

ABSCHLUSSBERICHT

1 Allgemeine Angaben

DFG-Geschäftszeichen: BO 3288/18-1, GE 3075/8-1, STU 530/10-1

Projektnummer: 460478737

Titel des Projekts: Integration von Kitodo und OCR-D zur produktiven
Massendigitalisierung

Namen der Antragstellenden: Dr. Achim Bonte, Dr. Sabine Gehrlein, Katrin Stump
(Hinweis: Amtsnachfolgerin von Dr. Achim Bonte in Dresden ist Katrin Stump,
Amtsnachfolger von Katrin Stump in Braunschweig ist Robert Strötgen)

Dienstanschriften:

Sächsische Landesbibliothek – Staats- und Universitätsbibliothek Dresden,
Zellescher Weg 18, 01069 Dresden

Universitätsbibliothek Mannheim, Schloss Schneckenhof West, 68131 Mannheim

TU Braunschweig – Universitätsbibliothek, Universitätsplatz 1, 38106 Braunschweig

Name(n) der Mitverantwortlichen: –

Name(n) der Kooperationspartnerinnen und -partner: –

Berichtszeitraum (gesamte Förderdauer): 10.2021 – 03.2024

2 Zusammenfassung/Summary

Die DFG strebt mit der „Koordinierten Förderinitiative zur Weiterentwicklung von Verfahren der Optical Character Recognition“ (OCR-D) die Transformation der im deutschen Sprachbereich erschienenen Drucke des 16. bis 18. Jahrhunderts (VD 16, VD 17 und VD 18) in maschinenlesbare Form an. Das vorliegende Projekt ist Teil der dritten Förderphase, in der die bisher entwickelte Technologie in die praktische Anwendung integriert werden sollte. Im Fokus stand daher die Integration von OCR-D in die Arbeitsabläufe bei der Digitalisierung mit der Webapplikation Kitodo.Production und in die Visualisierung der Digitalisate mit Kitodo.Presentation.

Das Projekt wurde in ständigem Kontakt mit den potenziellen Endnutzern aus der Community entwickelt. Um ihre Wünsche und Bedürfnisse zu berücksichtigen, wurden mehrere Umfragen, Konferenzen und kontinuierliche praktische Tests vorbereitet und durchgeführt. Im Ergebnis entstand ein passendes Anwendungsschema und ein produktiver Workflow für die automatische Texterkennung und Publikation, der von anderen Kitodo-Anwendern nachgenutzt werden kann.

Kitodo.Presentation und der darauf aufbauende DFG-Viewer können dank der Projektergebnisse nun für beliebige Digitalisate direkt Volltexte erzeugen, wenn noch keine OCR-Ergebnisse vorliegen oder diese nicht die gewünschte Qualität aufweisen. Die öffentliche Testinstanz des DFG-Viewers an der UB Mannheim bietet hierfür einen OCR-D-Prozess an, der den im Projekt entwickelten OCR-Server nutzt. Alternativ können OCR-Prozesse mit Tesseract oder Kraken gewählt werden.

Summary

With the “Coordinated Funding Initiative for the Further Development of Optical Character Recognition Methods” (OCR-D), the DFG is striving to transform the prints published in the German language area from the 16th to 18th centuries (VD 16, VD 17 and VD 18) into machine-readable form. The present project is part of the third funding phase, in which the technology developed so far should be integrated into practical application. The focus is therefore on the integration of OCR-D into the workflows for digitization with the web application Kitodo.Production and in the visualization of digitization with Kitodo.Presentation.

The project was developed in constant contact with potential end users from the community. To consider their wishes and needs, several surveys, conferences and continuous practical tests were prepared and carried out. The result was a suitable application scheme and a productive workflow for automatic text recognition and publication that can be reused by other Kitodo users.

Thanks to the project results, Kitodo.Presentation, and the DFG Viewer based on it, can now directly generate full texts for any digital copy if OCR results are not yet available or do not have the desired quality. The public test instance of the DFG Viewer at the UB Mannheim offers an OCR-D process for this purpose, which uses the OCR server developed in the project. Alternatively, OCR processes with Tesseract or Kraken can be chosen.

3 Arbeits- und Ergebnisbericht

3.1 Ausgangslage und Zielsetzung des Projektes

Technischer Ausgangspunkt waren die beiden unabhängig voneinander existierenden Open-Source-Software-Suiten OCR-D und Kitodo: OCR-D als Arbeitsplatzsystem-Kommandozeilen-Prototyp ohne Netzwerkschnittstellen für Verteilung und Skalierung, ohne Fehlerbehandlung, ohne dynamische Workflow-Verarbeitung, ohne dynamische Qualitätsabschätzung und ohne Unterstützung bei der Workflow-Konfiguration – Kitodo wiederum ohne eigenes OCR-Modul und ohne eine einfache Möglichkeit zur Einbindung von OCR-D.

Folglich hatte das Projekt die folgenden Ziele: OCR-D sollte zu einem verteilten System weiterentwickelt werden, bestehend aus einem zentralen Server und angebotenen Servern für Prozessierungsaufgaben, jeweils als flexibel einsetzbare Softwarecontainer. Weiterhin sollte durch eine automatische Qualitätsabschätzung von Zwischenergebnissen eine qualitätsbezogene Workflow-Optimierung realisiert werden, sowie darauf aufbauend eine manuell optimierte Vorauswahl von Konfigurationen. Zudem sollte OCR-D als Modul oder Workflow-Schritt in Kitodo integriert werden - mit Workflow-Auswahl, Datenübergabe, Ausführungsüberwachung und Fortschrittsanzeige in Kitodo.Production, bzw. Versionierung und direkter Anzeige der Ergebnisse in Kitodo.Presentation.

3.2 Arbeitsschritte im Berichtszeitraum

Im Projekt wurde OCR-D so in die Kitodo-Suite (bestehend aus Kitodo.Production und Kitodo.Presentation) integriert, dass es den ermittelten Anforderungen der Kitodo-Community gerecht wird (siehe 3.2.2). Hierbei wurde gemäß den Planungen eine verteilte Architektur konzipiert und umgesetzt. Durch die Verwendung definierter Schnittstellen wurde eine enge technische Verzahnung der beiden Systeme vermieden, um zukünftige Weiterentwicklungen nicht durch technische Abhängigkeiten zu beeinträchtigen.

Die ursprüngliche Architektur wurde in den identifizierten Funktionsbereichen (technische Administration, fachliche Verantwortlichkeit und Rechentechnik) ausdifferenziert. Neben dem Server für die OCR-D-Prozessierung (*Controller*) wurden ein administratives Backend (*Mana-*

ger) und ein fachliches Frontend (*Monitor*) implementiert (siehe 3.2.3). Letzteres enthält Komponenten zur Steuerung von OCR-D-Workflows sowie zur Visualisierung von Jobs und Ergebnissen. Sowohl in Kitodo.Production als auch Kitodo.Presentation wurden die gleichen Infrastrukturkomponenten integriert und nutzbar gemacht. Im speziellen Anwendungsfall des DFG-Viewers lag der Fokus der Implementierung auf der Usability einerseits und einer leichtgewichtigen OCR (auch mit anderen OCR-Engines als Optionen) andererseits, da in diesem Kontext keine institutionsspezifischen Anforderungen zur Anwendung kommen (siehe 3.2.10). Es wurde eine DFG-Viewer Instanz mit den Projektergebnissen zugänglich gemacht. Die Integration in Kitodo.Production wurde kontinuierlich getestet und im Betrieb erprobt (siehe 3.2.8).

Durch die Kapselung und Virtualisierung der Module, eine flexible Konfiguration und eine ausführliche Dokumentation konnte die Komplexität des Systems eingegrenzt werden. Jedoch war die Implementierung durch Probleme in der Zusammenarbeit mit dem Koordinierungsprojekt beeinträchtigt (siehe 3.2.1), und weiterführende Ziele konnten aufgrund massiver ungeplanter Mehraufwände in der Entwicklung von OCR-D Kernkomponenten nicht zufriedenstellend umgesetzt werden (siehe 3.2.8).

Die Implementierung ermöglicht eine funktionale Integration in einem breiten Anwendungsspektrum. Auf Basis der noch fertig zu stellenden WebAPI muss die bisherige provisorische Implementierung für den Einsatz als nationaler Dienst für die Massendigitalisierung noch einmal nach Projektende gründlich überarbeitet werden.

3.2.1 AP 1: Projektkoordination und -kommunikation (SLUB Dresden)

Im Rahmen des Projekts bestand qua Konstitution die Notwendigkeit für eine intensive Abstimmung, die von erheblichen externen Abhängigkeiten geprägt war. Dies betraf insbesondere die Anwendergemeinschaft von Kitodo in organisatorischer Hinsicht (siehe 3.2.2), das Release-Management von Kitodo in technischer Hinsicht (siehe 3.2.5) sowie technisch wie organisatorisch das Koordinierungsprojekt von OCR-D. Zudem waren auch Mitwirkende in parallelen Projekten (Modul- und Implementierungsprojekte) sowie der OCR-D-Beirat in diesen Abstimmungsprozess eingebunden.

Gemeinsam und arbeitsteilig mit dem OCR-D-Koordinierungsprojekt sollten in der ersten Phase des Projektes die Schnittstellen und Formate weiterentwickelt werden, die die Grundlage für die im Projekt geplanten Integrationen sind. Dies sollte über die in Phase 2 bewährte, GitHub-basierte Kollaboration erfolgen. Zudem waren von Seiten des OCR-D-Koordinierungsprojektes noch Nacharbeiten an den prototypisch vorliegenden Kernkomponenten erforderlich.

Da die projektierte Mitarbeit des OCR-D-Koordinierungsprojektes nicht ausreichend und zu spät erfolgte, entstand im Projekt ein hoher Mehraufwand für die Weiterentwicklung sowohl der Kernkomponenten als auch der Schnittstellen, der zu Abstrichen an den Projektzielen führte (s. 3.2.3 und 3.2.4). Die aus Sicht dieses Projektes unzureichende Unterstützung wurde mehrfach an das OCR-D-Koordinierungsprojekt und den OCR-D-Beirat adressiert.

Aufgrund der bis zum Ende dieses Projektes fehlenden, vom OCR-D-Koordinierungsprojekt zu implementierenden WebAPI wurde eine einfache Zwischenlösung implementiert, die im Sinne der Produktreife als vorläufig einzustufen ist und den Anforderungen an Robustheit und Skalierbarkeit im Produktivbetrieb nicht gerecht wird. Auf Basis der ggf. noch nach Ende dieses Projektes finalisierten WebAPI müsste die bisherige provisorische Implementierung nach Projektende überarbeitet werden. Ob und wie das außerhalb des Projekts möglich ist, müssen Kitodo-Verein und -Community und ORC-D-Einrichtungen noch klären.

3.2.2 AP 2: Kitodo-Community-Management (UB Braunschweig)

Im Rahmen dieses APs wurde für die Mitglieder des Kitodo-Vereins und der Kitodo-Community im November 2021 der Workshop „Anforderungen und Rahmenbedingungen für den Einsatz von OCR-D in Kitodo“ durchgeführt, welcher von der Umfrage „Anforderungserhebung Kitodo/OCR-D“ begleitet war. Die Ergebnisse der Umfrage sind entsprechend beschrieben, grafisch dargestellt und detailliert analysiert¹. Diese Ergebnisse dienten später der Entwicklung der Kitodo/OCR-D-Integration und haben es ermöglicht, die integrierte Lösung benutzerfreundlich auszubauen.

Die nächste Phase des Arbeitspakets adressierte die aktive Zusammenarbeit mit den Digital Humanities (DH) und relevanten Communities. Um die Anforderungen solcher Communities zu ermitteln, wurde eine DH-spezifische Umfrage erstellt. Bereits bei der Vorbereitung dieser Umfrage gab es mehrere Rücksprachen mit Mitarbeitenden von DH-Institutionen (z. B. Georg-Eckert-Institut Braunschweig, Herzog-August-Bibliothek Wolfenbüttel). Als Ergebnis dieser Rücksprachen konnten Fragestellungen ausgearbeitet werden, die nicht nur für erfahrene Kitodo-Anwender, sondern auch für Nutzende von Kitodo-fremder Digitalisierungssoftware und für Anwender ohne umfassende Erfahrung in den Bereichen Digitalisierung und Texterkennung anwendbar sind. Dies hat es ermöglicht, die Anforderungen eines breiten Nutzerkreises aus dem Bereich „Digital Humanities“ zu untersuchen. Die Umfrage „Kitodo/OCR-D Integra-

¹ <https://doi.org/10.24355/dbbs.084-202205311213-0>

tion für Digital Humanities“ wurde in verschiedenen Communities verteilt (Georg-Eckert-Institut Braunschweig, NFDI4Culture, Text+, NFDI4Memory, DHd-AG OCR usw.). Die Ergebnisse der Umfrage sind entsprechend zugänglich².

Den Abschluss des APs bildete der Workshop "Integration von OCR-D in Kitodo: vom Digitalisat bis zur Veröffentlichung der Texte". Hier wurden die Workflowauswahl, das Starten der Texterkennung als Kitodo-Aufgabe, Prozessstatus, Visualisierung und Bewertung der Ergebnisse sowie die anschließende (automatische) Veröffentlichung der Texte behandelt.

Alle Projektergebnisse wurden entsprechend dokumentiert. Dazu gehört die Anwender- und Administratordokumentation als GitHub-Website³ und im GitHub-Wiki⁴, in der alle notwendigen Schritte beschrieben sind, um die integrierte Kitodo/OCR-D Lösung effektiv nutzen zu können sowie die Texterkennung und Veröffentlichung der digitalisierten Medien zu automatisieren. Die Einbindung der Community wurde somit gut umgesetzt, die noch eingeschränkte Produktreife (siehe Bericht AP 1) hat im AP eine weitere Unterstützung bei der Einführung aber noch nicht möglich gemacht und ist zukünftig durch den Kitodo-Verein zu leisten.

3.2.3 AP 3-4: Entwicklung einer technischen Detailspezifikation, Entwicklung eines OCR-D-Servers für den verteilten Betrieb (SLUB Dresden)

Die technische Spezifikation konnte ausgehend vom Projektantrag ausgearbeitet und den ermittelten Anforderungen entsprechend ergänzt werden. Sie ermöglicht eine Kopplung ohne wechselseitig technische Abhängigkeiten und ist so generisch in Kitodo.Production (siehe 3.2.9), Kitodo.Presentation (siehe 3.2.10) und in weitere Quellsysteme integrierbar. Neben der zentralen OCR-D-Komponente (=Controller⁵) mit allen OCR-D-Modulen wurden zusätzlich eine koordinierende Komponente (=Manager⁶) und eine Anwendung zur fachlichen Überwachung (=Monitor⁷) eingeführt. Letztere ist vollständig als Webserver implementiert, die vorigen hingegen setzen auf die zu Projektbeginn verfügbare bewährte Kommandozeilen-Lösung⁸ und realisieren die Netzwerkschnittstelle mittels eines speziell gekapselten OpenSSH-Servers⁹.

Alle Komponenten sind individuell erweiterbar und austauschbar. Eine Container-Virtualisierung ermöglicht einfaches und flexibles Ausrollen, Continuous Integration mit Integrationstest

² <https://doi.org/10.24355/dbbs.084-202305090908-0>

³ https://slub.github.io/ocrd_kitodo/

⁴ https://github.com/slub/ocrd_kitodo/wiki

⁵ https://github.com/slub/ocrd_controller

⁶ https://github.com/slub/ocrd_manager

⁷ https://github.com/slub/ocrd_monitor/

⁸ https://github.com/OCR-D/ocrd_all, <https://ocr-d.de/core/api/ocrd/ocrd.cli.html>

⁹ https://github.com/slub/ocrd_controller/blob/master/start-sshd.sh

und partiellen Funktionstests garantiert Wartbarkeit. Insbesondere der *Controller* ist stark von den Entwicklungen in OCR-D abhängig und kann durch zukünftige Entwicklungen profitieren. Die Aufteilung der Funktionsbereiche und der spezifizierte Datenaustausch ermöglichen den Einsatz in verschiedenen Szenarien, sei es komplett lokal in einer Einrichtung, teilweise verteilt oder vollständig extern. Institutionen mit umfangreichem IT-Knowhow können alle Teile der Infrastruktur selbst hosten. Kleinere Einrichtungen mit begrenzten IT-Ressourcen können sich hingegen auf fachliche Aspekte konzentrieren und Rechenkapazität sowie Expertise auslagern. Ein dokumentiertes GitHub-Repository¹⁰ steht als Demonstrator für alle Anwendungsszenarien und als Zusammenführung der Einzelkomponenten zur Verfügung. Es dient als Proof-of-Concept der Projektziele und zeigt die Integration beispielhaft an einem vorkonfigurierten Kitodo.Production.

3.2.4 AP 5-7: Konzeption einer automatischen Prozesssteuerung in OCR-D, Implementierung einer automatischen Prozesssteuerung in OCR-D, Entwicklung von Metriken für eine automatische Qualitätsabschätzung (SLUB Dresden)

Die unzureichenden technischen Voraussetzungen (s. 3.2.1) wirkten sich besonders auf die APs 5-7 aus. Eine automatische Prozesssteuerung ist ohne API-Änderungen für eine Fehlerbehandlung¹¹ in Modulprozessoren und ohne ein Workflow-Format¹² mit Kontrollstrukturen und dynamischer Ausführung nicht umsetzbar; Werkzeuge für eine automatische Qualitätsabschätzung sind im Funktionsmodell ohne dezidierte Evaluator-Schnittstellenspezifikation¹³ nicht direkt einsetzbar. Da dieser Projektteil keine essenzielle Funktionalität bereitstellen würde, mussten angesichts der Mehraufwände für Planung, Entwicklung, Pflege und Betreuung der OCR-D-Kernkomponenten (s. 3.2.1) die Ziele stark reduziert werden.

Als Kompensation wurde – maßgeblich unterstützt durch die UBBS – den Nutzenden in der Weboberfläche des Monitors eine Möglichkeit gegeben, Workflows sukzessive manuell zu optimieren und variieren: Zum einen durch direkte Integration des OCR-D-Browsers¹⁴ für die vollständige Visualisierung jeglicher struktureller und visueller (Zwischen-)Ergebnisse, zum anderen durch den Workflow-Editor. Experten-Anwendende sollen damit in die Lage versetzt werden, spezialisierte Kenntnisse über OCR-Workflows zu sammeln und entsprechende Erfahrungen auszutauschen.

¹⁰ https://github.com/slub/ocrd_kitodo

¹¹ <https://github.com/OCR-D/core/issues/322>, [core#1015](#)

¹² [spec#241](#), [core#1069](#)

¹³ [spec#172](#)

¹⁴ <https://github.com/hnesk/browse-ocrd>, Integration der Gtk-UI in die Web-UI per [Broadway-Backend](#)

Aufgrund der im parallelen OCR-D-Modulprojekt der UB Mannheim entstandenen, qualitativ stark verbesserten generischen OCR-Modelle sind spezifische Workflows zudem seltener nötig. Eine Ausnahme dabei bildet die Layoutanalyse.

Als Ergebnis von AP 5 liegt bei Projektende ein Konzept¹⁵ für Evaluatoren und Workflow-Kontrollstrukturen vor, die im DFG-Projekt zur Verbesserung der Layoutanalyse¹⁶ zu implementieren sind. Für AP 7 wurden einzelne Metriken für die textuelle Qualitätsabschätzung nachnutzbar in Form von Prozessoren implementiert: `ocrd-cor-asv-ann-mark`¹⁷ (Wort-Trefferquote geknüpft an externe Wörterbücher) und `ocrd-keraslm-rate`¹⁸ (Bewertung und Pfadsuche mit Sprachmodellen auf Zeichen-Ebene, mit auf Daten aus dem Deutschen Textarchiv¹⁹ vortrainiertem Modell). Darüber hinaus sind für die manuelle Workflow-Optimierung Werkzeuge entstanden, darunter Prozessoren für die Layoutanalyse-Nachverarbeitung (`ocrd-segment-repair`²⁰, `ocrd-segment-project`²¹), die OCR-Nachkorrektur (`ocrd-cor-asv-ann-align`²²) und wiederkehrende Spezialaufgaben (`ocrd-page-transform`²³); aber auch für die statische Evaluierung gegen Ground-Truth (`ocrd-segment-evaluate`²⁴, `ocrd-cor-asv-ann-evaluate`²⁵). Diese Beiträge sollen die gezielte Anpassung von OCR-Workflows an bestimmte Materialien erleichtern und werden im Rahmen des sich anschließenden DFG-Projektes zur Verbesserung der Layoutanalyse in OCR-D für die phänomenologische und datenexplorative quantitative Analyse aufgegriffen und weiterentwickelt.

3.2.5 AP 8: Aufbau eines verteilten OCR-D-Dienstes in der Kitodo-Community (UB Braunschweig)

Die in AP 4 beschriebenen Komponenten *Manager* und *Controller* sind als Docker Container verfügbar und werden in der UBBS praktisch eingesetzt. Eine entsprechende Konfiguration für das Container-Orchestrierungswerkzeug *Docker Compose* wird in den jeweiligen GitHub Repositories bereitgestellt. Durch die Verwendung von Docker Containern sind nutzende Einrichtungen in der Lage, die Komponenten leicht selbständig einzusetzen, ohne zusätzliche Systemabhängigkeiten lokal zu installieren. An der UBBS werden die Auslieferung und Aktu-

¹⁵ [spec#172](#)

¹⁶ <https://gepris.dfg.de/gepris/projekt/517459941>

¹⁷ <https://github.com/ASVLeipzig/cor-asv-ann#ocr-d-processor-interface-ocrd-cor-asv-ann-mark>

¹⁸ https://github.com/OCR-D/ocrd_keraslm#ocr-d-processor-interface-ocrd-keraslm-rate

¹⁹ <https://deutsches-textarchiv.de>

²⁰ https://github.com/OCR-D/ocrd_segment/blob/master/ocrd_segment/repair.py

²¹ https://github.com/OCR-D/ocrd_segment/blob/master/ocrd_segment/project.py

²² <https://github.com/ASVLeipzig/cor-asv-ann#ocr-d-processor-interface-ocrd-cor-asv-ann-align>

²³ <https://github.com/bertsky/workflow-configuration#ocrd-page-transform>

²⁴ https://github.com/OCR-D/ocrd_segment/blob/master/ocrd_segment/evaluate.py

²⁵ <https://github.com/ASVLeipzig/cor-asv-ann#ocr-d-processor-interface-ocrd-cor-asv-ann-evaluate>

alisierung der einzelnen Komponenten zusätzlich durch den Einsatz des Konfigurationsverwaltungswerkzeugs *Ansible* geregelt. Da die Komponenten Manager und Controller an der UB Braunschweig auf verteilten Systemen eingesetzt werden, werden durch den Einsatz von Ansible die notwendigen Arbeitsschritte zur Konfiguration und Auslieferung maßgeblich vereinfacht.

Vor Ort können an der UB Braunschweig nun praktisch und teil-automatisch verschiedene Werke in den Kitodo-Projekten mit OCR prozessiert und publiziert werden. Es bleibt zu erwähnen, dass aufgrund der Verzögerungen im Projektmanagement (siehe Bericht AP1) sich auch hier erhebliche Verzögerungen ergeben haben und daher Teilaufgaben nicht vollumfänglich erreicht werden konnten. Das Erzeugen von OCR-D Ergebnissen ist an der UB Braunschweig dank diverser Systemanpassungen dennoch nahezu kontinuierlich möglich. Eine Texterkennung wird für die lokalen Bestände aktuell und auch künftig angewendet. Dank der durchgeführten Tests an diversen (historischen) Medientypen konnten passende Texterkennungs-Szenarien und Workflows ausgearbeitet werden²⁶.

Eine öffentlich angebotene Schnittstelle wie geplant existiert derzeit zwar nicht, dennoch ist geplant, zeitnah im individuellen Austausch mit einzelnen interessierten Institutionen eine OCR-D Bearbeitung extern unter den gegebenen technischen Umständen einzurichten und anzubieten (siehe Abschn. 5). Dies ist nicht zuletzt möglich, da das notwendige Wissen auch über das Projekt hinaus im Haus bleibt. Für ein großflächiges OCR-Angebot für weitere Institutionen ist die aktuelle Produktreife und Performance zwar nicht ausreichend, die bereitstehenden (technischen) Ressourcen liefern aber einen wesentlichen Grundstein, sodass ein nachhaltiges Betriebskonzept und Angebot entwickelt werden könnte.

3.2.6 AP 9: Integration von OCR-D in den Workflow von Kitodo.Production (SLUB Dresden)

Ziel der Integration war es, Kitodo.Production und OCR-D leichtgewichtig miteinander zu verbinden, ohne direkte technische Abhängigkeiten zu erzeugen. Damit wird zum einen eine Erhöhung der Komplexität von Kitodo.Production vermieden. Zum anderen erlaubt es beiden Systemen, unabhängig voneinander verbessert und erweitert zu werden. Die Ansteuerung des *Managers* wird mithilfe der in Kitodo.Production bereits vorhandenen Erweiterungsmöglichkeiten – den sogenannten Script-Tasks, die in Digitalisierungsworkflows eingebunden sein können – realisiert. Im Rahmen des Projekts wurden dazu entsprechende Skripte angelegt, die eine generische Anbindung realisieren und individuell an die Bedürfnisse der Einrichtungen angepasst werden können. Dabei geht es vor allem um spezifische Ordnerstrukturen und

²⁶ https://github.com/slub/ocrd_kitodo/wiki/

die Übergabe von Metadaten (wie Sprache und Schrifttyp), die je nach Konfiguration des Regelsatzes an unterschiedlichen Stellen im internen Datenformat abgelegt sein können.

Während die Umsetzung der Anbindung als dezidiertes Plugin mit eigener Bedienoberfläche nicht sinnvoll erschien, erwiesen sich im Laufe des Projektes dennoch in zwei Bereichen Erweiterungen von Kitodo.Production als notwendig: zum einen für eine saubere Modellierung des Ergebniszustands von Tasks²⁷, zum anderen für die Auswahl von OCR-Workflows auf Projekt- oder Vorgangsebene²⁸ – beide jedoch ohne spezifischen Bezug zu OCR-D.

Bisher nutzte Kitodo.Production, um interne Aufgaben durch extern gestartete Skripte abzuschließen, eine ActiveMQ-Schnittstelle²⁹ ohne Unterscheidung zwischen Erfolgs- und Fehlerfall. Die hier zusätzlich implementierte Schnittstelle³⁰ bietet neben dem reinen Abschluss von Aufgaben Möglichkeiten zur Rückmeldung des Ergebniszustands. Dies ermöglicht eine rudimentäre Fehlerbehandlung in Kitodo.Production, da Anwendende über Infomeldungen auf die Prozessierungsergebnisse auf dem *Monitor* verwiesen werden.

Abhängig von den zu digitalisierenden Materialien muss gewährleistet sein, dass ein jeweils geeigneter OCR-D-Workflow zum Einsatz kommen kann, wenn die Standard-Konfiguration erwartbar keine qualitativ zufriedenstellenden Ergebnisse liefert. Mit der o.g. Erweiterung ist es nun möglich, in Kitodo.Production vorinstallierte OCR-Workflows auszuwählen. Diese Auswahl wird durch einen Aufrufparameter des *Managers* in Kitodo.Production übergeben³¹.

Abgesehen von zwei gezielten, auch jenseits von OCR-D sinnvollen Erweiterungen an den Übergabepunkten zwischen den Systemen konnte die Integration von OCR-D in Kitodo.Production mit den dort bereits vorhandenen Schnittstellen erreicht werden. Das Release 3.7.0, welches alle Entwicklungen des Projektes enthält, wird voraussichtlich im ersten Halbjahr 2024 veröffentlicht.

3.2.7 AP 10: Integration von OCR-D in die Nutzerdienste von Kitodo.Presentation (UB Mannheim)

2020 konnte eine Machbarkeitsstudie im Rahmen des *KIT Summer of Code for Society*³² nachweisen, dass „OCR on Demand“ für Kitodo.Presentation³³ und den DFG-Viewer³⁴ realisierbar ist. Diese Implementierung bot dafür nur die absolut notwendigen Funktionen und wies

²⁷ <https://github.com/kitodo/kitodo-production/pull/5657>

²⁸ <https://github.com/kitodo/kitodo-production/pull/5809>

²⁹ https://github.com/kitodo/kitodo-production/wiki/Developer_3.x-Active-MQ#finalizestep-queue

³⁰ https://github.com/kitodo/kitodo-production/wiki/Developer_3.x-Active-MQ#taskaction-queue

³¹ <https://github.com/kitodo/kitodo-production/wiki/Platzhalter>, s. *ocrdworkflowid*

³² <https://soc4s.ipd.kit.edu/home/projekte/>

³³ <https://github.com/KIT-SOC4S/kitodo-presentation>

³⁴ <https://github.com/KIT-SOC4S/dfg-viewer>

noch technische Fehler auf, konnte aber nach Sichtung als Ausgangsbasis übernommen werden.

Schon Anfang 2023 wurde eine öffentlich zugängliche Demo-Instanz des DFG-Viewers an der Universitätsbibliothek Mannheim angelegt und ständig um den neuesten Entwicklungsstand ergänzt.³⁵ So war es möglich, Dritten die Funktionalität zu präsentieren, Rückmeldungen einzuholen und Anforderungen mit der Implementierung abzugleichen.

Gewünscht war, auch bei bereits vorhandenen Volltexten trotzdem eine neue OCR zu ermöglichen – beispielsweise um unbefriedigende OCR-Ergebnisse zu verbessern. Diese "re-OCR"-Funktion ist jetzt realisiert. Zudem können Nutzende zwischen unterschiedlichen OCR-Prozessen wählen. Die Demo-Instanz bietet OCR-D mit dem im Projekt implementierten OCR-Server, aber zusätzlich auch Tesseract und Kraken zur Auswahl. Auch zwischen mehreren OCR-Modellen – konkret den an der UB Mannheim entwickelten Modellen *frak2021*, *german_print*³⁶ oder *german_handwriting* – kann man wählen. Die Implementierung dieser Funktionen ist nicht nur im Webfrontend, sondern auch auf der Serverseite benutzerfreundlich und erfordert kein tiefes technisches Wissen. Es genügt, eine einfache Konfigurationsdatei anzupassen, um die Liste der angebotenen OCR-Prozesse zu ergänzen und zu modifizieren.

Darüber hinaus gibt es eine praktische Übersicht, für welche OCR-Prozesse bereits Volltexte erzeugt wurden bzw. lokal oder remote verfügbar sind. Benutzerinnen und Benutzer können nun nahtlos zwischen verschiedenen OCR-Ergebnissen wechseln und sie so vergleichen.

Optional lassen sich METS-Daten erzeugen. Nach der Erzeugung eines neuen Volltextes kann optional die Indizierung angestoßen werden. Diese und alle weiteren Einstellungen erfolgen in der TYPO3-eigenen Extension Configuration, wodurch die Benutzerfreundlichkeit und Anpassbarkeit des Systems erheblich vereinfacht wird.

Im Zuge der Entwicklung wurden auch allgemeine Arbeiten wie Fehlerkorrekturen und Programmerweiterungen an Kitodo.Presentation³⁷ und dem DFG-Viewer³⁸ vorgenommen und zurückgeführt. Insbesondere in den letzten sechs Monaten der Projektlaufzeit lag der Fokus daher verstärkt darauf, ein stabiles Release der beiden Projekte zu erreichen und die Projektergebnisse dort ordnungsgemäß integrieren zu können.

³⁵ <https://dfg-viewer.bib.uni-mannheim.de/>

³⁶ vgl. DFG-Modulprojekt OCR-D Training, <https://gepris.dfg.de/gepris/projekt/460547474>

³⁷ <https://github.com/kitodo/kitodo-presentation/pull/892>, [#912](#), [#917](#), [#922](#), [#1029](#), [#1062](#), [#1067](#), [#1080](#), [#1087](#), [#1111](#), [#1156](#), [#1185](#), [#1186](#) und [#885#comment](#)

³⁸ <https://github.com/slub/dfg-viewer/pull/220>, [#221](#), [#224](#), [#232](#), [#255](#), [#257](#), [#261](#), [#262](#), [#264](#), [#265](#), [#266](#), [#273](#), [#275](#), [#277](#), [#279](#), [#282](#), [#284](#) und https://github.com/slub/slub_digitalcollections/pull/64, [#65](#), [#70](#)

Zum Zeitpunkt dieses Berichts steht die Rückführung zur Codebasis noch aus. Das Release Kitodo.Presentation 5.0.0³⁹ wird voraussichtlich im ersten Halbjahr 2024 veröffentlicht.

3.2.8 AP 11: Speicherung und Datenübernahme in Quellsysteme (UB Mannheim)

Geplant war, die durch „OCR on Demand“ erzeugten Volltexte den Institutionen, von denen die Digitalisate stammen, zur Verfügung zu stellen, damit sie ihre eigenen digitalen Sammlungen um diese Volltexte ergänzen können. Da der DFG-Viewer mit „OCR on Demand“ bisher kaum genutzt wird, gibt es aber nur wenige neu erzeugte Volltexte, so dass es sich nicht lohnt, dafür einen Prozess aufzusetzen, der diese automatisch verteilt.

Da die erzeugten Volltexte frei zugänglich sind, lassen sie sich aber von jeder Institution und anderen Interessenten herunterladen.

Allerdings musste die UB Mannheim die Erfahrung machen, dass selbst sehr hochwertige Volltexte, die dort im Rahmen von Schulungen und Praktika erzeugt wurden, bei den besitzenden Institutionen zwar auf Interesse stießen, aber praktisch nie übernommen wurden. Offenbar scheitert die Datenübernahme oft an fehlenden Kapazitäten oder organisatorischen Voraussetzungen.

3.2.9 AP 12: Produktivsetzung der „OCR on Demand“ im DFG-Viewer (SLUB Dresden / UB Mannheim)

Um eine einfache und einheitliche Einrichtung zu ermöglichen, wurden Kitodo.Presentation und der DFG-Viewer sowie die entwickelten „OCR on Demand“-Versionen dockerisiert.⁴⁰ Dies erleichtert nicht nur das Zurücksetzen oder Wechseln zwischen Versionen während der Entwicklungsphase, sondern liefert auch ein vollständiges Setup inklusive aller benötigten OCR-Software und ihrer Modelle.

Die Quellcodeübernahme in die Hauptentwicklungszweige des DFG-Viewers und Kitodo.Presentation steht noch aus. Zudem ist die UB Mannheim mit der SLUB Dresden im Austausch, inwieweit die OCR-Funktionalität nach Projektende in den Betrieb der produktiven DFG-Viewer-Instanz aufgenommen und verfügbar gemacht werden kann.

Vorerst wird die UB Mannheim ihre Demo-Instanz des DFG-Viewers mit „OCR on Demand“ und OCR-D weiter zur Verfügung stellen.

³⁹ <https://github.com/kitodo/kitodo-presentation/milestone/6>

⁴⁰ <https://github.com/UB-Mannheim/kitodo-presentation-docker>

4 Öffentlich zugängliche Projektergebnisse

4.1 Publikationen mit wissenschaftlicher Qualitätssicherung

Keine.

4.2 Weitere Publikationen und öffentlich gemachte Ergebnisse

- Onlinesammlung von Vorträgen, Workshops und Quellcode-Repositoryen⁴¹

⁴¹ https://github.com/slub/ocrd_kitodo/wiki/Ver%C3%B6ffentlichungen