Ugly is better than nothing:
Generieren von Titelaufnahmen aus den METS/MODS-Daten digitalisierter Handschriften zur Übernahme in den K10Plus

Ulrike Mehringer / 2023

„Ja gibt's denn keinen Knopf dafür?" – wer kennt es nicht, das verwunderte Kopfschütteln, wenn man anfängt, Arbeitsabläufe zu erklären. Tatsächlich gibt es im Bibliotheksalltag sehr viel weniger Knöpfe als gemeinhin angenommen und viele Bibliotheksanwendungen existieren unverbunden nebeneinander her. Dies ist der Versuch, in der UB Tübingen einen bisher fehlenden „Knopf" zu schaffen, oder zumindest ein Surrogat.

1. Ausgangssituation

Die Universitätsbibliothek Tübingen setzt im Bereich der Digitalisierung eine angepasste Version der Workflowsoftware DWork[1] ein, eine Eigenentwicklung der Universitätsbibliothek Heidelberg. DWork unterstützt u.a. die Erfassung von bibliografischen und Strukturdaten, DOI-Vergabe, das Einlesen der Scan-Dateien und den Export aller Daten zur Online-Präsentation OpenDigi. Ein Digitalisat in Dwork/OpenDigi wird als „Projekt" bezeichnet. Der Projektname ist frei wählbar und wird nach Tübinger Konventionen aus der Signatur gebildet.

Nachdem die Bearbeitung in DWork abgeschlossen ist und das Projekt nach OpenDigi exportiert wurde, wird das Handschriftendigitalisat im K10Plus katalogisiert, damit es überregional und im lokalen Bibliothekssystem auffindbar ist. Die Katalogisierungssoftware für die Erfassung im K10Plus ist WinIBW. Es ermöglicht die Erstellung von Datenmasken, d.h. von Vorlagen mit den benötigten Feldern und gleichbleibenden Inhalten. Die Bearbeiterin ruft die Datenmaske in WinIBW auf und ergänzt aufwändig manuell bzw. per copy+paste die verstreuten variablen Feldinhalte von verschiedenen Stellen in DWork und OpenDigi. Außerdem muss bei Personen, Orten und Schlagwörtern noch die Verknüpfung zum Normsatz hergestellt werden, bei mehrteiligen Werken zum übergeordneten Werk.

2. Aufgabenstellung

Ziel ist ein Python-Skript, das nach Eingabe eines Projektsnamens alle nötigen bibliografischen Angaben aus den METS/MODS-Daten ausliest und daraus eine fertige Titelaufnahme im Pica3-Format für den K10Plus erstellt. Das Skript soll bei Vorliegen einer GND-Nummer die entsprechende K10Plus-ID (PPN) des Normdatensatzes einfügen. Das Skript soll außerdem Stücke eines mehrteiligen Werkes erkennen, und – wenn schon vorhanden – die Verknüpfung zum übergeordneten Werk herstellen. Die fertige Titelaufnahme kann dann von der Bearbeiterin durch ***einmaliges*** copy+paste ohne weitere Nachbearbeitung in die WinIBW übernommen werden. Ein zweites Skript soll eine Konkordanz mit Projektnamen und PPNs der neuen Titelaufnahmen eines Zeitraumes erstellen, anhand derer die PPNs dann wieder in DWork eingepflegt werden können.

---

[1] https://www.ub.uni-heidelberg.de/helios/digi/dwork.html

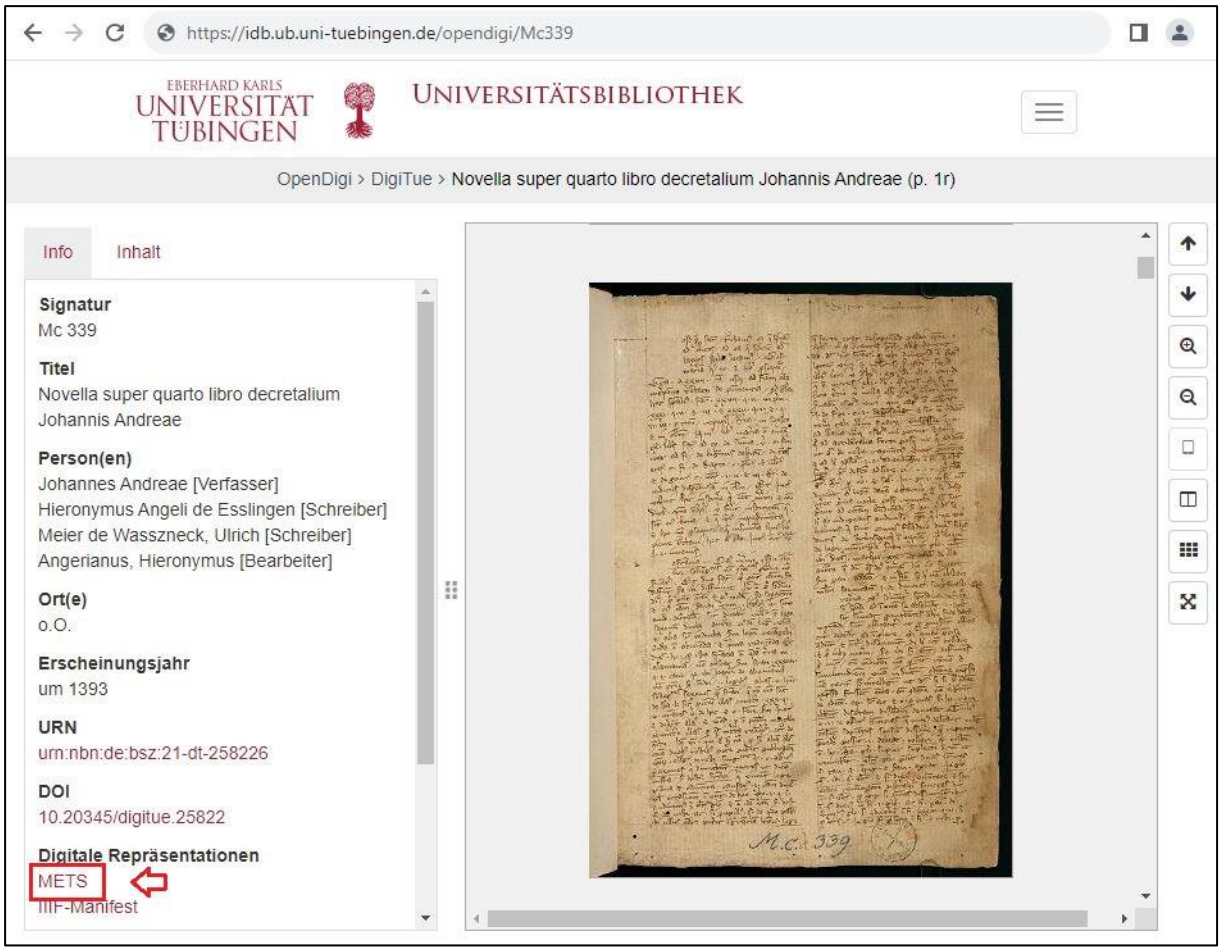

Abbildung 1 (oben): Präsentation des Projekts Mc339 in OpenDigi, Link auf XML-Datei mit METS/MODS-Daten

```
0500 Gattung
0501 Text$btxt
0501 unbewegtes Bild$bsti
0502 Computermedien$bc
0503 Online-Ressource$bcr
1100 Entstehungsjahr$nVorlageform Entstehungsjahr
1101 crxcn ---apaup
1109 Jahr der Digitalisierung
1131 !10457187X!
2050 URN
2051 DOI
1500 Sprache$aSprache
1505 $erda
3010 !PPN!Person$BFunktion$4Kürzel
4000 Titel/Handschrift – Universitätsbibliothek Tübingen, Signatur$dZusatz
4022 Online-Ausgabe
4040 !PPN!Entstehungsort$4prp
4046 Entstehungsort$nErzeugername$hDatierung
4048 Tübingen$nUniversitätsbibliothek
4060  Umfang
4065 XA-DE$cUB Tübingen$aSignatur
4068 1 Online-Ressource (xyz Seiten)
4150 Titel übergeordn. Werk$lZählung
4160 !PPN!übergeordn. Werk$lZählung
4201 Kurzaufnahme einer Handschrift
4233 $aaa$cDigitalisierungsjahr$5DE-21
4950 http://idb.ub.uni-tuebingen.de/opendigi/Projektname$xD$3Volltext$4LF$534
4950 http://nbn-resolving.de/URN$xR$3Volltext$534
4950 https://doi.org/DOI$xR

7100 $Dn
8012 ditu$adbok$ahssa
```

Abbildung 2 (links): Zielformat Pica3 für die Eingabe in K10Plus

***rot** = variable Feldinhalte*

***schwarz** = gleichbleibende Feldinhalte*

***blau** = Feldnamen*

3. Beschreibung der Ausarbeitung

3.1 Eingabe

Der Projektname wird über ein GUI eingegeben und zunächst auf einfache Syntaxfehler geprüft: leere Eingabe, Eingabe eines oder mehrerer Leerzeichen, Eingabe von Umlauten und ß. Gibt es unter dem eingegebenen Namen keine METS/MODS-Datei, öffnet sich ein weiteres Fenster mit einer Fehlermeldung. Im Eingabefenster bleibt die falsche Eingabe zur Information stehen. Ist der Projektname korrekt, wird die Eingabezeile geleert und der nächste Projektname kann eingegeben werden. Über den Projektnamen werden die METS/MODS-Daten heruntergeladen und in eine XML-Datei gespeichert.

Das GUI wird mit der Python-Library Tkinter realisiert. Tkinter ist das Interface zwischen Python und Tcl/Tk, einem toolkit für die Programmierung plattformunabhängiger Benutzer-oberflächen. Tkinter ist leicht verständlich und leicht verfügbar, und ist hier das Mittel der Wahl, um ein GUI ohne besondere Anforderungen in kurzer Zeit umzusetzen. Für den Ablauf ist zu beachten, dass ein Python-Skript beim Aufruf der Eventloop zunächsts gestoppt wird und erst weiterläuft, wenn das Fenster beendet wird. Deshalb wird für die gesamte Verarbeitung der Eingabe und die Ausgabe der Titelaufnahme die Funktion mets_to_k10plus() definiert und als command beim Anklicken des Buttons innerhalb der Eventloop ausgeführt.

3.2 Funktionen und Verarbeitung

XML-Dateien sind hierarchisch aufgebaut und weisen eine Baumstruktur auf. Die Struktur gliedert sich in Elemente auf, die Werte und Attribute enthalten können. Der ElementTree-Parser in Python analysiert und segmentiert die XML-Datei mit den METS/MODS-Daten so, dass die einzelnen Elemente, ihre Werte und Tabellen, abgefragt und verarbeitet werden können. Eine Problematik liegt darin, dass METS/MODS keine leeren Elemente liefert. Da in Python Variablen nicht vordefiniert werden, produziert eine Abfrage auf eine nicht vorhandene Variable einen Fehler. So ist man in vielen Fällen gezwungen, zunächst das Vorhandensein einer Variablen abzufragen, bevor der Wert abgefragt werden kann. Dass es bei der Erfassung in DWork keine Prüfung auf Pflichtfelder gibt (Pflichtfelder sind nur „verbal“ formuliert), ist eine große Fehlerquelle für das Skript.

Nach dem Parsing werden aus den Elementen die Inhalte für die einzelnen K10Plus-Felder generiert.

Die Erfassung von Personen, normierten Orten und Schlagwörtern erfordert im K10Plus die Verknüpfung zur K10Plus-ID (PPN). In DWork werden für Personen und Schlagwörter aber nur GND-Nummern erfasst, und dies nicht obligatorisch. Liegt eine GND-Nummer vor, liest das Skript sie mit der Funktion gnd_to_ppn() aus, macht damit eine Abfrage über die SRU-Schnittstelle des K10Plus nach dem Normsatz, und liest aus diesem die PPN aus. Für den normierten Ort wird bisher leider keine GND-Nummer in DWork erfasst, so dass die Verknüpfung ein Problem darstellt. Eine Lösung über den Namen des Ortes konnte noch nicht realisiert werden, weil zunächst eindeutige Merkmale für den richtigen Datensatz gefunden werden müssen. Die künftige Erfassung der GND-Nummer in DWork wäre eine Lösung.

Die SRU-Schnittstelle zum K10Plus ist hier am besten geeignet, weil sie die benötigte Abfrage erlaubt (nach einem bestimmten Suchattribut über den gesamten Bestand) und keine Authentifizierung erfordert.

Bei mehrteiligen Werken wird für jeden Band und die Gesamtaufnahme (GA) jeweils ein eigenes Projekt in DWork angelegt, dies enstpricht der hierarchischen Aufnahme im K10Plus. Die Titelaufnahme von Band und GA sind über die PPN der GA verknüpft. Die Funktion mw_ppn() liest aus den METS/MODS-Daten der GA die PPN aus (falls vorhanden) und fügt sie in die Titelaufnahme des Bandes ein.

3.3 Ausgabe und das ergänzende Skript PPN_List.py

Alle Felder und Inhalte, die in der Funktion mets_to_k10plus() generiert werden, werden in eine Textdatei ausgegeben. Die Textdatei wird nach Feldnamen sortiert. Dies wäre für die Eingabe in K10Plus nicht notwendig, erleichtert der Beabeiterin aber die Sichtkontrolle und das Erkennen von Fehlern und Hinweisen. So wird z.B. über das Attribut „swb-ppn“ geprüft, ob schon eine Titelaufnahme im K10Plus vorhanden ist. Ist das der Fall, wird die Titelaufnahme vom Skript trotzdem erzeugt, enthält aber eine Warnung. Fehlende Elemente wie der normierte Ort oder eine nicht vorhandene Gesamtaufnahme werden durch *** markiert. Hier muss die Bearbeiterin im K10Plus manuell nacharbeiten.

Das ergänzende Skript PPN_List.py sucht alle neuen Titelaufnahmen eines Zeitraums und erstellt eine Liste mit Projektnamen und PPN der Titelaufnahmen. Die PPNs müssen abschließend von der Bearbeiterin wieder in DWork ergänzt werden, dank der Liste ist das aber „am Stück“ möglich.

4. Folgeaktivitäten

Obwohl das Skript METS_to_K10plus.py jetzt schon gut einsetzbar ist und seinen Zweck im wesentlichen erfüllt, gibt es noch 23 Punkte auf der Do-List, die zu erledigen sind. Darunter einige, die unabdingbar für den Echtbetrieb sind:

- Portierung nach Windows und der Start des Skripts über ein Icon auf dem Desktop
- Verbindungsfehler sauber abfangen, Fehlermeldung für Endnutzer ergänzen
- Bei der Eingabe von Projektnamen alle Sonderzeichen außer Bindestrich und Unterstrich abfangen
- Umgang mit persönlichen Namen, die Komma enthalten: Christoph, Württemberg, Herzog
- Verknüpfung mit dem normierten Ort in Feld 4040
- Personen mit Funktionsbezeichnung „Schreiber“ in Feld 4046 $n ausgeben
- Bei mehreren Entstehungsorten Feld 4046 wiederholen oder nur den ersten Ort ausgeben
- Probleme mit Sachschlagwörtern lösen/reparieren lassen
- Umsetzung von sprechenden Entstehungsjahren, z.B. „circa 15. Jahrhundert“, wenn Sortierzählung fehlt
- Körperschaften als Urheber und Mitwirkende in Feld 3110 ausgeben
- Links auf Répertoire International des Sources Musicales (RISM) und andere Onlinequellen auslesen und umsetzen, RISM-Nummer nach Feld 2277
- Seitenzahlen bei Sammelhandschriften (Problem mehrfach beginnender Seitenzählung)
- Konkordanz PPN_List.py: Fenster für Benutzereingabe von Zeitraum, Datum oder „heute“

Auf der Longlist stehen Aufgaben wie das Einlesen mehrerer Projektnamen in Listenform (die aus DWork erzeugt werden könnte), die Umsetzung von Titeln in nicht-lateinischen Schriften und die Identifizierung von enthaltenen Werken in den Strukturdaten für Feld 4222.

5. Ergebnis und Bewertung

- **METS_to_K10plus.py** (erzeugt eine Titelaufnahme aus METS/MODS-Daten): https://github.com/U-Mehringer/2022-2023-Data_Librarian_Ulrike_Mehringer/blob/main/Modul_6/prog/METS_to_K10plus.py
- **PPN_List.py** erzeugt eine Konkordanz Projektname-PPN: https://github.com/U-Mehringer/2022-2023-Data_Librarian_Ulrike_Mehringer/blob/main/Modul_6/prog/PPN_List.py
- **Projektnamen zum Testen:** Mc339, MaVII50, Mn1-245, Md2, MaI390, MaVI68, MaIX11

Das Skript METS_to_K10plus ist weit davon entfernt, ein „Knopf" zu sein, der alles von alleine macht oder wirklich eine Verbindung schafft, und vielleicht wird es das nie. Im Moment sind weiterhin manuelle Tätigkeiten im Ablauf vorhanden: Eingabe der Projektnamen, copy+paste, Verknüpfung mit dem normierten Ort, und die Eingabe der PPN in DWork. Aber die Katalogisierung von Handschriften-Digitalisaten wird durch die beiden Skripte enorm beschleunigt und der gute Rat „Automate the boring stuff"[2] konnte zufriedenstellend umgesetzt werden. Wie so oft hängt die Qualität der erzeugten Titelaufnahme nicht unerheblich von der Qualität der Datenerfassung ab.

Ob sich der Programmieraufwand gelohnt hat, wird man an der Langlebigkeit des Skriptes messen müssen, und diese wiederum hängt nicht nur von der Bereitschaft ab, das Skript zu pflegen und aktuell zu halten, sondern auch von Faktoren wie künftig verfügbaren Schnittstellen, IT-Infrastruktur, oder veränderten Datenflüssen und Arbeitsabläufen.

6. Hilfreiche Quellen

- Schnittstellen zum K10Plus: https://wiki.k10plus.de/display/K10PLUS/Schnittstellen
- DDB-METS/MODS: https://wiki.deutsche-digitale-bibliothek.de/pages/viewpage.action?pageId=19006651
- Python.org für ElementTree u.a.: https://docs.python.org/3/library/xml.etree.elementtree.html
- Python-lernen.de für Tkinter u.a.: https://www.python-lernen.de/tkinter-button.htm
- Python Tkinter Youtube-Tutorial von "Programmieren starten": https://youtu.be/d2HT7i9OrHE
- Stack Overflow: https://stackoverflow.com/
- Python-forum.de: https://www.python-forum.de/
- DelftStack: https://www.delftstack.com/de/

[2] Sweigart, Al: Automate the Boring Stuff with Python. - https://automatetheboringstuff.com/

Anhang: Funktion gnd_to_ppn()

```
# Funktion gnd_to_ppn: K10Plus-ID (PPN) über die GND-Nummer im K10Plus
abrufen
def gnd_to_ppn(gnd_nr):
    # Normsatz aus K10Plus auslesen
    url = "http://sru.k10plus.de/opac-de-
           627!rec=2?&operation=searchRetrieve&
           query=pica.nid="+gnd_nr+"&maximumRecords=1
           &recordSchema=mods"
    gnd_xml = "../data/gnd.xml"

    try:
        urllib.request.urlretrieve(url,gnd_xml)
        # XML parsen
        gnd_tree = ET.parse(gnd_xml)
        gnd_root = gnd_tree.getroot()
        # K10Plus-PPN aus gnd_xml auslesen
        k10plus_ppn =
        gnd_root.find(".//{http://www.loc.gov/mods/v3}recordIdentifier")
        if k10plus_ppn is None:
            k10plus_ppn ="nn"
        else:
            k10plus_ppn = k10plus_ppn.text

    except urllib.error.HTTPError as err:
        k10plus_ppn = "nn"

    return(k10plus_ppn)
```