

Textkodierung mit TEI (Titel des Tutorials)

Aufgabe 10b: Mit Roma ein Schema erstellen und in Oxygen zur Textkodierung nutzen (Nietzsche-Beispiel)

Christof Schöch

DARIAH-DE

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

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.

Aufgabe 10b: Mit Roma ein Schema erstellen und in Oxygen zur Textkodierung nutzen (Beispiel: Nietzsche)

1. Lernziele

- * Verständnis für den Zusammenhang zwischen Schema und Kodierung entwickeln
- * Roma für das Erstellen eines Schemas nutzen können

2. Überblick

In dieser Übung geht es darum, für eine bestimmte Textsorte ein Schema zu erstellen und dieses für die Kodierung eines Textes zu nutzen. Die Übung lässt sich in drei Teile gliedern:

- * Teil 1: das formlose Erstellen der Schema-Regeln auf der Grundlage einer Beispieldkodierung.
- * Teil 2: das Zusammenstellen einer Schema-Datei mit Hilfe der Web-Anwendung Roma.
- * Teil 3: das Kodieren eines Textes unter Berücksichtigung des Schemas im XML-Editor Oxygen.

3. Materialien

- * `Nietzsche-Ergebnis.xml`

Teil 1: Festlegen, welche Schema-Regeln gebraucht werden

Die Grundlage der Übung ist der Text von Nietzsche, der in einer vorigen Übung bereits kodiert wurde. Schauen Sie also noch einmal die Kodierung dieses Textes an, so wie sie in der Musterlösung vorgeschlagen wurde (Datei: `Nietzsche-Ergebnis.xml`). Diese Kodierung dient uns als eine gemeinsame Grundlage. Zunächst müssen Sie überlegen, welche Elemente in der Kodierung benötigt werden. Grundsätzlich wird natürlich das Element `TEI` als „root element“ benötigt; halten

wir das gleich einmal fest; dieses Element befindet sich im Modul „textstructure“.

* TEI (textstructure)

1. Welche Elemente werden für den `teiHeader` benötigt?

Gehen Sie durch den `teiHeader` der Nietzsche-Beispielkodierung und notieren Sie alle notwendigen Elemente (kleiner Tipp: es sind mindestens 14).

2. Welche Elemente im `body` werden benötigt?

Gehen Sie durch den `body` der Nietzsche-Beispielkodierung und notieren Sie alle notwendigen Elemente (kleiner Tipp: es sind mindestens 8 Elemente).

3. Attribute und Werte

Diesen Aspekt des Schemas lassen wir zunächst einmal aus.

Teil 2: Erstellen des Schemas in Roma

1. Der Webservice „Roma“ der Text Encoding Initiative

Gehen Sie mit dem Browser zu folgender Adresse: <http://www.tei-c.org/Roma/>. Wählen Sie dort die Option `Build up` aus.

2. Ein neues Schema anlegen (Reiter `Customize`)

- * Name angeben, zum Beispiel: TEI für Nietzsche
- * Dateiname angeben, zum Beispiel: `tei_ntz`.
- * Alles andere lassen, und die Änderungen speichern (roter `Save`-Button).

3. Für jedes Modul die passenden Elemente aktivieren („Modules“)

- * Zum Reiter `Modules` wechseln. Links ist die Liste der verfügbaren Module (cf. die Kapitel in den Guidelines und die zweite Sitzung.) Rechts ist die Liste der aktiven Module. Wir ändern hier erst einmal nichts.
- * Klicken Sie in der rechten Spalte auf das Wort `Header`. Dann erscheint die Liste der Elemente, die in diesem Modul enthalten sind.
- * Klicken Sie auf `Exclude`, um erst einmal alle Elemente zu deaktivieren.
- * Jetzt können Sie diejenigen Elemente aktivieren (einzeln als `include` markieren), die Sie benötigen (siehe Listen oben, Teil 1). Nicht alle Elemente, die Sie benötigen, sind in diesem ersten Modul enthalten.
- * Danach nicht vergessen, die Änderungen zu speichern (ganz unten).
- * Gehen Sie zurück zum Reiter `Modules`. Wiederholen Sie das gleiche Verfahren für das Modul „core“ und das Modul „textstructure“.
- * Am Ende sollten Sie alle Elemente, die Sie oben notiert haben, aktiviert haben.

4. Überprüfen, ob das Schema ein gültiges TEI-Schema ist („Sanity Checker“)

- * Klicken Sie auf den Reiter `Sanity Checker`. Diese Funktion von Roma überprüft, ob das von Ihnen gerade generierte Schema ein formal sinnvolles Schema ist.
- * Wenn Sie kein positives Ergebnis haben, melden Sie sich bitte! Wenn Sie ein positives Ergebnis haben, können Sie weitermachen.

5. Masterdatei und Schema abspeichern

- * Jetzt können Sie die Masterdatei (ODD-File) und das Schema selbst abspeichern. Die Masterdatei enthält alle Informationen über das Schema, die notwendig sind, um ein Schema in verschiedenen Formaten zu generieren, oder um eine Dokumentation zu generieren, oder um das Schema zu einem späteren Zeitpunkt in Roma weiter zu bearbeiten. (Beides geht mit einer bestimmten Schema-Datei nicht!)
- * Die Masterdatei erhalten Sie, wenn Sie auf den Reiter `Save Customization` klicken. Speichern Sie die Datei unter einem geeigneten Namen ab.
- * Die Schema-Datei erhalten Sie, wenn Sie auf den Reiter `Schema` klicken.
- * Wählen Sie dort das Format `Relax NG Compact Syntax` aus, das Sie schon ein wenig kennen.
- * Speichern Sie auch das Schema unter einem geeigneten Namen ab.
- * Vorsicht! Schließen Sie die Roma-Seite noch nicht!

Teil 3: Nutzen des Schemas in Oxygen

1. Ein neues XML-Dokument in Oxygen anlegen

- * Öffnen Sie Oxygen.
- * Wählen Sie `Datei > Neu`. Wählen Sie als Dateityp: `TEI P5 All`. Ein neues TEI-Dokument wird geöffnet, das mit dem Standard-Schema `tei_all` verknüpft ist.
- * Speichern Sie das neue Dokument in dem gleichen Ordner ab, wo sich auch das neue Schema befindet.

2. Testen Sie, wie sich Oxygen mit diesem Schema verhält

- * Tippen Sie nach dem Endtag `title` einmal das Zeichen „<“ und warten Sie kurz ab. Sie sollten nun eine Liste aller erlaubten Elemente sehen. Wie lang ist diese Liste? Welche Elemente werden aufgelistet? Welche Elemente sind eventuell im Kontext unseres Nietzsche-Textes nicht sinnvoll?
- * Gehen Sie nun mit dem Cursor einmal in den Bereich `body`, und klicken Sie dort auf eine Stelle innerhalb des kurzen Textes („Some text here“). Tippen Sie auch hier einmal das Zeichen „<“ und warten Sie kurz ab. Wie lang ist die Liste hier? Welche Elemente sind eventuell im Kontext unseres Nietzsche-Textes nicht sinnvoll?

3. Das Schema mit dem XML-Dokument verknüpfen

- * Löschen Sie die Zeilen 4-6 in dem neuen XML-Dokument.
- * Löschen Sie außerdem die Zeile 2 in dem neuen XML-Dokument.
- * Assoziieren Sie nun das Schema, das Sie gerade erstellt haben, mit dem XML-Dokument: `Dokument > Schema > Schema assoziieren`; dann das neue Schema auswählen, und bestätigen.
- * Prüfen Sie noch einmal, welche Elemente an den beiden Stellen im Dokument nun erlaubt sind. Ist das sinnvoll so, wenn wir unser Nietzsche-Dokument mit Hilfe dieses Schemas kodieren wollen?

4. Die Kodierung des Dokuments beginnen

- * Ergänzen Sie in dem neuen Dokument einzelne Informationen aus der Musterkodierung. Beobachten Sie dabei genau, wie sich Oxygen verhält, und überprüfen Sie regelmäßig, ob Ihr Dokument valide ist.
- * Fügen Sie beispielsweise, ohne Copy & Paste zu verwenden, die Informationen über den Autor des Textes zum Header hinzu.
- * Fügen Sie außerdem, wieder ohne Copy & Paste zu verwenden, die Informationen zum `publicationStmnt` hinzu.
- * Fügen Sie außerdem den Titel und Untertitel des Nietzsche-Textes hinzu, und probieren Sie einmal, welche Attribute Sie vergeben können.

5. Wenn noch Zeit bleibt

Wenn noch Zeit bleibt, können Sie weitere Elemente aus der Musterkodierung verwenden. Oder probieren Sie mal aus, was passiert, wenn Sie ein zusätzliches Element vergeben wollen! Außerdem: Sie könnten einmal einen Blick in das Relax-NG-Schema werfen. Ab etwa Zeile 680 finden Sie Dinge wieder, die Sie vielleicht schon kennen. Ende der Übung!

6. Weiterführende Hinweise

Möchten Sie mehr über Schemas, Roma und ODD herausfinden? Dann nutzen Sie die folgenden Ressourcen:

- * Implementation of an ODD System in den TEI Guidelines;
- * Kapitel "Documentation Elements" in den TEI Guidelines;
- * Die entsprechende Übung in TEI by Example.