

Notizen zur 6. Sitzung (14.05.2018)

Besprechung der Hausaufgabe

Aufgabenstellung: TEI-Codierung des Texts der Schlusszene von Sandberger, *Der Tod des Kaisers* gemäß Particellskizze von 1923

Formales

Die Textcodierung in den abgegebenen Dateien war zum Teil formal unsauber. Strenggenommen handelt es sich hauptsächlich um Verstöße gegen das TEI-Schema, sodass die Dateien die Schema-Validierung nicht bestehen würden und es bei der maschinellen Weiterverarbeitung Probleme gäbe. Alle diese „Fehler“ können jedoch leicht ausgebessert werden.

Die meisten dieser Fehler lassen sich aufspüren, indem man die Datei gegen das TEI-all-Schema validiert (in *jEdit* über Plugins – XML – Schema setzen... und die Funktion Parsen in der *Sidekick*-Leiste).

Position von übertragenem Text

Der TEI-Standard erlaubt Text nur innerhalb bestimmter Elemente, während andere leer sind oder nur weitere Kindelemente, aber keinen bloßen Text enthalten dürfen. Beispiele aus den abgegebenen Dateien:

- Das `<text>`-Element benötigt Unterelemente, zumindest einen `<body>`. Dieser muss wiederum in `<div>`-Elemente unterteilt sein (zum Beispiel für einzelne Szenen), und erst innerhalb dieser sind die eigentlichen „Textelemente“ wie `<p>` erlaubt. (Die TEI-Rumpfdati enthält daher bereits das Grundgerüst `<text> – <body> – <div> – <p>`.)
- Die `<sp>`-Elemente dienen nur dazu, weitere Elemente wie `<speaker>`, `<p>`, `<stage>` etc. zu einer Redeeinheit zusammenzufassen. Der Text steht nicht innerhalb des `<sp>`-Elements selbst, sondern im jeweils semantisch sinnvollen Unterelement.
Achtung: Gerade wenn ein Sprecher seinen Satz nach einer Regieanweisung fortsetzt, vergisst man gern ein erneutes `<p>` (besonders wenn eigentlich der vorhergehende Absatz fortgesetzt wird).
- Das `<gap>`-Element darf keinen Text enthalten; in der Regel ist es leer. Wollte man definieren, wie die Transkriptionslücke dargestellt werden soll, könnte man ein Attribut wie `rend="curly-brace-and-dots"` setzen, was hier aber nicht nötig ist (siehe unten).

<speaker> vs. <rs>

Will man die Sprechernamen mit einem Element `<rs type="character">` zusätzlich annotieren, benötigt man weiterhin ein `<speaker>`-Element. `<speaker>` zeigt nämlich die semantische Gliederungsebene des Texts an (analog dem `<p>` für Absätze), während `<rs>` eine Zusatzinformation liefert, die erste Angabe aber nicht ersetzt. Das TEI-Schema erlaubt `<rs>` daher nicht direkt unterhalb eines `<div>`-Elements, sondern nur innerhalb der texttragenden Elemente wie `<p>`, `<stage>`, `<speaker>`. Korrekt ist also:

```
<speaker><rs type="character">Stefania</rs></speaker>
```

Inhalt von <spGrp>

Die gleichzeitig gesungenen Ausrufe gegen Ende der Schlusszene können am einfachsten mit `<spGrp type="simultaneous">` codiert werden. Hierbei muss jedoch beachtet werden: Als Kindelemente einer `<spGrp>` („speech group“) sind im Wesentlichen nur `<sp>`-Elemente vorgesehen; der Text selbst muss innerhalb dieser wieder in `<speaker>`, `<p>` etc. gegliedert sein.

XML-Einrückung

Eine inkonsistente Einrückung der XML-Elemente widerspricht zwar nicht dem TEI-Schema, kann aber die Lesbarkeit des codierten Texts erschweren. Als generelle Faustregel bietet es sich an, die Hierarchie der Elemente mit der Einrückung abzubilden und dabei zwischen „texttragenden“ und „nicht-texttragenden“ Elementen zu unterscheiden, zum Beispiel so:

```
<div>
  <sp>
    <speaker><rs type="character">Stefania</rs></speaker>
    <stage>zu Orso</stage>
    <p>Rette meinen Gemahl!</p>
  </sp>
</div>
```

Anfangs- und Endtag nicht-texttragender Elemente stehen in einer eigenen Zeile, während zwischen den Tags der texttragenden Elemente außer dem eigentlichen Text kein überflüssiges Leerzeichen, Tabstop oder Zeilenumbruch steht. Das dem `<div>` untergeordnete `<sp>` sowie die Kindelemente von `<sp>` sind um jeweils einen Tabstop weiter eingerückt. Der Inhalt des `<p>`-Elements ist zwar semantisch der Regieanweisung untergeordnet, steht aber aus syntaktischer Sicht auf derselben Ebene wie `<stage>`; daher keine weitere Einrückung.

Editorische Fragen im Hinblick auf die Codierung

Einige Faustregeln und (teils offene) Fragestellungen, die in der Hausaufgabe auftauchten und auch in anderen digitalen Editionsprojekten von Belang sind:

TEI-Auszeichnungen ersetzen editorische Markierungen im Text

In der Rohtranskription werden bestimmte editorische Phänomene mit zusätzlichen Zeichen markiert, die auf der ersten Seite aufgelistet werden. Eine TEI-Transkription kommt dagegen ohne solchen Zeichen aus, sondern markiert editorische Phänomene mit speziellen XML-Elementen. „Übersetzt“ man nun die Rohtranskription in eine TEI-Codierung, geht die Information der Zeichenfolge { . . . } komplett in

```
<gap reason="illegible" extent="1 word">
```

auf. Ebenso genügt für `furcht[bar{?}]` die Codierung

```
furcht<supplied cert="medium">bar</supplied>
```

(`<supplied>` ist mit dem zusätzlichen `cert`-Attribut versehen, da das Fragezeichen hier andeutet, dass sich der Transkribent unsicher war, ob seine Ergänzung korrekt ist).

Normalisierung: Sprecherbezeichnungen

Sprecherbezeichnungen werden in der Rohtranskription (meist) in Kapitälchen gesetzt und mit Punkt beendet; in der handschriftlichen Originalquelle ist kein genaues System erkennbar. Es kann argumentiert werden, dass der Punkt für die Textsemantik unerheblich ist und außerdem bereits das `<speaker>`-Element zur Kennzeichnung ausreicht. Hier braucht sich die TEI-Transkription somit nicht eng an die Quelle(n) zu halten. Wichtig ist jedoch, das gewählte System beizubehalten (und am besten in Editions- und Codierungsrichtlinien zu dokumentieren). (Die abgegebenen Dateien gehen ebenfalls jeweils anders, aber in sich schlüssig, mit diesem Phänomen um.)

Normalisierung: Regieanweisungen

Eine ähnliche Argumentation kann man für die Klammern in Regieanweisungen führen: Werden diese nicht durch die Einfassung in `<stage>` überflüssig? Hier sticht jedoch ein Sonderfall hervor: Für die Anweisung [Schrei] verwendet Sandberger eckige Klammern. Somit gibt es drei editorische Möglichkeiten:

- Übertragung der runden und eckigen Klammern im Text trotz TEI-Element:

```
<stage>(deutlich)</stage>  
<stage>[Schrei]</stage>
```

- Abbildung der Besonderheit in Attributwerten, eventuell Recherche der Bedeutung der eckigen Klammern bei Sandberger:

```
<stage type="normal">deutlich</stage>  
<stage type="special">Schrei</stage>
```

- Weglassen der Klammern als unwesentliche Information (auch das ist eine editorische Entscheidung):

<stage>deutlich</stage>
<stage>Schrei</stage>

Seitenwechsel

Die Seitenwechsel im Particell können mit <pb>-Elementen codiert werden. Die in der Rohtranskription angegebenen Seitenzahlen haben jedoch keinen besonderen Informationswert, da sie sich auf die Bildnummern der BSB-Digitalisate bezieht. Allerdings kann stattdessen die konsistente Seitenzählung Sandbergers in einem n-Attribut mitgeführt werden. Aus {S. 94} würde so <pb n="74"> werden.

Codierung von Lesarten in TEI

In einer TEI-Datei können Texte aus mehreren Quellen zusammengeführt und Abweichungen zwischen den Quellen dokumentiert werden. Beispielsweise existiert zu Sandbergers Opernfragment *Der Tod des Kaisers* neben der Particellskizze von 1923 auch ein handschriftliches Libretto von 1939/41. Hier kann eine gemeinsame Codierung anzeigen, wo Sandberger nachträglich Änderungen am Gesangstext vorgenommen hat, was wiederum bei der Werkanalyse hilfreich sein kann.

Das bereits bekannte `<choice>`-Element erlaubt, zu einer Textstelle mehrere Transkriptionsalternativen anzubieten. Allerdings erwähnen die *TEI Guidelines* erwähnt, dass es sich dabei um editorische Alternativen zu einer einzigen Quelle handelt: Herausgeberkorrekturen, Abkürzungsaufösungen, unsichere Transkriptionen etc. Für den Vergleich mehrerer Quellen (sogenannter „Textzeugen“) ist dagegen das `<app>`-Element („apparatus“ = textkritischer Apparat) vorgesehen, das an den abweichenden Stellen die verschiedenen Lesarten bündelt.

Zunächst müssen die in der Transkription der TEI-Datei verwendeten Textzeugen deklariert werden. Dies geschieht im TEI-Header durch eine `<listWit>` („list of witnesses“) innerhalb der `<sourceDesc>` („source description“). Eine einfache Textzeugenliste im Sandberger-Beispiel ist etwa:

```
<sourceDesc>
  <listWit>
    <witness xml:id="P">Particellskizze (1923)</witness>
    <witness xml:id="L">Libretto (1939/41)</witness>
  </listWit>
</sourceDesc>
```

Beispiel für eine abweichende Textstelle: In der Particellskizze auf Seite 74 hat Stefania den Text „O furcht[bar{?}] rächt sich was ich tōricht plante“, im Libretto auf Seite 33 dagegen „O, furchtbar rächt sich, was im Haß ich plante“. Die Teile „furchtbar“ und „im Haß ich“ sowie die Stellen mit abweichender Kommasetzung werden – ähnlich wie bei `<choice>` – bei der Codierung in jeweils ein `<rdg>`-Element („reading“, Lesart) eingefasst, und die aufeinander zu beziehenden `<rdg>`-Elemente in einem `<app>`-Element gebündelt. Um zu markieren, welche Lesart in welchem `<rdg>` wiedergegeben ist, erhält das Element ein Attribut `wit` („witness“, Textzeuge) mit einem Verweis auf die in der Textzeugenliste definierte ID. Die Codierung der gesamten Stelle lautet damit:

```

<p>0<app>
  <rdg wit="#P"/>
  <rdg wit="#L">,</rdg>
</app> <app>
  <rdg wit="#P">furcht<supplied cert="medium">bar</supplied></rdg>
  <rdg wit="#L">furchtbar</rdg>
</app> rächt sich<app>
  <rdg wit="#P"/>
  <rdg wit="#L">,</rdg>
</app> was <app>
  <rdg wit="#P">ich tőricht</rdg>
  <rdg wit="#L">im Haß ich</rdg>
</app> plante</p>

```

Beachte:

1. Ein <rdg>-Element kann leeren Text haben, wenn der andere Textzeuge an dieser Stelle zusätzliche Zeichen hat (hier die zusätzlichen Kommata).
2. Sonstige TEI-Elemente wie <supplied> können innerhalb eines <rdg> weiterhin verwendet werden.
3. Da die Zeichen zwischen den <app>- und <rdg>-Tags irrelevant sind, kann man den „Platz“ benutzen, um den XML-Code zur besseren Lesbarkeit einzurücken. Innerhalb von <rdg> und zwischen zwei <app>-Elementen zählt jedoch jedes Zeichen zum transkribierten Text! Korrekte Leerzeichensetzung ist hier besonders wichtig.
4. Beim Verweis auf die Textzeugendefinition nicht das #-Zeichen vergessen.

Projektarbeit Sandberger: *Der Tod des Kaisers*

Das Digitalisat des Librettos zur Schlusszene steht hier bereit:

http://www.musikwissenschaft.uni-muenchen.de/personen/mitarbeiter/leipold/lehre-sole18/material/sandberger_schluss_libretto.zip

Ziel ist es, die Abweichungen zwischen Particellskizze und Libretto wie oben beschrieben in die eigene Teilcodierung einzuarbeiten (Gruppeneinteilung unverändert). Eine Zeichentabelle zum Lesen der Kurrentschrift findet sich beispielsweise unter http://www.kurrent.de/_html/schreibschrift.htm.

Bei Leseschwierigkeiten kann jederzeit eine Transkriptionslücke codiert werden:
 <gap reason="illegible"/>.