

Textkodierung mit TEI (Titel des Tutorials)

## **Aufgabe 10 - TEI anpassen: Mit Roma ein Schema definieren und es verwenden (Beispiel: Jespersen)**

Christof Schöch

DARIAH-DE

Lizenz: Creative Commons Attribution 3.0; TEI @ Oxford , July 2012 (CC BY-SA).

Das vorliegende Dokument beruht weitgehend 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.

### **Überblick über die Aufgabe**

#### **1. Inhalte**

In dieser Übung werden wir TEI so anpassen, dass nur die Elemente verfügbar sind, die wir auch verwenden wollen. Um ein TEI-Schema anzupassen oder zu generieren, muss man wissen, welche Elemente man brauchen wird und welche nicht. Bis man soweit ist, benötigt man oft einen längeren Prozess der Dokumentenanalyse. In unserem Fall wird das nicht so schwierig sein, weil wir mit einem sehr einfachen Dokument arbeiten.

#### **2. Lernziele**

Nachdem Sie diese Übung abgeschlossen haben, werden Sie folgende Dinge tun können:

- \* Analysieren, welche Elemente, Attribute und Werte für ein TEI Dokument notwendig sind
- \* Mit dem Dienst Roma ein TEI Schema passgenau für ein bestimmtes TEI Dokument anpassen: Module, Elemente, Attribute auswählen und Werte vordefinieren
- \* In Oxygen ein eigenes Schema verwenden
- \* Mit Roma eine lesbare Spezifikation des spezifischen Schemas generieren
- \* Die Funktionsweise des zugrundeliegenden TEI ODD XML-Formats verstehen

#### **3. Materialien**

- \* Bilddatei `Jespersen-5117.jpg`;
- \* TEI-Datei `Jespersen-Start.xml`.

### **Aufgabenstellung**

#### **1. Vorbereitung: Analyse (gemeinsam)**

Bevor wir loslegen, müssen wir einen genaueren Blick auf unseren Text, einen kur-

zen Brief von Otto Jespersen an Hugo Schuchardt, werfen. Öffnen Sie die Bilddatei (Jespersen-5117.jpg) und die TEI-Datei (Jespersen-Start.xml – mit Oxygen öffnen); beide finden Sie in dem Ordner Übung 03. Was wollen wir kodieren? Machen wir auf einem separaten Blatt eine Liste; links das Phänomen, rechts das entsprechende TEI-Element. (Hinweis: Die Elemente sollten sich vor allem auf die Funktion einzelner Passagen und auf die Struktur des Gedichts beziehen, mit Attributen und Werten können wir aber auch die Darstellung ein wenig festhalten.) Bevor Sie weitermachen, sollten wir soweit sein, dass wir eine gemeinsame Liste der Phänomene und Elemente, Attribute und Werte vorliegen haben.

## 2. Ihr aktuelles Schema

Oxygen weiß gewissermaßen über TEI Bescheid; man kann einige Typen von TEI-Dateien neu erstellen und sie werden direkt mit einem TEI-Schema verbunden. Oder Oxygen erkennt, wenn es sich bei einer XML-Datei um ein Dokument in XML handelt und lädt das Schema, das in der Datei vermerkt ist. In unserem Fall ist das Dokument mit dem Default-Schema `tei_all` verknüpft. Testen Sie einmal, was das bedeutet:

- \* Ist das Dokument valide? Testen Sie dies.
- \* Gehen Sie in dem Dokument, das eine einfache Transkription enthält, an irgendeine Stelle innerhalb eines Absatzes, nach einem öffnenden `p` und setzen Sie den Cursor dort an.
- \* Wenn Sie nun "`<`" eingeben und warten (das kennen Sie bereits), öffnet Oxygen eine Dropdown-Liste mit allen Elementen, die an dieser Stelle eingefügt werden können.
- \* Gehen Sie einmal diese Liste durch. Dort sind Elemente wie `address`, `camera`, `incident`, `metamark`, und `notatedMusic` aufgelistet. Lesen Sie sich zu einigen der Elemente einmal die Tooltips rechts neben der Liste durch, die Oxygen ebenfalls aus dem TEI Schema aufruft.
- \* Drücken Sie `ESC` und löschen Sie das "`<`", das Sie eingefügt hatten.

Wie Sie sehen, kann man in einem Absatz sehr viele Elemente einfügen. Das sind mehr Möglichkeiten, als man in einem bestimmten Projekt benötigt. Es ist sogar eine Fehlerquelle, so viele Optionen offenzuhalten. Je weniger Wahlmöglichkeiten es gibt, desto kohärenter wird die Kodierung am Ende.

## 3. Mit Roma ein neues Schema anlegen

Mit Roma kann man ein TEI Schema anpassen und alles löschen, was man nicht braucht. Gehen Sie im Browser zu <http://www.tei-c.org/Roma/> (wenn Sie nicht schon dort sind); es gibt vier Optionen:

- \* 1. Build up: Hiermit kann man ein neues Schema anpassen, indem man einem minimalen Schema Elemente hinzufügt.
- \* 2. Reduce: Hiermit kann man ein Schema anpassen, indem man Elemente u.a. vom vollen Schema mit allen Elementen entfernt
- \* 3. Template: Hiermit kann man eine der häufig benutzten Schemata als Ausgangspunkt nehmen und Elemente hinzufügen und/oder entfernen

- \* 4. Open: Hiermit kann man ein vorhandenes Schema öffnen und weiter bearbeiten.

Da wir gerade in Oxygen `tei_all` in Aktion gesehen haben, wählen wir die Option `Reduce` und klicken auf `Start`. Setzen Sie nun die folgenden Parameter:

- \* Title: Geben Sie Ihrem Schema einen Namen!
- \* Filename: Schlagen Sie einen Dateinamen vor; er sollte möglichst einfach sein, bspw. `tei-jespersen`.
- \* Author name: Setzen Sie hier Ihren Namen ein.
- \* Die Beschreibung lassen wir einfach vorerst wie sie ist.
- \* Speichern Sie Ihre Einstellungen. Oben rechts erscheint der Titel dieses Schemas.

#### 4. Module löschen und hinzufügen

TEI Elemente sind in Modulen gruppiert, die teils thematische, teils strukturelle Gemeinsamkeiten haben. Jedes Modul hat ein Kürzel (bspw. `header`), und für jedes Modul gibt es ein Kapitel in den Guidelines (z.B. `The TEI Header`). Dies sind die Module und Kapitel, die im vorhergehenden Inputreferat vorgestellt wurden.

- \* Klicken Sie auf den Reiter `Modules`, um die Liste der verfügbaren und aktivierten Module zu sehen.
- \* Links steht die Liste der verfügbaren Module mit einer Kurzbeschreibung; rechts steht die Liste der im aktuellen Schema momentan aktivierten Module. Weil wir mit einem maximalen Schema begonnen haben, sind die Listen im Moment identisch.
- \* Viele der aktivierten Module können Sie von vorneherein löschen; dazu gehören: `'analysis'`, `'certainty'`, `'corpus'`, `'dictionaries'`, `'drama'`, `'gaiji'`, `'iso-fs'`, `'linking'`, `'nets'`, `'spoken'`, `'textcrit'`, and `'tagdocs'`! Löschen Sie alle Module außer `'core'`, `'header'` und `'textstructure'`.
- \* Damit haben wir sehr viel gelöscht, und das Schema passt schon etwas besser für unser Dokument.

#### 5. Elemente entfernen oder hinzufügen

Wir haben das Schema bereits stark reduziert, aber die vorhandenen Elemente beinhalten immer noch viele Elemente, die wir nicht brauchen.

- \* Klicken Sie auf `core` (nicht auf `remove`, sondern auf das Wort `core` selbst). Daraufhin öffnet sich eine Liste aller Elemente im Modul `'core'`.
- \* Jede Zeile dieser Tabelle hat mehrere Einträge: Element, Included/Excluded, Name, Link zur Definition des Elements, kurze Beschreibung des Elements, Link zu den Attributen.
- \* Verwenden wir eine neue Strategie: markieren Sie alle Elemente als `exclude`, indem Sie oben auf `exclude` klicken.
- \* Anschließend gehen Sie die Liste durch und aktivieren diejenigen Elemente,

die wir oben definiert haben, und die hier auftauchen: mit dem `RadioButton` können Sie für jedes Element seinen Status separat ändern.

\* Speichern Sie Ihre Arbeit!

Jetzt haben wir eine erste, nützliche Fassung des Schemas definiert.

## 6. Das Schema generieren

Klicken Sie auf den Reiter `Schema`; es wird eine Liste möglicher Schema-Formate angezeigt. Intern verwendet TEI ein "Meta-Schema", das ODD heißt ("One Document Does it All") und auf dessen Grundlage verschiedene Schemata generiert werden können.

\* Wählen Sie das Schema-Format `Relax NG Compact Syntax` oder `Relax NG XML Syntax` aus.

\* Klicken Sie auf `generate`, und Sie sollten das Schema dann abspeichern können.

\* Finden Sie die entsprechende Datei (sie sollte `tei-jespersen.rng` oder `tei-jespersen.rnc` heißen) und kopieren Sie sie in den gleichen Ordner, in dem auch Ihre Datei `Jespersen.xml` liegt.

\* Den Browser nicht schließen, sonst müssen Sie wieder von vorne anfangen!

## 7 . Das Schema in Oxygen mit der TEI-Datei assoziieren

Gehen Sie nun wieder zu Oxygen, wo die Datei `Jespersen.xml` geöffnet ist. Am Besten speichern Sie die Datei einmal unter einem neuen Namen ab, bspw. einfach `Jespersen.xml`. Bis jetzt hat Oxygen das Schema `tei_all` verwendet; jetzt assoziieren Sie das Schema, das wir gerade generiert haben.

\* Gehen Sie zum Menü `Document > Schema > Associate Schema`.

\* Im Dialogfeld wählen Sie `Klick` auf das kleine Ordner-Symbol die Datei `tei-jespersen.rng` aus, bestätigen Sie mit `OK`.

Schon fertig! Oxygen hat übrigens fast ganz oben eine Zeile in das XML-Dokument eingefügt, die auf das Schema verweist. Probieren wir also aus, wie sich Oxygen jetzt verhält.

\* Gehen Sie irgendwo innerhalb eines Absatzes hinein und tippen Sie wieder "`<`": ist die Liste der erlaubten Elemente kleiner geworden? Wenn ja, gut!

\* Kodieren Sie einige der Merkmale, die wir vorher unterschieden haben; verwenden Sie insbesondere:

\* Zu Beginn des Briefs: `opener`, `dateline`, `date`

\* Für den Haupttext: `p`, `abbr`, `q`, `hi`

\* Für den Schluss des Briefs: `closer`, `salute`, `signed`

## 8. Auswählen von Attributen, definieren von Werten

Der nächste Schritt ist, für bestimmte Elemente Attribute und Werte zu definieren, um genauere Angaben machen zu können. Beispielsweise kann das Element `div`

auch genutzt werden, um die Gattung oder den Typ eines solchen Abschnitts anzugeben. Außerdem haben wir das Element `hi` verwendet, das eine Hervorhebung markiert, ohne anzugeben, wie diese Hervorhebung denn genau aussieht. Um uns in beiden Fällen vor Willkür und Inkonsistenzen zu bewahren, sollten wir für diese Attribute und Werte eine kleine Typologie erstellen und diese im Schema festhalten. Für eine Edition von Korrespondenz, wäre es beispielsweise denkbar, Briefe und Postkarten zu unterscheiden. Dies könnten wir mit einem Attribut "type" für das Element `div` angeben, und nur die beiden Werte "letter" und "postcard" zulassen. Machen wir das einmal!

- \* Gehen Sie wieder zu Roma
- \* Wählen Sie den Reiter `Moduls`, klicken Sie dort auf das Modul `textstructure`, und finden Sie den Eintrag `div`.
- \* Klicken Sie ganz rechts im Eintrag `div` auf `Change attributes`. Nun wird eine Seite mit allen erlaubten Attributen für `div` angezeigt; das sind ganz schön viele; zu viele!
- \* Schließen Sie alle Attribute aus, indem Sie ganz oben auf `exclude` klicken.
- \* Aktivieren Sie nur das Attribut "type", indem Sie bei dem entsprechenden Eintrag den Radiobutton wieder auf `include` stellen.
- \* Speichern Sie Ihre Arbeit.

Jetzt darf das Element `div` nur noch ein Attribut tragen, nämlich "type". Das reicht aber noch nicht, wir sollten auch die möglichen Werte festlegen.

- \* Klicken Sie auf das Attribut `type`; eine Seite mit einem Formular für das Attribut führen; belassen Sie die meisten Einstellungen, wie sie sind.
- \* Ändern Sie aber den Punkt `closed list` zu `Yes`. So sind nur die Werte erlaubt, die wir gleich angeben.
- \* Im Feld `List of values` fügen Sie die beiden Werte ein: "postcard" und "letter", getrennt nur durch ein Komma, d.h. ohne Leerzeichen.

Das war der erste Teil. Führen Sie nun die gleichen Schritte für zwei weitere Elemente durch (Sie finden beide im Modul `core`).

- \* Für `hi`: Attribut "rend", Werte "underline" oder "italic". Dieses Attribut können Sie obligatorisch machen (`potional: no`), denn andernfalls hat es keinen Sinn.
- \* Für `add`: Attribut "place", Werte "above" oder "below" (Description: `Position of an addition.`)
- \* Denken Sie daran, jeweils zu speichern!

## 9. Was passiert nun?

Jetzt generieren wir wieder das Schema, speichern es ab, benennen es um (`tei_jespersen2.rng`) und assoziieren dieses neue Schema mit der XML-Datei. Machen Sie das einfach so, wie oben in den Abschnitten 8. Schema generieren und 9. Das Schema in Oxygen mit der TEI-Datei assoziieren beschrieben. Prüfen

Sie nach, was nun passiert, wenn Sie den Elementen Attribute geben.

- \* Gehen Sie an das Ende des öffnenden `div` Elements: der Cursor muss genau zwischen dem `v` und dem `>` sein. Fügen Sie ein Leerzeichen ein.
- \* Oxygen schlägt vor, ein Attribut einzufügen; nur "type" wird angeboten, wählen Sie es also aus, indem Sie mit `Return` bestätigen.
- \* Oxygen schlägt Ihnen nun vor, einen der beiden definierten Werte auszuwählen; wählen Sie den richtigen Wert! (Beachten Sie, dass Oxygen das `=` und die Anführungszeichen für Sie automatisch korrekt setzt.)

Das war der erste Schritt; machen wir nun weiter.

- \* Prüfen Sie, ob das Dokument weiter valide ist; wenn nicht, woran liegt es? Was müssen Sie tun, um wieder ein valides Dokument zu haben?
- \* Fügen Sie außerdem für das Element `add` das Attribut "rend" mit einem geeigneten Wert ein. Ist Ihr Dokument wieder valide?

## 10. Schon fertig? (Optionalen Aufgabenteil; sonst direkt weiter mit Punkt 11)

In der Analyse des Dokuments am Anfang der Übung haben wir ein paar Phänomene und die passenden Elemente erst einmal beiseite gelassen.

- \* Versuchen Sie einmal, eines dieser Elemente in das TEI-Dokument einzufügen. Was passiert?

Sie können eine weiteres Elemente zum Schema hinzufügen.

- \* Fügen Sie in Roma ein Element zum Schema hinzu; Speichern Sie das Schema erneut ab
- \* Assoziieren Sie sie mit dem XML-Dokument
- \* Überprüfen Sie, ob Sie nun diese weiteren Elemente einfügen können.

Was passiert nun, wenn Sie die zusätzlichen Elemente einfügen? Grünes Quadrat? Sehr gut!

## 11. Ein selbst definiertes Schema für Roma speichern

Jetzt sollten wir das Schema noch in einer Form abspeichern, in der wir es später wieder mit Roma öffnen und weiter bearbeiten können.

- \* Gehen Sie wieder zu Roma.
- \* Klicken Sie auf den Reiter `Save Customization`; eine XML-Datei sollte zum Abspeichern angeboten werden; speichern Sie die Datei an einem geeigneten Ort ab.
- \* Schließen Sie den Browser noch nicht!
- \* Öffnen Sie die Datei `tei-jespersen.xml` mit Oxygen und formatieren Sie das Dokument neu (`Strg+Shift+P`).
- \* In dem Dokument finden Sie eine Reihe von Elementen `moduleRef`; sie

beinhalten die Definition der Module, die in dem Attribut "key" jeweils angegeben werden. Wie Sie sehen, ist sehr viel ausgeschlossen! (Wert: "except")

\* Schauen Sie sich auch das erste Element `elementSpec` an, das sich auf `div` bezieht; Sie sehen eine lange Liste von Attributen, die ausgeschlossen sind (`mode="delete"`).

\* Weiter unten in der Liste taucht das Attribut "type" auf, das nicht ausgeschlossen ist, sondern speziell definiert wurde (`mode="change"`). Hier sind die beiden möglichen Werte eingetragen (`valltem="postcard"` und `valltem="letter"`).

## 12. Letzter Schritt: eine Dokumentation generieren

Mit Roma kann man nicht nur Schemata definieren, sondern auch gleich die passende Dokumentation dazu produzieren. Das machen wir nun zum Abschluss noch.

\* Gehen Sie im Browser auf `Documentation`

\* Wählen Sie dort `HTML web page` aus und klicken Sie auf `Generate`

\* Der Browser wird entweder die HTML-Seite direkt anzeigen, oder aber Sie können Sie abspeichern und mit dem Browser öffnen.

\* Hier ist nun die Dokumentation aus den Guidelines zusammengestellt, die für das von Ihnen definierte Schema relevant ist.

Geschafft! Das ist das Ende dieser Übung.