

Bastian Blömer (2455180)
Bergische Universität Wuppertal
Modelle der textgenetischen Darstellung
Sommersemester 2025

Computer-Programme und Textgenese

Automatische Erzeugung textgenetischer Darstellungen am Beispiel des Dokumentes H P1 der digitalen Faust-Edition

Inhalt

Einleitung.....	2
Experiment.....	4
Auswahl: Der Weg zu H P1.....	4
Vorbereitung: Präparieren der Codierungen.....	6
Durchführung: Automatische Apparaterzeugung.....	9
Übertragbarkeit: Allgemeine Computer-Programme?.....	14
Reflexion.....	16
Literatur.....	20
Internetquellen.....	20

Einleitung

Im Sommersemester 2025 habe ich an der Bergischen Universität Wuppertal das Seminar *Modelle der textgenetischen Darstellung* besucht. In diesem Seminar haben wir uns vier verschiedene Apparate¹ angeschaut, mit denen man Textgenese darstellen kann. Unser Fokus in dem Seminar lag vor allem auf der mikrogenetischen Ebene und ich werde mich auch in dieser Arbeit ausschließlich darauf konzentrieren. Die vier Apparate waren der Einblendungsapparat, der Einzelstellenapparat, der Treppenapparat und der synoptische Apparat. Wir haben uns die Frage gestellt, welche textgenetischen Phänomene diese Apparate abbilden können und welche nicht. Als wir uns dann verschiedene digitale Editionen angeschaut haben, mussten wir feststellen, dass diese digitalen Editionen bei der Abbildung von Textgenese hinter Bucheditionen zurückzufallen scheinen². In der Geschichte der Buchedition wurde der Treppen- sowie der synoptische Apparat entwickelt, da man mit diesen beiden Apparaten besonders gut komplexe Textgenese abbilden kann. In den digitalen Editionen haben wir allerdings ausschließlich Apparate vorgefunden, die sich am Einblendungs- oder Einzelstellenapparat orientieren. Warum also findet man in digitalen Editionen keine Apparate, die sich am Treppen- oder synoptischen Apparat orientieren?

Eine Antwort auf diese Frage lässt sich an drei verschiedenen Stellen des Editionsprozesses suchen. Man könnte untersuchen, ob die edierten Texte möglicherweise keine komplexen genetischen Phänomene aufweisen und deshalb die Gestaltung eines Treppen- oder synoptischen Apparates gar nicht nützlich gewesen wäre. Man könnte außerdem untersuchen, ob Einblendungs- und Einzelstellenapparate besonders gut zu den technischen Möglichkeiten digitaler Interfaces passen und in digitalen Editionen zu bevorzugen sind. Diesen beiden Fragen werde ich hier allerdings nicht weiter nachgehen. Stattdessen stelle ich mir die Frage, ob es unlösbare Probleme dabei gibt, einen Treppen- und synoptischen Apparat automatisch aus den repräsentierenden Daten einer digitalen Edition zu erzeugen. Die repräsentierenden Daten in digitalen Editionen haben meistens die Form von TEI/XML-Dokumenten³. Solche Dokumente bestehen aus dem Originaltext sowie aus TEI/XML-Elementen, die Eigenschaften dieses Textes wie Durchstreichungen und Hinzufügungen beschreiben. Durch diese Elemente, Elena Pierazzo nennt sie "Explicit Codes"⁴, kann der Text durch Computer-Programme weiterverarbeitet werden. Zu diesen Weiterverarbeitungen gehört, dass der Text automatisch in ein bestimmtes Layout gebracht werden kann, beispielsweise Durchstreichungen und Hinzufügungen tiefgestellt werden und dadurch textgenetische Apparate erzeugt werden. Technisch gesehen werden dafür die TEI/XML-Dokumente in HTML-Dokumente umwandeln. Die HTML-Dokumente können automatisch durch in XSLT geschriebene Computer-Programme erzeugt werden und in ihnen stehen dann die Layout-Anweisungen für den Internetbrowser.

¹ für eine übersichtliche und schematische Darstellung dieser vier Apparate vgl. Scheibe, 1991

² vgl. Nutt-Kofoth, S. 20f., 2019

³ für eine ausführliche Erläuterung dessen, was TEI/XML-Dokumente sind, vgl. Sahle, S. 304ff., 2013

⁴ siehe Pierazzo, S. 307f., 2016

Um der Antwort auf die Frage näher zu kommen, ob es grundsätzliche Probleme bei der automatischen Erzeugung von Treppen- und synoptischen Apparaten gibt, habe ich in dieser Arbeit mit dem Schreiben solcher XSLT-Computer-Programme experimentiert. Ich habe mir die TEI/XML-Dokumente zu einem einzelnen Originaldokument der digitalen Faust-Edition aus dem Internet heruntergeladen, sie für das Experiment präpariert und dann Computer-Programme geschrieben, die solche HTML-Dokumente erzeugen, sodass der Interbrowser digitale Treppen- und synoptische Apparate anzeigt⁵.

In der digitalen Faust-Edition gibt es pro Originaldokument zwei verschiedene Codierungen, das dokumentarische und das textuelle Transkript. Diese beiden Codierungen konzentrieren sich auf verschiedene Aspekte des Originaldokumentes. Ich habe in meinem Experiment beide Codierungen verarbeiten lassen und versucht dadurch Erkenntnisse darüber zu gewinnen, ob sich eine der beiden Codierungen besser für die automatische Erzeugung von Treppen- und synoptischen Apparaten eignet. Hierüber schreibe ich insbesondere in der Reflexion am Ende dieser Arbeit. In meinem Experiment habe ich außerdem nur die Codierungen zu einem einzelnen Originaldokument verarbeitet und die Computer-Programme speziell dafür geschrieben. Ich konnte daher nur eingeschränkt Aussagen darüber machen, ob grundsätzlich auch Computer-Programme möglich gewesen wären, die alle Codierungen der digitalen Faust-Edition wie gewünscht verarbeiten. Hierüber schreibe ich in dieser Arbeit insbesondere in dem Kapitel *Übertragbarkeit*.

Indem ich in meinem Experiment Computer-Programme schreibe, verorte ich mich inmitten des digitalen Wandels, wie Inga Hanna Ralle ihn beschreibt:

“Der Wandel technischer Bedingungen und Möglichkeiten sowie die Digitalisierung von Kulturgut und Forschung gehen mit beträchtlichen Schritten voran [... Djie Beobachtung [ist], dass sich der größte Teil der neu entwickelten Methoden und Werkzeuge auf Daten und Maschinenlesbarkeit konzentriert⁶”

Auch wenn ich versuche, mir die Maschinenlesbarkeit von TEI/XML-Dokumenten zunutze zu machen, sollte aber nicht vergessen werden, dass diese Dokumente immer auch menschenlesbar sind⁷. Es wäre mir nicht möglich gewesen, meine Computer-Programme zu schreiben, hätte ich die TEI/XML-Dokumente nicht lesen und verstehen können.

⁵ für die vollständigen Dokumente zu meinem Experiment, den präparierten XML-Dokumenten, den XSLT-Skripten und den HTML-Dokumenten, siehe das von mir eingerichteten GitHub-Projekt https://github.com/BastianBloemer/Experiment_XSLT-Transformation_Genetischer-Apparat

⁶ siehe Ralle, S. 144, 2019

⁷ vgl. Sahle, S. 35, 2010

Experiment

Auswahl: Der Weg zu *H P1*

Für mein Experiment habe ich mir die TEI/XML-Dokumente zu der Notiz *Prosaentwurf zu Faust*⁸ mit der Sigle *H P1* von der Internetseite der digitalen Faust-Edition heruntergeladen. Im Folgenden möchte ich kurz den Auswahlprozess wiedergeben, der mich zu *H P1* geführt hat.

Zunächst habe ich in dem Editions-Browser *a catalog of Digital Scholarly Editions* nach möglichen digitalen Editionen gesucht, deren TEI/XML-Dokumente für mein Experiment in Frage kommen könnten. Meine Überlegung war es, eine digitale Edition mit genetischem Fokus zu suchen, da in ihren Codierungen Phänomene wie Hinzufügungen und Streichungen beschrieben sein sollten. Die Möglichkeit von genetischen Editionen ist davon abhängig, dass Dokumente mit genetischen Informationen überliefert sind, und da es diese Überlieferungslage vor allem bei moderner Literatur gibt, habe ich in dem Editions-Browser nach eben dieser gefiltert. Um meine Auswahlmöglichkeiten noch weiter einschränken zu können, habe ich außerdem nach deutschsprachiger, moderner Literatur gefiltert. An sich hätte ich aber auch ein Dokument in einer Sprache wählen können, die ich nicht verstehe, da ich in meinem Experiment weniger den Originaltext als vielmehr die Struktur und die Codes in den Codierungen verstehen musste. Den Text selbst habe ich beim Schreiben der Computer-Programme mehr als bedeutungsleere Zeichenketten denn als bedeutungsvollen Text gelesen.

Durch diese ganzen Filterprozesse blieben am Ende nur noch sechs Editionen übrig und von diesen sechs Editionen haben nur zwei die technische Möglichkeit angeboten, die TEI/XML-Dokumente direkt aus dem Internet herunterzuladen. Die übrigen vier Editionen habe ich wegen ihren TEI/XML-Dokumenten angeschrieben. Von der Edition *Wolfgang Koeppen. Jugend* habe ich keine Rückmeldung erhalten. Von der Edition *Thomas Bernhard. Heldenplatz* habe ich eine ausführliche und erhellende Beschreibung ihrer Editionsarbeit erhalten, die TEI/XML-Dokumente durften mir aber aus urheber- und verlagsrechtlichen Gründen nicht herausgegeben werden. Die Edition *Arthur Schnitzler digital* befindet sich noch im Entstehungsprozess und plant in Zukunft ein Herunterladen der TEI/XML-Dokumente aus dem Internet zu ermöglichen. Die Edition *Friedrich Dürrenmatt. Das Stoffe-Projekt* hat mich zurück gefragt, welche Dokumente ich mir wünsche und mir dann die gewünschten Dokumente via E-Mail zugeschickt. Zu diesem Zeitpunkt war mein Experiment allerdings schon beendet.

Die Editionen, die ein Herunterladen ihrer TEI/XML-Dokumente direkt im Internet angeboten haben, waren die digitale Faust-Edition⁹ und die digitale Edition *Theodor Fontane: Notizbücher*¹⁰. Da ich nach jeweils dreißigminütigen Durchklicken durch die Faksimiles der beiden Editionen bei *Theodor Fontane: Notizbücher* keine und bei der digitalen Faust-Edition viele Dokumente mit vielen genetischen Phänomenen gefunden habe, fiel meine Entscheidung auf die Faust-Edition. Durch Zufall bin ich dadurch in die Lage gekommen, wie in der Einleitung schon beschrieben, gleich zwei Codierungen pro Originaldokument zu haben und bei meinem

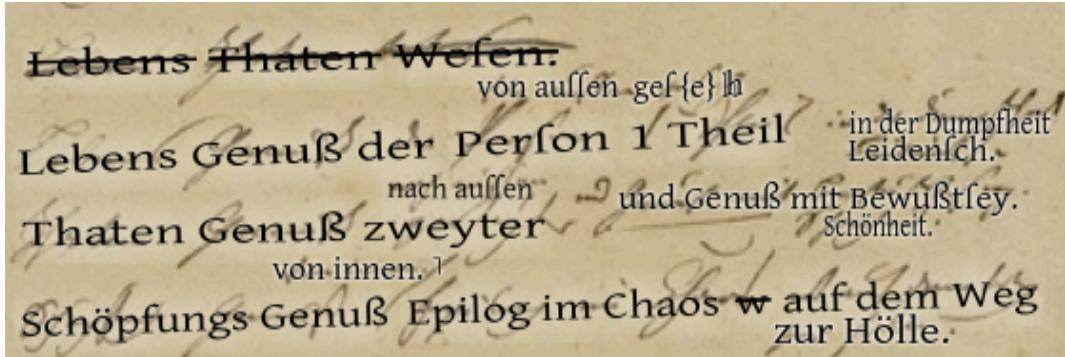
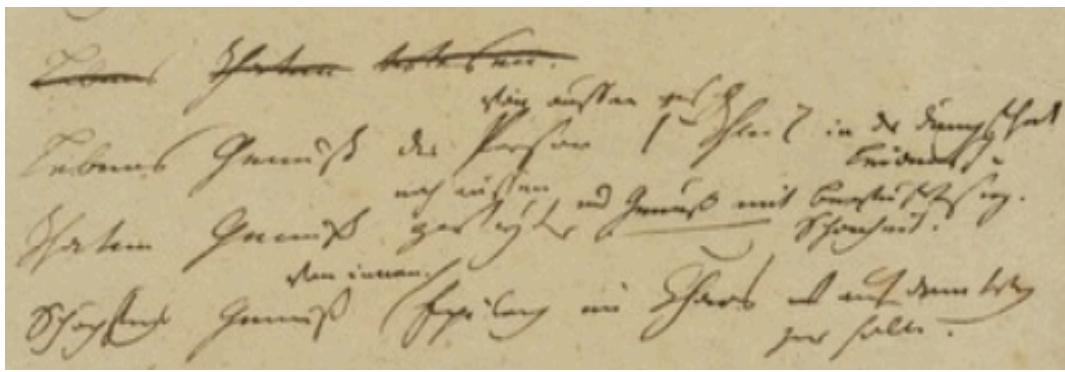
⁸ siehe https://faustedition.net/document?sigil=H_P1

⁹ siehe <https://faustedition.net/>

¹⁰ siehe <https://fontane-online.de/digitale-notizbuchedition-der-ganze-fontane/>

Experiment die Verarbeitung von verschiedenartigen Codierungen miteinander vergleichen zu können.

Bei der Auswahl des konkreten Originaldokumentes innerhalb der digitalen Faust-Edition habe ich dann nach einem Dokument mit hinreichend vielen genetischen Phänomenen gesucht. Ein weiteres Kriterium war, dass beide Codierungsmethoden auch tatsächlich durchgeführt wurden und zur Verfügung standen. Dies war nicht für alle Originaldokumente der Fall. Schließlich habe ich mich für die Codierungen zu dem Dokument *H P1* entschieden und diese Codierungen dann noch auf einen Ausschnitt mit besonders vielen genetischen Phänomenen reduziert. Der überwiegende Teil der genetischen Phänomene in diesem Abschnitt waren spätere Hinzufügungen. Außerdem gab es auch eine Sofortkorrektur und eine spätere Streichung:



11

¹¹ Zu sehen ist hier der für mich relevante Ausschnitt aus dem Faksimile des Dokumentes *H P1*. Oben ist der Ausschnitt ohne und unten mit eingeblendeter dokumentarischer Transkription zu sehen. Für das vollständige Faksimile siehe https://faustedition.net/document?sigil=H_P1&page=1&view=facsimile.

Vorbereitung: Präparieren der Codierungen

Die digitale Faust-Edition stellt zwei Codierungen in Form von TEI/XML-Dokumenten zur Verfügung, das dokumentarische und das textuelle Transkript. Diese beiden verschiedenen Codierungsmethoden möchte ich kurz vorstellen.

Auf der Internetseite der digitalen Faust-Edition – in dem Intro¹² und in den Guidelines¹³ – wird nicht direkt über die Transkripte gesprochen, sondern über die sichtbaren Transkriptionen. Den Nutzer:innen der Edition werden zwei verschiedene Transkriptionen angeboten, die dokumentarische und die textuelle Transkription. Bei der dokumentarischen Transkription handelt es sich um eine diplomatische Transkription. Hier liegt der Fokus auf dem materiellen Dokument und der räumlichen Ordnung des Textes und es wird sichtbar, an welcher Raumposition die verschiedenen Textteile auf dem Papier stehen. Im Vergleich dazu abstrahiert die textuelle Transkription von der räumlichen Ordnung und fokussiert sich auf die Abbildung der chronologischen Ordnung der Textgenese. Die textuelle Transkription funktioniert dabei wie ein Einblendungsapparat:

Lebens Thaten Wesen:
von außen gef{e} [¶]
Lebens Genuß der Person 1 Theil in der Dumpfheit
nach außen und Genuß ^{mit} Bewußtsey.
Thaten Genuß zweyter von innen. Schönheit.
Schöpfungs Genuß Epilog im Chaos ^w auf dem Weg
zur Hölle.

LebensThaten Wesen.(*tilgt*)
Lebens Genuß der Person **(von außen gef{e})**
h erg>**in der Dumpfheit Leidenſch. erg> 1**
Theil
Thaten Genuß **(nach außen und Genuß mit**
Bewußtsey. Schönheit. erg> zweyter [*]**
Schöpfungs Genuß **(von innen. erg> Epilog**
im Chaos <(w > auf dem Weg zur Hölle. erg> 14

Während die linke dokumentarische Transkription aus dem entsprechenden dokumentarischen Transkript erzeugt wurde, wurde die rechte textuelle Transkription aus dem entsprechenden textuellen Transkript erzeugt. Aufgrund dieses Zusammenhangs lässt sich erschließen, welche Informationen in den jeweiligen Transkripten codiert sind. So lassen sich in dem dokumentarischen Transkript beispielsweise Informationen über die Raumpositionen finden, während dies in dem textuellen Transkript nicht der Fall ist. Wie in den hier gezeigten Ausschnitten nicht unmittelbar sichtbar wird, sind in den textuellen Transkripten der digitalen Faust-Edition beispielsweise auch Abkürzungen codiert, was in den dokumentarischen Transkripten nicht der Fall ist¹⁵. Die Unterschiede zwischen den beiden Codierungsmethoden werden in diesem und in den folgenden Kapiteln noch klarer werden. Ob in den textuellen Transkripten auch textgenetische Informationen codiert sind, die sich nicht zumindest auch

¹² siehe <https://faustedition.net/intro>

¹³ siehe https://faustedition.net/transcription_guidelines

¹⁴ Zu sehen ist links ein Ausschnitt aus der dokumentarischen Transkription und rechts der entsprechende Ausschnitt aus der textuellen Transkription. Für die beiden vollständigen Transkriptionen siehe https://faustedition.net/document?sigil=H_P1&page=1&view=document_text.

¹⁵ für eine ausführlichere Beschreibung der beiden Transkripte siehe Brüning, Henzel und Pravida, Abs. 3f u. Abs. 20f., 2013

implizit in den dokumentarischen Transkripten finden lassen, hat sich mir im Laufe meiner Arbeit aber nicht erschlossen. Damit die Computer-Programme für mein Experiment nicht unnötig komplex werden und nur genetisch relevante Informationen verarbeiten müssen, habe ich die originalen Transkripte präpariert. Im Folgenden möchte ich lediglich die Präparation des textuellen Transkripts beschreiben. Die Präparation des dokumentarischen Transkripts funktionierte nach den gleichen Prinzipien.

Zunächst habe das Transkript auf die Repräsentation des mich interessierenden Textausschnittes reduziert:

```
<p>
    <del f:revType="soon">Lebens< milestone unit="refline" n="pl_8"/>Thaten Wefen.</del>
</p>
<!-- Nach Morris, S. 156 eine Sofortrevision, Ersetzung durch das folgende. -->
<p>Lebens Genuß der Perfon <add f:revType="late">von auffen gef<unclear cert="high"
    >e</unclear>h</add>< milestone unit="refline" n="pl_9"/><add f:revType="late">
    >in der Dumpfheit <abbr>Leidenfch.</abbr></add> 1 Theil</p>
<p>Thaten Genuß <add f:revType="late">nach auffen < milestone unit="refline" n="pl_10"
    />und Genuß mit Bewußtfey. Schönheit.</add> zweyter <space/></p>
<p>Schöpfungs < milestone unit="refline" n="pl_11"/>Genuß <add f:revType="late">von
    innen.</add> Epilog im Chaos <add><del f:revType="instant">w</del> auf dem Weg
    zur Hölle.</add></p>
```

16

In diesem Ausschnitt gibt es noch vier Elementtypen¹⁷, die für mein Experiment keine Rolle spielten und die ich daher gelöscht habe:

- das abbr-Element, mit dem Abkürzungen codiert werden
- das unclear-Element, mit dem Unsicherheiten in den Transkriptionen codiert werden
- das space-Element, mit dem bedeutsame Leerräume codiert werden
- das milestone-Element, mit dem Grenzpunkte codiert werden

Die Funktion von textgenetischen Apparaten ist es, Textstadien voneinander zu unterscheiden und in eine genetische Ordnung zu bringen. Da Abkürzungen im Originaltext und Unsicherheiten beim Transkribieren in diesen Apparaten keine unmittelbare Rolle spielen, mussten meine Computer-Programme diese Informationen auch nicht verarbeiten und ich konnte sie aus der Codierung löschen. Bei den bedeutsamen Leerräumen hätte es sich anders verhalten können, da sie an sich verschiedene Textstadien voneinander trennen können. Dies schien in diesem Ausschnitt aber nicht der Fall zu sein und auch bei einem Vergleich mit dem Faksimile hat sich mir nicht erschlossen, warum nach zweyter ein Leerraum codiert ist und beispielsweise nach 1 Theil nicht. Auch die Regeln hinter der Codierung der Grenzpunkte hat sich mir anhand des Dokumentes *H P1* nicht vollständig erschlossen. Die Grenzpunkte befanden sich zwischen verschiedenen Hinzufügungen wie zwischen *auffen gef(e)h* und *in der Dumpfheit Leidenfch*, aber auch zwischen Textteilen, die als Teile einer einzelnen Hinzufügung interpretiert wurden. Letztendlich schienen mir weder die Leerräume noch die Grenzpunkte in

¹⁶ Zu sehen ist ein Ausschnitt aus dem Originaltranskript. Der hier in Grün erscheinende Teil ist ein Kommentar. Dieser weist darauf hin, dass eine Textstelle, die die Editor:innen als eine späte Streichung codierten, von Morris als eine "Sofortrevision" interpretiert wurde.

¹⁷ für eine genauere Beschreibung der offiziellen Richtlinien für den Gebrauch der Elemente siehe <https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html>.

diesem Ausschnitt textgenetischen Informationen zu geben, die nicht sowieso schon in der Codierung enthalten sind. Ich habe sie daher ebenso gelöscht und folgendes Präparat für mein Experiment erstellt:

```

<p>
    <del revType="soon">Lebens Thaten Wefen.</del>
</p>
<p>Lebens Genuß der Perfon
    <add revType="late">von auffen gefeh</add>
    <add revType="late"> in der Dumpfheit</add>
    Leidenſch. 1 Theil
</p>
<p>Thaten Genuß
    <add revType="late">nach auffen und Genuß mit Bewußtfey. Schönheit.</add>
    zweyter
</p>
<p>Schöpfungs Genuß
    <add revType="late">von innen.</add>
    Epilog im Chaos
    <add>
        <del revType="instant">w</del>
        auf dem Weg zur Hölle.</add>
    </p>

```

18

Wie zu sehen ist, wurde der gesamte Textausschnitt durch vier p-Elemente in Paragraphen unterteilt. Diese vier Paragraphen enthielten verschiedene Textstadien wie beispielsweise die Grundschichten und den ihnen zugeordneten Hinzufügungen. Während die Grundschichten nicht weiter codiert sind, wurden die Hinzufügungen durch das add-Element codiert und die Streichungen durch das del-Element. Diese beiden Textmanipulationen wurden durch das revType-Attribut näher bestimmt und beispielsweise als späte Hinzufügungen und baldige Streichungen codiert. Außerdem wurde durch die Kombination aus einem add-Element und einem del-Element mit dem revType-Wert *instant*, der Textteil *w auf dem Weg zur Hölle* als eine Sofortkorrektur codiert¹⁸. Außerdem ist auffällig, dass die Textteile *nach auffen* und *und Genuß mit Bewußtfey. Schönheit* als eine einzige Hinzufügung codiert wurden, obwohl es im Faksimile eher zwei voneinander getrennte Textteile zu sein scheinen.

Die von mir geschriebenen Computer-Programme mussten schließlich nur noch folgende Elemente verarbeiten können:

- p-Elemente, mit denen Paragraphen codiert werden*
- add-Elemente, mit denen Hinzufügungen codiert werden*
- del-Elemente, mit denen Streichungen codiert werden*
- Verschachtelte add- und del-Elemente, mit denen Sofortkorrekturen codiert werden*

¹⁸ Zu sehen ist die von mir präparierte Codierung. Aus technischen Gründen enthieilt das vollständige Präparat noch einrahmende text- und xml-Elemente, die hier nicht mit abgebildet sind.

¹⁹ Bei der späteren Überarbeitung dieser Arbeit fiel mir auf, dass dieser Textteil an sich auch als eine Hinzufügung codiert worden sein könnte, in der dann eine Soforttilgung stattfindet. Wäre dies der Fall, wäre die Hinzufügung aber wahrscheinlich ebenso wie die anderen Hinzufügungen direkt mit einem Attribut näher bestimmt worden.

Durchführung: Automatische Apparaterzeugung

Nach der Präparierung des dokumentarischen und textuellen Transkripts habe ich mich an das Schreiben der XSLT-Computer-Programme gemacht, um automatisch digitale Treppen- und synoptische Apparate zu erzeugen. Aus zeitökonomischen Gründen habe ich mich dafür entschieden, die Apparate so einfach wie möglich zu gestalten und beispielsweise die Raumpositionen von Textteilen nicht abzubilden. Auch ohne es ausprobiert zu haben, lässt sich aber aufgrund der Informationen in den verschiedenen Codierungen sagen, dass die Abbildung von Raumpositionen nur durch eine Verarbeitung des dokumentarischen Transkripts möglich gewesen wäre. Das textuelle Transkript hätte die dafür notwendigen Informationen nicht enthalten.

Zunächst habe ich das textuelle Transkript verarbeitet und folgende Apparate erzeugen lassen:

1. **1. Lebens Thaten Wesen.**
- 2.
2. **1. Lebens Genuss der Person**
 2. Lebens Genuss der Person von außen geseh
 3. Lebens Genuss der Person von außen geseh in der Dumpfheit Leidensch. 1 Theil
3. **1. Thaten Genuss**
 2. Thaten Genuss nach außen und Genuss mit Bewußtsey. Schönheit. zweyter
4. **1. Schöpfungs Genuss**
 2. Schöpfungs Genuss von innen. Epilog im Chaos **1. w**
 2. auf dem Weg zur Hölle.

1. [Lebens Thaten Wesen.]

2. **Lebens Genuss der Person** **Leidensch. 1 Theil**
 von außen geseh in der Dumpfheit
3. **Thaten Genuss** **zweyter**
 nach außen und Genuss mit Bewußtsey. Schönheit.
4. **Schöpfungs Genuss** **Epilog im Chaos [w] auf dem Weg zur Hölle.**
 von innen.

20

Zu dem Treppenappart ist zu sagen, dass hier beispielsweise im zweiten Paragraph an erster Stelle (2.1.) nur *Lebens Genuss der Person* steht, obwohl die Grundschicht des Originaltextes *Lebens Genuss der Person 1 Theil* ist. Ich habe mich aus zwei Gründen für diese Umsetzung entschieden.

Zunächst einmal war es leichter, ein Computer-Programm zu schreiben, dass in der ersten Zeile des zweiten Paragraphen den Text zunächst einmal bis zur ersten Hinzufügung ausgab und den letzten Teil der Grundschicht schließlich erst am Ende nach allen Hinzufügungen. Ein wichtiger Grund war aber, dass mir diese Apparatgestaltung am ehesten einem teleologischen Modell von Textgenese entsprach. Ich werde in meiner Reflexion noch

²⁰ Zu sehen ist oben der Treppenapparat und unten der synoptische Apparat, wie sie von meinem Internet-Browser (Firefox) angezeigt wurden.

weiter darauf eingehen, dass dem ursprünglichen Treppenapparat ein teleologisches Genesemodell und dem ursprünglichen synoptischen Apparat ein chronologisches Genesemodell zugrunde lag. Durch meine Umsetzung konnte ich diese beiden verschiedenen Genesemodelle vergleichen und mir Fragen darüber stellen, inwiefern diese Modelle die Computer-Programme beeinflussen und inwiefern sie schon in den Codierungen enthalten waren. Im Folgenden möchte ich aber zunächst in Teilen beschreiben, wie die Computer-Programme für die Verarbeitung des textuellen Transkriptes funktioniert haben.

Das Computer-Programm für den Treppenapparat hat im Wesentlichen Listen ineinander verschachtelt und automatisch nummeriert. Zunächst hat es die vollständige Codierung gelesen und für jeden der vier Paragraphen eine Listenzeile erzeugt (1.-4.). Dann fand die erste Verschachtelung statt, indem das Programm für jeden dieser Paragraphen eine weitere, untergeordnete Listenzeile (1.1., 2.1., 3.1., 4.1.) erzeugt und den Textinhalt bis zur ersten Hinzufügung ausgegeben hat. Da der erste Paragraph keine Hinzufügung enthält, hat es den vollständigen Text ausgegeben. Nach der Ausgabe der ersten untergeordneten Listenzeilen hat das Programm etwas gemacht, das ich als Beispiel dafür anführen möchte, dass mein Programm sehr auf die Verarbeitung der Codierung von *H P1* spezialisiert war:

```
<xsl:for-each select="del[@revType = 'soon']">
  <li/>
</xsl:for-each>
```

21

Wie der Ausschnitt zeigt, hat das Computer-Programm nun für jeden Paragraphen geprüft, ob er unmittelbar²¹ eine Streichung mit dem Attribut *soon* enthält, und wenn dies der Fall ist eine nächste leere Listenzeile erzeugt (1.2.). Bei der Verarbeitung des textuellen Transkripts hat das Programm für den vierten Paragraphen keine leere Listenzeile erzeugt, da die Streichung sich hier nicht unmittelbar in dem Paragraphen sondern in der Hinzufügung befindet. Wenn aber beispielsweise in einem ersten Paragraphen einer anderen Codierung nicht der ganze Text, sondern nur ein Wort durchgestrichen wäre, hätte das Programm ebenfalls die leere Listenzeile 1.2. erzeugt, obwohl dies nicht den Regeln des Treppenapparates entsprochen hätte. Dies wäre also ein beispielhafter Fall, der die Spezialisierung des Programmes auf *H P1* anschaulich gemacht hätte. Eine weitere spezialisierte Verarbeitungsregel des Programmes ist die Erzeugung der Listenzeile 4.2. Als allerletzten Verarbeitungsschritt prüft das Programm, ob ein Paragraph eine Sofortkorrektur enthält. Ist dies der Fall, erzeugt es eine Listenzeile mit dem Text bis zur Sofortkorrektur (4.2.) und dann eine Extraliste (4.2.1. - 4.2.2.), die diese Sofortkorrektur abbildet. Dies erzeugt bei *H P1* nur deshalb einen richtigen Treppenapparat, da hier der vierte Paragraph mit der Sofortkorrektur endet. Ich könnte hier noch weitere Verarbeitungsschritte anführen, die bei anderen Codierungen mit anderen genetischen Phänomenen zu falschen Ergebnissen führen würden, werde es aber einfacheitshalber bei diesen zwei beispielhaften Fällen belassen. Stattdessen möchte ich nun dazu übergehen, die Funktionsweise des Programmes für die Erzeugung des synoptischen Apparates zu beschreiben.

²¹ Zu sehen ist ein Ausschnitt aus dem XSLT-Skript *text_trans_treppenapparat_HP1.xsl*.

²² In dem XSLT-Diskurs wird zu dieser Unmittelbarkeit gesagt, dass das Streichungselement ein Kind des Paragraphen-Elementes ist.

Die Erzeugung des synoptischen Apparates funktionierte im Wesentlichen so, dass das Programm einzelne Textteile klassifizierte und diese in der Ausgabe dann tiefer stellt, wobei die Stärke der Tiefstellung von der jeweiligen Klasse abhängig ist. Schrittweise beschrieben funktioniert dies so, dass das Programm zunächst für jeden Paragraph eine Listenzeile (1. - 4.) mit dem vollständigen Textinhalt erzeugt hat. Dann wurden Streichungen, Hinzufügungen und Sofortkorrekturen klassifiziert, wobei die Klassifizierung teilweise aus dem textuellen Transkript übernommen werden kann, in dem diese genetischen Phänomene codiert sind. Teilweise musste das Programm die Klassen aber auch weiter ausdifferenzieren und zwischen erster und zweiter Hinzufügung unterscheiden. Das Programm musste also zählen:

```

<xsl:variable name="index"
  select="count($p//add[not(.//del[@revType = 'instant'])])
  [. &lt;&lt; current()] + 1"/>
<span class="add-{$index}-soon">
  <xsl:apply-templates/>
</span>
```

23

Der Ausschnitt oben zeigt, dass das Programm zählte, um die wievielte Hinzufügung in einem Paragraphen es sich bei einer Hinzufügung handelt. Bei der Klassifizierung wird der gezählte Wert der jeweiligen Hinzufügung zugeordnet, sodass Klassennamen wie *add-[Zahlenwert]-soon* entstehen. Die späteren Tiefstellungen der Hinzufügungen fanden dann abhängig von diesem Zahlenwert statt und spätere Hinzufügungen wurden stärker tiefergestellt als frühere Hinzufügungen.

Nachdem ich mit der Verarbeitung des textuellen Transkripts experimentiert hatte, habe ich Computer-Programme für die Verarbeitung des dokumentarischen Transkripts geschrieben. Da sich die Struktur des dokumentarischen Transkripts wesentlich von der Struktur des textuellen Transkripts unterschied, mussten auch die jeweiligen Programme wesentlich anders funktionieren. Bei dem Vergleich der Programme möchte ich mich auf die Programme für die Erzeugung des synoptischen Apparates konzentrieren und darin einführen, indem ich zunächst die jeweils zweiten Zeilen dieser Apparate einander gegenüberstelle:

2. Lebens Genuss der **Person 1 Theil**
von außen gefeh

2. Lebens Genuss der Person	Leidensch. 1 Theil
von außen gefeh	
	in der Dumpfheit

24

Die Unterschiede in den zweiten Zeilen der synoptischen Apparate sind, dass in der oberen Zeile, die ich aus dem dokumentarischen Transkript erzeugt habe, sowohl die Hinzufügung *in der Dumpfheit* als auch das Wort *Leidensch.* aus der Grundschicht fehlt und dort außerdem die Hinzufügung *von außen gefeh* vor *Person* steht. Nun möchte ich zeigen, wie diese Unterschiede

²³ Zu sehen ist ein Auszug aus dem XSLT-Skript *text_trans_syntoptischerapparat_HP1.xsl*.

²⁴ Zu sehen sind jeweils die zweite Zeile der synoptischen Apparate. Die obere Zeile wurde aus dem dokumentarischen Transkript erzeugt, die untere Zeile aus dem textuellen Transkript.

zustande gekommen sind, indem ich die Struktur der beiden Transkripte und die Funktionsweise der beiden Programme einander gegenüberstelle.

In dem textuellen Transkript waren die verschiedenen Textteile, die in dem synoptischen Apparat in einer Zeile stehen, in einem einzelnen Paragraphen enthalten. Der Paragraph für die zweite Zeile enthielt hier also sowohl die Grundschicht als auch die Hinzufügungen von *auffen gefeh* und *in der Dumpfheit*. Außerdem standen diese verschiedenen Textteile in dem Transkript auch schon in derselben Reihenfolge, in der sie am Ende im synoptischen Apparat erschienen. Die Editor:innen, die das Transkript erstellt haben, haben hier also schon im Transkript eine horizontale Ordnung erzeugt und das Programm hat die zweite Zeile in eben dieser Ordnung einfach ausgegeben. Ich hätte zwar ein Programm schreiben können, das die verschiedenen Textteile in eine andere horizontale Ordnung bringt als die vorgefundene, dies wäre aber nur dann begründet gewesen, wenn ich der vorangegangenen Arbeit der Editor:innen widersprochen hätte. Im Vergleich zum textuellen Transkript scheint im dokumentarischen Transkript auf den ersten Blick noch keine horizontale Ordnung vorgegeben zu sein:

```

<zone type="main">
  <line>
    <mod rend="strikethrough">Lebens</mod>
    <mod rend="strikethrough">Thaten</mod>
    <mod rend="strikethrough">Wefen.</mod>
  </line>
  <line left="#Person" type="inter">von auffen gefeh </line>
  <line xml:id="Lebens-Theil">Lebens Genuß der <anchor xml:id="Person"/> Perfon 1 Theil</line>
  <line type="inter"/>
  <line left="#zweyter" type="inter" xml:id="nach-aussen">nach auffen</line>
  <line xml:id="Thaten-Theil">Thaten Genuß <anchor xml:id="zweyter"/> zweyter </line>
  <line type="inter">
    <ins at="#Epilog" orient="left">von innen.</ins>
  </line>
  <line>Schöpfungs Genuß <anchor xml:id="Epilog"/> Epilog im Chaos <mod rend="strikethrough">
    <w/></mod> auf dem Weg </line>
    <line rend="indent-70">zur Hölle.</line>
  </zone>
  <zone left-right="#Lebens-Theil" top="#Lebens-Theil">
    <line type="inter">in der Dumpfheit</line>
    <line rend="indent-10" type="inter"> Leidenfch. </line>
  </zone>
  <zone left-right="#nach-aussen" top="#nach-aussen">
    <line type="inter"> und Genuß mit Bewußtfey. </line>
    <line left-right="#Thaten-Theil" type="inter">Schönheit.</line>
  </zone>

```

25

Das dokumentarische Transkript war in eine Hauptzone und in zwei Extrazonen eingeteilt. Die Hauptzone enthielt sowohl normale Zeilen als auch Zwischenzeilen und die Extrazonen enthielten ausschließlich Zwischenzeilen. Die Durchstreichungen sind explizit als Durchstreichungen codiert, während die Hinzufügungen nicht explizit codiert zu sein scheinen. Stattdessen sind die Hinzufügungen implizit dadurch codiert, dass die Zwischenzeilen hier Hinzufügungen andeuten und mittels Identifikatoren (beispielsweise *xml:id="Person"*) und Referenzierungen (beispielsweise *left="#Person"*) bestimmten Stellen in den normalen Zeilen

²⁵ Zu sehen ist die von mir präparierte Codierung des dokumentarischen Transkripts. Aus technischen Gründen enthält die vollständige Codierung noch umklammernde xml-Elemente, die hier aber keine weitere Rolle spielen.

zugeordnet werden²⁶. Um die horizontale Ordnung der einzelnen Zeilen des synoptischen Apparates zu erzeugen, musste sich das Programm die einzelnen Textteile zusammensuchen und an ihre zugewiesene Stelle setzen:

```

<xsl:template match="anchor">
  <xsl:variable name="anchor-id" select="concat('#', @xml:id)" />
  <xsl:variable name="line-refs" select="//line[@type = 'inter'] | //ins" />
  <xsl:variable name="zone-refs" select="//zone[not(@type = 'main')]" />
  <xsl:for-each select="$line-refs[@* = $anchor-id]">
    <xsl:variable name="line-refs-id" select="concat('#', @xml:id)" />
    <span class="add-1">
      <xsl:apply-templates select="node()" />
    </span>
    <xsl:if test="$zone-refs[@* = $line-refs-id]">
      <span class="add-2">
        <xsl:value-of select="$zone-refs[@* = $line-refs-id]" />
      </span>
    </xsl:if>
  </xsl:for-each>
</xsl:template>

```

27

Der Ausschnitt oben zeigt, wie dieses Zusammensuchen funktionierte. Das Programm sucht in den normalen Zeilen nach Ankerpunkten. Diese Ankerpunkten haben eine Identifizierung und das Programm sucht nach darauf refenzierenden Zwischenzeilen und klassifiziert sie für die spätere Tiefstellung als *add-1*. Dann sucht es nach weiteren einer weiteren Zwischenzeile, die auf die erste referenziert und klassifiziert diese für die Tiefstellung als *add-2*.

Ich habe oben geschrieben, dass im dokumentarischen Transkript auf den ersten Blick keine horizontale Ordnung vorgegeben sei, da sich dieser Eindruck auf den zweiten Blick revidiert. Auch wurden die Stellen der Textteile durch Identifizierung und Referenzierung eindeutig bestimmt. Für die zweite Zeile des synoptischen Apparates hat das Programm also in der normalen Zeile *Genuß der Person 1 Theil* vor *Person* einen Ankerpunkt gefunden und die erste Hinzufügung hier eingesetzt. Die beiden Transkripte widersprechen sich hier also in ihrer Zuordnung, da die erste Hinzufügung im textuellen Transkript nach *Person* stand. Im dokumentarischen Transkript fand das Programm in der normalen Zeile kein *Leidenſch.*, wodurch der zweite Unterschied in den Apparaten entstand. Das Faksimile von *HP1* zeigt hier eindeutig, dass *Leidenſch.* nicht zur Grundsicht gehört und die Zuordnung zur Grundsicht im textuellen Transkript scheint an dieser Stelle fehlerhaft zu sein. Dass das Programm den Textteil *in der Dumpfheit* im dokumentarischen Transkript gar nicht erst verarbeitet hat, ist dagegen in meiner eigenen Programmierung begründet. Diese Hinzufügung wird der vollständigen normalen Zeile zugeordnet und nicht einer bestimmten Stelle innerhalb dieser Zeile und dies schien mir fehlerhaft. Für eine Auflösung dieses Fehlers hätte ich sehr spezielle Funktionen programmieren müssen und dies würde dem eigentlichen Ziel, ein möglichst allgemeines Programm zu schreiben, widersprechen. Auf die Möglichkeit von allgemeinen Programmen, die mehr als eine Codierung verarbeiten und richtige Treppen- und synoptische Apparate erzeugen können, werde ich im folgenden Kapitel eingehen.

²⁶ Es gibt zwar ein ins-Element, dass explizit eine Hinzufügung markiert, dies ist aber nicht der Normalfall und ich konnte mir nicht erschließen, warum von *innen* explizit als Hinzufügung codiert worden ist und die anderen Hinzufügungen nicht.

²⁷ Zu sehen ist ein Ausschnitt aus dem XSLT-Skript *dok_trans_synoptischerapparat_HP1.xsl*.

Übertragbarkeit: Allgemeine Computer-Programme?

Wie oben angedeutet, ist für die Brauchbarkeit eines Computer-Programmes, das automatisch digitale genetische Apparate erzeugt, eine hinreichende Allgemeinheit notwendig. Damit ist gemeint, dass es nicht auf ein oder ein paar Codierungen spezialisiert ist, sondern eine größere Menge an Codierungen verarbeiten kann. In einem wirklichen Editionsprojekt verbietet es sich aus arbeitsökonomischen Gründen spezielle Programme zu schreiben, da die manuelle Erzeugung digitaler genetischer Apparate im Vergleich dazu wesentlich effektiver wäre. Ich möchte auf zwei Bedingungen eingehen, die für die Möglichkeit einer automatischen Verarbeitung einer größeren Menge an Codierungen notwendig sind.

Die erste Bedingung betrifft nicht die Computer-Programme sondern die zu verarbeitenden Codierungen. Damit eine breite Verarbeitung stattfinden kann, ist es notwendig, dass diese Codierungen denselben Richtlinien folgen. Hiermit ist gemeint, dass die Regeln für den Gebrauch von Elementen, Attributen und Werten die gleichen sind. Für die Erfüllung dieser Bedingung arbeitet die *Text Encoding Initiative (TEI)*²⁸ an Codierungs-Standards:

"XML-Kodierungen wie , <deletion>, <streichung> [...] sind für den Menschen gleichbedeutend, nicht jedoch für eine Software. Seit 1987 arbeite[n] deshalb in der Text Encoding Initiative [...] Geisteswissenschaftlerinnen daran, Kodierungsstandards auch für die editorische Arbeit vorzuschlagen."²⁹

Die TEI-Initiative stellt im Internet sogenannte Schemata³⁰ bereit, die mit TEI/XML-Dokumenten so verlinkt werden³¹, dass es bei einer Verletzung der Gebrauchsregeln zu Fehlermeldungen kommt. Schemata und Fehlermeldungen bedeuten aber nicht, dass keine Fehler mehr möglich sind. Die Schemata regeln nur Codierungsintern beispielsweise welche Elemente welche Attribute haben dürfen. Die Beziehung zwischen Codierung und externem Originaldokument, ob die Codierung am Ende auch wirklich das Originaldokument repräsentiert, ob sich beispielsweise eine Hinzufügung auch wirklich an der in der Codierung angegeben Stelle befindet, wird dagegen nicht geregelt.

Die zweite Bedingung für eine breite Verarbeitung von Codierungen ist die Allgemeinheit des Computer-Programms. Das Programm muss so gestaltet sein, dass es mit den Besonderheiten einer großen Menge an Codierungen umgehen kann. Ich habe oben schon darauf hingewiesen, dass die von mir geschriebenen Programme eher spezifisch sind und beispielsweise mit Sofortkorrekturen am Ende eines Paragraphen umgehen können, nicht aber mit Sofortkorrekturen am Anfang oder in der Mitte eines Paragraphen. Bei solchen Phänomenen würde der digitale genetische Apparat am Ende falsch gestaltet sein. Aus den Ergebnissen

²⁸ für eine ausführliche Erläuterung dessen, was die TEI ist, vgl. Sahle, S. 304ff., 2013

²⁹ siehe Vogeler, S. 125, 2019

³⁰ siehe für eine Liste der Schemata <https://www.tei-c.org/release/xml/tei/custom/schema/xsd/>

³¹ Die digitale Faust-Edition hat mit einem modifizierten TEI-Schema gearbeitet. Die Transkripte der digitalen Faust-Edition sind durch das im sogenannten XML-Prolog befindliche `href="http://faustedition.net/schema/1.0/faust-tei.rng"` mit diesem Schema verlinkt. Für eine Erläuterung der Gründe, warum die Editor:innen das TEI-Schema für ihre genetische Codierung erweitern mussten, siehe Brüning, Henzel und Pravida, Abs. 16f., 2013

meines Experiments kann ich daher noch nicht den Schluss ziehen, dass ich ein allgemeines Programm schreiben könnte, dass alle dokumentarischen und textuellen Transkripte wie gewünscht verarbeiten kann. Die dokumentarische und die textuelle Transkription der digitalen Faust-Edition wurden zwar ebenfalls durch Programme erzeugt, sie konnten aber aus Codierungen erzeugt werden, die extra für diesen Zweck geschrieben wurden. Auch hieraus kann ich daher nicht den Schluss ziehen, dass meine eigenen Programme hinreichend allgemein werden können, da ich die Codierungen quasi zweckentfremde. Letztendlich kann ich also nur meine Meinung formulieren.

Meiner Meinung nach sollten Computer-Programme, die aus den Codierungen der digitalen Faust-Edition digitale Treppen- und synoptische Apparate erzeugen, grundsätzlich möglich sein. Der Grund dafür ist, dass die Gestaltung von Treppen- und synoptischen Apparaten nach Regeln funktioniert, die an sich formalisierbar sein sollten:

"Die wissenschaftliche Edition kann nur die Durchführung eines kritischen Prozesses sein, der im besten Fall in so klare Regeln gefasst ist, dass er auch von einem Computer durchgeführt werden könnte."³²

Würde ich mein Experiment fortführen und das Schreiben von allgemeinen Computer-Programmen versuchen, würden sich mir im Voraus zwei Fragen aufdrängen. Die eine Frage ist, ob sich dieses Ziel besser durch deduktives oder durch induktives Programmieren erreichen lässt. Beim induktiven Programmieren würde man erst ein Programm für eine einzelne Codierung schreiben. Dann würde man das Programm auf eine nächste Codierung anwenden, die auftretenden Fehler beheben und diesen Prozess solange fortführen, bis keine Fehler mehr auftreten. Beim deduktiven Programmieren würde man sich dagegen zunächst das gebrauchte Codierungsschema anschauen, sein Programm passend zu diesem Schema schreiben und demnach versuchen, direkt ein Programm für alle Codierungen zu schreiben.

Die zweite Frage wäre, ob sich bei der digitalen Faust-Edition das dokumentarische oder das textuelle Transkript besser für die automatische Erzeugung von digitalen Treppen- und synoptischen Apparaten eignet. Ich hatte den Eindruck, dass das Computer-Programm, das ich für die Verarbeitung dieses Transkripts geschrieben habe, von Anfang an allgemein war. Ein Grund dafür könnte sein, dass die Zuordnung der verschiedenen Textteile eines Paragraphen auf der horizontalen Achse dort über die oben beschriebene Identifizierung und Referenzierung funktioniert. Dieses Prinzip könnte sich für allgemeine Programme besser eignen als das Prinzip, dass diese Textteile schon in der Codierung in ihrer horizontalen Ordnung stehen. Solche Zusammenhänge näher zu erforschen wäre eine interessante Fortführung meines Experiments.

³² siehe Sahle, Abs. 9.7, 2005

Reflexion

Im vorangegangen Kapitel habe ich geschrieben, dass ich die automatische Erzeugung von digitalen Treppen- und synoptischen Apparaten für möglich halte. Ich habe ich meinem Experiment keine grundsätzlichen Probleme erkannt, die erklären könnten, warum diese Apparate in digitalen genetischen Editionen nicht umgesetzt werden. Im Folgenden möchte ich darüber reflektieren, welche Codierungsmethode der digitalen Faust-Edition sich besser für die automatische Apparaterzeugung eignete und welche Informationen Codierungen für diese Zwecke mindestens enthalten müssten.

In der digitalen Faust-Edition gab es das dokumentarische und das textuelle Transkript und an sich konnte ich in meinem Experiment beide Transkripte wie gewünscht verarbeiten. Wie oben schon gesagt, ließen sich aber nur aus dem dokumentarischen Transkript genetische Apparate mit Angabe der Raumpositionen erzeugen, weil nur in diesem Transkript die dafür notwendigen Informationen enthalten sind. Aber aus welchem Transkript ließen sich die von mir gewählten einfachen Apparate leichter erzeugen und welches Transkript ist daher anwendungsneutraler? Die beiden Transkripte wurden in der digitalen Faust-Edition spezifisch für die dokumentarische und textuelle Transkription erstellt und es würde für ihre Anwendungsneutralität sprechen, wenn auch andere Abbildungen leicht aus ihnen erzeugt werden könnten. Inga Hanna Ralle schreibt über Anwendungsneutralität:

“[Es wird] festgestellt, dass dem traditionellen Konzept von Edition – die Edition als Vorschlag [und Vermittlung] – als Gegenpol das Konzept von Edition als Datenbank entgegensteht [...] Das sogenannte ‚anwendungsneutrale Format‘, d.h. keine Festlegung auf eine Präsentationsform, wird als Vorteil formuliert [...]”³³

Ich selbst habe in meinem Experiment aber keinen Widerspruch zwischen anwendungsneutralen Codierungen und den Vermittlungszielen von Editionen erkannt. Im Gegenteil dazu glaube ich sogar, dass gerade eine anwendungsneutrale Codierung eine größere Menge an Vermittlungen und Abbildungen ermöglichen könnte. Die Vermittlungsarbeit innerhalb von Editionsprojekten würde dann eher im Schreiben von Computer-Programmen liegen, die aus den Codierungen Abbildungen erzeugen. Hinsichtlich dem dokumentarischen und dem textuellen Transkript könnte das dokumentarische Transkript anwendungsneutraler gewesen sein, da es wegen seiner Struktur und seinen Elementen befundsnäher³⁴ zu sein schien. Auch die Editor:innen der Faust-Edition beschreiben das dokumentarische Transkript als editorischen Befund und das textuelle Transkript als editorische Deutung³⁵. Mein Eindruck der stärkeren Befundsnähe entstand zunächst dadurch, dass das dokumentarische Transkript die einzelnen Textteile weniger als das textuelle Transkript in eine zeitliche und horizontale Ordnung zu bringen und stattdessen vielmehr die bloße räumliche Ordnung des Dokumentes zu repräsentieren schien. Es lässt sich allgemein sagen, dass die zeitliche und horizontale Ordnung von Textteilen erst dadurch entsteht, dass Editor:innen den räumlichen Befund deuten.

³³ siehe Ralle, S. 146f., 2019

³⁴ für eine ausführliche Unterscheidung zwischen Befund und Deutung im Kontext von editorischem Handeln siehe Zeller, S. 45ff, 1971

³⁵ vgl. Brüning, Henzel und Pravida, Abs. 18, 2013

Ich möchte hier der Bewertung des dokumentarischen Transkripts als editorischen Befund aber nicht uneingeschränkt zustimmen, da es explizite genetische Informationen enthält, die notwendigerweise Ergebnisse von Deutungen sind, auch wenn diese Deutungen nicht wirklich angezweifelt werden können. So sind beispielsweise die Zonen in Haupt- und Nebenzonen sowie die Zeilen in normale Zeilen und Zwischenzeilen ausdifferenziert. Dadurch werden chronologische Aussagen entweder schon gemacht oder zumindest sehr nahegelegt. Außerdem gibt es durch Identifikatoren und Referenzierungen schon eine horizontale Zuordnung der einzelnen Textteile, die über die bloße Angabe der Raumposition hinausgeht. Ich bewerte hier so pedantisch, da ich das Experiment interessant gefunden hätte, aus einer Codierung mit ausschließlich räumlichen Informationen einen genetischen Apparat zu erzeugen. Ein Computer-Programm zu schreiben, dem diese Verarbeitung gelingt, würde einen dazu zwingen viele der Annahmen über Schreibkonventionen zu explizieren, die das Erschließen von Textgenese erst ermöglichen³⁶. Zu diesen Annahmen gehört beispielsweise, dass ein Textteil, der zwischen zwei weiteren Textteilen steht und eine wesentlich kleinere Schriftgröße hat als diese, als eine Hinzufügung wahrgenommen wird. Da ich diese Explizierungsprozesse beim Schreiben von Programmen für wissenschaftlich wertvoll halte, erschließt sich mir die Kritik von Gerrit Brünning an allzu befundsnahen Codierungen nicht ganz:

"Die Beschränkung auf [befundsnaher Repräsentation] ist im digitalen Raum [...] alles andere als ein Ausweis methodischer Reflektiertheit – im Gegenteil: Bestimmte Entscheidungen, denen der Editor aus dem Weg geht, werden bei der späteren maschinellen Verarbeitung in der einen oder anderen Weise automatisch nach mechanischen Regeln getroffen"³⁷

Brünning übt seine Kritik an dem Beispiel, dass eine Beschränkung auf befundsnaher Codierung das gute Funktionieren von Volltextsuche verhindern würde. Auf diese Funktion möchte ich hier nicht weiter eingehen und ich habe darüber zu wenig Wissen, um Brünning hier zu widersprechen.

Zum Schluss möchte ich nochmal auf die von mir geschriebenen Computer-Programme und die Abhängigkeit ihrer Funktionsweise von den Codierungen und den genetischen Apparaten eingehen. Um die genetischen Apparate zu erzeugen, mussten die Programme horizontale und vertikale Ordnungen herstellen. In der horizontalen Ordnung haben die Textteile eine syntagmatische Stelle und in der vertikalen Ordnung eine paradigmatische Stelle. Dies möchte ich kurz am Beispiel des zweiten Paragraphen für die Herstellung der beiden digitalen genetischen Apparate näher erklären. In diesem Paragraphen lautet das letzte Textstadium *Lebens Genuß von außen gefeh in der Dumpfheit Leidenfch. Perlon 1 Theil*, wobei in der Grundsicht die späteren Hinzufügungen von *außen gefeh* und *in der Dumpfheit Leidenfch.* noch nicht vorhanden sind.

Da diese syntagmatischen Positionen in der Grundsicht leer sind, gibt es oben in der horizontalen Achse des synoptischen Apparates Leerstellen, die eben dies abbilden. Die Hinzufügung von *außen gefeh* nimmt nun beispielsweise dieselbe syntagmatische Position ein

³⁶ für eine nähere Erläuterung von logischen Prämissen bei der Erschließung von Textgenese siehe Lukas, S. 29ff., 2019

³⁷ siehe Brünning, S. 89, 2019

wie die erste Leerstelle. Sie ist eine paradigmatische Variante zu dieser Leerstelle und dies bildet der Apparat dadurch ab, dass diese Hinzufügung weiter unten auf derselben vertikalen Achse wie die erste Leerstelle steht.

Der Treppenapparat verhält sich anders zu der Abbildung von syntagmatischen und paradigmatischen Positionen. Auf der oberen horizontalen Achse steht hier beispielsweise die Grundsicht bis zur ersten Hinzufügung. Auf der vertikalen Achse eine Stufe darunter steht dann derselbe Teil der Grundsicht und zusätzlich die erste Hinzufügung. Der Treppenapparat bildet also anders als der synoptische Apparat keine Leerstellen ab und er zeigt Textgenese als ein Wachstum. Wolfgang Lukas schreibt hierzu folgendes:

“Es ist [...] ein Akt der gezielten Modellbildung, und je nach dem zugrundegelegten Konzept von Genese – z. B. ob sie teleologisch als „ideales Wachstum“ (sensu Friedrich Beißner) [Treppenapparat] gedacht wird oder aber rein chronologisch als Transformationsprozess verschiedener Systemzustände (sensu Hans Zeller) [synoptischer Apparat] – ist das Ergebnis ein anderes”³⁸

In der von mir erzeugten einfachen Form wurde der Treppenapparat außerdem dafür kritisiert, dass er weniger befundsgenau sei als der synoptische Apparat³⁹ und er zeigt beispielsweise nicht die vollständige Grundsicht des zweiten Paragraphen, sondern nur bis zur ersten Hinzufügung. Meine Computer-Programme mussten daher aus den zwei Transkripten zwei Apparate mit verschiedenen horizontalen und vertikalen Ordnungen herstellen, was letztendlich zu folgenden Funktionsweisen geführt hat:

Apparat \ Transkript	Textuell (Deutung)	Dokumentarisch (Befund)
Treppen (teleologisch)	Horizontale Ordnung: Ausschneiden Vertikale Ordnung: Testen	Horizontale Ordnung: Verketten Vertikale Ordnung: Testen
Synopse (chronologisch)	Vertikale Ordnung: Klassifizieren	Horizontale u. Vertikale Ordnung: Klassifizierende Verkettung

Die Tabelle zeigt, dass die Programme für den Treppenapparat die horizontale und vertikale Ordnung durch zwei voneinander getrennte Funktionen herstellen mussten. Die vertikalen Ordnungen wurden dabei dadurch hergestellt, dass die Programme bei jeder gefundenen Hinzufügung eine neue Listenzeile erzeugt haben. Die jeweiligen horizontalen Ordnungen der einzelnen Listenzeilen wurden dabei entweder durch das Ausschneiden oder durch das Verketten von Textteilen erzeugt. Im Gegensatz zu den Treppenapparaten wurden die synoptischen Apparate nicht durch getrennte Funktionen erzeugt. Das Programm, das das textuelle Transkript verarbeitet hat, konnte ohne Extrafunktion die jeweiligen Paragraphen aus dem Transkript ausgeben und musste daher gar keine horizontale Ordnung herstellen. Die Tiefstellungen der vertikalen Ordnung konnte es durch Klassifizierung erzeugen. Das Programm, das das dokumentarische Transkript verarbeitet hat, musste dagegen auch die horizontale Ordnung herstellen und benötigte daher eine etwas komplexere Funktion. Es konnte

³⁸ siehe Lukas, S. 23, 2019.

³⁹ vgl. Nutt-Kofoth, S. 16, 2019

die horizontale und vertikale Ordnung letztendlich trotzdem durch eine einzelne Funktion erzeugen, indem es die Textteile beim Verketten direkt für die Tiefstellungen klassifiziert hat.

Ich finde es schwer zu erkennen, ob möglicherweise Aussagen über die verschiedenen Apparate aus den eben beschriebenen Funktionsweisen abgeleitet werden können. Dies liegt auch daran, dass mir unklar ist, inwiefern auch anders funktionierende Computer-Programme dieselben Ergebnisse hätten erzeugen können. Hinsichtlich meiner eigenen Programme ist aber auffällig, dass die Ordnungen des Treppenapparates aus zwei getrennten Funktionen erzeugt werden mussten, während die Ordnungen des synoptischen Apparates aus einer Funktion erzeugt werden konnten. Das wirft zumindest Fragen darüber auf, wie diese verschiedenen Funktionsweisen begründet sind. Sie könnten beispielsweise dadurch begründet sein, dass die horizontalen und vertikalen Ordnungen der Treppenapparate an sich weniger eng miteinander verknüpft sind als die Ordnungen der synoptischen Apparate. Sie könnten außerdem noch dadurch begründet sein, dass die Ordnungen des Treppenapparates weniger befundsnahe sind und dieser Apparat daher weiter von den Transkripten entfernt ist, die die Programme verarbeiten mussten. Ich selbst habe an dieser Stelle weder die Zeit noch den Raum, um diesen Zusammenhängen weiter nachzugehen. Ich hoffe aber Fragen aufgeworfen zu haben, die interessant genug sind, dass andere oder ich selbst zu einem späteren Zeitpunkt nach ihren Antworten suchen werden.

Literatur

Brüning Gerrit; Was ist und wozu codiert man Textgenese?; in: Bosse und Fanta (Hrsg.); Textgenese in der digitalen Edition; De Gruyter, 2019

Brüning Gerrit, Henzel Katrin und Pravida Dietmar ; Multiple Encoding in Genetic Editions: The Case of "Faust"; <https://doi.org/10.4000/jtei.697> (Publiziert 2013)

Lukas Wolfgang; Archiv - Text - Zeit. Überlegungen zur Modellierung und Visualisierung von Textgenese im analogen und digitalen Medium; in: Bosse und Fanta (Hrsg.); Textgenese in der digitalen Edition; De Gruyter, 2019

Nutt-Kofoth Rüdiger; Textgenese analog und digital: Ziele, Standards, Probleme; in: Bosse und Fanta (Hrsg.); Textgenese in der digitalen Edition; De Gruyter, 2019

Pierazzo Elena; Textuel Scholarship and Text Encoding; in: Schreibma, Siemens und Unsworth (Hrsg.); A New Companion to Digital Humanities; Wiley & Sons, 2016

Ralle Inga Hanna, Maschinenlesbar - menschenlesbar. Über die grundlegende Ausrichtung der Edition; in: Nutt-Kofoth und Plachta (Hrsg.); EDITIO Band 30; De Gruyter, 2016

Vogeler Georg; Digitale Editionspraxis. Vom pluralistischen Textbegriff zur pluralistischen Softwarelösung; in: Bosse und Fanta (Hrsg.); Textgenese in der digitalen Edition; De Gruyter, 2019

Sahle Patrick; Digitale Editionsformen Teil 3: Textbegriffe und Recodierung; BoD; 2013

Sahle Patrick; Zwischen Mediengebundenheit und Transmedialisierung; in: Nutt-Kofoth und Plachta (Hrsg.); EDITIO Band 24; De Gruyter, 2010

Sahle Patrick; Digitales Archiv - Digitale Edition. Anmerkungen zur Begriffsklärung; https://www.germanistik.ch/publikation.php?id=Digitales_Archiv_und_digitale_Edition (Publiziert Oktober 2005)

Scheibe Siegfried; Editorische Grundmodelle; in: Scheibe und Laufer (Hrsg.); Zu Werk und Text. Beiträge zur Textologie; Akademie-Verlag; 1991

Zeller Hans; Befund und Deutung. Interpretation und Dokumentation als Ziel und Methode der Edition; in: Martens und Zeller (Hrsg.); Texte und Varianten; C.H. Beck; 1971

Internetquellen

Editions-Broser; <https://www.digitale-edition.de/exist/apps/editions-browser/index.html>⁴⁰

Faust-Edition; <https://faustedition.net/>

GitHub; https://github.com/BastianBloemer/Experiment_XSLT-Transformation_Genetischer-Apparat

Text-Encoding-Initiative; <https://tei-c.org/>

Theodor Fontane: Notizbücher: <https://fontane-online.de/digitale-notizbuchedition-der-ganze-fontane/>

⁴⁰ Diese und die folgenden Internetquellen habe ich zuletzt am 04.09.2025 aufgerufen.