Mittelniederdeutsche Urkunden des ostniederdeutschen Sprachraums (MndUrk)

Eine TEI-Datenpublikation der Urkundentranskriptionen aus dem DFG-Projekt *Atlas frühmittelniederdeutscher Schreibsprachen* (1994–2005)

Herausgegeben von Andreas Bieberstedt und Hanna Fischer.
Bearbeitet von Mareike Krause, Karsten Labahn,
Robert Stephan und Katharina Wiebe.

Universität Rostock, 2025

https://doi.org/10.18453/rosdok_id00004746

Kontakt:

Andreas Bieberstedt	https://orcid.org/0009-0005-7423-8454
Hanna Fischer	https://orcid.org/0000-0002-8157-5716
Mareike Krause	https://orcid.org/0009-0008-1031-3867
Karsten Labahn	https://orcid.org/0000-0002-8482-807X
Robert Stephan	https://orcid.org/0000-0001-7605-7415

Zitiervorschlag:

Bieberstedt, Andreas und Hanna Fischer (Hrsg.). 2025. *Mittelniederdeutsche Urkunden des ostniederdeutschen Sprachraums. Eine TEI-Datenpublikation der Urkundentranskriptionen aus dem DFG-Projekt Atlas frühmittelniederdeutscher Schreibsprachen* (1994–2005). Bearbeitet von Mareike Krause, Karsten Labahn, Robert Stephan und Katharina Wiebe. Universität Rostock. https://doi.org/10.18453/rosdok_id00004746

Abstract

Die Datenpublikation umfasst 1942 Urkundentranskriptionen aus dem DFG-Projekt *Atlas frühmittelniederdeutscher Schreibsprachen* (1994–2005, Standort Rostock). Es handelt sich um mittelniederdeutsche Urkunden des 14. und 15. Jahrhunderts aus dem ostniederdeutschen Sprachraum, die als RTF-Dateien vorlagen. Für die Datenpublikation wurden die Dateien kontrolliert und ihre Metadaten vereinheitlicht und ergänzt. Fehlerhafte Sonderzeichen wurden durch Unicode-Zeichen ersetzt. Im Rahmen der Datenpublikation stehen die Urkundentranskriptionen in den Formaten TEI, HTML und PDF zur Verfügung. Die Datenpublikation wurde finanziell durch die Universität Rostock gefördert.

Inhaltsverzeichnis

1 Projektbeschreibung

Ziel der Datenpublikation ist es, die Quellengrundlage des DFG-Projekts *Atlas frühmittelniederdeutscher Schreibsprachen* (1994–2005) für den ostniederdeutschen Sprachraum digital zur Verfügung zu stellen. Hierfür wurden insgesamt 1942 Transkriptionen der historischen Projektquellen kontrolliert, im Hinblick auf Sonderzeichen korrigiert, in standardisierte Dateiformate gebracht und zusammen mit ihren Metadaten veröffentlicht.

1.1 DFG-Projekt Atlas frühmittelniederdeutscher Schreibsprachen

1.1.1 Hintergrund

Das hier vorgelegte Datenmaterial ist das Ergebnis eines sprachhistorischen Forschungsprojekts, das mit Hilfe einer Langzeitförderung der Deutschen Forschungsgemeinschaft von 1994 bis 2005 am Institut für Germanistik der Universität Rostock durchgeführt wurde. Das Projekt ist unter dem Titel *Atlas frühmittelniederdeutscher Schreibsprachen (Untersuchung regionaler Schreibsprachen im mittelniederdeutschen Sprachraum)* und der Projektnummer 5342480 in der GEPRIS-Datenbank verzeichnet.[1] Ziel des DFG-Projekts war die systematische Erfassung der Schreibsprachen des mittelniederdeutschen Sprachraums des 14. und 15. Jahrhunderts und ihre Beschreibung in Form eines historischen Sprachatlasses, mit dessen Hilfe die diatopische und diachrone Varianz der mittelniederdeutschen Schreibsprachenlandschaft dargestellt werden sollte. Die Datenbefunde sollten unter anderem dazu beitragen, die in der älteren Forschung vertretene These von der sogenannten „Lübischen Norm“ kritisch zu hinterfragen. Diese These geht davon aus, dass der Schreibusus der Hansestadt Lübeck aufgrund ihrer ökonomischen und politischen Bedeutung im 15. Jahrhundert beginnt, normative Wirkung für den gesamten mittelniederdeutschen Schreibsprachenraum zu entfalten und von anderen Hansestädten im Sinne eines überregionalen Akkommodationsprozesses übernommen wird. Diese seit den Anfängen der Niederdeutschen Philologie im ausgehenden 19. Jahrhundert ventilierte Vorstellung einer vormodernen Sprachstandardisierung wurde in der jüngeren Forschung zwar kritisch gesehen, blieb jedoch bis in die 1980er Jahre und darüber hinaus als sprachhistorisches Erklärungsmodell populär. Das neue Datenmaterial sollte vor diesem Hintergrund helfen, sprachliche Konvergenz- und Divergenzprozesse innerhalb der mittelniederdeutschen Schreibsprachlandschaft präzise zu konturieren und somit genauere Aufschlüsse über sprachhistorische Entwicklungsprozesse in einem relevanten Bereich der mittelniederdeutschen Schriftlichkeit liefern.

1.1.2 Konzeptioneller Ansatz

Für diese Zielsetzung verfolgte das Projekt einen variablenlinguistischen Ansatz, indem variationsrelevante sprachliche (genauer: phonographematische, morphologi-

[1] https://gepris.dfg.de/gepris/projekt/5342480

sche, lexikalische und im Ausnahmefall auch syntaktische) Merkmale des Mittelniederdeutschen auf Basis eines Variablenkatalogs erfasst und auf Einzelkarten abgebildet werden sollten. Grundlage bildete der in den 1980er Jahren von Robert Peters erarbeitete „Katalog sprachlicher Merkmale zur variablenlinguistischen Erforschung des Mittelniederdeutschen“, der im Rahmen des Projekts modifiziert und erweitert wurde:

Peters, Robert (1987–1990): Katalog sprachlicher Merkmale zur variablenlinguistischen Erforschung des Mittelniederdeutschen. Teil I in: Niederdeutsches Wort 27 (1987), 61–93; Teil II in: Niederdeutsches Wort 28 (1988), 75–106; Teil III in: Niederdeutsches Wort (1990), 1–17.

Zugleich wurde ein korpuslinguistischer Ansatz gewählt, der darauf abzielte, durch die Erhebung einer möglichst großen und dichten Datenmenge aus einem überlieferungsstarken Bereich der mittelniederdeutschen Schriftlichkeit quantitativ basierte Aussagen über sprachliche Verteilungsmuster generieren zu können. Gewählt wurde der Bereich des städtischen Verwaltungsschrifttums in Form vor allem von Urkunden und Stadtbucheinträgen, da dieses Schrifttum diatopisch und diachron eine hohe Überlieferungsdichte aufweist und aufgrund seines institutionellen Charakters weniger stark von ideolektalen, stilistischen oder ähnlichen Varianzen beeinflusst ist. Dadurch sollte insgesamt eine gute Vergleichbarkeit des Sprachmaterials gewährleistet werden. Zudem wurde bei der Textsortenwahl berücksichtigt, dass das Konzept der Lübischen Norm und des vormodernen Schriftsprachenausgleichs mit Blick auf genau diesen Bereich der historischen Schriftlichkeit entworfen wurde. Der Fokus auf das städtische Verwaltungsschrifttum bedeutete zugleich, dass fürstliche und geistliche (zum Beispiel bischöfliche) Urkunden keine Berücksichtigung fanden, unabhängig davon, ob sie am selben Schreibort ausgefertigt worden waren.

1.1.3 Standorte

Geplant war das Projekt zunächst als Kooperationsprojekt, das an den beiden Standorten Rostock und Münster gemeinsam durchgeführt werden sollte. Hierbei war der Standort Münster für die Erfassung des Datenbestands des sogenannten niederdeutschen Altlandes (im Wesentlichen Westfalen, Ostfalen, Nordniedersachsen), der Standort Rostock dagegen des niederdeutschen Neulands (im Wesentlichen Ostelbisch, Märkisch-Brandenburgisch, baltisches Niederdeutsch) verantwortlich. Leiter des Standorts Münster war Robert Peters, Leiterin des Standorts Rostock die hiesige Professorin für Niederdeutsche Sprache und Literatur, Irmtraud Rösler.

Im Verlauf der Datenerhebung wurde beschlossen, das Gesamtprojekt an beiden Standorten als separate Projekte weiterzuführen mit dem Ziel zweier Sprachatlanten, zum einen für das niederdeutsche Altland und angrenzende (im Wesentlichen niederländische) Gebiete und zum anderen für das niederdeutsche Neuland. Der Standort Münster präsentierte die Ergebnisse seines Teilprojekts 2017 in Form des „Atlas spätmittelalterlicher Schreibsprachen des niederdeutschen Altlandes und angrenzender Gebiete (ASnA)“:

Peters, Robert (2017). Atlas spätmittelalterlicher Schreibsprachen des niederdeutschen Altlandes und angrenzender Gebiete (ASnA): Band I: Einleitung, Karten; Band II: Verzeichnis der Belegtypen; Band III: Verzeichnis der Schreibformen und der Textzeugen (Ortspunktdokumentation). In Zusammenarbeit mit Christian Fischer und Norbert Nage. Berlin, Boston: De Gruyter, 2017. https://doi.org/10.1515/9783110417623

Am Standort Rostock konnte die Datenbearbeitung bisher nicht bis zu diesem Punkt gebracht werden.

1.1.4 Korpus

Auswahlkriterien für das Korpus waren neben der Textsortenzugehörigkeit (Verwaltungsschrifttum in Form von Urkunden, Stadtbucheinträge sowie Briefe) die Lokalisierbarkeit und Datierbarkeit der Texte. Zur möglichst präzisen Erfassung phonographematischer Phänomene wurden zudem ausschließlich Originaltexte erfasst, keine Transkriptionen aus Urkundenbüchern.

Die diachrone Varianz wurde durch die Erfassung von Texten aus zwei Jahrhunderten (14. und 15. Jahrhundert) und hier insgesamt aus drei Zeiträumen berücksichtigt:

Zeitraum I: 14. Jahrhundert (1300–1400)
Zeitraum II: Mitte 15. Jahrhundert (1446–1455)
Zeitraum III: Ende 15. Jahrhundert (1491–1500)

Teilweise wurden auch Texte aus anderen Zeitabschnitten erfasst; diese blieben bei der späteren Auswertung zwar unberücksichtigt, wurden aber in die vorliegende Datenpublikation aufgenommen (siehe Abschnitt 1.2).

Die diatopische Varianz wurde durch die möglichst umfassende Einbeziehung sämtlicher relevanter mittelalterlicher Schreiborte des niederdeutschen Neulands in dessen Ausdehnung vom Süden (Halle, Wittenberg) bis in den Nordwesten (Lübeck, Kiel) und Nordosten (Kolberg, Danzig, Riga) überprüfbar gemacht. Zudem wurden als Überschneidungsbereiche zwischen Alt- und Neuland auch angrenzende (elbostfälische) Schreibsprachenregionen berücksichtigt.

Hinsichtlich der Textmenge wurde eine Mindestanzahl von 50 Einzeltexten pro Ort festgelegt. Mehrere Schreiborte, die diese Anforderung nicht erfüllten, mussten im Laufe des Projekts aus dem Korpus ausgeschlossen werden. Die im Vergleich zum niederdeutschen Altland schlechtere Überlieferungslage hatte zum einen historische Gründe (spätes Städtewachstum, wenig ausdifferenzierte städtische Verwaltung, verzögerter Übergang vom Lateinischen zum Niederdeutschen), war zum anderen hohen Verlustzahlen geschuldet (mittelalterliche Stadtbrände, Verluste im Zweiten Weltkrieg sowie durch Verschleppung von Archivalien als Beutegut durch die sowjetischen Besatzer). Insgesamt wurden ca. 2000 Einzeltexte aus den folgenden Schreiborten in das Korpus integriert:

Berlin	Lübeck	Salzwedel
Burg	Magdeburg	Schwerin
Brandenburg	Parchim	Stettin
Danzig	Prenzlau	Stralsund
Greifswald	Quedlinburg	Wismar
Halle	Reval	Wittenberg
Kiel	Riga	Zerbst
Kolberg	Rostock	

1.1.5 Projektphasen

Geplant war eine Durchführung des Projekts in drei Phasen: (1) Korpuserstellung, (2) Überführung des Textbestands in eine ACCESS-basierte Datenbank und semiautomatische Lemmatisierung und (3) Kartierung ausgewählter Lemmata. Vollständig abgeschlossen werden konnte im Rahmen der Projektlaufzeit lediglich die Phase 1. Hierbei wurden auf Basis einer umfangreichen Literatur- und Archivrecherche die jeweiligen Urkundenbestände vor Ort gesichtet und abfotografiert. Anschließend erfolgte ihre computergestützte Transkription. Gewählt wurde ein Transkriptionsverfahren, dass den originalen Text möglichst detailgetreu, inklusive sämtlicher Satzzeichen, diakritischer Zeichen und Abbreviaturen, erfasst. Hierfür wurden unter anderem projektintern ein eigener Schriftsatz sowie ein Set an diakritischen Zeichen (= Sonderzeichen) entworfen. Die Transkriptionen wurden im *Double-Keying*-Verfahren Korrektur gelesen. Allerdings ist der Korrekturstand bei einzelnen Ortspunkten – nach nochmaliger Sichtung – als unterschiedlich zuverlässig zu bewerten. Die Unterlagen und Materialien des Projektes befinden sich am Institut für Germanistik der Universität Rostock, Professur für Niederdeutsche Sprache und Literatur.

1.2 Datenpublikation der Urkundentranskriptionen

Die digitale Publikation der Urkundentranskriptionen wurde in Zusammenarbeit des Instituts für Germanistik und der Universitätsbibliothek der Universität Rostock vorgenommen. Das Projekt wurde im Zeitraum 01.09.2024 bis 31.12.2024 durch die Forschungsförderung des Departments „Wissen – Kultur – Transformation (WKT)“ der Interdisziplinären Fakultät (INF) finanziell gefördert. Die Projektleitung erfolge durch Andreas Bieberstedt und Hanna Fischer in Zusammenarbeit mit Karsten Labahn und Robert Stephan und unter Mitarbeit von Mareike Krause und Katharina Wiebe. Als Datengrundlage diente ein Projektordner mit dem Datennachlass aus dem DFG-Projekt, der auf einem Institutscomputer verfügbar war. Die im Rahmen des Projekts durchgeführten Arbeiten werden im Folgenden kurz beschrieben.

1.2.1 Auswahl der Urkundentranskriptionen für die Publikation

Zunächst wurden alle RTF-Dateien mit Transkriptionen gesichtet. Dabei wurden Dateien entfernt, die aufgrund folgender Kriterien als ungeeignet eingeschätzt wurden:

- fehlerhafte Verortung
- unklare Datierung
- hochdeutsche Urkunden
- fehlerhafte Angaben zu Absenderort oder Zielort
- Transkription auf Grundlage einer Abschrift, nicht der Originalquelle

In die Datenpublikation wurden zusätzlich Transkriptionen aufgenommen, die außerhalb des Zeitraums des Atlasprojekts datieren oder die die zuvor festgelegte Mindestzahl von 50 Texten pro Ort unterschreiten. Für die Datenpublikation wurden **insgesamt 1942 Transkriptionen aus 27 Orten** ausgewählt.

1.2.2 Zusammenführung, Kontrolle und Recherche von Metadaten

Im Atlasprojekt wurden die Metadaten der Urkunden ortsweise in sog. Ortsprotokollen erfasst. Im Rahmen der Datenpublikation wurden diese Metadaten aus allen Ortsprotokollen in eine Gesamttabelle überführt und dort vereinheitlicht. Die Metadaten wurden zudem mithilfe von Archivkatalogen überprüft und ggf. ergänzt oder aktualisiert.

- Die Recherche erfolgte in den digitalen Katalogen der entsprechenden Archive oder im Archivverbund Mecklenburg-Vorpommern (*ARIADNE*).
- Überprüft wurden die Angaben zu Signatur, Datierung und Inhalt der Urkunden, fehlende Informationen wurden mit „k.A.“ („keine Angabe“) ausgewiesen. Für Orte wurden GNDs recherchiert und ergänzt.
- Ergänzt wurden auch URLs zu digitalen Katalogeinträgen. Für die Verlinkung wurden die Webseiten und Onlinekataloge der entsprechenden Archive herangezogen. Falls ein Archiv über eine eigene Datenbank verfügt, wurde diese genutzt und ein Link generiert.
- Bei vielen Archiven war kein Online-Zugriff möglich. Die hinterlegten Signaturen konnten in diesem Fall nicht überprüft werden. Sie werden dennoch aufgeführt, aber mit „n.f.“ („nicht findbar“) in der Kommentarspalte gekennzeichnet.
- Im Zuge der Recherchen wurde auch die Liste der Archive aktualisiert.

Für ca. 300 Transkriptionen waren keine „Ortsprotokolle“ vorhanden. Hier wurden die Metadaten aus dem Kopfbereich der ursprünglichen RTF-Dateien übernommen.

1.2.3 Korrektur von Sonderzeichen

Im Rahmen des Atlasprojekts wurden Diakritika über eigens erstellte Sonderzeichen in einer eigenen Schriftartdatei abgebildet. Diese Datei wurde speziell für das Alt-Projekt erstellt und kann von heutigen Programmen nicht mehr gelesen werden. Entsprechend war eine Konvertierung der Zeichen in Unicode-Schriftarten notwendig. Zur Vorbereitung der Konvertierung wurde eine Tabelle mit allen verwendeten Sonderzeichen erzeugt und die ihnen entsprechenden Unicode-Codepoints ermittelt.

1.2.4 Konvertierung der Daten

Die Konvertierung erfolgte in einem mehrstufigen Verfahren. In verschiedenen Schritten waren Ausnahmefälle zu behandeln, die häufig schon als Fehler in den RTF-Dateien vorgefunden wurden. So wurden beispielsweise einige Fußnoten nicht mit der Formatvorlage „Fußnote“, sondern nur als hochgestellte Zahlen erfasst. Sofern es sich bei dem Phänomen um Einzelfälle handelte, wurden die Korrekturen direkt in die Programmierung des jeweiligen Konvertierungsschritt integriert. Dadurch konnte ein mehrfaches Verarbeiten und Testen der Transformationsschritte ohne direkte Modifikation der Ausgangsdateien ermöglicht werden.

1.2.4.1 Sonderzeichen in der RTF-Datei markieren

Im RTF-Dateiformat sind Sonderzeichen einfach zu identifizieren, da sie mit einem Steuerzeichen gefolgt von einem Hexadezimalwert (Syntax: `\'xx`) abgespeichert sind. Alle diese Vorkommen wurden sichtbar gemacht, indem ihnen ein RTF-codierter Backslash vorangestellt wurde (zum Beispiel: `\'f1 → \'5c'f1`). Durch diesen Schritt wurde sichergestellt, dass die Sonderzeichen bei der anschließenden TEI-Konvertierung nicht verloren gehen und in einem späteren Schritt durch ihre korrekte Unicode-Variante ersetzt werden können.

1.2.4.2 Automatische Konvertierung von RTF zu TEI

Für die eigentliche Datei-Konvertierung wurde zunächst geprüft, ob diese vollständig in *Java* nachimplementiert werden kann. Dazu hätten alle Phänomene (z.B. Formatierungen, Absätze, Fußnoten) in der RTF-Ausgangsdatei identifiziert und eine Entsprechung in TEI-Zielformat gefunden werden müssen. Dieser Ansatz wurde auf Grund der komplexen Struktur des RTF-Formates verworfen.

Stattdessen wurde nach einigen Tests die Konvertierung mit dem Online-Konverter von TEI-Garage[2] durchgeführt. Dieser bietet neben einem Webformular für die Konvertierung von Einzeldateien auch einen Webservice[3] als REST-API an. Darüber lassen sich die RTF-Dateien an den Dienst schicken, der diese wiederum als konvertierte TEI-XML-Datei ausgibt. Im Rahmen der Konvertierung erfolgt auch die Umstellung des Zeichensatzes von Windows-1252 (RTF) zu UTF-8 (TEI-XML).

Nach Abschluss dieses Schrittes lagen XML-Dateien vor, die mit verschiedenen etablierten Tools und Softwarebibliotheken weiterverarbeitet werden können.

1.2.4.3 Sonderzeichen auflösen

In den konvertierten TEI-XML-Dateien findet man die Sonderzeichen noch in der zuvor definierten Form mit vorangestelltem Backslash (zum Beispiel: `\'f1`). Diese wurden in sämtlichen Dateien automatisch ausgelesen, ihre Häufigkeit gezählt und in einer Tabelle abgelegt. Das Zeichensymbol wurde anhand des Codepoints in der im Altprojekt verwendeten Schriftart ermittelt und ein entsprechendes Unicode-Symbol gesucht.

[2] https://teigarage.tei-c.org/
[3] https://teigarage.tei-c.org/ege-webservice/

Für die Recherche wurden die Unicode-Zeichenübersichten von Compart[4] und MUFI[5] verwendet. Auf der Webseite Shapecatcher[6] können Zeichen mit der Maus skizziert werden und das System schlägt die am ähnlichsten aussehenden Unicodezeichen vor.

Nachdem das Unicode-Zeichen ermittelt wurde, wurde die Codierung festgelegt. Viele Zeichen gibt es sowohl als zusammenhängendes als auch als zusammengesetztes Zeichen. So lässt sich beispielsweise das Symbol ū über den Codepoint `&#x016B;` (Latin Small Letter U with Macron) oder als zusammengesetztes Zeichen `u&#x304;` (Latin Small Letter U AND Combining Macron) in Unicode darstellen. In der Regel wurde die letztere Variante verwendet, weil dadurch weniger unterschiedliche Codepoints in den XML-Dateien vorkommen. Unicode-fähige Programmiersprachen (z.B. Java oder Python) bieten Methoden an, mit denen sich die beiden Codepoint-Varianten für die weitere Verarbeitung ineinander konvertieren lassen.

Das Ergebnis dieser Recherche war eine Tabelle, die jeweils den aus der Ausgangsdatei übernommenen RTF-Codepoint und den neuen Unicode-Codepoint enthält. Sie bildete die Grundlage für die automatische Ersetzung der Codepoints in den Urkundentranskriptionen.

1.2.4.4 XML-Struktur bereinigen

In diesem Schritt wurden die Struktur und Formatanweisungen in den generierten XML-Dateien untersucht und vereinheitlicht. Dazu wurden unter Zuhilfenahme der Anfragesprache XPath[7] Listen mit unterschiedlichen Phänomenen erzeugt, analysiert und deren Vorkommen gegebenenfalls vereinheitlicht. Beispielsweise wurden Farbinformationen entfernt, da die Analyse ergeben hat, dass ausschließlich Schwarz als Farbe verwendet wurde. Weiterhin wurde die Formatierung Blocksatz entfernt, da diese im Gesamtkorpus uneinheitlich und nicht systematisch verwendet wurde. Absätze und Zeilenumbrüche wurden in Absätzen zusammengefasst, vereinheitlicht und leere Absätze wurden gelöscht.

1.2.4.5 Steuerzeilen und Kopfbereich entfernen

In den RTF-Dateien waren der Kopfbereich mit Metadaten sowie Start und Ende des Textes durch Zeilen mit speziellen Steuerzeichen (`§, @, @@`) codiert. In diesem Schritt wurde der Kopfbereich entfernt, weil die darin enthaltenen bibliographischen Angaben bereits vorab ausgelesen wurden. Später wurden diese Informationen wieder in den TEI-Header aus der externen Tabelle mit den Metadaten eingefügt. Außerdem wurden die Textstart- und Textende-Markierungen gelöscht, da sich diese Phänomene geeigneter durch die entsprechende TEI-XML-Syntax ausdrücken lassen.

---

[4] https://www.compart.com/de/unicode
5 MUFI: The Medieval Unicode Font Initiative, https://mufi.info/
[6] https://shapecatcher.com/
[7] https://www.w3.org/TR/xpath-10/

1.2.4.6 Fußnoten bereinigen

Im Rahmen des Altprojektes wurden einige Fußzeilen nicht mittels der entsprechenden Formatvorlage formatiert und markiert, sondern das Fußnotensymbol nur als hochgestellte Zahl in den Text geschrieben. Häufig wurde eine Fußzeile mehrfach im Text referenziert, so wurden beispielsweise Phänomene wie Durchstreichungen oder Zeilenumbrüche mit nur einer Fußzeile erwähnt und das Fußnotensymbol im Text wiederholt.

Über verschiedene Heuristiken und XPath-Ausdrücke wurden diese Stellen ermittelt und automatisch durch die korrekte TEI-XML-Auszeichnung für Fußnoten und Referenzen ersetzt.

1.2.4.7 Metadaten ergänzen

Die Metadaten aller Urkundentranskriptionen wurden in einem vorausgehenden Projektschritt bereits in einer Exceltabelle zusammengetragen, normiert und teilweise um Angaben aus dem Kopfbereich der RTF-Dateien ergänzt (siehe Kapitel 51.2.2). Für diese Daten wurde die entsprechenden Elemente im TEI-Header ermittelt und mittels Java JDOM2-API[8] in die TEI-Datei eingefügt.

1.3 Erzeugung von Präsentationsderivaten in HTML und PDF

Für die Generierung der **HTML-Ansichten** wurde ein XSLT 1.0 Stylesheet[9] erstellt. Mit diesem Stylesheet wurden die Metadaten der Urkunden aus dem TEI-Header in eine tabellarische Ansicht überführt und die wenigen XML-Elemente aus dem TEI-Body mit den ihnen entsprechenden HTML-Elementen ersetzt (z.B. `<tei:p>` zu `<html:p>` oder `<tei:hi>` zu `<html:span>`). Am Ende des Dokuments wurden die Fußnotenreferenzen und -inhalte in tabellarischer Form ausgegeben.

Da alle heutigen Webbrowser die Fähigkeit besitzen, XSLT 1.0 zu verarbeiten, wurde der Inhalt der Datenpublikation so aufbereitet, dass er unter Verwendung eines Webservers angezeigt werden kann (siehe Kapitel 2.3). Dafür war es notwendig, einen Verweis auf das Stylesheet in jede TEI-Datei zu integrieren.

Zur Generierung der **PDF-Dateien** wurde ein Java-Programm geschrieben, welches unter Verwendung der Java-Bibliotheken JDOM2[10] und OpenPDF[11] die Transformation durchführt. Der Quellcode und eine ausführbare Datei mit allen abhängigen Bibliotheken sind Bestandteil dieser Datenpublikation.[12]

Für die Anzeige der mittelalterlichen Texte in HTML- und PDF-Dateien wurde die Schriftart JUnicode2[13] verwendet. Diese ist unter einer Open Font License frei verfügbar und enthält alle für die Darstellung notwendigen Sonderzeichen. Über ein GitHub-

[8] http://jdom.org/
[9] Datei: `99_Technik\xslt\MndUrk_Transkription_html.xsl`
[10] vgl. Fußnote 8.
[11] https://github.com/LibrePDF/OpenPDF
[12] Ordner: `99_Technik\java`
[13] https://psb1558.github.io/Junicode-font/

Ticket[14] wurde mit dem Autor der Schriftart Kontakt aufgenommen, welcher daraufhin unkompliziert eine fehlende Zeichendarstellung ergänzte.

2 Beschreibung der Daten

2.1 Ordnerstruktur und Dateien

Das publizierte ZIP-Archiv enthält sämtliche Daten (Ausgangsdaten, Projektergebnis und Hilfsdateien) aus dem Projekt in folgender Ordnerstruktur:

2.1.1 Ordner `01_Urkundentranskriptionen_Daten_TEI`

Dieser Ordner enthält die Transkriptionen im XML-Format gemäß den Richtlinien der Text Encoding Initiative (TEI). Sie sind das Ergebnis der Transkription der RTF-Dateien des Altprojektes und wurden mit kontrollierten, normierten und teilweise nachträglich recherchierten Metadaten angereichert.

Die Datei `MndtUrk_Korpus.tei.xml` enthält im TEI-Header die Projekt-Metadaten und listet alle Dateien (Urkundentranskriptionen) gruppiert in 27 Ordnern (nach Herkunftsorten) auf.

2.1.2 Ordner `11_Urkundentranskriptionen_Ansicht_PDF`

Dieser Ordner enthält PDF-Dateien die eine Ansicht der Urkundentranskriptionen ermöglichen. Sie wurden aus den TEI-Dateien generiert und ebenfalls in 27 Ordnern (nach Herkunftsorten) gruppiert. Das dazu verwendete Java-Programm liegt dieser Datenpublikation bei.

2.1.3 Ordner `12_Urkundentranskriptionen_Ansicht_HTML`

Dieser Ordner enthält HTML-Dateien, die eine weitere Ansicht der Urkundentranskriptionen ermöglichen. Sie wurden mittels XSL-Stylesheets aus den TEI-Dateien automatisch generiert und ebenfalls in 27 Ordnern (nach Herkunftsorten) gruppiert. Das XSL-Stylesheet und ein Java-Programm, welches die Transformation durchführt, sind in dieser Datenpublikation enthalten.

2.1.4 Ordner `21_Altdaten_RTF`

In diesem Ordner befinden sich die Originaldateien des Ausgangsprojekt im RTF-Format und die für die Erstellung und Anzeige benötigte Schriftart (TTF-Datei). Diese Dateien bildeten die Grundlage für die Konvertierung nach TEI. Sie sind in Ordnern gruppiert, die nach den Schreiborten benannt sind. Hier wurde die ursprüngliche Benennung beibehalten, weshalb die Namen von den normierten Ortsbezeichnungen in den zuvor beschriebenen Ordnern abweichen.

[14] https://github.com/psb1558/Junicode-font/issues/308

2.1.5 Ordner `99_Technik`

In diesem Ordner befinden sich Hilfsdateien, die vor allem für die Anzeige der Daten benötigt werden, sowie Tools, die für die Erzeugung der Daten genutzt wurden.

Im Unterordner `xslt` befinden sich die XSLT1.0-Stylesheets für die HTML-Anzeige einzelner Urkunden und von der Übersichtsdatei `MndUrk_Korpus.tei.xml` .

Der Unterordner `web` enthält Javascript und CSS-Dateien für die Darstellung und Formatierung der Texte im Browser sowie eine Kopie der Javascript-Bibliothek `prism.js`[15], die für eine formatierte Anzeige der XML-Texte verwendet wird. Die Anzeige wird über den Javascript-Code in der Datei `view-tei.html` realisiert. Außerdem wurde die Schrift Junicode2[16] für die Anzeige mittelalterlicher Sonderzeichen im Browser beigefügt.

Im Ordner `java` befinden sich die Quellcode-Dateien und eine ausführbare Version des Programms, welches die Transformation der Daten nach HTML und PDF vornimmt. Sofern ein Java-Laufzeitumgebung (mindestens Version 21) auf dem Rechner installiert ist, kann es mit folgendem Kommando gestartet werden:

```
> java -jar mndurk_transformer-1.0.0.jar <Basisverzeichnis>
```

Die Datei `MndUrk_Transkriptionen.json` enthält eine Auflistung aller Urkunden mit ihren IDs, Orts- und Dateinamen und wird für die Generierung von Auswahllisten und Menüs auf den dynamischen Webseiten verwendet.

2.1.6 Datei `MndUrk_Uebersicht.csv`

Diese `csv`-Datei im Format UTF-8 (TAB-getrennt) ist aus den TEI/XML-Daten generiert und bietet eine tabellarische Übersicht über alle 1942 Urkundentranskriptionen mit zugehörigen Metadaten: ID, Typ, Ort, Datierung, Archiv, Signatur, Kataloglink (wenn vorhanden) sowie den Pfaden zur jeweiligen XML/TEI-Datei und zur ursprünglichen RTF-Datei.

2.2 TEI-Elemente

Die Auswahl der in der Transkription verwendeten TEI-Elemente erfolgte anhand der TEI P5-Guidelines[17]. Wir beschränken uns im Folgenden auf die Beschreibung wesentlicher Aspekte und verweisen für weitere Details auf die Spezifikation.

[15] https://prismjs.com/
[16] Junicode, vgl. Fußnote 13.
[17] TEI: Guidelines for Electronic Text Encoding and Interchange P5 Version 4.9.0., https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html

2.2.1 Header

Im TEI-Header der Urkundentranskriptionen wurden die Metadaten der jeweiligen Urkunde in strukturierter Form erfasst. Sie beginnen mit einem fingierten Titel im `titleStmt`, der den Dokumenttyp und einen Identifier enthält, z.B. „*Transkription von Testament HLK 1498a*“. Es folgen Institution, Ort und Jahr der Veröffentlichung, sowie ein Verweis auf die Datenpublikation (DOI). Ein weiterer Verweis referenziert die TEI-XML-Datei mit der Beschreibung des Gesamtkorpus. Es folgen die lokale ID des Dokumentes in maschinenlesbarer Form, z.B.

```
<idno type="local">HLK_1498a</idno>
```

und die Angabe der Lizenz (CC-BY-NC 4.0).

Ein `notesStmt` enthält den Pfad zur Ausgangsdatei im RTF-Format.

In der `sourceDesc` werden die Archiv-Signatur (`msIdentifier`) und weitere bibliographischen Angaben zur Quelle erfasst. Sie enthalten

- in `repository` Angaben zum Archiv, in dessen Besitz sich die Urkunde heute befindet (mit interner ID im `key`-Attribut und ISIL im `sameAs`-Attribut)
- in `idno` die Archivsignatur des Originals
- in `date` das Ausstellungsdatum, in dessen `when`-Attribut noch einmal das Jahr in normierter Form (4-stellig) erfasst wurde, und
- im `settlement`-Element den Ausstellungsort mit einem internen Identifier im `key`-Attribut, der dem Namen des Ordners entspricht, in dem die Urkundentranskriptionen des jeweiligen Ortes abgelegt sind, und im `sameAs`-Attribut ein Link auf den Orts-Normdatensatz der GND[18].

Das folgende Beispiel demonstriert die zuvor beschriebene Struktur:

```
<msIdentifier>
   <repository key="STA HL"
      sameAs="http://ld.zdb-services.de/resource
              /organisations/DE-2392">
      Stadtarchiv Hansestadt Lübeck
   </repository>

   <idno type="sigle">Interna Appendix XVIII,
       Testamente, 1498-03-01
   </idno>
</msIdentifier>
<date when="1498">1498-03-01</date>
<settlement key="HLK__Lübeck"
   sameAs="http://d-nb.info/gnd/4036483-5">Lübeck</settlement>
```

[18] Gemeinsame Normdatei (GND), https://gnd.network/

Am Ende befindet sich eine `profileDesc`, in der in einem `keywords`-Element noch einmal der Dokumenttyp in maschinenlesbarer Form abgelegt wurde:

```
<keywords><term type="genre" key="TST">
   Testament</term></keywords>
```

Der TEI-Header der Datei `MndUrk_Korpus.tei.xml` enthält die Metadaten des Transkriptions-Projektes. Sie umfassen Projekttitel, beteiligte Personen, Institution, Ort und Jahr der Publikation, Angaben zur Lizenz und die DOI der Datenpublikation.

2.2.2 Body

Im TEI `<body>` wurden nur wenige Elemente zur Strukturierung des Textes verwendet. Absätze wurde durch `<p>`-Elemente gruppiert.

In Einzelfällen wurden in einer Datei mehrere Urkunden oder Urkundenfragmente transkribiert. Zur Trennung wurde ein `<milestone>`-Element mit einem Bezeichner im `n`-Attribut verwendet, z.B:

```
<milestone unit="part" n="Zweite Urkunde" />
```

Fußnoten wurden im Text mit dem `<note>`-Elemente ausgezeichnet und das Label im n-Attribut gespeichert, z.B.:

```
<note xml:id="ftn1" place="foot" n="1">
  Seitenrand beschädigt</note>
```

Über das `<ref>`-Element werden Verweise auf andere Fußnoten innerhalb eines Dokumentes realisiert.

Eine Besonderheit in diesem Projekt ist die mehrfache Verwendung von Fußnoten. Man sieht in einem Text, dass dasselbe Fußnotenlabel an mehreren Stellen verwendet wird. Im TEI-XML wurden für das zweite und weitere Vorkommen leere `<note>`-Elemente mit dem Attribute `sameAs` benutzt, die auf die zuvor definierte Fußnote verweisen, z.B.:

```
<note place="foot" sameAs="#ftn1" />
```

Hervorhebungen im Text wurden mittels `<hi>`-Element realisiert. Für die Art der Hervorhebung wurde das `rend`-Attribut verwendet. Es kann einen oder mehrere der folgenden Werte annehmen:

`bold, italic, underline, strikethrough, smallcaps, sup.`

2.3 Software für die HTML-Anzeige

Die Datenpublikation wurde so aufbereitet, dass sie mit einem gängigen Webbrowser angezeigt werden kann. Für die Präsentation werden XSLT 1.0 Stylesheets verwendet, die von heutigen Webbrowsern ohne weitere Hilfsmittel verarbeitet werden können. Sicherheitsbeschränkungen der Browserhersteller verhindern jedoch, dass dynamische Webseiten (mit Javascript) direkt von der Festplatte des lokalen PCs geladen werden können. Deshalb ist es notwendig, die ZIP-Datei zu entpacken und das Verzeichnis über einen etablierten Webserver (z.B. Apache2[19] oder NginX[20]) bereitzustellen.

Alternativ besteht die Möglichkeit, einen einfachen Webserver auf dem eigenen PC zu starten. Wenn eine Java-Umgebung (mind. Version 21) auf dem PC installiert ist, kann in dem entpackten Ordner das Programm `jwebserver`[21] über die Datei `start.bat` ausgeführt werden. Alternativ kann in einer Python-Umgebung das Programm `http.server`[22] in dem entpackten Ordner gestartet werden. Anschließend lässt sich die Publikation im Browser über die Adresse http://localhost:8000/ öffnen.

Es öffnet sich eine Startseite, auf der eine bestimmte Transkriptionsdatei ausgewählt werden kann. Über das darin verknüpfte XSL-Stylesheet wird eine einfache HTML-Ansicht generiert. Die Formatierung erfolgt über CSS-Dateien. Für die Interaktion (Menüs, Anzeige von XML-Quellcode) wurde Javascript verwendet.

[19] https://httpd.apache.org/
[20] https://nginx.org/
[21] https://docs.oracle.com/en/java/javase/21/docs/specs/man/jwebserver.html
[22] https://docs.python.org/3/library/http.server.html#command-line-interface