seite gehört. Auch das geschieht nun in einem eher topographischen als logischen Sinn, d.h. nicht als Absatz `p`, sondern als Zeilen, `lines`.

- * Klicken Sie in Oxygen auf "`Strg+Shift+P`", um eine Standard-Formatierung zu erhalten
- * Löschen Sie am Ende des `<zone/>` Elements den Schrägstrich `"/`; das leere Element wird wieder in Anfangs- und End-Tag getrennt. Fügen Sie zwischen die beiden `zone`-Tags eine `line` ein.
- * Öffnen Sie nun mit Oxygen die Datei `Durrenmatt-Text.txt`; sie enthält die Transkription des Texts, der sich auf unserer Bilddatei befindet. (Die Feinheiten der Kodierung, um die sich zweifelsohne zu kümmern wäre, lassen wir hier einmal beiseite.)
- * Kopieren Sie den Text, der der bisher definierten `zone` entspricht, zwischen die beiden `line`-Tags; klicken Sie auf "`Strg+Shift+P`".
- * Trennen Sie nun den Text jeweils in die topographischen Zeilen auf dem Faksimile. Gehen Sie dazu an die passende Stelle und Wählen Sie "`Alt+Shift+D`", sodass die vorhandene `line` dort in zwei `lines` getrennt wird. Wiederholen Sie dies für den gesamten Text.
- * Sie können an jeder Grenze zwischen zwei `lines` einmal `return` drücken, und dann noch einmal mit "`Strg+Shift+P`" formatieren, damit das Ganze

übersichtlicher wird.

7. Text und Bild zusammenführen

Für den ersten Bildbereich haben wir nun die Kodierung abgeschlossen und können das digitale Faksimile selbst, das definierte Rechteck, und den kodierten Text zusammenführen. Das machen wir mit einer Transformation, die auf XSLT beruht. Dabei werden die drei Komponenten in eine SVG- Datei, also eine Bilddatei, umgewandelt.

- * Richten Sie eine Transformationsroutine ein; im Menü `Document > Transformation > Configure Transformation Scenario`
- * Im Dialog wählen Sie `New`, wählen dann eine `XML Transformation with XSLT` aus
- * Im Reiter `XSLT` behalten Sie die `XML URL` bei; in der Zeile `XSL URL` klicken Sie auf das kleine Ordnersymbol rechts, und wählen dann die Datei `xml2svg.xsl` aus (im Ordner "Übung 06")
- * Wählen Sie im Dropdown-Menü für `Transformer` den Eintrag `Saxon EE 9.4.0.3` (evtl. auch eine andere Versionsnummer) aus. Bestätigen Sie das Ganze mit `OK`.
- * Klicken Sie nun im Dialogfenster ganz unten auf "Apply Associated", um die Transformation auszulösen. Fertig!

Im Ordner "Übung 06" finden Sie nun einen neuen Ordner, "out" (für "output"), dort wiederum einen Ordner "time". In diesem Ordner befindet sich die von der Transformation generierte SVG-Datei.

- * Öffnen Sie diese Datei mit dem Browser (am Besten: mit Safari oder Chrome), und kontrollieren Sie, ob das Faksimile mit der Zone und dem dazugehörigen Text angezeigt wird.

8. Weitere zone anlegen

Wir belassen das Dokument vorerst so, wie es ist, und üben das Anlegen der `zone` noch ein wenig! Es gibt ja noch mindestens drei weitere Bereiche in dem digitalen Faksimile, von denen wir zumindest einen noch kodieren sollten. Wiederholen Sie also die vorhergegangenen Schritte für einen weiteren Bereich, vielleicht den kleinen Bereich am rechten Rand.

- * Markieren Sie den Bereich in Inkscape (wie in Abschnitt 4 beschrieben)
- * Tragen Sie die Koordinaten in das TEI-Dokument ein (siehe Abschnitt 5)
- * Fügen Sie die Transkription des Bereichs ein (siehe Abschnitt 6)
- * Führen Sie erneut die Transformation durch (siehe Abschnitt 7).

9. Schon fertig? Weitere zones anlegen (optional)

Wer schon fertig ist, kann weitere `zones` nach dem gleichen Prinzip anlegen, und erneut die SVG- Datei generieren.

10. Die Schreibsequenz definieren

Die SVG-Datei, wie wir sie nun generiert haben, ist eine statische Datei, d.h. man kann das Rechteck und den Text nicht interaktiv ein- und ausblenden. Dafür hat Elena ein Javascript-Modul programmiert, das Sie uns ja am Mittwoch gezeigt hat, und das an dem Proust-Beispiel ja auch online verfügbar ist, unter der URL: http://research.cch.kcl.ac.uk/proust_prototype/index.html. Diese JavaScript-Erweiterung ist relativ komplex, und wir werden Sie in dieser Übung nicht anwenden können. Wir können aber zumindest eine (hypothetische) Schreibsequenz für die verschiedenen Bild-/Textbereiche definieren. Welche Schreibsequenz könnte man denn annehmen? Auf welche Informationen im Text könnte man sich stützen?

- * 1. Relativ sicher ist, dass in einem ersten Schritt die maschinengeschriebenen Teile des Textes vorhanden waren, d.h. der zentrale Textblock und der kleine Bereich mit der Seitenzahl. Welche der beiden Teile zuerst kam, lässt sich wohl nicht festlegen.
- * 2. Vermutlich sind dann die handschriftlichen Streichungen im Textblock und die Ergänzungen am rechten Rand und unterhalb des Textblocks gemacht worden. Ob dies eine Überarbeitungsphase war, oder ob zu unterscheiden ist zwischen den (früheren) Ergänzungen am rechten Rand und den (späteren?) Ergänzungen unterhalb des Textblocks, ist sicherlich schwer zu entscheiden.
- * 3. Man kann es nur annehmen, aber vermutlich sind die Zeichnungen in der unteren Bildhälfte anschließend angefertigt worden. (Als klar war, dass der Platz für keine weiteren Überarbeitungen oder Notizen benötigt wird.)

Um diese drei Schritte im Entstehungsprozess im TEI-Dokument festzuhalten, verwenden wir den Mechanismus, den das Element `change` bereitstellt.

- * Damit wir alle wieder auf dem gleichen Stand sind, öffnen Sie bitte das Dokument `Durrenmattchange.xml` mit Oxygen. Sie finden dort das TEI-Dokument, wie es am Ende des Schritts 9 aussehen sollte; zusätzlich sind dort einige der mikrostrukturellen Phänomene kodiert (mit `del` und `add`), und die Zählung der `zones` kann evtl. abweichen. Nehmen Sie sich ruhig einen Moment Zeit, um sich das alles anzuschauen.
- * Gehen Sie dann im `teiHeader` ans Ende der `fileDesc`. Nach der `fileDesc` fügen wir einen neuen Bereich ein: eine `profileDesc`, die der Dokumentation nicht-bibliographischer Aspekte eines Manuskripts dient. Wir nutzen Sie, um die vermutete Schreibsequenz zu dokumentieren.
- * In diese `profileDesc` sollten Sie das Element `listChange` einfügen. Das Element dient dafür, eine Liste aller Überarbeitungsschritte (Ergänzungen, Löschungen) im Schreibprozess, d.h. eine Liste jedes identifizierbaren `change` anzuführen.
- * Das Element `listChange` sollte zwei Attribute bekommen: `@ordered` mit Wert `true`, und `@n` mit Wert `writing`. Damit dokumentieren wir, dass wir von einer definierbaren zeitlichen Abfolge der verschiedenen Zonen bzw. Textpassagen ausgehen, und dass wir den Schreibprozess dokumentieren, nicht die Abfolge der Textpassagen bei der (idealisierten) Lektüre des Textes (das ist aber auch möglich!).

Jetzt sind wir soweit, dass wir die eigentlichen `changes` eintragen können.

- * Fügen nun Sie also im Element `listChange` das erste Element `change` ein. Es steht für den ersten Schritt in der Entstehung des nun auf dem Blatt vorhandenen Textes; geben wir ihm also für das Attribut `@xml:id` den Wert `stage-1`.

- * Sie sollten kurz beschreiben, was diese erste Schreibphase beinhaltet; schreiben Sie eine kurze Charakterisierung in den Inhalt des Elements: bspw. "Maschinengeschriebener Text liegt vor, einschließlich der Seitenzahl."

So sollte Ihr erster `change` aussehen: `<profileDesc> <creation> <listChange ordered="true" n="writing"> <change xml:id="stage-1">Maschinengeschriebener Text liegt vor, einschließlich der Seitenzahl.</change> </listChange> </creation> </profileDesc>` Fügen Sie nun zwei weitere `change`-Element ein, für die handschriftlichen Löschungen und Ergänzungen, sowie für die Zeichnungen.

- * Geben Sie jedem `<change>` wieder eine `@xml:id` ("stage-2" bzw. "stage-3") und eine kurze Beschreibung

- * Zweiter Schritt: "Streichungen im maschinengeschriebenen Text, Ergänzungen am rechten Rand, sowie nicht direkt verbundene Ergänzungen unterhalb des Textblocks.

- * Dritter Schritt: "Zeichnungen in der unteren Hälfte des Blatts; unklarer Zeitpunkt."

Jetzt müssen wir die verschiedenen Zonen noch mit diesen drei Phasen assoziieren:

- * Fügen Sie jeder der fünf `<zone>`-Elemente nach den vier räumlichen Attributen noch ein weiteres Attribut hinzu, nämlich `@change` mit dem Wert, der den gerade definierten `<change>`s entspricht.

- * "zone-1" und "zone-2" müssen als Wert für das Attribut `@change` "stage-1" bekommen

- * "zone-3" und "zone-4" bekommen "stage-2"

- * "zone-5" bekommt "stage-3"

Damit ist dokumentiert, in welcher Reihenfolge die topographisch definierten Bildbereiche mit dem Text oder den Zeichnungen, die mit ihnen assoziiert wurden, entstanden sind. Mit ein paar weiteren Tricks könnte Elenas Javascript daraus nun eine per Klick interaktive Grafik zaubern. Ende der Übung.

Anhang: Spoilers

```
<sourceDesc> am Ende von Teil 5 <sourceDoc> <surfaceGrp> <surface ulx="0" uly="0" lrx="800" lry="1220" xml:id="surface-1"> <graphic url="Durrenmatt.jpg"/> <zone xml:id="zone-1" ulx="120" uly="160" lrx="536" lry="230"/> </surface> </surfaceGrp> </sourceDoc>
```