Integration von Textsammlungen in die DTA-Infrastruktur an der BBAW. Eine Schritt-für-Schritt-Anleitung am Beispiel der „Digitalen Bibliothek“ (CLARIAH-DE, AP1)

Version 1.0

Matthias Boenig, Marius Hug

31.3.2021

Inhaltsverzeichnis

1 Einleitung

Ein Text-Repositorium ist immer nur so gut wie die darin befindlichen Korpora. Ganz ohne Zweifel profitieren die jeweiligen Textsammlungen von Neueinspielungen und Erweiterungen des Bestandes. Der Mehrwert ist jedoch umso größer, wenn dabei etablierte Standards beachtet werden und die Interoperabilität von Datenbeständen sichergestellt wird.

Anfang 2019 schlossen sich die Forschungsinfrastrukturen CLARIN-D und DARIAH-DE zu CLARIAH-DE zusammen. Innerhalb des vom BMBF geförderten Projekts CLARIAH-DE ist das Arbeitspaket „Forschungsdaten, Standards und Verfahren“ im Rahmen der Zusammenführung und Homogenisierung digitaler Datenbestände u. a. zuständig für die Konvertierung ausgewählter TEI-Ressourcen des TextGrid Repositories sowie deren Integration in die Infrastruktur des Deutschen Textarchivs (DTA). Die Verständigung auf gemeinsame Standards und Verfahren für die Erstellung, Aufbereitung und Archivierung von Daten und Werkzeugen soll den Forschenden im Bereich der Geistes- und Sozialwissenschaften zugute kommen. Das Ziel besteht darin, substantielle Teile der „Digitalen Bibliothek“ in das Basisformat des Deutschen Textarchivs (DTABf) zu transformieren. Durch diese Transformation der „Digitalen Bibliothek“ wird die Erschließung des TextGrid-Bestandes zugleich auch durch die innerhalb der vom DTA bzw. von CLARIN-D entwickelten Werkzeuge ermöglicht. Nicht zuletzt profitiert die (wissenschaftliche) Community von der umfassenden Metadatenanreicherung des Datensatzes.

Die folgende Schritt-für-Schritt-Anleitung dokumentiert die Harmonisierung der Daten im Detail. Dazu gehört neben der umfänglichen Anreicherung der Metadaten auch ein Blick auf die Eigenheiten des Textbestandes.[1] Für eine Konversion von Daten in diesem Umfang kann festgestellt werden, dass es kein generisches Konzept geben kann, stattdessen sind viele spezifische Entscheidungen und textsortenabhängige Anpassungen notwendig und vorzunehmen. Ziel der vorliegenden Anleitung ist es, die einzelnen Schritte so konkret wie nötig und so allgemein wie möglich zu beschreiben. Abschließend werden zusammenfassend ein paar allgemeine Schlussfolgerungen formuliert.

2 Evaluation der „Digitalen Bibliothek“ im TextGrid Repository

Grundstock des TextGrid Repositories bildet die ursprünglich von Zeno.org publizierte „Digitale Bibliothek“,[2] genauer: Der Literatur-Ordner der Digitalen Bibliothek. Dieser Grundstock liegt in drei verschiedenen TextGrid-Versionen vor (2011, 2012, 2017). Evaluiert wurden die Versionen von 2012 und 2017. Die Versionen von 2011 (Version I) und 2012 (Version II) werden zum Download bereitgestellt, wobei darin gemäß der Zeno.org-Vorlage jedem Autor eine XML-Datei entspricht. Die Online-Version von TextGrid entspricht der Version II. Da für diese Version im Gegensatz zu den im zip-Ordner enthaltenen zum

1 Einen guten inhaltlichen Einstieg zur „Digitalen Bibliothek“ bietet auch der Blogartikel: „Geschichte der Digitalen Bibliothek“. URL: <https://sprache.hypotheses.org/?p=2436>.

2 https://textgridrep.org/?lang=de.

Download bereitgestellten Dateien die komplexe TEI-Corpus-Mantelung aufgelöst wurde, liegen die Werke in knapp 100.000 einzelnen TEI-Dateien aufgesplittet vor, wobei jede Datei eine PID enthält.[3] Bei einer solchen Datei kann es sich im Falle eines Romans z. B. um ein ganzes Buch handeln, im Falle eines Gedichtbandes aber auch um einen Vierzeiler.

2.1 Statistische Auswertungen/Evaluierung

2.1.1 Gesamtkorpus Version II (2012)

Diese Version entspricht den aktuell online angezeigten Daten.[4] Für die Online-Version wurden die TEI-Corpus-Dateien in Einzel-Dateien aufgelöst und mit PIDs versehen. Evaluiert wurde das zum Download angebotene Gesamtkorpus, das skriptbasiert in einzelne TEI-Dateien gesplittet und zur besseren Prozessierbarkeit nach einheitlichem Schema umbenannt wurde (Sonderzeichen in Dateinamen wurden bspw. entfernt).

Voraussetzung wissenschaftlicher Arbeit ist die Zitierbarkeit und die Vollständigkeit der edierten Texte. Aus diesem Grund werden die Qualität der Metadaten sowie die Umsetzung der ursprünglich Zeno-XML-formatierten Texte in TEI betrachtet. Um einen Einblick in die Qualität der Metadaten zu bekommen, wurden die TEI-Header aller Dateien mittels XPath-Abfragen evaluiert. Die Ergebnisse dieser Auswertung werden hier tabellarisch wiedergegeben und anschließend interpretiert:

Nr	Einheit	Anzahl
1	Dateien gesamt (TEI-Corpus)	698
2	Text-Dateien extrahiert aus TEI-Corpus	106675
3	Autor_innen gesamt	690
3a	Autor_innen mit pnd	639
3b	fileDesc/titleStmt ohne author	89920
4a	Verwendete Zeno.org-Quellen: `<BOOKCITE>`	9
4b	Verwendete Zeno.org-Quellen: `<BOOKDESCR>`	1547
4c	milestone[@unit="sigel"]	1667
5	`<sourceDesc>` leer	610
6a	`<sourceDesc>` ohne `<title>`	16738
6b	`<sourceDesc>` enthält 1 `<title>`	89937
6c	`<sourceDesc>` enthält 2 `<title>`	47
7	Dateien mit leerem date	36777
8	Dateien mit `<publisher>`	0

3 Leider gehen die PIDs (aus technischen Gründen) verloren, wenn das zum Download angebotene Gesamtkorpus heruntergeladen wird.

4 Die Version I von 2011, die ebenfalls zum Download angeboten wird, wurde nicht evaluiert.

Interpretation der Ergebnisse

Ad 1) Der Ordner enthält eine Datei pro Autor mit einer Ausnahme. Zu J.W.v. Goethe gibt es vier Dateien.

Ad 2) Siehe dazu unten die Anmerkungen zur Version von 2017.

Ad 3) Die Autoren der Texte wurden in der Regel – entgegen der Empfehlung durch die Text Encoding Initiative (TEI) – nur in die `<sourceDesc>` mit aufgenommen und fehlen im `<titleStmt>` der `<fileDesc>`. Die Differenz zu den im Ordner enthaltenen Dateien – siehe 1) – erklärt sich u.a. durch die in vier Dateien aufgeteilten Texte von Goethe sowie zwei Dateien zu Winckelmann und Nietzsche, die allerdings leer sind.

Ad 4) Nachdem die Metadaten-Katalogdatei von Zeno.org rund 2500 Einträge enthält, und hier die Zählung der Quellennachweise distinkt erfolgte, fehlen in den Daten also Verweise auf rund 1000 Quellen. Zwar werden über das `<milestone>`-Element rund 100 weitere Datensätze referenziert, eine Differenz von etwa 900 Werken bleibt aber auch hier bestehen, wobei sich beinahe die Hälfte davon auf Texte von Karl May bezieht.

Ad 5) Dabei handelt es sich um die Biographien (tg2.item.xml).

Ad 6) In der Regel wird in der Version von 2012 nur ein `<title>`-Element verwendet. Dabei handelt es sich um das `<BOOKDESCR>`-Element aus dem Zeno.org-Katalog. Die Zuordnung zur Bandangabe geht damit in den davon betroffenen Werken verloren. Rund 15 % der Dateien enthalten überhaupt keinen `<title>` in der `<sourceDec>`.

Ad 7) In rund einem Drittel der Dateien fehlt die Datumsangabe per `<date>`.

Ad 8) Die Verlage wurden nicht annotiert.

2.1.2 Nicht veröffentlichte (verbesserte Text-) Version von 2017

Nr	Einheit	Anzahl
1	Dateien gesamt	566719
2	Text-Dateien (item.xml)	91826
3	Autor_innen gesamt	690
3a	Autor_innen mit pnd (ohne)	588 (306)
4a	Verwendete Zeno.org-Quellen: `<BOOKCITE>`	1847
4b	Verwendete Zeno.org-Quellen: `<BOOKDESCR>`	1531
4c	milestone[@unit="sigel"]	0
5a	`<sourceDesc>` leer	500
5b	sourceDesc/p enthält den string "Biographie"	611
6	`<sourceDesc>` enthält 1 `<title>` (gesamt)	18059
6a	`<sourceDesc>` enthält 1 `<title>` mit dem string "kein Title"	17979
6b	`<sourceDesc>` enthält 1 fehlerhaftes `<title>`	80
7	`<sourceDesc>` enthält 2 `<title>`, 1) `<BOOKCITE>`, 2) `<BOOKDESCR>`	72658
8a	Dateien mit leerem `<date>`	7157
8b	Dateien mit `<date>` "0000"	19306
9a	Leere `<pubPlace>`	24181
9b	Fehlerhafte `<pubPlace>`	ca. 4900

Interpretation der Ergebnisse

Ad 1) Es gibt verschiedene XML-Typen: aggregation, edition, item, work. Der Ordnerstruktur ist zunächst nicht anzusehen, welche XML-Dateien enthalten sind. Die eigentlichen Texte sind in den item-XMLs zu finden. Siehe dazu auch das Metadata Cheatsheet des Textgrid Repository: https://wiki.de.dariah.eu/download/attachments/12189756/Metadata-Cheatsheet.pdf?api=v2

Ad 2) Vgl. dazu die 106675 Dateien in der 2012er Version. In der Version von 2017 sind nur zwei Goethe-Ordner mit insgesamt 1703 TEI-Dateien enthalten. In der 2012er Version befinden sich im Ordner mit den gesplitteten Dateien insgesamt 16551 Goethe-Texte. Höchst wahrscheinlich handelt es sich bei den nicht enthaltenen Dateien also um einen Großteil des Goethe-Korpus.

Ad 3) Die Autoren entsprechen zwar teilweise denjenigen mit pnd, allerdings wurde ihnen kein *Identifier* zugeordnet.

Ad 4) title[1] entspricht dem von Zeno.org verwendeten `<BOOKCITE>`, als title[2] wurde das Element `<BOOKDESCR>` aus der Zeno-Katalogdatei übernommen. Während in den online bereitgestellten Daten noch `<milestone unit="sigel">` für die Zuordnung zu den Zeno.org-Metadaten verwendet wird, s.o., wird in der Version von 2017 darauf verzichtet.

Ad 6b) Dabei handelt es sich in 74 Fällen um Texte von Annette von Droste-Hülshoff, in

denen nur <BOOKDESCR> übernommen wurde. In 6 anderen Fällen (betroffen sind Texte von Hugo Ball, Paul Scheerbart und Johann Gottfried Herder) enthält dieses <title> den nicht aufgelösten Zeno.org-Identifier als Wert des @book-Attributs im Element <sigel>.

Ad 8b) Fehlerhafte <pubPlace>s sind bspw.: “a.M.”, “u. a.”, “a.d.S.”, “[um” etc.

Zusammenfassung:

- Ca. 20 % der Dateien enthalten keine Quelle (18059 von insg. 91824),
- knapp 30 % der Texte sind ohne Datum (26436 von insg. 91824),
- und über 30 % ohne Erscheinungsort (29080 von insg. 91824).

2.2 Das Problem der Kategorisierung

Neben der einfachen und erweiterten Suche über den kompletten Bestand bietet das TextGrid Repository für folgende Kategorien entsprechende Suchfilter an:[5]

Genre	Anzahl
verse	118060
other	58792
prose	6663
drama	1461
=> gesamt	184976

Aus den Zeno.org-Metadaten lassen sich jedoch deutlich feiner spezifizierte Textgattungen extrahieren. Hier die Top 20:

Genre	Anzahl (gesamt: 91826)
Gedichte	53746
Sagen	12175
Märchen und Sagen	9724
Briefe	1798
Werke	1767
Lyrik	1575
Werk	1311
Gesamtausgabe der Werke	1197
Märchen	1155
Erzählungen	1055
Biographie	610

5 Die Genreübersicht findet sich im Menü „Content“, siehe https://textgridrep.org/facet/work.genre. Da diese Filterfunktion jedoch zunächst alle im Repository enthaltenen Korpora berücksichtigt, muss im Anschluss an die Auswahl eines bestimmten Genres die Suche noch explizit auf den Bestand aus der „Digitalen Bibliothek“ eingeschränkt werden.

Genre	Anzahl (gesamt: 91826)
Prosa	541
Tagebücher	524
Fabeln	486
Dramen	450
Romane	384
Gedichte und Prosa	291
Lyrik und Prosa	169
Liedsammlung	169
Poetische Werke	159

Gleichzeitig gibt es aber auch eine ganze Reihe von „Gattungs“-Zuordnungen, die in der kompletten Textsammlung „Literatur-Ordner der Digitalen Bibliothek“ nur einmal verwendet werden, das sind z. B. (Auszug):

- Aphorismen und Fragmente
- Autobiographische Roman-Trilogie
- Dichtung
- Flugblatt
- Lehrgedicht
- Lyrisches Drama
- Novellenzyklus
- Predigtliteratur
- Puppenspiel
- Reisebilder
- Romanze
- Satirischer Traktat
- Schauspiel
- Sinnsprüche
- Tragödie

Ohne dass dokumentiert wäre, worin bspw. der Unterschied zwischen „Werk“ und „Werke“ oder zwischen „Dichtung“ und „Gedicht“ besteht, liegt es doch nahe, die Anzahl der Gattungen zu beschränken (s. dazu die Ausführungen im folgenden Kapitel). Bei der hier gezeigten Aufstellung erscheint eine Reduktion auf nur vier Kategorien allerdings unterkomplex.

3 Anreicherung der Metadaten

Die Anreicherung der Metadaten umfasst die Korrektur, Nachrecherche und tiefere Annotation der bibliographischen Metadaten einerseits, andererseits die Anreicherung über den Abgleich mit weiteren Datensätzen wie GerDraCor oder das i5-Korpus des IDS, mit den Zeno.org-permalinks und bspw. den Autoren-pnds, die im Projekt TextGrid integriert wurden. Die Anreicherung wurde im Bereich der Metadaten mit Hilfe von in Python programmierten Skripten sowie durch den Einsatz von dafür angefertigten XSLT-Skripten realisiert.

3.1 Manuelle Anreicherung

Nach eingehender Evaluierung der TextGrid-TEI-Header im Abgleich mit den zugrunde liegenden bibliographischen Metadaten wurde festgestellt, dass die Erstellung von DTABf-konformen TEI-Headern eine zentrale Herausforderung darstellt und nicht ohne manuelle Anreicherung bewerkstelligt werden kann. Die vergleichsweise strengen Regeln für die Generierung der DTABf-Header haben sich bewährt und der Mehrwert für die wissenschaftliche Nutzung der Daten durch die Community rechtfertigt den Aufwand. In einem umfangreichen Bearbeitungsschritt wurden rund 2500 bibliographische Datensätze manuell angereichert und nach gängigem Muster annotiert.

In den Headern der TextGrid-Dateien werden die Quellen innerhalb der `<sourceDesc>` in einem `biblFull/titleStmt/title` nachgewiesen. Im Optimalfall enthält jeder Datensatz zwei solcher `<title>`. Diese wiederum wurden der bibliographischen Katalogdatei entnommen, die folgender Systematik entspricht:

```
<DEFBOOK>
    <BOOKNAME>
        Byron-Werke Bd. 4
    </BOOKNAME>
    <BOOKDESCR>
        Lord Byrons Werke. 6 Bände in dreien, übers. v. Otto Gildemeister,
        Berlin: Verlag von G. Reimer, 1877.
    </BOOKDESCR>
    <BOOKCITE>
        Lord Byrons Werke. Berlin 1877, Band 4
    </BOOKCITE>
 ...
</DEFBOOK>
```

In die TextGrid-Header wurden demnach die Inhalte von `<BOOKDESCR>` und `<BOOKCITE>` übernommen. Für den DTA-Header eigneten sich weder das Eine noch das Andere im Kontext von `<biblFull>`. Dort werden vielmehr explizit annotierte Elemente benötigt, das sind in der Regel der Autor, evtl. ein Herausgeber, ein Titel, Verlag, Verlagsort und das Erscheinungsjahr.

Da <BOOKCITE> im Gegensatz zu <BOOKDESCR> die zentralen bibliographischen Informationen in einer besser strukturierten Form enthält, war die semiautomatische Anreicherung von <BOOKCITE> naheliegend. Die dabei nicht berücksichtigten Angaben wie Herausgeber oder Verlag wurden in einem nachgelagerten Arbeitsschritt eingepflegt.

Das Ergebnis entspricht folgendem Datensatz:

```
<bibl bnid="Byron-Werke Bd. 4">
    <persName role="author">
        <roleName>Lord</roleName>
        <surname>Byron</surname>
    </persName>
    <persName role="translator">
        <forname>Otto</forename>
        <surname>Gildemeister</surname>
    </persName>
    <title level="m" type="main">Lord Byrons Werke</title>
    <title level="m" type="volume" n="4">Band 4</title>
    <pubPlace>Berlin</pubPlace>
    <publisher>Verlag von G. Reimer</publisher>
    <date>1877</date>
</bibl>
```

Den Personen wurde über das @role-Attribut eine Funktion zugewiesen (author, editor, composer, translator). Die Titel wurden ebenfalls tiefenstrukturiert, einerseits wurde per @level-Attribut die Textart deklariert, andererseits wurde per @type-Attribut zwischen Haupt-, Untertitel und Bandangabe unterschieden. Orte wurden wie üblich per <pubPlace>, die Verlage per <publisher> und Datumsangaben per <date> ausgezeichnet. Lücken in den Metadaten (s.o.) wurden nachrecherchiert und konnten in den allermeisten Fällen geschlossen werden. Was sich für den oben gezeigten Eintrag vergleichsweise einfach gestaltet, machte in vielen anderen Fällen jedoch sehr viel Handarbeit und Recherche nötig.

3.2 Skript-basierte Anreicherungen der Katalogdatei

Im Anschluss wurden diese Daten aus externen Datensätzen weiter angereichert. Besonders große Anstrengungen wurden in einer Zusammenarbeit mit dem »Zentrum für digitale Lexikographie der deutschen Sprache« (ZDL) im Bereich der Datumsangaben unternommen, die u. a. für die linguistische und historische Forschung wichtig sind. Zur Erinnerung: Rund 30 % der TextGrid-Headern wurden keine <date>s zugeordnet. Ein Grund dafür ist im zugrunde liegenden Metadatensatz zu finden, in dem es viele fehlende Jahresangaben (o. J.) gibt. Diese wurden nun, wenn immer möglich, nachrecherchiert. Andererseits entsprechen vorhandene Jahresangaben der Textausgaben in vielen Fällen nicht dem Datum der Erstausgabe. Um möglichst vielen Werken neben dem vorhandenen Erscheinungsjahr auch das Datum der Erstveröffentlichung mitzugeben, hat sich das Pro-

jektteam auf ein automatisches Verfahren verständigt, mittels dessen verschiedene externe Datensätze abgefragt und evtl. Übereinstimmungen in die Metadaten der DTABf-Header mit eingespielt werden.

Konkret wurde im Bereich der Dramen auf die angereicherten Metadaten des Projekts GerDraCor zurückgegriffen. Außerdem wurde das i5-Korpus des IDS berücksichtigt, die Anreicherungen vom TextGrid-Projekt per @notBefore und @notAfter (wobei es sich hier in vielen Fällen um eine Annäherung an die Entstehungsdaten des Werkes mittels Lebensdaten der Autoren handelt) sowie das ebenfalls in den TextGrid-Headern abgebildete und aus den Zeno-Daten übernommene `<note>`-Element, das in vielen Fällen Angaben zum Entstehungszeitraum und der Ersterscheinung des Werks enthält. Die so aus verschiedenen Ressourcen gewonnenen Datumsangaben wurden manuell evaluiert mit dem Ziel, ein eigenes, projektspezifisches `<date type="firstPublication">` zu generieren, welches das in den verschiedenen Datensätzen früheste genannte Jahr enthält.

Den Autoren wurden per PND eindeutige Identifier zugewiesen. Ein Großteil der Personen-IDs entstammt den TextGrid-Daten, alle anderen wurden nachrecherchiert und skriptbasiert in die Daten integriert.

Außerdem wurden sowohl die Orte wie auch die Herausgeber harmonisiert. Aus einer Reihe von Varianten des o.g. Verlags wurde so bspw. immer der dann normalisierte Datensatz:

```
<publisher>Verlag Georg Reimer</publisher>
```

Die automatische Anreicherung der bibliographischen Metadaten erfolgt über ein Python-Skript:

```
python3 update_bibliography.py
```

3.3 Korrekturen/Änderungen an den Originaldaten

Die Evaluierung des Zeno.org-Datensatzes machte – für die anschließende Verarbeitung der Daten – in einigen Fällen auch Änderungen bzw. Korrekturen an den original Textdaten erforderlich. Im Folgenden werden exemplarisch eine Reihe typischer Inkonsistenzen dargestellt.

Bei den Texten von Adele Schopenhauer hat sich ein falsches Attribut `@sigel` eingeschlichen. Dieses wurde in das eigentlich zu verwendende Attribut `@book` geändert. In der Datei mit Texten von Johannes Proelß waren sämtliche Metadaten enthalten. Da diese in allen anderen Fällen in der eigens dafür vorgesehenen Katalogdatei Literatur-000a.xml gespeichert wurden, wurde das für Proelß entsprechend angepasst.

Alle Dateien wurde als UTF-8 kodiert und die Dateinamen von Umlauten, Kommata, Sonderzeichen etc. bereinigt. Außerdem machten manche Änderung in der Katalogdatei wiederum Änderungen in den Textdaten nötig. Wenn ein neuer Eintrag hinzugefügt

wurde und dieser eine neue ID erhielt, musste die Referenzierung daraufhin überarbeitet werden:

Dieses Vorgehen betrifft bspw. folgende Werke:

- Christian Weise: Sämtliche Werke. Berlin und New York 1971 ff.
- Dramen des deutschen Naturalismus. Herausgegeben von Roy C. Cowen, München 1981.
- Gottfried Keller: Sämtliche Werke in acht Bänden, Band 1, Berlin 1958–1961.
- Johann Rist: Sämtliche Werke. Berlin und New York 1972.
- Justinus Kerner: Werke. 6 Teile in 2 Bänden, Band 1, Berlin 1914.
- Praetorius, Johannes: Anthropodemus plutonicus. Das ist eine neue Welt-beschreibung [...] 1–2, Magdeburg 1666/67.
- Reinhard Johannes Sorge: Werke in drei Bänden. Nürnberg 1964.
- Alexander Schöppner: Sagenbuch der Bayer. Lande 1–3. München: Rieger 1852–1853.

In manchen Dateien gab es Inkonsistenzen in den Titeln, die korrigiert wurden, das betrifft bspw.:

- Buchholtz: Des Christliche Teutschen Herkules [...] Wunder-Geschichte. Braunschweig 1659/60.
- Ebner-Eschenbach: Bozena. München 1956-58.
- Grimmelshausen: Der abenteuerliche Simplicissimus. München 1956.
- Sienkiewicz: Quo vadis. Leipzig [o.J.]

Der Band Gottsched-Werke Bd. 8 wurde aufgesplittet in Gottsched-Werke Bd. 8.1 und Gottsched-Werke Bd. 8.2. Dazu wurde sowohl die Katalogdatei angepasst, wie auch die entsprechenden Referenzen in der Textdatei. Da „Das Kloster bei Sendomir“ eindeutig dem dritten Band aus Grillparzers Sämtlichen Werken zugeordnet werden konnte, wurde die entsprechende Referenz von „Grillparzer-SW“ auf „Grillparzer-SW Bd. 3“ geändert. Statt der ID „SuD-Nicolai Bd. 1“ wurde in zwei Fällen fälschlicherweise auch „Sud-Nicolai Bd. 1“ verwendet, auch das wurde in den Originaldaten angepasst.

Schließlich enthielt die besagte Katalogdatei eine ganze Reihe von Einträgen (insgesamt 25), auf die aus den Daten überhaupt nicht verwiesen wird. Diese Einträge wurden entfernt.

Outtakes Aufgrund von Inkonsistenzen in den Daten, die dazu führen, dass der zwar textgattungspezifisch entwickelte aber möglichst generisch angelegte Workflow Fehler produziert, wurden folgende Datensätze zunächst nicht berücksichtigt:

- Literatur_Busch___Wilhelm.xml
- Literatur_Goethe___Johann_Wolfgang-002.xml
- Literatur_Tucholsky___Kurt.xml

4 Transformation der „Digitalen Bibliothek“

Der im TextGrid Repository zur Verfügung gestellte Literatur-Ordner wurde ursprünglich von Directmedia Publishing digitalisiert und war Bestandteil der „Digitalen Bibliothek“. Die Editura GmbH & Co. KG hat die Daten von insgesamt 690 verschiedenen Autoren später kostenfrei auf www.zeno.org veröffentlicht. Bei der TextGrid-Version handelt es sich um XML- bzw. TEI P5-Dateien, die im projektspezifischen TextGrid Baseline Encoding vorliegen. Voraussetzung für die Einspielung des Literatur-Ordners der „Digitalen Bibliothek“ in die DTA-Infrastruktur an der Berlin-Brandenburgischen Akademie der Wissenschaften (BBAW) sind drei zentrale Schritte. 1) Die Generierung DTABf-konformer TEI-Header, 2) die Zusammenführung der Dateien auf Werkebene und 3) eine Konversion in das Basisformat des Deutschen Textarchivs (DTABf). Vor allem die Erstellung der TEI-Header stellte sich als eine besondere Herausforderung dar, was nicht zuletzt an den zugrunde liegenden, nicht spezifisch annotierten Metadaten liegt.

Ziel des Vorhabens war die Integration der Datensammlung in die Infrastruktur des Deutschen Textarchivs. Für die Integration in das DTA wird die Verfügbarkeit von Text und Bild empfohlen. Neben den digitalisierten und per DTABf kodierten Texten werden die Faksimiles der gescannten Seiten in einer Text-Bild-Ansicht angezeigt.

Da für den Literatur-Ordner der Digitalen Bibliothek keine Bilddigitalisate vorlagen, werden diese Texte als eigenständiges dstar-Korpus des Zentrums Sprache bereitgestellt. Die möglichst verlustfreie Konversion ist schrittweise auf Basis der Analyse der Daten angelegt. Die einzelnen XSL-Transformationen erfolgen textgattungs- wie aufgabenspezifisch.

4.1 Einleitung: Die Ordnerstruktur

Das DTABf sieht für den <text>-Bereich eine dreiteilige Strukturierung vor:[6]

1. Der <front>-Bereich, mit der Titelei,
2. der <body>-Bereich, dem eigentlichen Text der Publikation, der sich in der Regel in einzelne Kapitel, Abschnitte und Absätze gliedert[7] und
3. der <back>-Bereich, der in manchen Fällen vorhanden sein kann. Dieser enthält u. a. Register, Indizies oder Anhänge.

Um diese Struktur zu erreichen, sind mehrere Schritte der Konversion notwendig, die im Zusammenspiel der Erstellung der Metadaten für den TEI-Header harmonieren müssen. Um die einzelnen Konversionsschritte zu kontrollieren und in einer bestimmten Reihenfolge durchzuführen, wurde eine Ordnerstruktur definiert, in der die Ergebnisse der Konvertierung gespeichert werden. Da in jedem Schritt einer Konversion Text verloren gehen kann, dient diese Ordnerstruktur auch zur Qualitätssicherung.

6 https://deutschestextarchiv.de/doku/basisformat/TEIStruktur.html
7 https://deutschestextarchiv.de/doku/basisformat/div.html

Die Ordnerstruktur umfasst:

- 01-split-kickoff
- 02-sigel-pop
- 03a-distilled-raw
- 03b-distilled-control
- 03c-distilled-clean
- 04a-split_again-no
- 04b-split_again-all
- 04c-split_again-valid1
- 04d-split_again-fault
- 04e-split_again-valid2
- 04f-split_again-control
- 05-split-final
- 06-dta-raw
- 07-dta-fine
- 08-header
- 09-merge
- 10-final
- 11-txt
- 12-rename

In den folgenden Schritten wird auf diese Struktur mit den jeweiligen Ergebnissen der Transformation verwiesen.

4.2 Generierung der TEI-Header

Für die vollautomatische Generierung der TEI-Header werden folgende Datensätze berücksichtigt bzw. benötigt:

1. dibilit-meta.xml Hier handelt es sich um den im Projekt CLARIAH-DE angereicherten Metadatenkatalog, der die zentralen für die Generierung der TeiHeader benötigten Infos enthält (s.o. Anreicherung der Metadaten bzw. Katalogdatei).

2. digibib-distilled Es handelt sich dabei um XML-Dateien, die die zentralen das XML strukturierenden Elemente des Original-Datensatzes (des Zeno.org-Literatur-Ordners) enthalten. Die Datei wird mit XSL-Skripten automatisch erstellt. Entscheidend wird dabei auf die interne Referenzierung zu den Metadaten per eindeutigem Identifier zurückgegriffen. Die `<sigel>`-Elemente enthalten als Wert des Attributs @book den im Metadatenkatalog Literatur-000A.xml im Element `<BOOKNAME>` ebenfalls verwendeten eindeutigen String, der die Zuordnung zu den Quellenangaben erst möglich macht. Da es

Datensätze gibt, die mehrere solcher `<sigel>`-Elemente enthalten, dient digibib-distilled ebenfalls für die tiefere Strukturierung von bspw. mehrbändigen Werken (s.u. split-again). Außerdem enthalten diese XML-Dateien die als `<note>`-Element in die TextGrid-Header übernommenen Informationen zur Quelle. Diese werden wiederum für die Annotation im Bereich der Datumsangaben ausgewertet.

3. dtabf-header-dummy Hierbei handelt es sich um eine projektspezifische Vorlage für einen DTABf-validen TEI-Header. Dieser TEI-Header wird skriptbasiert jeweils werkspezifisch erweitert bzw. mit Daten befüllt.

4. tgr.csv, tgr-oai.csv Diese CSV-Dateien enthalten zentrale Metadaten der „Digitalen Bibliothek“ aus dem TextGrid Repository: Dateiname, Anzahl der Token, Textgattung, Autorname, Autor-PND, Titel etc. Die Datei dient als Grundlage für eine zeilenbasierte Zuordnung von Autor, Genre, Titel und Dateipfad. Außerdem werden daraus die PNDs der Autoren extrahiert. Die Datei wird v. a. für die skriptbasierte Anreicherung der bibliographischen Metadaten verwendet.

Da die zum Download bereitgestellten Dateien der „Digitalen Bibliothek“ leider – im Gegensatz zur Online-Version – keine TextGrid-IDs bzw. -HDL enthalten, wurde über die von TextGrid zur Verfügung gestellte OAI-PMH-Schnittstelle der Metadatensatz geharvestet. Die für die eindeutige Referenzierung der TextGrid-Daten benötigten Informationen wurden daraus extrahiert (tgr-oai.csv) und bei der Generierung der DTABf-Header berücksichtigt.

5. ids.csv, gerdracor.csv, zeno_perma.csv Ähnlich wurde mit Ressourcen des Leibniz-Instituts für Deutsche Sprache (IDS) in Mannheim (`<creatDate>`), des Projekts GerDraCor (`<date type="print">`) und schließlich den Permalinks auf www.zeno.org verfahren. Die Daten wurden evaluiert und skriptbasiert bei der Erstellung der Header verwendet.

6. Vollautomatische Erstellung der Header Das Skript `make_header.py` iteriert über die oben als digibib-distilled eingeführten XML-Dateien und legt für jeden in diesen aufgefundenen Identifier (`sigel/@book`) zunächst einen Dummy-DTABf-Header an. Dieser wird dann durch Rückgriff auf die verschiedenen oben genannten Metadatensätze werkspezifisch befüllt.

Der Aufruf erfolgt textgattungspezifisch per:

```
python3 make_header.py --genre=(roman|drama|...)
```

Die erzeugten TEI-Header werden textgattungspezifisch in Unterordner des Verzeichnisses `08-header` gespeichert.

4.3 Zusammenführung auf Werkebene

Das Kernkorpus des Deutschen Textarchivs basiert auf den Digitalisaten vollständiger Bücher, die allesamt per DTABf kodiert sind und dementsprechend avancierte TEI-Header enthalten. Mittlerweile wurden zwar auch nichtselbständige Werke in die Textsammlung mit aufgenommen, bei diesen handelt es sich jedoch ausnahmslos um vollständige Werke (z. B. Artikel), die dann für die Print-Publikation mit anderen Texten zusammengestellt wurden.

Im TextGrid Repository liegen die Texte (der Digitalen Bibliothek) dagegen aufgesplittet in der kleinstmöglichen Einheit vor. Während das bspw. bei den Romanen oder Dramen gänzlich unproblematisch ist, wird durch dieses Vorgehen der Gedichtband mit über 3000 Einzelgedichten (des gleichen Autors) in dann über 3000 einzelne TEI-Dateien mit jeweils einem eigenen TEI-Header aufgesplittet. Ein Grund für die Aufsplittung der zunächst Autor-basiert vorliegenden Zeno-XML-Dateien in die kleinstmögliche Texteinheit im TextGrid-Workflow liegt in der komplexen Verschachtelung von immer gleichen `<article>`-Elementen im Zeno-XML. Das TextGrid Repository stellt hier jedoch keine weiteren Anforderungen an das Datenformat und hat demnach keine Schwierigkeiten, auch mit Einzelgedichten oder einer Widmung umzugehen.

Ein weiteres sehr projektspezifisches Problem neben den Gedichtbänden stellt die mehrbändige Monographie dar. Aufgrund der Vorlage des Zeno-XMLs wurden diese Monographien als Werk interpretiert und nicht in die Einzelbände aufgeteilt. Die Wiederherstellung der Werkebene, die im Projekt CLARIAH-DE ganz am Anfang des Konversionsworkflows erfolgt, bedeutet also im einen Fall die weitere Aufsplittung der mehrbändigen Publikation in ihre Einzelbände, im anderen Fall die Rückführung der getrennt vorliegenden Bestandteile der Quelle in den ursprünglichen Werkkontext.[8] Technisch umgesetzt wird diese Rekontextualisierung bzw. Wiederherstellung der Quellen, aus denen die Texte der Digitalen Bibliothek stammen, per mehrmaliger XSL-Transformation.

Im Folgenden werden beispielhaft die einzelnen Schritte der Texttransformation dokumentiert.

8 Die Entscheidung für die Zusammenführung auf Werkebene hat auch für die Generierung der TEI-Header weitreichende Konsequenzen: Die Originaldaten enthalten in vielen Fällen Informationen zur Entstehung der einzelnen Texte, teilweise granuliert bis auf die Ebene einzelner Gedichte. Damit diese Informationen für den zusammengeführten Gedichtband nicht verloren gehen, wurden sie in den TEIHeader übernommen. Im `<notesStmt>` werden die Metainformationen, wenn möglich, für jedes einzelne Gedichte in einem eigenen `<note>`-Element gespeichert und über eine ID dem entsprechenden Textteil zugeordnet. Dieser aufwändige, spezielle Workflow wurde in einer Kooperation mit dem ZDL umgesetzt.

4.3.1 Splitten der Ausgangsdaten

Die Ausgangsdaten liegen pro Autor*innen vor. Dabei enthält eine Datei in der Regel mehrere identifizierbare Publikationen.

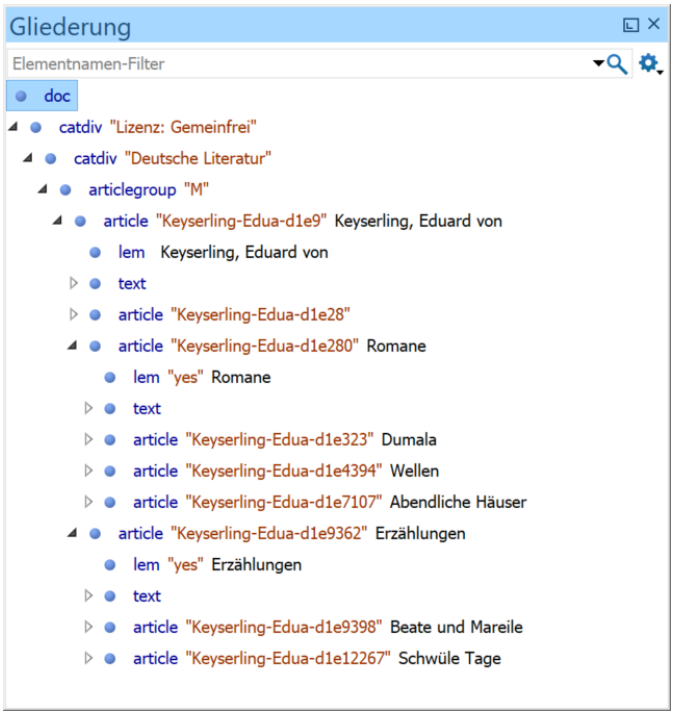

Abbildung 1: Romane und Erzählungen von Eduard von Keyserling.

Die Ausgangsdaten sind untergliedert in Textsorten.[9] Diese Kategorisierung wird homogenisiert und für die textsortenspezifische Speicherung der Ergebnisdaten genutzt.[10]

Ausgangs-Textsorte	Verzeichnis der Ergebnisdaten
Roman	roman
Romane	roman
Romane und Erzählungen	roman

```
java -jar saxon9ee.jar  -xsl:splitterxsl
                        -s:ausgangsdaten
                        -o:dtabf-konversion/01-split-kickoff
```

9 http://www.zeno.org/Kategorien/T/Literaturgattung
10 Die Verzeichnisse werden automatisch bei der Transformation im Filesystem angelegt.

4.3.2 Split-again

Ein Sonderfall stellt das mehrbändige Werk dar. Dort ist eine weitere Untergliederung in Einzelbände erforderlich. Dieses Aufsplitten der Publikation in Bände erfolgt textsortenspezifisch auf Grundlage der durch den `splitter` erstellten Verzeichnisstruktur, wobei die nochmalige Split-Routine zweimal durchzuführen ist.

1. Im ersten Durchlauf werden die weiter zu splittenden Dateien identifiziert und im Ordner: `04d-split_again-fault` gespeichert. Die in diesem Ordner befindlichen Dateien sind manuell zu korrigieren und im Ordner: `valid1/` zu speichern: Die Romane also bspw. in `04c-split_again-valid1/roman`.
2. Im zweiten Durchlauf wird die Transformationsroutine erneut gestartet. Das Input-Verzeichnis ist nun `04c-split_again-valid1`. Werden weitere Probleme beim Splitten erkannt, werden die davon betroffenen Dateien wieder im Verzeichnis `04d-split_again-fault` gespeichert.

Die Korrektur umfasst:

Abbildung 2: XML-Struktur im Verzeichnis 04d-split_again-fault. Der Fehler: Das article-Element [Motto] bildet keinen separaten Band.

Abbildung 3: XML-Struktur im Verzeichnis 04c-split_again-valid1 (nach der Korrektur). Es wurden zusätzliche article-Elemente zur Gliederung der Bände eingefügt. Das Motto wurde dem ersten Band zugeordnet. Das Ergebnis spiegelt nun die korrekte Bandeinteilung der zwei Bände wieder.

```
<article id="Schopenhauer-Jo-d1e8335">
    <lem>Die Tante</lem>
    <cat name="Roman" scope="all"/>
    <text>
        <sigel book="Schopenhauer-Tante Bd. 1"/>
        <sigel book=""/>
        <h4>Johanna Schopenhauer</h4>
        <h2>Die Tante</h2>
        <h4>Ein Roman</h4>
        <sigel book="Schopenhauer-Tante Bd. 1"/>
    </text>
    <article id="Schopenhauer-Jo-d1e8360">
        <lem>[Motto]</lem>
        <text/>
    </article>
    ....
</article>
```

```
<article id="Schopenhauer-Jo-d1e8335">
    <lem>Die Tante</lem>
    <cat name="Roman" scope="all"/>
    <text>
        <sigel book="Schopenhauer-Tante Bd. 1"/>
        <sigel book=""/>
        <h4>Johanna Schopenhauer</h4>
        <h2>Die Tante</h2>
        <h4>Ein Roman</h4>
        <sigel book="Schopenhauer-Tante Bd. 1"/>
    </text>
    <article>
        <article id="Schopenhauer-Jo-d1e8360">
            <lem>[Motto]</lem>
            <text/>
        </article>
    ...
    </article
</article
```

Erster Durchlauf, Identifikation der mehrbändigen Werke

```
java -jar saxon9ee.jar   -xsl:splitteragain.xsl
                         input_dir=file:dtabf-konversion/clariah/03c-distilled-clean/roman/
                         splitter_fault=dtabf-konversion/04d-split_again-fault/
                         output_default=dtabf-konversion/05-split-final
                         -s:dtabf-konversion/01-split-kickoff/roman
                         -o:dtabf-konversion/log
```

Zweiter Durchlauf, Aufsplitten der mehrbändigen Werke in einzelne Bände

```
java -jar saxon9ee.jar   -xsl:splitteragain.xsl
                         input_dir=file:dtabf-konversion/clariah/03c-distilled-clean/roman/
                         splitter_fault=dtabf-konversion/04d-split_again-fault/
                         output_default=dtabf-konversion/05-split-final
                         -s:dtabf-konversion/04c-split_again-valid1/roman
                         -o:dtabf-konversion/log
```

Parameter	Erklärung
-s:	Angabe des Input-Verzeichnisses der zu transformierenden Dateien.
-o:	Angabe des Verzeichnisses, in dem die Log-Dateien gespeichert werden.
input_dir	Angabe des Verzeichnisses, in dem die Kontroll-Dateien zu finden sind. Diese Dateien enthalten Informationen darüber, ob Bände vorliegen.
splitter_fault	Angabe des Verzeichnisses, in das die fehlerhaften Dateien gespeichert werden, die eventuell weiter zu splitten sind. Die Angabe der Textsorte ist nicht notwendig, da diese Information aus den Ausgangsdaten extrahiert wird.
output_default	Angabe des Verzeichnisses, in das die gesplitteten Dateien gespeichert werden. Die Angabe der Textsorte ist nicht notwendig, da diese Information aus den Ausgangsdaten extrahiert wird.

4.4 Schrittweise Konversion in das DTABf

4.4.1 rawDTABf

`rawDTABf.xsl` transformiert die nun als einzelne Werke vorhandenen Publikationen in ein rohes DTABf-ähnliches Format. Es wird die TEI-Grobstruktur mit `front`, `body` und gegebenenfalls `back` aufgebaut und die `article`-Struktur wird in eine `div`-Kapitel-Struktur umgewandelt. Die Ergebnis-Datei ist eine wohlgeformte XML-Datei, die mit dem Namensraum `http://www.tei-c.org/ns/1.0` verbunden ist. Die Datei ist nicht valide gegenüber dem TEI-Schema und dem DTABf-Schema.

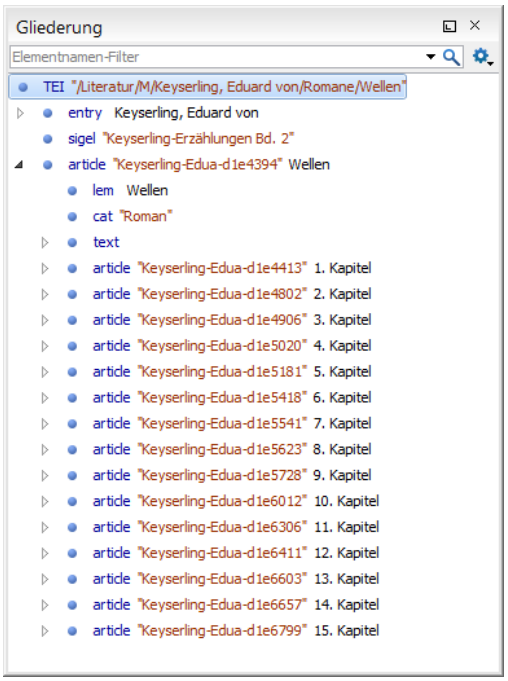

Abbildung 4: XML-Struktur im Verzeichnis 05-split-final. Die Dokumentstruktur besteht im wesentlichen aus einzeln ineinander geschachtelten article-Elementen. Eine Unterscheidung von Front (Titelei) und Body (Textkörper) ist nicht auszumachen.

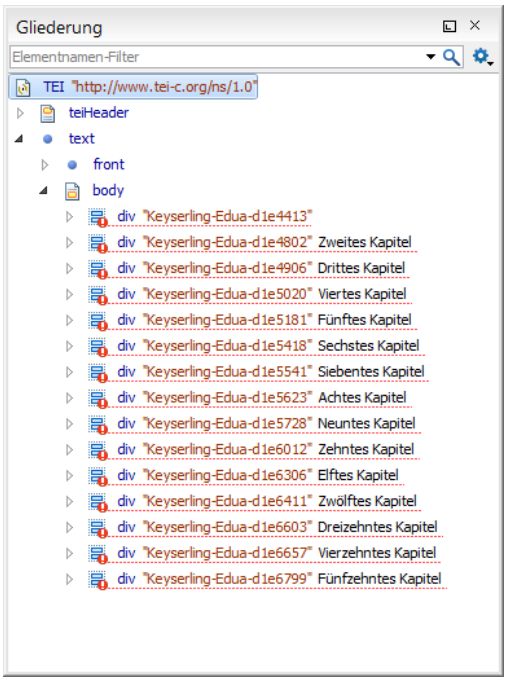

Abbildung 5: XML-Struktur im Verzeichnis 06-dta-raw. Die TEI-Grobstruktur ist deutlich zu erkennen. Die Kapitel befinden sich in div-Containern.

```
java -jar saxon9ee.jar  -xsl:rawDTABf.xsl
                        -s:dtabf-konversion/05-split-final/roman/
                        -o:dtabf-konversion/06-dta-raw/roman
```

4.4.2 fineDTABf

Wurden die vorangegangen Konversions-Routinen nur auf die textsortenspezifischen Verzeichnisse mit den darin enthalten Dateien angewendet, werden nun bei der Feinabstimmung der Ausgangsdaten spezielle Transformationen angewendet. Dies wird hier am Beispiel der Textsorte Roman dokumentiert. Weitere spezifische XSLT-Stylesheets befinden sich im Git-Repositorium. Das Ergebnis dieser Transformation ist eine valide XML-Datei, die dem DTABf-Schema mit einfachem Header entspricht.

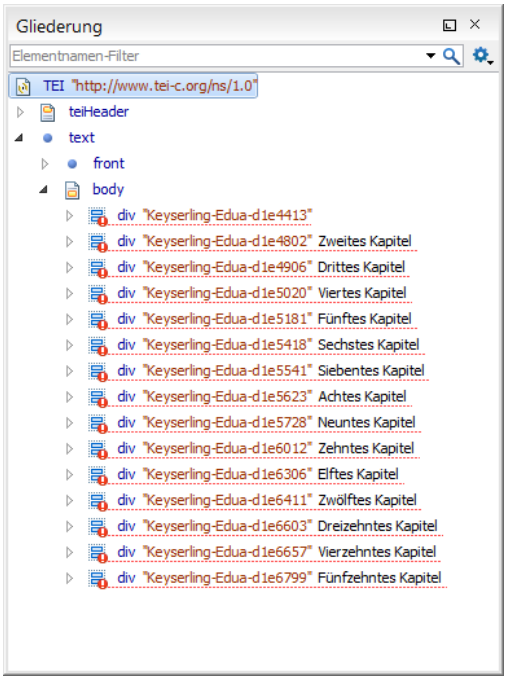

Abbildung 6: XML-Struktur im Verzeichnis 06-dta-raw. Die TEI-Grobstruktur ist deutlich zu erkennen. Die Kapitel befinden sich in div-Containern, die keine Typisierung aufweisen.

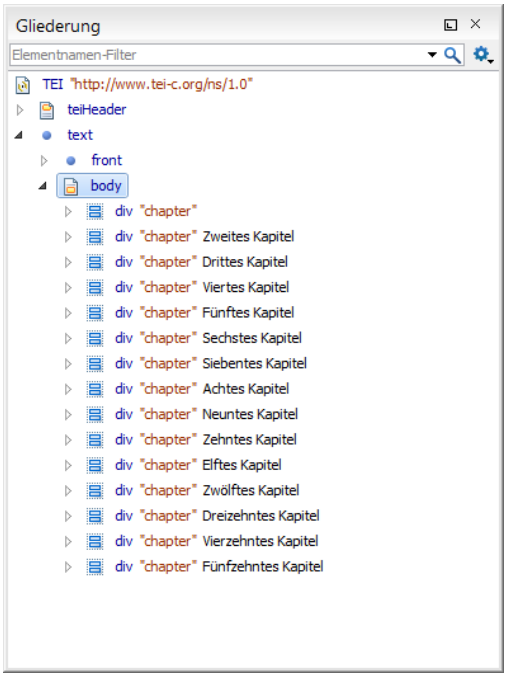

Abbildung 7: XML-Struktur im Verzeichnis 07-dta-fine. Die TEI-Grobstruktur und Feinheiten wie die Typisierung der div-Container ist deutlich zu erkennen.

```
java -jar saxon9ee.jar  -xsl:fineDTABf-roman.xsl
                        -s:dtabf-konversion/06-dta-raw/roman
                        -o:dtabf-konversion/07-dta-fine/roman/
```

4.4.3 addDTAHeader

Mit dem `addDTAHeader`-Stylesheet werden nun die Konversionsergebnisse mit dem DTABf-konformen und mit umfangreichen Metadaten angereichten Header verbunden. Dabei wird der zuvor verwendete „Dummy“-Header entfernt. Das Ergebnis der Transformation ist eine wohlgeformte XML-Datei. Im Element `<tei>` befindet sich ein Attribut `@n` das den zukünftigen Namen der Datei enthält. Dieses Attribut ist nicht TEI- oder DTABf-konform.

Da bei der Konversion von Dramen auf die Daten des German Drama Corpus (GerDraCor) zurückgegriffen wird, kommt für diese Textsorte ein spezielles XSLT-Stylesheet zur Anwendung.

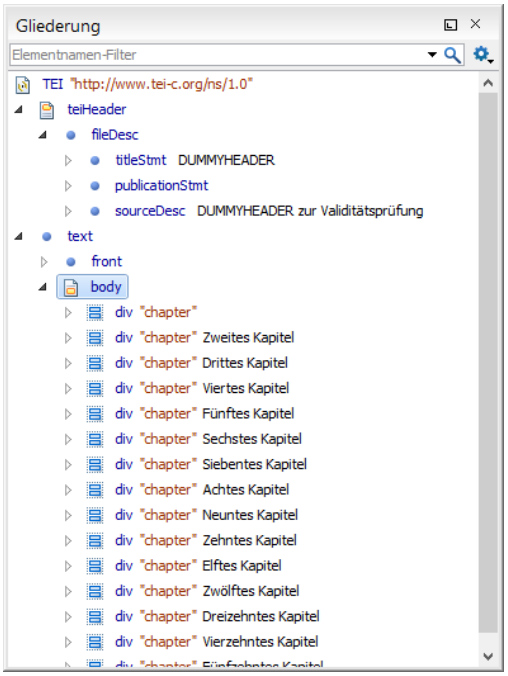

Abbildung 8: XML-Struktur im Verzeichnis 07-dta-fine. Hier wird der „Dummy“-Header verwendet.

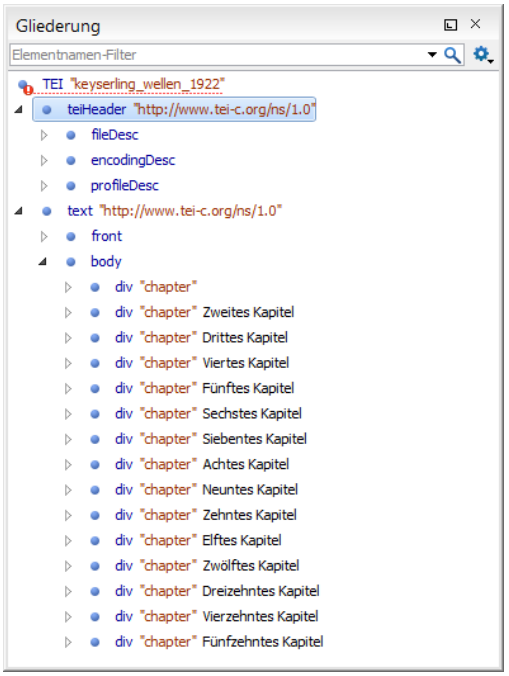

Abbildung 9: XML-Struktur im Verzeichnis 09-merge. Der TEI-DTABf-Header ist angefügt.

```
java -jar saxon9ee.jar  -xsl:addDTAHeader.xsl
                        -s:dtabf-konversion/07-dta-fine/roman/
                        -o:dtabf-konversion/09-merge/roman
```

4.4.4 sanitizer

Das `sanitizer`-Stylesheet ergänzt u. a. den TEI-Header um die Datumsangaben der Konversion, um den Kollationsvermerk (Seitenumfang) und die Textsorten und Inhaltskategorisierung des »Digitalen Wörterbuchs der Deutschen Sprache (DWDS).[11] Das Ergebnis ist eine DTABf-valide XML-Datei. Der `sanitizer` schließt die Konversion ab.

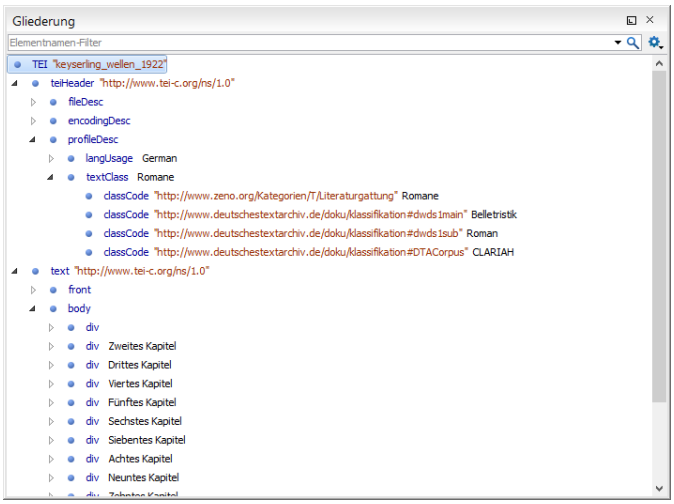

Abbildung 10: Der Roman „Wellen“ von Eduard v. Keyserling; DTABf-konform inkl. der DWDS-Textsorten- und Inhaltskategorisierung.

```
java -jar saxon9ee.jar  -xsl:sanitizer.xsl
                        -s:dtabf-konversion/09-merge/roman
                        -o:dtabf-konversion/10-final/roman/
```

4.4.5 rename

Das `rename`-Stylesheet ist eine nachgelagerte Konversion. Dabei wird die DTABf-valide Ausgangsdatei entsprechend der Namenskonvention des Deutschen Textarchivs umbenannt. Genutzt wird dabei ein vorgenerierter Dateiname.

Dateiname in 10-final:	Literatur_M_Keyserling-Eduard-von_Romane_Wellen-output.xml
Dateiname in 12-rename	keyserling_wellen_1922.xml

```
java -jar saxon9ee.jar  -xsl:rename.xsl
                        -s:dtabf-konversion/10-final/roman/
                        -o:dtabf-konversion/12-rename/roman
```

11 Die Anreicherung mit den Kategorien des DWDS ist ebenfalls in einer Zusammenarbeit mit dem ZDL entstanden.

5 Evaluierung

Evaluierung DTABf Alle nach den Textgattungen geordneten XML-Dateien im Ordner 11-rename werden schließlich final nochmals auf ihre DTABf-Konformität evaluiert.

Evaluierung Metadaten Für die Evaluierung der Metadaten kommt ein Python-Skript zum Einsatz. Dieses überprüft die zentralen Elemente in den TEIHeadern und generiert für jedes Werk eine 32-spaltige Zeile mit den entsprechenden Einträgen. Evtl. Lücken oder fehlerhafte Einträge werden durch entsprechende Sortierungen der csv-Datei sehr schnell sichtbar.

Evaluierung Textverlust Um sicherzustellen, dass im Laufe der schrittweisen Transformation der Ausgangsdateien keine Textteile verloren gegangen sind, wird ein automatisierter Abgleich mit einer txt-Version des Materials vorgenommen, die über eine gesonderte Routine generiert wurde.

Dazu werden die im Ergebnis vorliegenden XML-Dateien mittels eines einfachen XSLT-Skripts ebenfalls in eine txt-Version transformiert. Dieses kann über ein Shell-Skript aufgerufen werden, um flexibel auf die textgattungspezifische Ordnerstruktur reagieren zu können.

Für den darauf folgenden eigentlichen Textvergleich kommt ein Python-Skript zum Einsatz, das die beiden Pakete `scipy.spatial` und `sklearn.feature_extraction.text` nutzt, um die zu vergleichenden Dateien zunächst zu vektorisieren und anschließend die Cosinus Distanz zu ermitteln.

6 Zusammenfassung und Ausblick

Diese Schritt-für-Schritt-Anleitung dokumentiert im Detail die Homogenisierung und Zusammenführung verschiedener Derivate des Literatur-Ordners der Digitalen Bibliothek in das Basisformat des Deutschen Textarchivs. Das so entstandene DiBiLit-Korpus ist in die DTA-Infrastruktur der BBAW integriert: www.deutschestextarchiv.de/dibilit. So steht neben den linguistischen Analysetools wie Wortverlaufskurven oder DiaCollo auch die aus dem DTA-Kontext bekannte DDC-Suchmaschine für das neue und eigenständige Korpus zur Verfügung: https://kaskade.dwds.de/dstar/dibilit. Der daraus für die Community entstandene Mehrwert ist im BlogPost „Geschichte der Digitalen Bibliothek“ (https://sprache.hypotheses.org/?p=2436) bereits ausführlich beschrieben.

Rückblickend lässt sich sagen, dass die Entscheidung für einen modularen Workflow – Zusammenführung auf Werkebene, Generierung der DTABf-Header, Konversion der Texte – eine wichtige Voraussetzung für die erfolgreiche Umsetzung der herausfordernden Projektziele darstellte. Dadurch konnte die teils kleinteilige Arbeit – bspw. im Bereich der Anreicherung der Metadaten – innerhalb des Projektteams optimal verteilt werden. Die

Möglichkeit, einzelne Arbeitsschritte bspw. an eine studentische Hilfskraft delegieren zu können, darf im Rahmen der Arbeit an einer solch großen Textmenge nicht unterschätzt werden. Vor allem der Beschluss, die extern vorliegenden Metadaten möglichst lange getrennt von den eigentlichen Texten zu bearbeiten, könnte anderen Projekten mit vergleichbaren Datensätzen als Anregung dienen.

Als Desiderat könnte man die fehlenden Faksimiles ansehen. Die Verbindung von Text und Bild stellt bislang aus gutem Grund eine Voraussetzung für die Integration von Ressourcen in das Kernkorpus des Deutschen Textarchivs dar. Aufgrund der voranschreitenden Digitalisierungsbestrebungen der Bibliotheken ist zu erwarten, dass mehr und mehr Faksimiles der in der „Digitalen Bibliothek“ enthaltenen Werke frei zugänglich online verfügbar sind. Eine Verknüpfung mit diesen wäre eine zusätzliche Optimierung des Datensatzes.

Zum Ende der Projektphase wurden schließlich vielversprechende Tests bezüglich einer Nachnutzung im Kontext der anderen themenspezifischen Zeno.org-Ordner (Enzyklopädien, Märchen, Philosophie etc.) unternommen. Mit der Möglichkeit, den im Projekt CLARIAH-DE entwickelten Workflow für die Konversion weiterer Datensätze in das DTABf zu verwenden und diese so ebenfalls in die etablierte Infrastruktur zu integrieren, sind die gesteckten Ziele mehr als erreicht.

7 Links zu den Ressourcen

- Deutsches Textarchiv: https://www.deutschestextarchiv.de
- Deutsches Textarchiv, Erweiterung: https://www.deutschestextarchiv.de/dtae
- DiBiLit-Korpus: https://www.deutschestextarchiv.de/dibilit
- DiBiLit DDC-Korpussuche: https://kaskade.dwds.de/dstar/dibilit
- GerDraCor: https://github.com/dracor-org/gerdracor
- TextGrid Repository: https://textgridrep.org/browse/root
- Zeno.org, Literatur: http://www.zeno.org/Literatur