Aufgabe 4: Dokumentstruktur und Schreibweise in frühen Drucken: Stultifera Navis XXVII

Christof Schöch

DARIAH-DE

Übersicht

1. Lernziele

* Praktische Erfahrung mit der Anwendung von Elementen wie `app`, `rdg` und `lem`
* Verständnis, wie Informationen im `teiHeader` und im `body` zusammenhängen
* Weitere Vertrautheit mit Oxygen

2. Materialien (im ZIP-Archiv "Materialien-04")

* PDF-Datei (`StultiferaNavis-1497-XXVII.pdf`): enthält das digitale Faksimile der beiden Seiten aus dem Stultifera Navis, d.h. der Übersetzung des Narrenschiffs durch Jakob Locher, in der Ausgabe Basel, Johannes Bergmann, 1497, und zwar dem Nachdruck von vom August 1497, der Holzschnittbeischriften und Marginalien von Sebastian Brant enthält.
* TXT-Datei (`StultiferaNavis-Transkription.txt`) mit der Transkriptionen des Textes, der auf der ersten Seite zu finden ist.
* XML/TEI-Datei (`StultiferaNavis-Start.xml`) als Ausgangsbasis für die Kodierung des Textes.

3. Überblick

In dieser Übung geht es darum, ein Dokument mit komplexer Struktur zu kodieren, in unserem Fall eine Seite aus der lateinischen Fassung des Narrenschiffs, auf der man verschiedene strukturelle, semantische und layout-bezogene Textbereiche unterscheiden kann.

Erster Teil: Analyse des Dokuments

1. Analyse der ersten Seite

Entpacken Sie das ZIP-Archiv “Materialien-04.zip”; im darin enthaltenen Ordner befinden sich die Materialien für diese Übung. Schauen Sie sich einmal das digitale Faksimile (`StultiferaNavis-1497-XXVII.pdf`) an.

* Analysieren Sie nun zunächst diese erste Seite, indem Sie die verschie-

denen Elemente, aus denen die Seite besteht, unterscheiden, benennen und beschreiben.

2. Liste anlegen

Machen Sie hier (weiter unten und/oder auf der Rückseite des Blattes ist Platz) eine Liste dieser Elemente oder Teile der Seite, und unterscheiden Sie dabei insbesondere, wie sich diese Elemente unter struktureller, semantischer oder layoutbasierter Perspektive beschreiben lassen.

* "semantisch" meint dabei die inhaltliche Funktion der Elemente
* "strukturell" meint die hierarchische Organisation der Elemente
* "layout-bezogen" meint die räumliche Anordnung der Elemente auf der Seite.

* # semantisch strukturell layout
* ...
* ...
* ...
* ...
* ...
* ...
* ...
* ...
* ...
* ...

Wenn Sie die Analyse abgeschlossen haben, sollten wir kurz eine Pause einlegen und die Ergebnisse gemeinsam besprechen.

Zweiter Teil: Das Dokument kodieren

3. Die Struktur der Seite abbilden

Öffnen Sie mit Oxygen die Datei `StultiferaNavis-Start.xml`. Sie werden sehen, dass hier bereits ein `teiHeader` vorhanden ist (den wir hier beiseite lassen), sowie ein `body`-Element, das fast leer ist. In den Konventionen der Text Encoding Initiative dominiert zunächst einmal die Kodierung struktureller Merkmale. Dies geschieht mit den folgenden Elementen:

* `body` - Für die Kodierung des Textes selbst (nicht der Metadaten)
* `front` und `back` - Für die Kodierung von Dingen wie Titelseite (front matter) oder Index (back matter)
* `div` - Für die Kodierung von strukturellen Einheiten in einem Text: Kapitel, Unterkapitel, Abschnitte; dieses Element kann auch ineinander geschachtelt werden.

* `lg` und `l` - Für die Kodierung von Gruppen von Zeilen und einzelnen Verszeilen (nicht typographischer Zeilen!).

* `p` und `ab` - Für die Kodierung von Prosaabsätzen sowie anderen, undefinierten typographischen Einheiten.

Auf der Grundlage der bereits erfolgten Analyse der Seite können Sie nun die Struktur des Dokuments kodieren (die Illustration lassen wir hier einmal außer Acht).

* Kodieren Sie mit Hilfe des Elements `div` die grobe Struktur des Dokuments; sie können `div`s auch ineinander verschachteln, wenn das sinnvoll erscheint!

4. Semantische Informationen hinzufügen

In der Liste der Elemente haben Sie auch semantische Merkmale notiert. Diese Merkmale sollten wir nun auch noch in der Kodierung festhalten. Dies können wir für jede bereits kodierte strukturelle Einheit mit dem Attribut "type" umsetzen.

* Entscheiden Sie für jede strukturelle Einheit, welche semantische oder inhaltliche Funktion sie hat, und geben Sie ihr ein Attribut "type" mit einem Wert, der die semantisch-inhaltliche Funktion definiert.

Hinweis: Bei der Kodierung eines längeren Textes wird es ratsam sein, hier eine Typologie zu entwickeln und diese zu dokumentieren. Eine solche Dokumentation sollte an zwei Orten stattfinden: einerseits in Prosa in der `encodingDesc`, die sich im `teiHeader` befindet; andererseits in maschinen-lesbarer Form im Schema, das der Kodierung zugrunde liegt.

5. Titel kodieren

Fügen wir nun auch die Transkriptionen hinzu und kodieren diese. Wir beginnen mit dem Titel.

* Kopieren Sie die Transkription des Titels aus der Datei `StultiferaNavis-Transkription.txt` heraus und fügen Sie sie an der richtigen Stelle in Ihr XML-Dokument ein.

* Überprüfen Sie, ob Ihr Dokument weiterhin valide ist (`Strg+Shift+V`). Vermutlich ist das nicht der Fall! Schauen Sie sich die Fehlermeldungen an.

* Die Fehlermeldung bezieht sich auf den "entity name" und das "&"-Zeichen. Dieses Zeichen hat eine besondere Funktion in XML, nämlich den Beginn von besonders kodierten Entitäten oder Zeichen anzugeben. Oxygen erwartet nach dem "&" den Code für eine solche Entität.

* Das bedeutet, wir müssen das "&" anders kodieren, nämlich als Entität. Ersetzen Sie also das & durch eine "entity reference", die folgendermaßen aussieht: "&" (ohne die Anführungszeichen. Ist Ihr Dokument nun valide?

6. Das Motto kodieren

Kodieren wir nun das Motto. (Damit sind die ersten vier Zeilen gemeint, die dem Titel folgen. Vielleicht haben Sie diesem Struktur-Element einen anderen Namen

gegeben).

* Kopieren Sie die Transkription der vier Zeilen und fügen Sie sie in das XML-Dokument ein.
* Es handelt sich um vier Verszeilen, die eine Gruppe bilden. Diese Struktur sollten Sie mit den Elementen `lg` (line group) und `l` (Verszeile) kodieren.

Dritter Teil: die "besonderen" Zeichen kodieren

7. Was geschieht mit den "besonderen" Zeichen?

Die vorliegende Transkription modernisiert oder normalisiert den Text im Vergleich zu dem, was man in dem digitalen Faksimile sehen kann. Insbesondere enthält der Originaltext mehrere verkürzte Schreibungen. Diese sollten wir nun genauer in den Blick nehmen und dann präzise kodieren. Im Motto findet man drei verkürzte Schreibungen:

* Zeile 1: Transkription "praeterit" / Faksimile "kleines e mit Häkchen (Ogonek).
* Zeile 2: Transkription "atque" / Faksimile "kleines q mit kleinem Et.
* Zeile 4: Transkription "interitum" / Faksimile: kleines u mit Tilde.

Zunächst einmal ist zu klären, ob die für die verkürzten Schreibungen verwendeten Zeichen in Unicode (oder sogar in UTF-8) vorhanden sind oder nicht. Das kann unter Umständen etwas Recherche notwendig machen.(Die Referenz-Seite hierfür ist die des UNICODE Konsortiums: http://www.unicode.org/charts/. Für die Frühdrucke besonders wichtig ist der Abschnitt "Medievalist Additions" im Bereich "Latin Extended D".) Heute bekommen Sie sofort das Ergebnis:

* Das "e mit Ogonek" und das "u mit Tilde" sind in UTF-8 vorhanden; sie können direkt in Oxygen als Zeichen eingegeben werden, und zwar mit Hilfe der Funktion: `Edit > Insert from character map`.
* Das "q mit kleinem Et" ist nicht in UTF-8 oder Unicode vorhanden, beide Einzelbuchstaben aber schon. Hier ist es notwendig, einen Glyph zu definieren. (Das lassen wir hier beiseite.)

8. Kodieren der ersten beiden verkürzten Schreibweisen

Die beiden verkürzten Schreibweisen repräsentieren jeweils mit einem besonderen Buchstaben mehrere andere Buchstaben. Es ist eine editorische Entscheidung, wie man dies dokumentiert (siehe Input-Referat). Hier entscheiden wir uns für eine relativ einfache Lösung, die aber immerhin verkürzte und vollständige Schreibweise gemeinsam dokumentiert. Kleine Erinnerung

* Mit `choice` kann man alternative Lesarten kodieren
* Innerhalb von `choice` kann man mit `am` (abbreviation marker) und `ex` (editorial expansion) die verkürzte und die vollständige Schreibweise dokumentieren.

Kodieren Sie nun erst einmal das "e mit Ogonek" und das "u mit Tilde" und ihre vollständigen Schreibungen als Alternativen.

* Gehen Sie zunächst mit dem Cursor an die richtige Stelle im ersten relevanten Wort, "preterit".
* Gehen Sie in Oxygen in das Menü `Edit > Insert from character map`
* Finden Sie dort das "e mit Ogonek" und fügen Sie es als Zeichen ein
* Schließen Sie die "character Map".

Jetzt haben Sie das spezielle Zeichen, dann können Sie jetzt die alternativen Lesarten kodieren.

* Umgeben Sie das "e mit Ogonek" mit einem `choice`-Element. Markieren Sie dazu das "e mit Ogonek", drücken Sie `Strg+E` und tippen Sie "choice" in das kleine Eingabefeld; bestätigen Sie mit `Enter`.
* Umgeben Sie jetzt das "e mit Ogonek" mit einem `am`-Element
* Unmittelbar nach dem schließenden </am>-Tag, aber noch vor dem schließenden </choice>-Tag, geben Sie nun die vollständige Schreibung an, nämlich "ae".
* Umgeben Sie das "ae" mit einem `ex`-Element.
* Prüfen Sie, ob das Dokument weiterhin valide ist (`Strg+Shift+V`).
* Formatieren Sie das Dokument neu (`Strg+Shift+P`).

Das wird in der Kodierung schon ziemlich komplex, oder? Oxygen hilft Ihnen aber durch die Formatierung, den Überblick zu behalten.

* Wiederholen Sie das Prozedere für "anime/ae" und "interitu/um"

9. Schon fertig, während die anderen noch beschäftigt sind?

Dann können Sie die verkürzte Schreibweise für "que" noch kodieren! Dazu muss man eine Definition des Glyphen in den `teiHeader` einfügen und dann im Text statt eines UNICODE-Zeichens das Element `g` verwenden.

* Fügen Sie im Header folgende Zeilen nach dem schließenden </fileDesc>-Tag, aber vor dem schließenden </teiHeader>-Tag ein: <encodingDesc> <charDecl> <glyph xml:id="br-que"> <glyphName>Small latin letter q with a latin small letter et attached. </glyphName> </glyph> </charDecl> </encodingDesc>
* Nun können Sie das "q mit et" als ein leeres <g/>-Element repräsentieren.
* Geben Sie dem `g`-Element ein Attribut "ref", und tragen Sie als Wert des Attributs die xml:id des Glyphen ein, also "#br-que" (mit dem "#", weil Sie auf die ID verweisen). Das sollte nun also so aussehen: <g ref="#br-que"/>
* Dieses leere Element können Sie nun verwenden, wie einen Buchstaben, und wieder mit `choice` sowie `am` und `ex` die Vorkommen von "atque" korrekt kodieren.

10. Die Kodierung visualisieren

Weitere Textabschnitte würde man nun in analoger Weise kodieren. Wir belassen es hier einmal, verknüpfen das Dokument aber noch mit einem Stylesheet.

* Öffnen Sie mit Oxygen die Datei `StultiferaNavis-Kodierung.xml`. Diese Datei ist enthält eine Musterkodierung, mit der wir alle auf dem selben Stand sind.

* Wählen Sie in Oxygen im Menü `Document > XML Document` den Eintrag `Associate CSS/XSLT Stylesheet` aus; im Dialog klicken Sie auf das kleine gelbe Ordner-Symbol.

* Gehen Sie zum Ordner `Materialien-04` und wählen Sie dort das Stylesheet `StultiferaNavis-Stylesheet-EX.css` aus (auf "Open" klicken). Achtung, es sollte wirklich das Stylesheet mit "EX" am Ende sein. Bestätigen Sie die Auswahl mit `OK`.

In der dritten Zeile des XML-Dokuments sollte jetzt ein Verweis auf das Stylesheet stehen.

* Gehen Sie im Menü auf `Document > File` und klicken Sie dort auf `Open in Browser`; das Dokument sollte nun im Browser angezeigt werden.

* Wie wird hier unsere alternative Kodierung angezeigt?

* Gehen Sie einmal im XML-Dokument in die dritte Zeile mit der Angabe des Stylesheets und ersetzen Sie im Name der Datei "EX" durch "AM". Speichern Sie, und laden Sie das Dokument erneut im Browser. Wie wird die alternative Kodierung nun angezeigt?

Das ist das Ende der Übung.