Elektronisches Publizieren von Digitalen Forschungsdaten am Beispiel des TextGrid Repositorys

Umsetzung von Digitalen Publikationsworkflows für die eHumanities

Stefan Edwin Funk

Masterarbeit
Bibliotheks- und Informationswissenschaft
Fakultät für Informations- und Kommunikationswissenschaften, Technische Hochschule Köln

Gutachter: Dr. Peter Kostädt
Zweitgutachter: Prof. Dr. Achim Oßwald

vorgelegt am 23. Februar 2018 von

Stefan Edwin Funk

Zur sprachlichen Gleichbehandlung von Frauen und Männern in dieser Arbeit schreibe ich nur der Leserlichkeit halber nicht jedes Mal von „Geisteswissenschaftlerinnen und Geisteswissenschaftlern", ich möchte außerdem weder von „GeisteswissenschaftlerInnen", noch von „Forschenden" oder „forschenden Personen" reden. Ich verwende unregelmäßig abwechselnd die weibliche und die männliche Form, es ist immer auch das jeweils andere Geschlecht angesprochen.

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Dies gilt auch für Quellen aus eigenen Arbeiten.

Ich versichere, dass ich diese Arbeit oder nicht zitierte Teile daraus vorher nicht in einem anderen Prüfungsverfahren eingereicht habe.

Mir ist bekannt, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs mittels einer Plagiatserkennungssoftware auf ungekennzeichnete Übernahme von fremdem geistigem Eigentum überprüft werden kann.

Göttingen, den 23. Februar 2018

Zusammenfassung

Die digitale Transformation führt auch bei Bibliotheken und bei Forschungsinfrastrukturen, die durch die Geistes- und Kulturwissenschaften genutzt werden, zu strukturellen Veränderungen. So werden kritische Editionen und Wörterbücher elektronisch publiziert, große Mengen an Büchern digitalisiert und deren Inhalt als elektronisch erschlossene Volltexte der Öffentlichkeit zur Verfügung gestellt. In den eHumanities etablieren sich durch die Anwendung computergestützter Verfahren neue Publikationsformen. Es werden neue Arbeitsabläufe für die Publikation und die langfristige Speicherung der Daten benötigt. Im Rahmen dieser Studie werden Entwicklungspotenziale dieser neuen fachwissenschaftlichen Anforderungen am Beispiel der Virtuellen Forschungsumgebung TextGrid analysiert. Es werden zugleich Konzepte und Lösungen entwickelt, die als Funktionserweiterungen in dieses System implementiert werden. Zugleich stehen neben der Funktionserweiterung, basierend auf konkreten, erhobenen Anforderungen, die ebenfalls im Rahmen dieser Arbeit vorgestellt und diskutiert werden, intuitiv bedienbare Implementierungen im Fokus. Darüber hinaus werden technische Erweiterungen für die Langzeitarchivierungs-Strategien des TextGrid Repositorys und die Verbesserung des Import- und Publikationsprozesses implementiert und beschrieben.

The digital transformation is leading to structural changes for libraries and research infrastructures used by the humanities and cultural sciences. Critical editions and dictionaries are published electronically, large numbers of books are digitised and their content is made available to the public as electronically accessible full texts. In the eHumanities, new forms of publication are becoming established due to the use of computer-aided methods. New workflows are required for publication and long-term storage of data. This study analyses the development potential of these new scientific requirements using the example of the TextGrid Virtual Research Environment. At the same time, concepts and solutions are being developed, which are implemented in this system as functional extensions. Besides the functional enhancement, that is based on concrete, collected requirements, which are also presented and discussed in this study, it also focuses on intuitive implementations. In addition, technical enhancements for the long-term archiving strategies of the TextGrid Repository and the improvement of the import and publication processes are implemented and described.

Schlagwörter

Forschungsdaten, Publikation, Digitale Geisteswissenschaften, Digitaler Publikationsworkflow, Publikationsprozess, Datenpublikation, TextGrid Repository

Keywords

Research Data, Publication, eHumanities, Digital Publication Workflow, Data Publication, TextGrid Repository

Inhaltsverzeichnis

Einführung

Die Digitale Transformation hat auch vor den Geisteswissenschaften nicht halt gemacht: Kritische Editionen und Wörterbücher werden digital veröffentlicht, große Mengen an Büchern werden digitalisiert und ihr Inhalt auch als Volltext der Öffentlichkeit zur Verfügung gestellt. Diese digitalen Daten können nicht nur von Menschen überall gelesen werden, es können große Mengen an Text auch maschinell verarbeitet werden. So ergeben sich neue – und nicht nur quantitative – Forschungsfelder für die Geisteswissenschaften. Diese neuen, digitalen Formen von Daten heißt es zu verwalten und zu bewahren, so dass sie für die Forschung auch langfristig und zuverlässig zur Verfügung stehen. Die eHumanities zeichnen sich durch die Anwendung computergestützter Verfahren und die systematische Verwendung von digitalen Ressourcen aus. Als Beispiel von Publikationsformen der Digitalen Transformation können kritische und hybride Editionen sowie Online-Publikationen wie die Digitale Bibliothek[1] von TextGrid angeführt werden. Es entstehen neue Anforderungen an Datenspeicher und an die Prozesse, mit denen digitale Daten verarbeitet und publiziert werden.

Geisteswissenschaftlerinnen, die mit Forschungsdaten arbeiten, diese nachnutzen und im Zuge ihrer Forschung neue erzeugen, benötigen Arbeitsabläufe für die Publikation und langfristige Speicherung ihrer Daten, die auf ihren Bedürfnissen basieren. Diese Forschungsdaten können für die Dokumentation, die Referenzierbarkeit und die Nachnutzbarkeit in einem öffentlichen Repositorium abgelegt werden, so dass sie zum einen sicher aufbewahrt und zum anderen auch für andere Wissenschaftler nachnutzbar sind und für diesen Zweck auch komfortabel nach ihnen gesucht und recherchiert werden kann. Vor diesem Hintergrund wird in dieser Arbeit den folgenden Fragen nachgegangen:

1. Wie kann ein digitaler Publikationsworkflow aussehen, der Wissenschaftlerinnen unterstützt und sie motiviert, ihre Daten in ein fachwissenschaftliches Repositorium zu überführen?
2. Inwieweit sind die vorhandenen Workflows aus der Virtuellen Forschungsumgebung TextGrid geeignet, die Anforderungen an digitale Publikationen aus den Geisteswissenschaften zu bedienen?
3. Welche technischen Erweiterungen sind notwendig, um die Workflows für die genannten Zwecke auszubauen und zu verbessern?

Eine Besonderheit der vorliegenden Arbeit ist ihr starker Bezug auf die Praxis und die konkrete Entwicklungsarbeit am TextGrid Repository. Anhand der beschriebenen Use Cases werden Anforderungen von aktiven Nutzern des Produktivsystems analysiert, konkrete Lösungsvorschläge erarbeitet und schließlich sowohl Module der existierenden Dienste verbessert als auch bei Bedarf neue Module implementiert.

Im ersten Kapitel werden die in dieser Arbeit grundlegenden Begriffe definiert und es wird ein thematischer Überblick gegeben. Kapitel 2 beschreibt die Inhalte und Ziele der Projekte DARIAH-DE und TextGrid. Die vorhandenen Arbeitsabläufe für den Import und das Publizieren von elektronischen Dokumenten in der Virtuellen Forschungsumgebung TextGrid werden inhaltlich wie technisch beschrieben und der Status Quo der Implementierung seitens des TextGrid Repositorys wird erläutert. Projekte aus

[1] Vgl. TextGrid Digitale Bibliothek (2016). https://textgrid.de/digitale-bibliothek

dem Umfeld von TextGrid, die das TextGrid Repository als Forschungsdatenrepositorium nutzen und dort ihre Forschungsergebnisse präsentieren, werden als Use Cases in Kapitel 3 beschrieben.

Weiterhin werden in Kapitel 4 Anforderungen an Publikationen im Allgemeinen und an elektronische Publikationen in den eHumanities im Speziellen dargestellt. Es werden die im vorhergehenden Kapitel beschriebenen Use Cases auf neue Anforderungen untersucht, ihre Workflows analysiert und die Anforderungen kategorisiert. Als Ergebnis dieser Analyse wird eine Liste von Anforderungen erstellt, die in Kapitel 5 als Grundlage für die Erstellung eines Konzepts für die Verbesserung des TextGrid Publikationsworkflows und der dem TextGrid Repository zugrunde liegenden Software dient. Notwendige Erweiterungen werden aus den Anforderungen hergeleitet und konzipiert. Die Implementierung der Verbesserungen wird in Kapitel 6 diskutiert und schließlich werden diese – sofern als möglich und sinnvoll erachtet – im Rahmen dieser Arbeit implementiert und die Implementierung beschrieben.

Abschließend werden in Kapitel 7 die Entwicklungen dieser Arbeit in den Kontext der momentanen Entwicklungen im Bereich Forschungsdatenrepositorien gestellt. Mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf zukünftige Entwicklungen im Rahmen des TextGrid Repositorys – aufbauend auf den vorliegenden Workflow- und Softwareverbesserungen – sowie auf die technische Entwicklung allgemein schließt Kapitel 8 die Arbeit ab.

1 Publikationen, digitale Forschungsdaten und Repositorien

1.1 Publikationen und das Publizieren

Der Begriff *Publikation* bezeichnet laut Duden das „publizierte Werk" als auch das „Publizieren", es stammt von dem (spät)lateinischen Begriff „publicatio" (Veröffentlichung) und dem französischen Begriff „publication" ab.[2] „Die Publizierung" bezeichnet laut Duden die „Veröffentlichung (eines literarischen oder wissenschaftlichen Werkes)", Synonyme für Publizierung sind: Herausgabe, Publikation und Veröffentlichung.[3] Publizieren bedeutet laut Duden

> „1. im Druck erscheinen lassen; veröffentlichen"[4]

und

> "2. publik machen, bekannt machen, veröffentlichen"[5]

Laut Wikipedia bezeichnet eine Publikation „(…) den Inhalt eines Mediums oder/und das konkrete Medium samt Inhalt sowie den Vorgang der öffentlichen Verfügbarmachung eines Mediums."[6] Eine inhaltliche Form der Publikation ist die *wissenschaftliche Publikation*[7].

Bibliothekswissenschaftliche Definitionen finden sich beispielsweise in Ewert und Umstätter (1997):

> „Dabei umfaßt der Sammelbegriff ‚publizierte Information' geschriebene bzw. gedruckte Dokumente sowie audiovisuelle Medien in analoger oder digitaler Form, die von Verlagen, politischen, gesellschaftlichen oder privaten Vereinigungen, Organisationen bzw. Institutionen hergestellt, vervielfältigt und für die Öffentlichkeit bzw. eine Teilöffentlichkeit bestimmt, herausgegeben werden."[8]

sowie in Gantert (2016):

> „Wenn man von ‚Publikationen' (Veröffentlichungen) spricht, meint man die Tatsache, dass Druckwerke zu einem bestimmten Zeitpunkt und in einer bestimmten Form ‚publiziert', d. h. der Öffentlichkeit zugänglich gemacht werden."[9]

Eine Motivation des wissenschaftlichen Publizierens ist der Nachweis der Forschungstätigkeit in Zahl und Qualität der veröffentlichten Werke eines Wissenschaftlers (Reputation).[10] Durch eine Publikation können Forschungsergebnisse nach Stock (2010)[11] sowie Schirmbacher und Müller (2009)[12] andere Wissenschaftlerinnen zu fachlichen Diskussionen sowie zur Untersuchung weiterer Forschungsfragen

[2] Vgl. Duden (2018a). https://www.duden.de/rechtschreibung/Publikation
[3] Vgl. Duden (2018b). https://www.duden.de/rechtschreibung/Publizierung
[4] Duden (2018c). https://www.duden.de/rechtschreibung/publizieren
[5] Ebd.
[6] Vgl. Wikipedia (2018a). https://de.wikipedia.org/wiki/Publikation
[7] Vgl. Wikipedia (2017). https://de.wikipedia.org/wiki/Wissenschaftliche_Publikation
[8] Ewert und Umstätter (1997), S. 10f.
[9] Gantert (2016), S. 85.
[10] Vgl. Schirmbacher und Müller (2009), S. 8.
[11] Vgl. Stock (2010), S. 243ff.
[12] Vgl. Schirmbacher und Müller (2009), S 8f.

anregen (Kommunikation). Weiterhin werden Forschungsergebnisse der wissenschaftlichen Fachcommunity zitierfähig vorgestellt und dienen als Nachweis, dass bestimmte Ergebnisse zu einem bestimmten Zeitpunkt bereits vorlagen (Nachweisinstrument). Zuletzt wird in Schirmbacher und Müller (2009) noch die Erlangung finanzieller Erträge genannt, die jedoch aus Sicht der Autoren „(…) als Motivation in vielen Wissenschaftsdisziplinen eine untergeordnete Rolle (…)"[13] spielt.

Wichtig ist hier insbesondere die Erzeugung neuer Forschungsfragen durch den Zugriff auf die veröffentlichten Forschungsergebnisse. Aus diesen entstehen wiederum neue Forschungsergebnisse sowie neue Publikationen und es ergibt sich ein gesellschaftlich bedingter Kreislauf.[14]

1.1.1 Der Publikationsprozess im Wandel

Funk (2014) gibt einen „kleinen Exkurs zur Geschichte der Publikation"[15]. Als Publikation von Schriftstücken ausgehend, wird beginnend bei den altägyptischen Hieroglyphen und der Keilschrift der Sumerer samt deren Trägermaterialien Ton oder Stein bzw. Papyrusrollen, Leder oder Leinen – und später dem Papier –, auch auf deren Inhalt eingegangen. Anfangs für reine Verwaltungsangelegenheiten genutzt, wurden später auch religiöse, politische und wirtschaftliche Inhalte festgehalten sowie letztendlich auch wissenschaftliche Schriften und unterhaltende Literatur verfasst.

Weiterhin wird beschrieben, wie mit der Erfindung des Buchdrucks im 15. Jahrhundert durch Johannes Gutenberg schließlich die Herstellung von bis dahin nicht erreichbaren Mengen identischer Abschriften ermöglicht wurde. Der Vorgang des Vervielfältigens wurde später nochmals durch die Erfindung der Maschinenpresse beschleunigt und es bildete sich schließlich ein Verlagswesen heraus, das „die Distribution für die Wissenschaft (…) weitgehend effizient organisierte".[16]

Auf die Anfänge der elektronischen Publikation wird in Funk (2014) ebenfalls kurz eingegangen, z. B. gibt es seit den späten 1960er Jahren ein Konzept für ein interaktives elektronisches Buch, das „Dynabook"[17] von Alan Key und Adele Goldberg. Überlegungen zur Veröffentlichung elektronischer wissenschaftliche Zeitschriften gibt es seit den frühen 1970er Jahren.[18]

Der wissenschaftliche Publikationsprozess ist ein wesentlicher Bestandteil der Wissenschaft selbst.[19] Er ist Grundlage für die Verbreitung von Forschungsergebnissen und deren dauerhafte Erhaltung, so dass sich trotz räumlicher und zeitlicher Verteilung von Forschenden und Forschungsergebnissen alle an der Forschung Beteiligten aufeinander beziehen können. Neue Forschung kann auf einmal gesicherten Erkenntnissen aufbauen, aus dem Publikationsprozess entwickelt sich ein *Publikationskreislauf*:

> „Beim wissenschaftlichen Publizieren spricht man gemeinhin von einem gesellschaftlich bedingten Kreislauf, beginnend mit der Darstellung des geistigen Werkes durch die Autoren, der eigentlichen Publikation, seiner Bewertung und Verwertung, in der Regel organisiert durch Verlage, der

[13] Ebd., S. 8.
[14] Vgl. ebd., S. 7f.
[15] Vgl. Funk (2014), S. 8f.
[16] Vgl. Stäcker (2013), S. 41.
[17] Vgl. Kay und Goldberg (1977), S. 392.
[18] Vgl. Riehm; Böhle und Wingert (2004), S. 554.
[19] Vgl. Schirmbacher und Müller (2009), S. 7.

Erschließung, Aufbewahrung und Bereitstellung durch Bibliotheken und des Rezipierens durch die wissenschaftliche Gemeinschaft, aus deren Mitte dann wiederum Autorinnen und Autoren ein nächstes geistiges Werk, eine Publikation, schaffen."[20]

Ein solcher Publikationskreislauf sieht basierend auf Schirmbacher und Müller (2009) wie folgt aus (siehe Abbildung 1):

1. **Forschung**: Grundlage sind Publikationen und Forschungsdaten
2. **Darstellung eines geistigen Werkes**: Erstellen einer Arbeit
3. **Bewertung des Werkes**: Prüfung und Akzeptanz der Arbeit
4. **Verwertung des Werkes**: Layout und Veröffentlichung
5. **Zugang zur Publikation**: Erschließung, Aufbewahrung und Bereitstellung
6. **Auswirkungen**: Rezipieren durch die wissenschaftliche Gemeinschaft

Abbildung 1: Publikationskreislauf angelehnt an Schirmbacher und Müller (2009)

[20] Schirmbacher und Müller (2009), S. 8.

Analoge wissenschaftliche Publikationsworkflows mit dem Ziel einer Publikation in Form eines Zeitschriftenartikels oder eines Buches sind seit vielen Jahren in der Wissenschaft etabliert. Die Anforderungen sind geklärt, die Aufgaben und Rollen sind definiert und es gibt klar verteilte Zuständigkeiten, derer sich Wissenschaftler, Herausgeber, Verlage und Bibliotheken angenommen haben. Der Publikationsprozess ist arbeitsteilig organisiert und hat eine möglichst hohe Qualität der Publikation durch Auswahl der Manuskripte sowie gestalterische Aufbereitung zum Ziel.[21] So können die verschiedenen Aufgaben des analogen Publikationsprozesses klar jeweils einzelnen Stationen des Publikationskreislaufes zugeordnet werden – wie in Abbildung 1 dargestellt. Weiterhin können den Aufgaben auch Personen zugeordnet werden: Die Punkte 1 und 2 sind klar Aufgabe der Wissenschaft, Punkte 3 und 4 sind Aufgabe des Verlags (wobei bei der Bewertung natürlich auch wieder Wissenschaftler eine große Rolle spielen), Punkt 5 kann als Aufgabe den Bibliotheken zugeordnet werden und schließlich ist Punkt 6 wieder Aufgabe der Wissenschaft.

Die Eindeutigkeit dieser Zuordnungen von Aufgaben zu Akteuren sowie der Abgrenzung der Phasen des Publikationsprozesses voneinander hat in den letzten Jahren nachgelassen: Der Prozess des wissenschaftlichen Publizierens wandelt sich. Dies hängt nach Schirmbacher und Müller (2009)[22] mit drei Entwicklungen zusammen:

1. Die technischen Möglichkeiten der *Vervielfältigung und Verbreitung von Publikationen* haben sich unter anderem durch den Digitaldruck[23] und die Möglichkeiten des elektronischen Publizierens über das Internet in einer Weise geändert, dass im Prinzip „(…) ein freier uneingeschränkter Zugriff auf das Wissen der Welt möglich wird"[24].
2. Auch die Vereinfachung des *Erstellens von Publikationen* mit den Möglichkeiten des elektronischen Publizierens erlaubt „(…) damit eine zeitnahe Veröffentlichung" und lässt „(…) eine unmittelbare weltweite Verbreitung realistisch erscheinen"[25].
3. Aufgrund der *vielfältigen Präsentationsformen* elektronischer Publikationen sind diese nun nicht mehr ausschließlich text- und grafikorientiert, sondern können um Audio- und Videoformate ergänzt werden. Auch Mischformen und interaktive Formen der Publikation sind damit möglich.

Schaut man sich die mit diesen Entwicklungen einhergehenden Verschiebungen der Zuständigkeiten der einzelnen Arbeitsschritte an, folgt daraus für die Autoren einer Publikation, dass auch die Punkte 3 und 4 (Bewertung und Verwertung des Werkes) nicht mehr zwingend von einem Verlag ausgeführt werden müssen. Die Autoren können das Werk selbst in einer Qualität – bzgl. Text, Layout und Format – erstellen, die für den Druck bzw. eine Veröffentlichung im Internet notwendig ist, und die Arbeit auch drucken lassen oder im Internet veröffentlichen, unter Einbeziehung aller heute möglichen Mischformate und Präsentationsformen.

Daraus folgt jedoch nicht unbedingt eine Vereinfachung des Publikationsprozesses, zumindest nicht, wenn die wissenschaftliche Qualität der Publikation nicht darunter leiden soll. Denn auch die Bewertung

[21] Vgl. Riehm; Böhle und Wingert (2004), S. 549.
[22] Vgl. Schirmbacher und Müller (2009), S. 11f.
[23] Vgl. Wikipedia (2018b). https://de.wikipedia.org/wiki/Digitaldruck
[24] Schirmbacher und Müller (2009), S. 11.
[25] Ebd., S. 11f.

und Verwertung sind anspruchsvolle Aufgaben und Voraussetzung für eine sinnvolle und möglichst verständliche Nachnutzung der publizierten Arbeiten, die auch bei einer elektronischen Publikation durch z. B. ein Peer Review-Verfahren sichergestellt werden kann. Die Verantwortung für die Qualität der Publikation oder auch der Forschungsdaten muss also nicht unbedingt ganz in der Hand der Autorinnen und Autoren liegen. Die Bereitstellung auch elektronischer Publikationen sollte wiederum in der Verantwortung einer vertrauenswürdigen und nachhaltig verwalteten Institution liegen, die auch langfristig Zugriff auf die Publikationen gewährleisten kann.

1.1.2 Elektronisches Publizieren

Durch die Möglichkeiten des *elektronischen Publizierens* entstanden und entstehen Veränderungen in der Struktur des gesamten Publikationswesens im Allgemeinen sowie in der Branchenstruktur des Verlagswesens im Speziellen. Als elektronische Publikationen werden

> „(…) diejenigen Veröffentlichungen bezeichnet, deren Informationen digital gespeichert sind, und für deren Verwendung man einen Computer benötigt".[26]

Nach Gantert (2016) können elektronische Publikationen in Offline-Publikationen, Online-Publikationen und Multimedia-Publikationen eingeteilt werden. Online-Publikationen, von denen in dieser Arbeit hauptsächlich die Rede ist, können wiederum unterschieden werden in elektronische Zeitschriften, E-Books, retrodigitalisierte Drucke, elektronische Hochschulschriften, elektronische Nachschlagewerke, bibliographische Datenbanken und Forschungsdaten.[27]

Viele Verlage veröffentlichen ihre Publikationen in steigendem Umfang auch oder sogar ausschließlich elektronisch. Das geschieht mittlerweile überwiegend per Netzpublikation im Internet. Aus oben genannten Gründen ist das elektronische Publizieren einfacher, schneller und auch kostengünstiger als das Veröffentlichen von gedruckten Medien. Deshalb sind es heutzutage auch nicht mehr nur Verlage, die wissenschaftliche Publikationen digital veröffentlichen, sondern unter anderem auch Universitäten und Bibliotheken, die ihren Wissenschaftlerinnen neue Wege des elektronischen Publizierens anbieten und sehr oft den Weg des Open Access gehen.[28]

Open Access wird der freie Zugang zu wissenschaftlichen Materialien genannt, zum Beispiel zu im Internet öffentlich zugänglicher wissenschaftlicher Fachliteratur in Form von Beiträgen in elektronischen Zeitschriften, Preprints oder elektronischen Versionen von Beiträgen in Büchern oder Zeitschriften. Die Open-Access-Bedingungen erlauben es jedem, die so veröffentlichten Publikationen ohne Entgelt zu benutzen und je nach Lizenz auch anderweitig zu verwerten.[29] Eine Besonderheit beim Open Access liegt insbesondere darin, dass sowohl die Urheber- als auch die Verwertungsrechte bei jeder einzelnen Publikation angegeben werden – beispielsweise Creative Commons[30] –, so dass lizenzrechtliche Sicherheit bzgl. der Nachnutzung und Verwertung besteht.

[26] Gantert (2016), S. 104.
[27] Vgl. ebd., S. 107ff.
[28] Vgl. ebd., S. 95.
[29] Vgl. Wikipedia (2018c). https://de.wikipedia.org/wiki/Open_Access
[30] Vgl. Creative Commons (2018). https://creativecommons.org/licenses

Es gibt verschiedene Publikationswege innerhalb von Open Access. Die zwei wichtigsten sollen hier genannt werden: Der *Goldene Weg* bezeichnet die primäre Veröffentlichung von wissenschaftlichen Arbeiten in einem Open-Access-Medium, z. B. in einer Open-Access-Zeitschrift. Diese setzen, ebenso wie konventionelle Zeitschriften, zur Bewertung der Beiträge ein Peer-Review-Verfahren ein. Als *Grüner Weg* wird eine parallele Veröffentlichung von wissenschaftlichen Arbeiten oder auch Primär- und Forschungsdaten beispielsweise auf privaten oder auch institutionellen Internetseiten oder Repositorien von Instituten oder Universitäten bezeichnet. Für eine nachhaltige Speicherung und die dauerhafte Möglichkeit der Zitation der Daten ist es sinnvoll, auf das Angebot eines institutionellen Repositoriums zurückzugreifen, denn ein solches kann auch dauerhaft sicherstellen, dass die veröffentlichten Daten der Wissenschaft zur Verfügung stehen.

Wichtig im Kontext der vorliegenden Arbeit ist zudem *Open Access für Primärdaten*, denn auch wissenschaftliche Primär- oder Forschungsdaten können, ermöglicht durch die bereits erwähnte technische Entwicklung, quantitativ und qualitativ hochwertig in den wissenschaftlichen Kommunikationsprozess integriert werden.[31] So können etwa Forschungsdaten als Sammlungen bereitgestellt werden, die in den wissenschaftlichen Arbeiten dann direkt oder indirekt referenziert werden und so auch dauerhaft nachgewiesen werden können.[32]

Die Veröffentlichung von Primärdaten hat noch weitere Vorteile, wie in Wicherts u. a. (2006) als Vorschlag für eine Modifikation im Publikationsprozess der Zeitschrift der American Psychological Association (APA) angeführt wird: Die Autoren würden aufgefordert, die (anonymisierten) Primärdaten, auf die sich ihr Artikel bezieht, als ASCII-Datei abzugeben und diese Daten würden als Anhang des Artikels im Internet veröffentlicht werden. Wicherts u. a. (2006) führen für dieses Verfahren eine Reihe von Vorteilen an, die durchaus auch auf andere Fachdisziplinen angewendet werden können:

- Es wird sehr viel einfacher, die Daten, auf denen eine Publikation basiert, schnell und effektiv nochmals zu analysieren und so Forschungsergebnisse zu re-validieren. Auf statistische Daten können beispielsweise auch andere Algorithmen angewendet werden. So können die in einer Publikation beschriebenen Ergebnisse bestätigt werden.[33]
- Die Möglichkeit der Durchführung von Meta-Analysen wird durch die Bereitstellung der Primärdaten deutlich vereinfacht.[34] Beispielsweise können Textkorpora aus verschiedenen Quellen zusammengeführt und darauf Operationen wie nutzerbasierte Recherchen oder statistische Berechnungen ausgeführt werden.[35]
- Fragen zu den Primärdaten oder deren Herkunft können direkt gestellt und auch deren Bewertung durch den Autor hinterfragt werden. So wird wissenschaftliches Fehlverhalten durch das Bewusstsein der Autoren erschwert, dass ihnen jederzeit jemand über die Schulter schauen kann.[36]

[31] Vgl. Mittler (2007), S. 168.
[32] Vgl. Wikipedia (2018c). https://de.wikipedia.org/wiki/Open_Access
[33] Vgl. Wicherts u. a. (2006), S. 727.
[34] Vgl. ebd.
[35] Als Beispiel vgl. Herrmann und Lauer (2017) und Lauer (2017). https://kolimo.uni-goettingen.de/about.html
[36] Vgl. Wicherts u. a. (2006), S. 727f.

- Durch die Publikation der Forschungsdaten wird eine neue „kritische Dimension" in der Wissenschaft ermöglicht, denn nicht nur des Wissenschaftlers Interpretation der Daten wird so Teil des wissenschaftlichen Diskurses, sondern auch die Forschungsdaten selbst.[37]
- Auch die Primärdaten sind öffentlich im Internet zugänglich, und so sind sie, sofern geeignete Maßnahmen getroffen werden, dauerhaft und nachhaltig verfügbar – auch für den Autor selbst.[38]

Als letzten und wichtigsten Punkt für die Veröffentlichung von Forschungsdaten führen Wicherts u. a. (2006) abschließend an:

> „Sixth, and perhaps most important, scientific evidence should be publicly accessible as a matter of principle; anybody who wants to play in the scientific arena will have to come in with open sight. A procedure like the one proposed would thus increase the openness of scientific research."[39]

Der Gewinn durch die Veröffentlichung von Primärdaten überwiegt den Aufwand um ein Vielfaches.[40] Diese Empfehlung von Wicherts u. a. (2006) kann sicherlich auch auf andere (oder gar alle) Herausgeber erweitert werden:

> „It seems to us that, considering the ratio of the benefits achieved in this manner to the costs involved in terms of extra work, this is a bargain. We therefore suggest that the APA journals incorporate the proposed procedure in the publication process."[41]

1.2 Digitale Forschungsdaten

Kindling und Schirmbacher (2013) definieren *digitale Forschungsdaten* und den Forschungsprozess wie folgt:

> „Unter digitalen Forschungsdaten verstehen wir dabei alle digital vorliegenden Daten, die während des Forschungsprozesses entstehen oder ihr Ergebnis sind. Der Forschungsprozess umfasst dabei den gesamten Kreislauf von der Forschungsdatengenerierung, z. B. durch ein Experiment in den Naturwissenschaften, eine dokumentierte Beobachtung in einer Kulturwissenschaft oder eine empirische Studie in den Sozialwissenschaften, über die Bearbeitung und Analyse bis hin zur Publikation und Archivierung von Forschungsdaten."[42]

Weiterhin können Forschungsdaten Daten sein, die „(…) je nach Fachkontext Gegenstand eines Forschungsprozesses sind"[43], müssen also nicht unbedingt im Rahmen eines Forschungsprozesses entstehen, sondern können auch nur Teil eines solchen sein. Forschungsdaten können alle nur erdenklichen Arten von Daten sein, die in allen nur erdenklichen Formaten vorliegen, z. B. Rohdaten aus einem

[37] Vgl. ebd., S. 728.
[38] Vgl. ebd.
[39] Ebd.
[40] Vgl. ebd.
[41] Ebd.
[42] Kindling und Schirmbacher (2013), S. 130.
[43] forschungsdaten.org (2018)

Messinstrument, Bilddaten aus einem Teleskop, Audiodaten aus Interviews, Bilddaten aus der Retro-Digitalisierung samt Volltextdaten, daraus aufbereitete und edierte TEI-Dokumente[44] etc. Forschungsdaten sind somit immer im Bezug zu der Fachdisziplin zu sehen, in der sie entstanden sind.[45]

Einen guten Überblick über die Definition und heterogene Vielfalt von Forschungsdaten geben auch Engelhardt; Funk und Veentjer (2013): Je nach den Erhebungsmethoden und je nach Wissenschaftsdisziplin entstehen sehr unterschiedliche Arten von Forschungsdaten und diese können Anhand ihrer Rolle oder nach Medienart und Datenformat beschrieben werden, ihre Charakterisierung ist grundsätzlich kontextabhängig. Auch im Bezug auf Richtlinien oder Metadatenstandards besteht eine große disziplinäre Vielfalt, so dass daraus jeweils unterschiedliche Anforderungen an das Forschungsdatenmanagement folgen.[46]

Das Projekt DARIAH-DE definiert den Begriff Forschungsdaten für die Geistes- und Kulturwissenschaften wie folgt:

> „Unter digitalen geistes- und kulturwissenschaftlichen Forschungsdaten werden innerhalb von DARIAH-DE all jene Quellen/Materialien und Ergebnisse verstanden, die im Kontext einer geistes- und kulturwissenschaftlichen Forschungsfrage gesammelt, erzeugt, beschrieben und/oder ausgewertet werden und in maschinenlesbarer Form zum Zwecke der Archivierung, Zitierbarkeit und zur weiteren Verarbeitung aufbewahrt werden können."[47]

Es soll so der speziellen Eigenschaften von geisteswissenschaftlicher Forschung und der Heterogenität der zugrunde liegenden Forschungsdaten Rechnung getragen werden.

Lässt man den Bezug zu den Geistes- und Kulturwissenschaften und den Begriff DARIAH-DE außen vor, ergibt sich eine sehr schöne allgemeine Definition von Forschungsdaten, die auch auf andere Fachdisziplinen zutrifft:

> Unter digitalen Forschungsdaten werden all jene Quellen/Materialien und Ergebnisse verstanden, die im Kontext einer Forschungsfrage gesammelt, erzeugt, beschrieben und/oder ausgewertet werden und in maschinenlesbarer Form zum Zwecke der Archivierung, Zitierbarkeit und zur weiteren Verarbeitung aufbewahrt werden können.

In dem von DARIAH-DE entwickelten *Research Data Lifecycle* (siehe Abbildung 2) sind alle in der Definition von Forschungsdaten für die Geistes- und Kulturwissenschaften erwähnten Arbeitsschritte als einzelne Arbeitsschritte enthalten.[48] Er stellt „(…) die Erweiterung eines traditionellen Forschungsworkflows dar, so dass neue Forschungsprozesse auf vorhergehenden Arbeiten aufbauen können und Zwischenschritte publiziert und archiviert werden können"[49]. In ihm sind alle Stationen des Publikationskreislaufs von Abbildung 1 (Seite 10) enthalten, nur sind dessen Schritte 6 (Auswirkungen) ⟶ 1 (Forschung) ⟶ 2 (Darstellung eines geistigen Werkes) feiner aufgeschlüsselt: Hier geht es von *Re-Use* über *Source Definition*, *Processing* und *Data Production* zu *Conclusion* und *Verbalisation*. Über *Peer-Review* (Schritt 3

[44] Vgl. TEI (2018). http://www.tei-c.org/index.xml
[45] Vgl. Oßwald; Scheffel und Neuroth (2012), S. 15.
[46] Vgl. Engelhardt; Funk und Veentjer (2013), S. 5ff.
[47] DARIAH-DE (2018a). https://de.dariah.eu/weiterfuhrende-informationen
[48] Vgl. ebd.
[49] Vgl. Puhl u. a. (2015), S. 43.

im Publikationskreislauf) geht es dann wieder zur *Publikation* (Schritte 4 und 5) und *Re-Use* (Schritt 6): Der Kreis schließt sich.

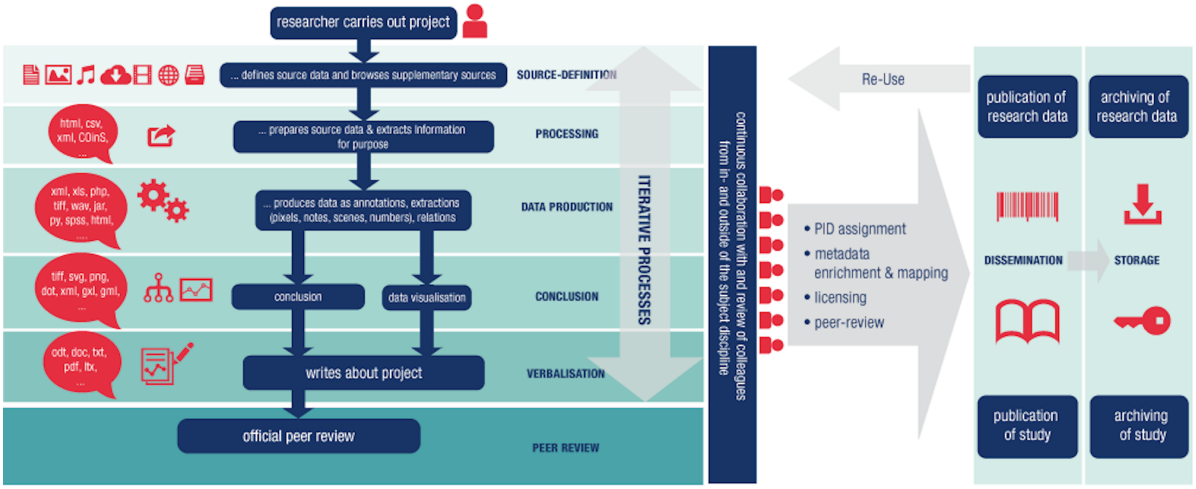

Abbildung 2: Der Research Data Lifecycle unter Einbeziehung von Publikation, Archivierung und Nachnutzung[50]

1.2.1 Publizieren von digitalen Forschungsdaten

Im Rahmen von Open Access und auch *Open Science*[51] ist es in den letzten Jahren immer wichtiger geworden, nicht nur Aufsätze zu Forschungsergebnissen zu veröffentlichen, sondern auch die den Forschungsarbeiten zugrunde liegenden Daten der Wissenschaft und der Öffentlichkeit zur Verfügung zu stellen (sprich: zu publizieren). So kann die Qualität und Korrektheit einer Forschungsarbeit einfacher – oder überhaupt erst – verifiziert werden. Außerdem stehen die Forschungsdaten auch anderen Forscherinnen für weitere, vielleicht auch fachfremde Forschungsfragen zur Verfügung.

In den letzten Jahren konnte in allen Wissenschaftsdisziplinen beobachtet werden, dass Verfügbarkeit und Verarbeitung von digitalen Forschungsdaten in hohem Maße angestiegen sind, und dieser Anstieg wird sich in den kommenden Jahren noch weiter fortsetzen.[52] Für die Zunahme sind laut Rat für Informations Infrastrukturen (2017) zunächst drei Entwicklungen verantwortlich: Zum einen die

Zunahme der Menge an digitalen Daten, die durch die technischen Entwicklungen in der Informationstechnologie möglich wurde. Es können immer größere Datenmengen gespeichert, prozessiert und auch über regionale und nationale Grenzen hinweg genutzt werden. Durch den wissenschaftlichen Erkenntnisgewinn, der durch die Bereitstellung und Nachnutzung dieser Forschungsdaten möglich wird, wird der wissenschaftliche Fortschritt maßgeblich gefördert. Weiterhin bietet die

Verknüpfung von Datenquellen entscheidende Möglichkeiten, in fachübergreifenden Zusammenhängen unterschiedliche Datenbestände miteinander zu verknüpfen. So werden Forschungsdaten

[50] Quelle: Puhl u. a. (2015), S. 43.
[51] Vgl. Wikipedia (2018d). https://de.wikipedia.org/wiki/Offene_Wissenschaft
[52] Vgl. Rat für Informations Infrastrukturen (2017), S. 3.

in neue Kontexte gestellt, klassische Projektgrenzen fließen ineinander und Forschungsdaten werden so „(…) auf unerwartete Weise neu bzw. mehrfach beforscht" und „(…) verändern aber auch ihre Bedeutung"[53]. Als dritter Punkt ist die

Fortentwicklung der Forschungsmethoden zu nennen, denn durch Verfügbarkeit und Verknüpfbarkeit der Forschungsdaten entwickeln sich auch diese weiter, sei es durch die Beobachtung größerer Zeiträume, die Möglichkeit der langfristigen Speicherung oder auch der algorithmischen Auswertung und Nachnutzung. So können Zusammenhänge in großen Datenmengen schnell erfasst werden und „tragen (…) zu einer Beschleunigung des Erkenntnisprozesses, aber auch zu einer enorm tiefgehenden Datennutzung bei"[54].

1.3 Digitale Repositorien

Der Begriff *Repositorium* stammt laut Duden vom Lateinischen Begriff „repositorium" (Aufsatz) ab. Der Duden definiert ein Repositorium als

1. ein Büchergestell oder einen Aktenschrank (veraltet) und
2. einen Ort zur Speicherung von Daten in der EDV und im Internet (Begriff aus der EDV)[55]

Im Zusammenhang mit Open Access (2018a) werden Repositorien definiert als

> „an Universitäten oder Forschungseinrichtungen betriebene Dokumentenserver, auf denen wissenschaftliche Materialien archiviert und weltweit entgeltfrei zugänglich gemacht werden."[56]

Grundsätzlich kann man zwischen institutionellen und disziplinären Repositorien unterscheiden, erstere sind institutionell betriebene Dokumentenserver, die meist von Universitätsbibliotheken, Forschungsorganisationen und anderen Infrastruktureinrichtungen betrieben werden und die Zugehörigen der jeweiligen Institution die Publikation digitaler Arbeiten anbieten. Disziplinär angebotene Repositorien sind eher thematisch fokussiert, und bedienen Wissenschaftler einer bestimmten Fachdisziplin institutsübergreifend.[57]

Das *Reference Model for an Open Archival Information System (OAIS)* ist mit Sicherheit eines der am häufigsten zitierten Dokumente, wenn digitale (Langzeit-)Archive und digitale Repositorien thematisiert werden. Das OAIS beschreibt nicht nur die einzelnen Teile der Infrastruktur eines digitalen Langzeitarchivs, sondern hat einen Blick auf den gesamten Langzeitarchivierungsworkflow, technisch wie organisatorisch. Es bietet eine Basis, digitale Langzeitarchivierung und die zugehörigen Prozesse zu diskutieren und einzelne Komponenten zu erläutern – auch über Fachgrenzen hinweg.[58]

---

[53] Ebd., S. 4.
[54] Ebd.
[55] Duden (2017). https://www.duden.de/rechtschreibung/Repositorium
[56] Open Access (2018b). https://open-access.net/informationen-zu-open-access/repositorien
[57] Vgl. ebd.
[58] Vgl. nestor – Kompetenznetzwerk Langzeitarchivierung (2012), S. 1.
[59] Quelle: Ebd., S. 33.

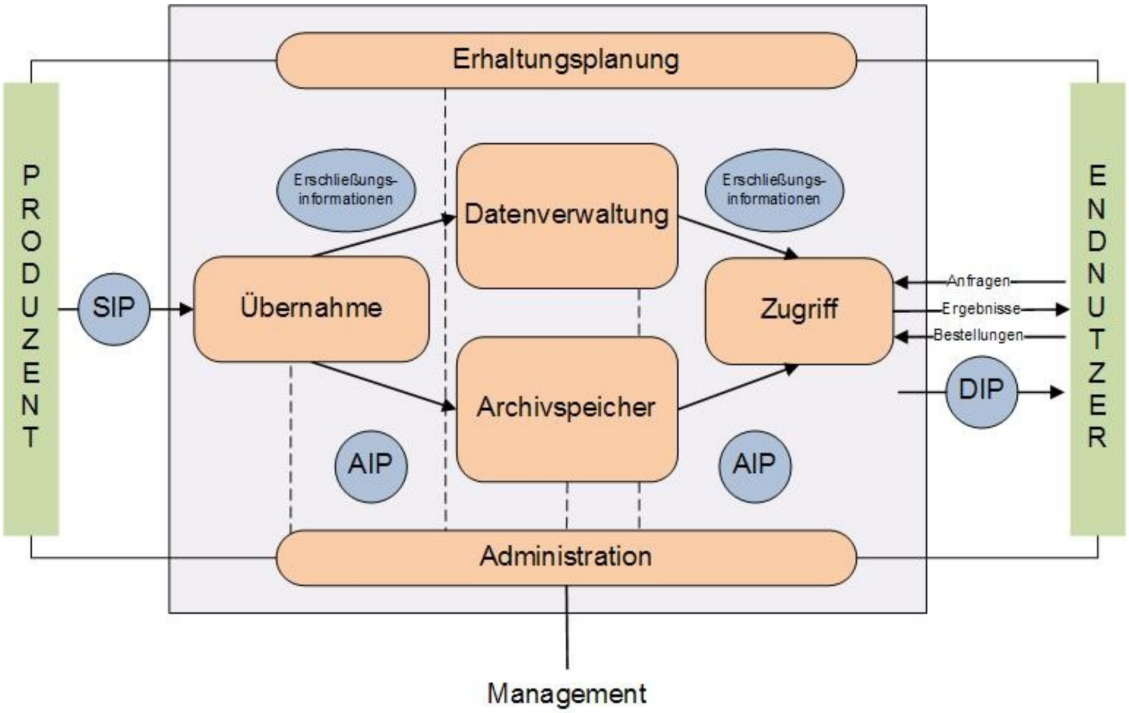

Abbildung 3: OAIS-Funktionseinheiten[59]

Die grundlegenden Funktionseinheiten eines *Open Archival Information Systems* sind in Abbildung 3 dargestellt und werden hier kurz erläutert:

Zur Übernahme von digitalen Daten dienen Funktionen, die die Daten als *Submission Information Package (SIP)* über eine Schnittstelle entgegennehmen und diese nach den Spezifikationen des Repositoriums validieren. Die Daten werden auf die Speicherinfrastruktur – *Archivspeicher* und *Datenverwaltung* – verteilt und liegen nun als *Archival Information Package (AIP)* vor.

Der Archivspeicher dient der sicheren Speicherung der Archivpakete. Diese müssen von der *Übernahme* in den Speicher aufgenommen, verwaltet und wieder an den *Zugriff* geliefert werden. Die Konsistenz der Archivpakete und auch der Datenträger, auf denen sie gespeichert sind, müssen regelmäßig kontrolliert werden.

Die Datenverwaltung hält Funktionen für den Zugriff auf deskriptive und administrative Metadaten vor, die für die Verwaltung benötigt werden und ermöglicht Suchanfragen, so dass Informationen über Archivdaten angefordert werden können.

Die Administration ist verantwortlich für den Betrieb des Repositoriums, was nicht nur technische Bereiche umfasst. Aufgaben der Administration sind die Überwachung des Betriebs, die Konfiguration von Hard- und Software sowie die Erstellung von Berichten über den Archivinhalt. Außerdem werden Archivstandards- und Policies gepflegt.

Mit der Erhaltungsplanung soll das Umfeld des Archivsystems und beinhalteter Daten beobachtet werden, um im Bedarfsfall Empfehlungen auszugeben und frühzeitig auf Entwicklungen zu reagieren, um so die Verfügbarkeit der Inhalte zu garantieren. Dies betrifft das Veralten von Medien- und Dateiformaten sowie Soft- und Hardware, von denen das Archiv abhängig ist.

Der Zugriff schließlich unterstützt die Nutzerin darin, Informationen und Daten im Repositorium zu finden, und Daten als *Dissemination Information Package (DIP)* zu erhalten. Dazu gehören unter Umständen auch Zugriffskontrolle und Zugriffsbeschränkung.

Ein digitales Repositorium sollte, wenn auch nicht zu 100% OAIS-konform, zumindest stark an das OAIS angelehnt sein, so dass alle grundlegenden Infrastrukturkomponenten implementiert sind. Kriterien, nach denen ein Archivsystem bewertet werden kann, liefert das OAIS nicht, es beschreibt lediglich das Modell eines vertrauenswürdigen Archivsystems mit seinen Funktionen und Verantwortlichkeiten.[60] Vertrauenswürdige Systeme für die Langzeitarchivierung forderten bereits Waters und Garrett (1996), für die Schaffung von Vertrauen seien Zertifizierungsprozesse notwendig.[61] Ein System (Repository, Archiv) ist vertrauenswürdig, sofern es „(…) gemäß seinen Zielen und Spezifikationen" operiert.[62]

Zwei Kriterienkataloge, die auf dem OAIS basieren, sind der *Kriterienkatalog vertrauenswürdige digitale Langzeitarchive* der nestor Arbeitsgruppe Vertrauenswürdige Archive[63] und das Dokument *Trustworthy Repositories Audit & Certification: Criteria & Checklist (TRAC)*[64] des Council on Library Resources[65] (CRL) und des Online Computer Library Center[66] (OCLC). Diese beiden Kriterienkataloge wurden 2012 als DIN 31644: *Information und Dokumentation – Kriterien für vertrauenswürdige digitale Langzeitarchive*[67] und ISO 16363: *Audit and certification of trustworthy digital repositories*[68] standardisiert. Nach diesen Standards können Repositorien zertifiziert werden. Die Anforderungen an ein digitales Langzeitarchiv werden in beiden Katalogen in drei Aspekte aufgeteilt: *Organisatorischer Rahmen*, *Umgang mit Objekten* und *Infrastruktur und Sicherheit*.[69]

Als weitere Qualitätsstandards für digitale Repositorien lassen sich das CoreTrustSeal (CTS)[70] und das DINI-Zertifikat[71] nennen. Im Rahmen von DARIAH-DE ist die Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal momentan in Vorbereitung und wird voraussichtlich noch im Laufe des Februars 2018 beantragt.

[60] Vgl. Engelhardt; Funk und Veentjer (2013), S. 8.
[61] Vgl. Waters und Garrett (1996), S. 43f.
[62] nestor – Kompetenznetzwerk Langzeitarchivierung (2008), S. 46.
[63] Vgl. ebd.
[64] Vgl. NARA Task Force on Digital Repository Certification (2007). http://www.crl.edu/PDF/trac.pdf
[65] Vgl. CRL – Center for Research Libraries (2018). http://www.crl.edu
[66] Vgl. OCLC – Online Computer Library Center (2018). https://www.oclc.org
[67] Vgl. DIN 31644:2012-04 (2012). https://www.beuth.de/de/norm/din-31644/147058907
[68] Vgl. International Organization for Standardization (2012). https://www.iso.org/standard/56510.html
[69] Vgl. Engelhardt; Funk und Veentjer (2013), S. 7ff.
[70] Vgl. CoreTrustSeal (2018). https://www.coretrustseal.org
[71] Vgl. Müller u. a. (2016)

2 Das Textgrid Repository

2.1 Die Projekte TextGrid und DARIAH-DE

Das vom Bundesministerium für Bildung und Forschung (BMBF) geförderte Projekt *TextGrid*[72] startete im Jahr 2006 und durchlief drei Förderphasen. Ziel des Forschungsverbunds war es, eine Virtuelle Forschungsumgebung[73] (VFU) für die Geistes- und Kulturwissenschaften zu schaffen. Diese wurde bis zum Projektende weiterentwickelt und die Pflege der technischen Infrastruktur verstetigt und stabilisiert. Die VFU besteht aus dem Frontend TextGrid Laboratory – *TextGridLab*[74] – und aus dem TextGrid Repository – *TextGridRep*[75]. Das TextGridLab ist implementiert als Eclipse RCP-Client[76] zur kooperativen Bearbeitung geisteswissenschaftlicher Daten bis hin zur deren Publikation – von digitalisierten Buchseiten und deren Volltexten über musikwissenschaftliche Daten im MEI-Format[77] oder Wörterbücher im TEI-Format bis hin zu hybriden Editionen. Als Backend und Middleware für die Datenspeicherung und den Zugriff dient das TextGridRep mit seinen Komponenten TG-crud, TG-auth*, TG-publish/TG-import und TG-search für die Speicherung, die Authentifizierung und Autorisierung, den Publikationsvorgang sowie für Recherche und Zugriff. Ein Publikationsworkflow für die Publikation von Daten in das TextGrid Repository existiert bereits auf mehreren Ebenen.[78]

Inzwischen wurde die gesamte Architektur des TextGrid Repositorys auf die DARIAH-DE Infrastruktur transferiert, so dass ein technischer und administrativer Betrieb im Rahmen des vom BMBF geförderten Projekts DARIAH-DE gewährleistet werden kann.[79] Auch die Betreuung und Pflege des TextGridLab sowie die Schulung von TextGrid-Nutzern wird von DARIAH-DE angeboten und von am Projekt beteiligten Einrichtungen und Wissenschaftlerinnen übernommen.[80]

Die Projekte *DARIAH-DE*[81] und der europäische Verbund DARIAH-EU[82] haben zum Ziel, eine digitale Forschungsinfrastruktur für die Geisteswissenschaften zur Verfügung zu stellen, so dass geisteswissenschaftliche Vorhaben durch digitale Methoden, Verfahren und Werkzeuge unterstützt werden. Hierzu zählen beispielsweise die Bereitstellung von Werkzeugen für kooperatives wissenschaftliches Schreiben wie Wiki und Etherpad sowie von Entwicklerwerkzeugen wie Subversion Repository, Bugtracker und Continuous Integration-Systeme. Weiterhin werden fachwissenschaftliche Dienste wie die Generische Suche, die Collection Registry oder das Visualisierungstool Geo-Browser verfügbar gemacht.[83] Zwei

[72] Vgl. TextGrid (2017a). https://textgrid.de/projekt
[73] Vgl. Schwerpunktinitiative Digitale Information (2011). https://www.allianzinitiative.de/fileadmin/user_upload/www.allianzinitiative.de/2011_VRE_Definition.pdf
[74] Vgl. TextGrid Laboratory (2018). https://textgrid.de/download
[75] Vgl. TextGrid (2018a). https://textgridrep.org
[76] Vgl. Eclipse (2018). https://wiki.eclipse.org/Rich_Client_Platform
[77] Vgl. MEI (2018). http://music-encoding.org
[78] Vgl. Funk (2014), S. 5f.
[79] Vgl. Schmunk und Funk (2016), S. 216.
[80] Vgl. TextGrid (2018b). https://textgrid.de/nachhaltigkeit
[81] Vgl. DARIAH-DE (2018b). https://de.dariah.eu
[82] Vgl. DARIAH-EU (2018). https://dariah.eu
[83] Vgl. DARIAH-DE (2018c). https://de.dariah.eu/list-services

Hauptpunkte, auf die DARIAH-DE speziell eingeht, sind die Authentifizierungs- und Autorisierungs-Infrastruktur[84] (AAI) sowie die Storage-Infrastruktur. Letztere wird durch den direkten Zugriff über eine Storage-API auf den DARIAH-DE Storage implementiert.[85] Weiterhin wurde Ende 2017 das DARIAH-DE Repository[86] in den Produktivbetrieb genommen, das als Publikationsplattform für geisteswissenschaftliche Forschungsdaten fungiert. Mit dem DARIAH-DE Publikator[87] ist es für die Nutzer einfach und intuitiv möglich, Forschungsdaten in das DARIAH-DE Repository einzuspielen. Ein Publikationsworkflow für das DARIAH-DE Repository ist aus den Erfahrungen der Erstellung des TextGrid Repositorys entwickelt und implementiert worden, von diesen wird auch der TextGrid-Publikationsworkflow profitieren.[88]

Geistes- und Kulturwissenschaftlerinnen steht mit den beiden Repositorien, die von DARIAH-DE und TextGrid angeboten werden, ein Angebot zur nachhaltigen und referenzierbaren Speicherung von Forschungsdaten zur Verfügung. Beide Repositorien fußen auf dem selben technologischen Framework und bieten sowohl verschiedene Modelle zu Datenpräsentation und -kuration als auch verschiedene Schnittstellen zur spezifischen Nachnutzung der enthaltenen Daten an.

Das *TextGrid Repository* ist für XML/TEI-Formate und das editorische Arbeiten sowie für die Publikation aus dem TextGrid Laboratory heraus optimiert, wohingegen das *DARIAH-DE Repository* eher für generische Datenformate konzipiert wurde und der Publikationsprozess über das DARIAH-DE Portal bzw. den Publikator angeboten wird. Beide Repositorien adressieren Wissenschaftler an Universitäten und Forschungseinrichtungen, die ihre Forschungsdaten langfristig im Rahmen einer nachhaltigen Lösung speichern wollen.[89]

2.2 Architektur

Das TextGrid Repository ist ein fachwissenschaftliches Langzeitarchiv, das die langfristige Verfügbarkeit und Zugänglichkeit von geisteswissenschaftlichen Forschungsdaten sichert. Es ist für ein kooperatives Bearbeiten und intuitives Publizieren von geisteswissenschaftlichen Forschungsdaten entwickelt worden. Die Architektur des TextGrid Repositorys ist in Abbildung 4 auf Seite 22 abgebildet. Der Kern der Middleware besteht aus den drei Komponenten für die Datenverwaltung und -kontrolle:

- dem rollenbasierten Rechtemanagement *TG-auth**[90],
- dem Suchindex für Metadaten und Volltexte (ElasticSearch) und dem RDF Triplestore für Beziehungsmetadaten (Sesame) *TG-search* sowie
- dem Speicherdienst *TG-crud*, der für mehr oder weniger atomare Speicheroperationen zuständig ist (in der Hauptsache CREATE, UPDATE, RETRIEVE und DELETE)

[84] Vgl. DARIAH-DE (2018d). https://wiki.de.dariah.eu/display/publicde/DARIAH+AAI+Documentation
[85] Vgl. Funk (2014), S. 6.
[86] Vgl. DARIAH-DE (2017a). https://de.dariah.eu/repository
[87] Vgl. DARIAH-DE (2017b). https://de.dariah.eu/publikator
[88] Vgl. Funk (2014), S. 26.
[89] Vgl. TextGrid (2018c). https://wiki.de.dariah.eu/display/publicde/Das+DARIAH-DE+Repository+und+das+TextGrid+Repository
[90] Das ***** steht stellvertretend für Authe**N**tication und Authori**Z**ation.
[91] Quelle: TextGrid

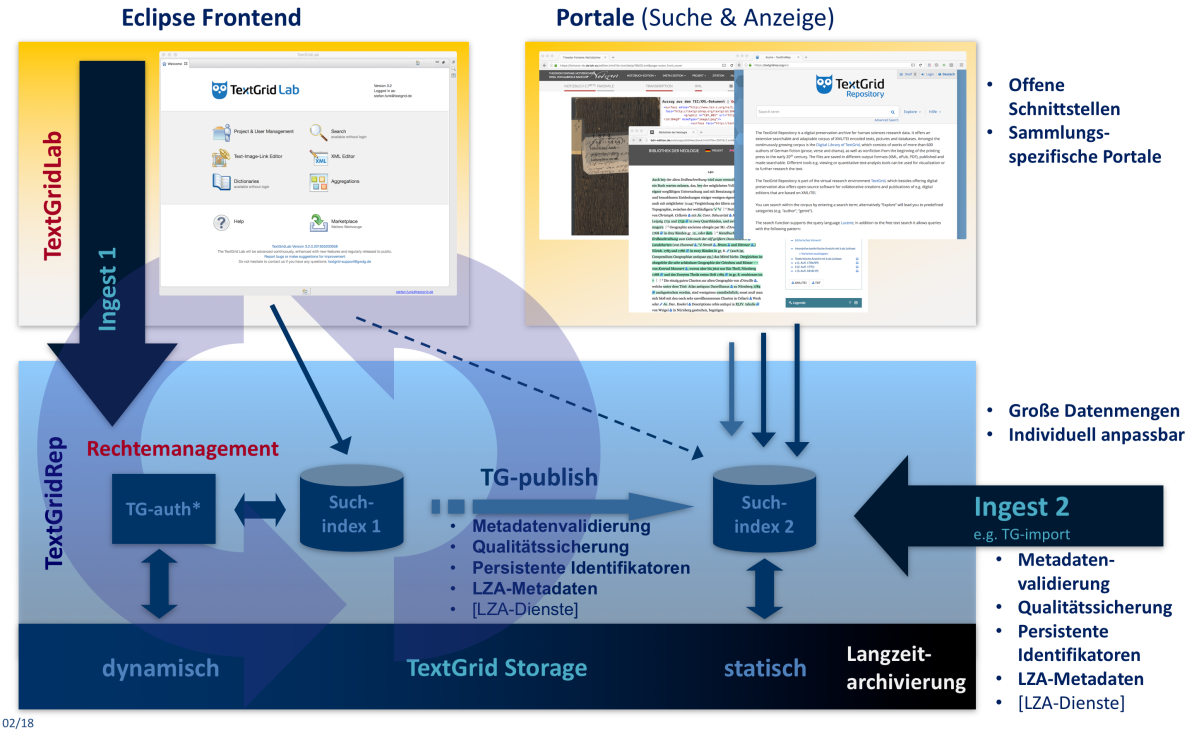

Abbildung 4: Die Architektur des TextGrid Repositorys[91]

Wenn Forscherinnen mit dem TextGridLab arbeiten und Dateien aufrufen, neu erstellen oder bearbeiten, operiert das TextGridLab über die TG-crud-API im TextGridRep (*Ingest 1*). Die Dateien sind durch das Rechtemanagement auf Projektebene geschützt (*Suchindex 1*) und die Daten werden im *dynamischen TextGrid-Storage* abgelegt. Der Ersteller des zu bearbeitenden Projekts (Rolle: *Projektleiter*) kann seine Kollegen als *Bearbeiter* oder *Beobachter* zu den eigenen Projekten einladen. Mit den Rollen sind bestimmte Rechte verbunden: Bearbeiter dürfen beispielsweise Objekte des Projekts lesen und bearbeiten, Beobachter dürfen diese nur lesen, Projektleiter dürfen Inhalte publizieren und Rechte delegieren, *Administratoren* dürfen Objekte löschen.

Sollen die im TextGridLab erarbeiteten Ergebnisse und Forschungsdaten (natürlich auch Zwischenergebnisse) veröffentlicht werden, wird ein Publikationsvorgang im TextGridLab angestoßen (*TG-publish*): Daten werden vom dynamischen Speicherbereich in den statischen verschoben, Indexdaten werden kopiert und die Objekte werden mit Persistenten Identifikatoren (ePIC-Handles) verknüpft, um sie langfristig nachzuweisen und referenzierbar zu halten. Die Daten sind dann im TextGrid Repository weltweit zugänglich – beispielsweise über den TextGrid Repository-Browser[92] – und nicht mehr veränderbar.

Eine Möglichkeit, auch große Datenmengen automatisiert in das TextGrid Repository zu importieren, kann durch eine Aufbereitung der Daten durch das Tool *TG-import* und anschließendes Einspielen über die Schnittstellen des *Ingest 2* erfolgen. TG-import basiert wie TG-publish auf der *kopal Library for Retrieval and Ingest*[93] (koLibRI).

[92] Vgl. TextGrid Repository (2017). https://textgridrep.org/repository.html
[93] Vgl. DP4lib (2018). http://dp4lib.langzeitarchivierung.de/index_koLibRI.php.de

2.3 Der TextGrid-Publikationsworkflow

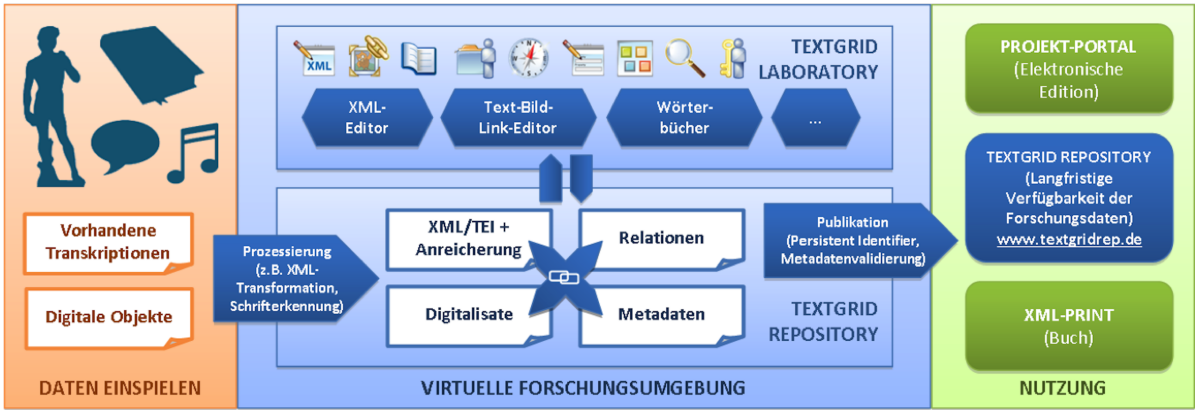

Abbildung 5: Workflow der Virtuellen Forschungsumgebung TextGrid[94]

In Abbildung 5 sind die einzelnen Komponenten der Virtuellen Forschungsumgebung TextGrid dargestellt: Auf der linken Seite beginnend mit den Daten, die bearbeitet werden sollen, mittig das TextGridLab als Einstiegspunkt und schließlich auf der rechten Seite die verschiedenen Möglichkeiten der Publikation. Die Arbeit mit der VFU TextGrid wird anhand eines beispielhaften Workflows im Folgenden kurz erläutert.[95]

Mit der Datenauswahl der zu publizierenden Daten beginnt der Workflow. Dies können eine Reihe von eingescannten Bücherseiten sein, für die Volltexte im XML/TEI-Format vorliegen oder noch erstellt werden sollen. Diese Daten werden nun in das TextGridLab eingespielt. Im TextGridLab können die Daten im geschützten Bereich gemeinsam von verschiedenen Nutzerinnen bearbeitet und für die Publikation vorbereitet werden. Für die Bearbeitung bieten sich der TextGrid XML-Editor[96], der Text-Bild-Link-Editor[97] (TBLE) sowie verschiedene digitale Wörterbücher an. Die Daten können kooperativ geordnet, strukturiert und mit Metadaten angereichert werden. Sobald die Bearbeitung abgeschlossen ist, können die Daten als eine Edition oder eine Kollektion publiziert werden.[98]

2.4 Importworkflows in TextGrid

Der Datenimport wird anhand von Abbildung 6 auf Seite 24 illustriert. Für das Publizieren aus dem TextGridLab heraus werden publizierfähige Daten vorausgesetzt. Diese können entweder mit der Import-Perspektive[99] des TextGridLab oder in einem Editor des TextGridLab erstellt und über den Speichervorgang importiert werden. In beiden Fällen durchlaufen die zu importierenden Daten den folgenden Importworkflow, dessen Kern der Dienst TG-crud ist. Dieser Workflow entspricht **Ingest 1** von Abbildung 4 auf Seite 22:

[94] Quelle: TextGrid
[95] Vgl. Funk; Veentjer und Vitt (2013), S. 12ff.
[96] Vgl. TextGrid XML-Editor (2018). https://wiki.de.dariah.eu/display/TextGrid/XML-Editor
[97] Vgl. TextGrid Text-Bild-Link-Editor (2018). https://wiki.de.dariah.eu/display/TextGrid/Text-Bild-Link-Editor
[98] Vgl. Funk; Veentjer und Vitt (2013), S. 12ff.
[99] Vgl. TextGridLab Nutzerhandbuch 2.0 (2015a). https://wiki.de.dariah.eu/pages/viewpage.action?pageId=40220393

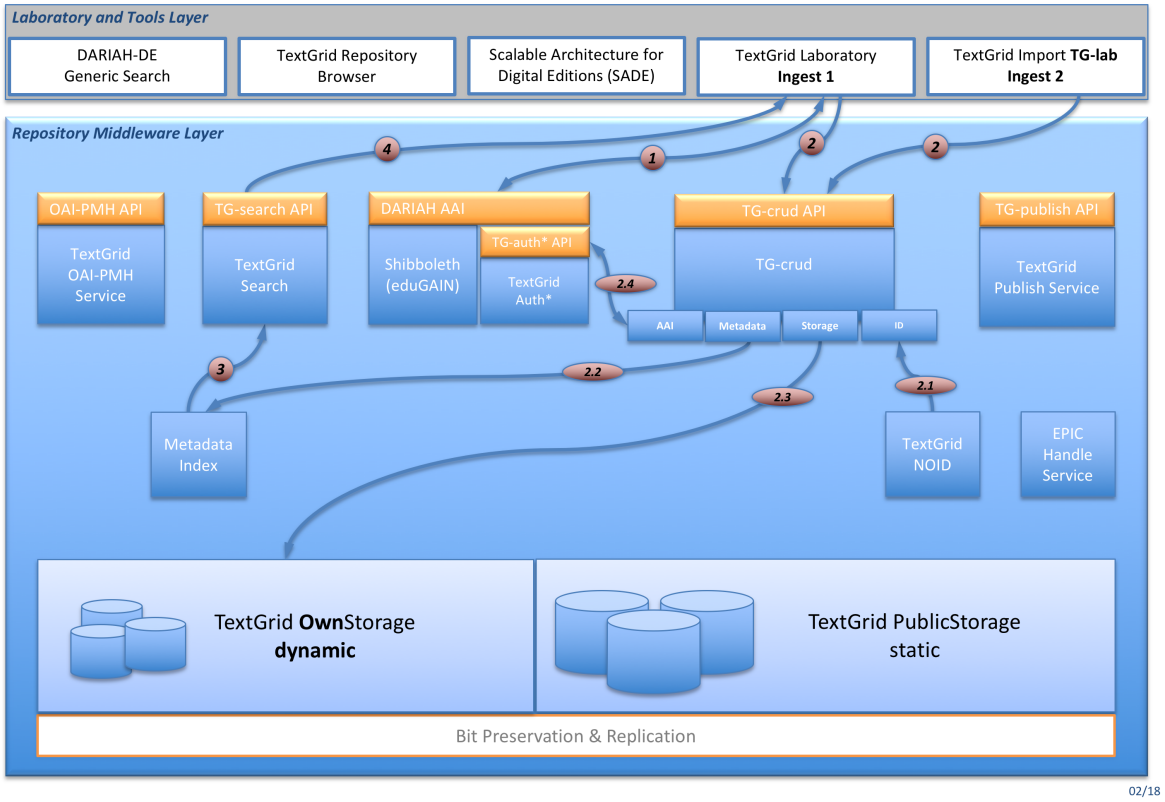

Abbildung 6: Importworkflows in TextGrid

1. Anmeldung über das TextGridLab an der *DARIAH AAI* (Authentifizierung) und *TextGrid Auth** (Autorisierung und Projekterstellung)
2. Hochladen und/oder Erstellen von Daten im TextGridLab. *TG-crud#CREATE*
    1. generiert eine TextGrid-URI für das Objekt per *TG-noid*,
    2. schreibt die Metadaten in den *Metadatenindex* (ElasticSearch für Volltext- und Metadaten, Relationsdaten und strukturelle Metadaten in die RDF-Datenbank Sesame),
    3. schreibt Daten und Metadaten in den *dynamischen TextGrid-Storage* (OwnStorage) und
    4. registriert die Ressource bei *TextGrid Auth** (OpenRBAC).
3. *TG-search* schließlich greift für Anfragen auf den Metadatenindex zu (ElasticSearch und Sesame) und
4. liefert die Ergebnisse an Klienten wie beispielsweise das *TextGridLab*.[100]

[100] Vgl. TG-search-Dokumentation: TextGrid Search (2017). http://textgridlab.org/doc/services/submodules/tg-search/docs/index.html

Ebenfalls in Abbildung 6 auf Seite 24 ist ein weiterer Weg des Imports von Daten in das TextGridLab aufgeführt, der nicht direkt in Abbildung 4 (Seite 22) illustriert ist. Dieser ist sinnvoll für den Import großer Datenmengen, die eine spezifische Präprozessierung benötigen – beispielsweise die Anreicherung von Metadaten aus externen Quellen – und für die Bearbeitung im TextGridLab vorgesehen sind. Hier geschieht der Import über das Tool TG-import und über *TG-crud*. Hierfür wird zum einen eine *SessionID* benötigt und zum anderen die *ProjectID* des Projekts, in das die Daten eingespielt werden sollen. Beides kann über das TextGridLab bezogen und in die Konfigurationsdatei des TG-import-Moduls eingetragen werden, das die *kopal Library for Retrieval and Ingest*[101] und die Module für den TextGrid-Import[102] nutzt:

1. Anmeldung an der *DARIAH AAI* und *TextGrid Auth** wie oben und Kopieren von SessionID und ProjectID in die TG-import-Konfiguration.
2. Importieren mit Hilfe des Tools *TextGrid Import*[103]. Letztendlich wird von TG-import direkt TG-crud#CREATE angesprochen, der dann wie oben die Daten importiert (siehe Punkte 2.1 bis 2.4), auch
3. *TG-search* wird wie oben über das TextGridLab angefragt und
4. liefert ebenfalls die Ergebnisse an die Klienten aus.

Sobald die Daten importiert wurden, sind diese im TextGridLab verfügbar und können dort mit allen zur Verfügung stehenden Werkzeugen bearbeitet und für die Publikation vorbereitet werden, beispielsweise mit dem XML-Editor oder auch dem Text-Bild-Link-Editor, je nach Anwendung.

2.5 Publikationsworkflows in TextGrid

Nach dem Import von Daten in das TextGridLab – und ihrer Bearbeitung – wird nun auf die beiden Publikationsprozesse eingegangen, die für eine Publikation von Daten in das Textgrid Repository bereitstehen: Es besteht zum einen die Möglichkeit, im TextGridLab vorhandene und für die Publikation aufbereitete Daten direkt von dort aus zu publizieren. Dies geschieht über den Dienst **TG-publish**. Die Daten sind nach dem Publizieren aus dem TextGridLab sofort öffentlich und können nicht mehr gelöscht oder korrigiert werden.

Zum anderen ist es möglich, Daten über die API von **TG-crud public**[104] direkt in das TextGrid Repository zu importieren und damit zu publizieren. Um die Daten sinnvoll aufzubereiten und zu strukturieren, bietet es sich an, hierfür das Tool **TG-import** zu nutzen. Diese Import-Bibliothek bietet dazu verschiedene Module, Policies und Konfigurationsmöglichkeiten an. Eine Besonderheit des Imports mit TG-import ist, dass alle Daten zunächst in die sogenannte *Sandbox* eingespielt werden. Nach dem Publikationsvorgang sind alle Daten bereits genauso öffentlich wie bei einem Import über das TextGridLab, jedoch sind sie

[101] Vgl. koLibRI (2017a). https://projects.gwdg.de/projects/kolibri/repository

[102] Vgl. koLibRI (2018a). https://projects.gwdg.de/projects/kolibri/repository/revisions/master/kolibri-addon-textgrid-import

[103] Vgl. TextGrid Import (2017a). http://textgridlab.org/doc/services/submodules/kolibri/kolibri-addon-textgrid-import/docs/index.html

[104] Vgl. TextGrid CRUD (2017). http://textgridlab.org/doc/services/submodules/tg-crud/service/tgcrud-webapp/docs/index.html#api-documentation

zunächst nicht über die öffentliche Suche auffindbar. Die Daten können nun entweder – nach interner Begutachtung – gelöscht oder aber endgültig publiziert werden.

2.5.1 Publikation über das TextGridLab

In Abbildung 7 ist der Publikationsprozess in das TextGrid Repository dargestellt, das im TextGridLab vorhandene Daten publiziert. Der Datenimport geschieht wie in Kapitel 2.4 skizziert über das TextGridLab oder über TG-import. Alle zu veröffentlichenden Objekte können in einer *Edition* oder einer *Kollektion* strukturiert und diese mitsamt aller enthaltenen Objekte publiziert werden.[105] Dieser Workflow entspricht dem Pfeil **TG-publish** auf Abbildung 4, Seite 22:

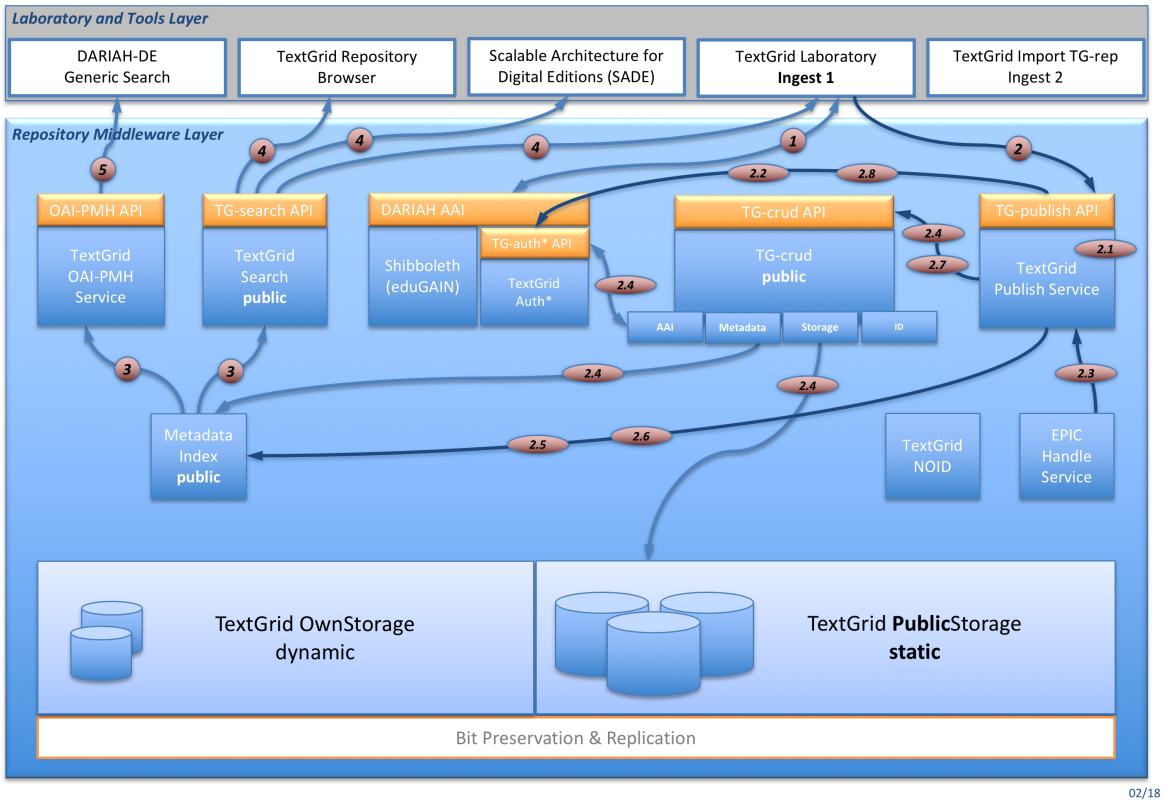

Abbildung 7: Publikationsworkflow über das TextGridLab

1. Hier auch Anmeldung über das TextGridLab an der *DARIAH AAI* und *TextGrid Auth** (siehe Kapitel 2.4).
2. Anstoßen des Publikationsprozesses vom TG-lab aus, liegen die SessionID und die TextGrid-URI der zu publizierenden Edition oder Kollektion vor, kann auch direkt über die TG-publish-API publiziert werden.[106] TG-publish führt folgende Module aus:

[105] Vgl. TextGridLab Nutzerhandbuch 2.0 (2015b). https://wiki.de.dariah.eu/pages/viewpage.action?pageId=40220493
[106] Vgl. TextGrid Publish (2017). http://textgridlab.org/doc/services/submodules/kolibri/kolibri-tgpublish-service/docs/index.html

1. `PublishCheckEdition` (TG-publish) – Als erstes wird geprüft, ob es sich bei dem zu publizierenden Objekt tatsächlich um eine Edition oder eine Kollektion handelt, denn nur diese beiden Objektarten sind für die Publikation vorgesehen. Das hat hauptsächlich den Grund, dass für beide Objektarten eine Reihe von Metadaten als erforderlich definiert sind, so dass die publizierten Objekte sinnvoll ausgezeichnet und in der Suche auffindbar sind.
2. `CheckIsPublic` (TG-auth*) – Alle Objekte, die publiziert werden sollen, werden darauf getestet, ob sie bereits publiziert wurden bzw. ob bereits publizierte Objekte referenziert werden. Falls ja, müssen diese nicht mehr publiziert werden, sie können als Referenz bestehen bleiben und aus dem Publikationsprozess herausgenommen werden.
3. `GetPids`(TG-crud) – Für jedes Objekt wird beim TG-pid-Service ein Persistenter Identifikator angefordert und generiert. Es wird für jedes Objekt eine Checksumme generiert, die Dateigröße ausgelesen und beide Werte werden in die PID-Metadaten übernommen.
4. `ModifyAndUpdate` (TG-crud) – Hier werden die TextGrid-Metadaten um Checksumme und PID angereichert und aktualisiert, bei Aggregationen außerdem die Daten selbst – sofern konfiguriert, da die Referenzen auf die enthaltenen Objekte von TextGrid-URIs zu PIDs umgeschrieben werden können.
5. `CopyElasticSearchIndex` (ElasticSearch) – Die in Schritt 4 aktualisierten Metadaten werden in den ElasticSearch-Index der öffentlichen Instanz kopiert. Sie verbleiben auch in der nicht-öffentlichen Instanz, damit sie im TextGridLab noch gefunden werden können.
6. `CopyRelationData` (Sesame) – Ebenso werden die Relationsdaten in die öffentliche Instanz der Sesame-Datenbank kopiert. Sie sind nun ebenfalls öffentlich zugänglich und auch weiterhin vom TextGridLab aus erreichbar.
7. `MoveToStaticGridStorage` (TG-crud) – In diesem Schritt werden die Daten selbst vom dynamischen TextGrid-Storage in den statischen verschoben. Das TextGridLab kann publizierte Objekte noch lesen, denn auch der non-public TG-crud liefert publizierte Objekte aus. So ist es nicht nötig, den Datenbestand zu verdoppeln.
8. `UpdateTgauth` (TG-auth) – Zum Schluss wird TG-auth* mitgeteilt, dass alle Objekte nun publiziert sind. Sie können dann nicht mehr verändert werden, haben den Status „publiziert" und als einzige Operation kann ein `read` auf ihnen ausgeführt werden.

3. *TG-search* und der *TextGrid OAI-PMH-Service*[107] greifen auf den Metadatenindex zu, um Suchergebnisse aus dem TextGrid Repository zu erstellen und auszuliefern.
4. Von TG-search wiederum werden der *TextGrid Repository-Browser*[108], etwaige SADE-Installationen[109] sowie das TextGridLab bedient, das auch die im TextGrid Repository publizierten Daten im Navigator[110] sowie in der Such-Perspektive[111] anzeigen kann.

[107] Vgl. TextGrid OAI-PMH (2018). http://textgridlab.org/doc/services/submodules/oai-pmh/docs_tgrep/index.html
[108] Vgl. TextGrid (2018a). https://textgridrep.org
[109] Vgl. TextGridLab Nutzerhandbuch 2.0 (2018a). https://wiki.de.dariah.eu/display/TextGrid/Publikationswerkzeug+SADE
[110] Vgl. TextGridLab Nutzerhandbuch 2.0 (2018b). https://wiki.de.dariah.eu/pages/viewpage.action?pageId=40220331
[111] Vgl. TextGridLab Nutzerhandbuch 2.0 (2018c). https://wiki.de.dariah.eu/display/TextGrid/Suche

5. Letztendlich greifen Klienten wie die *Generische Suche* von DARIAH-DE auf den TG-oaipmh-Service zu, so dass auch die Daten im TextGrid Repository von der Generischen Suche indiziert und dort recherchiert werden können.

2.5.2 Publikation über TG-import

Für die Publikation von Daten direkt in das TextGrid Repository müssen diese – je nach Import-Policy[112] – in geeigneter Form vorliegen:

- Die Policy `aggregation_import` erstellt TextGrid-Metadaten aus den zu importierenden Dateien. Die Daten müssen lediglich in einem Dateiordner liegen. Für jeden Unterordner wird jeweils eine TextGrid-Aggregation erzeugt, so dass die Ordnerstruktur in TextGrid nachgebildet wird.
- Bei Nutzung der Policy `complete_import` müssen die TextGrid-Metadaten für jede zu importierende Datei bereits vorliegen, genauso wie alle Strukturobjekte wie TextGrid-Aggregationen und deren Metadaten. TG-import importiert lediglich alle vorhandenen Dateien.
- Die dritte Policy `dfgviewermets_import` benötigt eine DFG-Viewer-METS-Datei[113], verarbeitet die dort enthaltenen Strukturinformationen für die Nachbildung in TextGrid und kopiert alle referenzierten Bilder.
- Weitere Policies steuern die finale Publikation und das Löschen von Objekten sowie die Fortsetzung eines unterbrochenen Import- oder Publikationsvorgangs: `publish_import`, `delete_import`, und `continue_import`.

Das Tool **TG-import** muss heruntergeladen und konfiguriert werden, bereitet die Daten geeignet auf und nutzt dann als letzten Schritt des Import-Vorgangs **TG-crud public** zum Importieren der Daten.

1. Hier auch Anmeldung über das TextGridLab an der *DARIAH AAI* und *TextGrid Auth** sowie Kopieren von SessionID und ProjectID in die TG-import-Konfiguration (siehe Kapitel 2.4, Seite 23).
2. Anstoßen des Publikationsprozesses per TG-import, es wird zusätzlich zu SessionID und ProjectID die TextGrid-URI der zu publizierenden Edition bzw. Kollektion aus Schritt 1 benötigt. Je nach verwendeter TG-import-Konfiguration (verschiedene Service-Endpunkte) können Daten entweder direkt in das TextGridRep oder auch für die weitere Bearbeitung in das TextGridLab eingespielt werden. TG-import führt folgende Module aus, beispielhaft wird hier die TG-import-Policy `aggregation_import` dokumentiert. Alle möglichen Policies und Konfigurationsmöglichkeiten sind in der TG-Import Dokumentation[114] ausführlich beschrieben.

[112] Vgl. TextGrid Import (2017b). https://textgridlab.org/doc/services/submodules/kolibri/kolibri-addon-textgrid-import/docs/import_and_configuration.html#editing-the-config-file

[113] Vgl. DFG-Viewer (2018). https://dfg-viewer.de und METS (2018). https://www.loc.gov/standards/mets

[114] Vgl. TextGrid Import (2017a). http://textgridlab.org/doc/services/submodules/kolibri/kolibri-addon-textgrid-import/docs/index.html

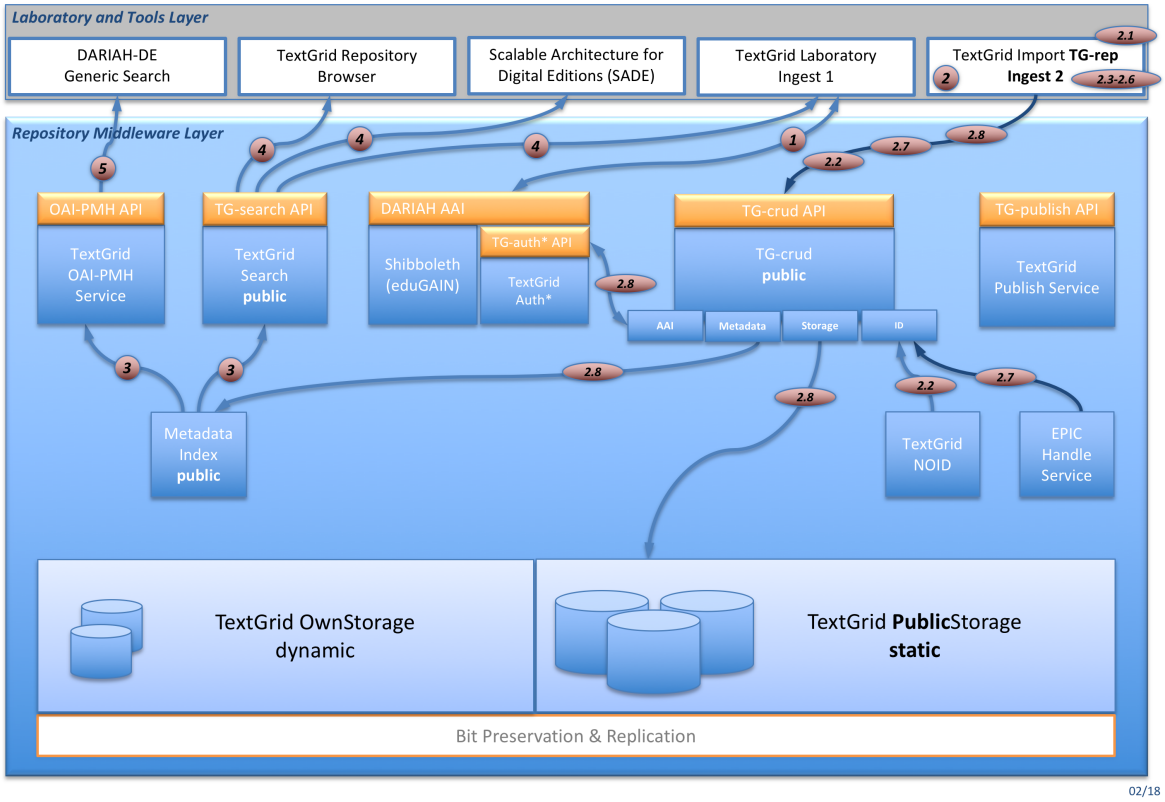

Abbildung 8: Der Publish-Workflow im TextGridRep

1. `FileCopyBase` (TG-import) – Da die zu publizierenden Dateien auf der Festplatte des publizierenden Nutzers liegen, wird durch ein Kopieren verhindert, dass die originalen Daten verändert werden, denn während des Publikationsprozesses werden diverse Umschreibe- und Anreicherungsoperationen vorgenommen, beispielsweise werden lokale Dateipfade in TextGrid-URIs umgeschrieben und die PIDs in die Metadaten eingetragen.

2. `GetUris` (TG-crud) – Für jede Datei, die importiert werden soll, wird eine TextGrid-URI von TG-crud angefordert und dort generiert.

3. `MetadataGenerator` (TG-import) – Dieses Modul untersuch jede Datei mit dem Tool JHOVE[115] (JSTOR/Harvard Object Validation Environment), um das Dateiformat zu erkennen.

4. `TextgridMetadataProcessor` (TG-import) – Jede Datei braucht für den Import in das TextGridLab und TextGridRep eine Metadatendatei. Diese wird in der Policy `aggregation_import` aus einer Vorlage erzeugt, die noch mit Titel und Format ergänzt wird. Der Titel wird aus dem Dateinamen erzeugt, das Format wird im vorhergehenden Modul (MetadataGenerator) von JHOVE erkannt und extrahiert.

5. `CreateAggregations` (TG-import) – Für jeden Ordner, der für die Publikation vorgesehen ist, wird in diesem Modul eine TextGrid-Aggregation erzeugt, so dass die Ordnerstruktur

[115] Vgl. JHOVE (2018). http://jhove.openpreservation.org

der einzuspielenden Daten im TextGrid Repository nachgebildet wird.

6. `RenameAndRewrite` (TG-import) – Um alle einzuspielenden Dateien korrekt referenzieren zu können, werden zunächst alle Dateipfade durch die generierten TextGrid-URIs ersetzt, denn die Dateipfade der Festplatte des Nutzers sind im TextGrid Repository nicht verwendbar, weil nicht eindeutig.

7. `GetPidsAndRewrite` (TG-import) – Als vorletzter Schritt werden PIDs vom TG-pid-Service angefordert und generiert, und es werden hier teilweise die TextGrid-URIs durch die ePIC PIDs ersetzt.

8. `SubmitFiles` (TG-crud public) – Die Dateien und ihre Metadaten, also das gesamte TextGrid-Objekt[116], werden über TG-crud in das TextGrid Repository eingespielt. Sie sind nun öffentlich verfügbar, mit einem Persistenten Identifikator versehen und können so dauerhaft und eindeutig referenziert werden.

3. *TG-search* und der *TextGrid OAI-PMH-Service* greifen auf den Metadatenindex zu, um Suchergebnisse aus dem TextGrid Repository zu erstellen und auszuliefern (siehe Kapitel 2.5.1 auf Seite 26).

4. Von TG-search werden der *TextGrid Repository-Browser*, etwaige *SADE-Installationen* sowie das *TextGridLab* bedient (siehe ebenfalls Kapitel 2.5.1).

5. Letztendlich greifen Klienten wie die DARIAH-DE *Generische Suche* auf den TG-oaipmh-Service zu (siehe ebenfalls Kapitel 2.5.1).

Nach der Beschreibung von Architektur sowie Import- und Publikationsprozessen des TextGrid Repositorys werden im folgenden Kapitel nun einige Use Cases beschrieben, die zuvor behandelte Funktionen und Workflows des TextGridRep im produktiven Betrieb anwenden und ihre digitalen Daten im Rahmen verschiedener Forschungsprozesse dort publiziert haben bzw. publizieren.

[116] Vgl. TextGridLab Nutzerhandbuch 2.0 (2018d). https://wiki.de.dariah.eu/display/TextGrid/TextGrid+Objects

3 Use Cases

In diesem Kapitel werden vier Use Cases vorgestellt, von denen sich drei direkt auf Arbeiten in Forschungsprojekten beziehen, die mit dem TextGrid Repository bzw. dem TextGridLab und angeschlossenen Tools arbeiten und auf diesem Weg ihre Forschungsdaten vorbereiten und schließlich veröffentlichen. Es werden die Projekte und ihre Ziele vorgestellt sowie die Art ihrer Arbeit mit dem TextGrid Repository. Der vierte Use Case bezieht sich auf die allgemeine Nutzung des TextGridLab und seiner Publikationsprozesse.

3.1 Use Case #1 – Die Digitale Bibliothek bei TextGrid

Die *Digitale Bibliothek* von TextGrid beinhaltet eine Sammlung von in XML/TEI erschlossenen Texten deutscher Autoren vom Beginn des Buchdrucks bis zu den ersten Jahrzehnten des 20. Jahrhunderts. Diese digitale Bibliothek wurde vom Forschungsverbund TextGrid im Jahr 2010 von der Volltextbibliothek zeno.org[117] erworben.[118] und steht nach einer umfangreichen Bearbeitung durch Mitarbeiterinnen und Mitarbeitern des Projekts TextGrid[119] nun auch zur freien wissenschaftlichen Nachnutzung unter der Lizenz CC BY 3.0 DE[120] zur Verfügung.

Die Texte der Sammlung „Literatur" liegen bereits seit der Release des TextGrid Repositorys im Jahr 2011 in digitaler Form vor[121], seit dem werden belletristische Texte von 693 Autoren für die wissenschaftliche Verwendung zur Verfügung gestellt. Die Texte wurden mit Hilfe des Import-Tools TG-import[122] mit der Policy `complete_import` in das Repository eingespielt.[123] Zur Zeit sind im TextGrid Repository 94.461 Werke[124] der Digitalen Bibliothek mit 106.832 XML/TEI-Dateien[125] nachgewiesen und so für die geisteswissenschaftliche Forschung nachnutzbar: Die statistische Auswertung der angebotenen Texte, die Zusammenstellung eigener Textkorpora sowie die Erstellung kritischer Editionen wird nicht zuletzt durch die Form des Angebots einfach und erstmalig für eine solche Menge an Texten ermöglicht. Die verfeinerte Auszeichnung aller Texte in TEI P5[126] bietet die Möglichkeit einer umfassenden maschinellen Auswertung. Neue Forschungsfragen können gestellt und durch feingranulare Recherche beantwortet werden.

[117] Vgl. Zeno.org (2018). http://www.zeno.org

[118] Vgl. TextGrid – Presseinformation (2009). https://www.uni-goettingen.de/de/3240.html?cid=3426

[119] Vgl. Die Digitale Bibliothek bei TextGrid. Arbeitsschritte. TextGrid Digitale Bibliothek (2016). https://textgrid.de/digitale-bibliothek

[120] Vgl. Die Digitale Bibliothek bei TextGrid. Lizenzierung. TextGrid Digitale Bibliothek (2016). https://textgrid.de/digitale-bibliothek

[121] Vgl. Betz (2015), S. 236.

[122] Vgl. koLibRI (2018a). https://projects.gwdg.de/projects/kolibri/repository/revisions/master/kolibri-addon-textgrid-import

[123] Vgl. Brodhun u. a. (2013). S. 14ff.

[124] Suche nach Werken in der Digitalen Bibliothek des TextGrid Repositorys. https://textgridrep.org/search?filter=format:text%2Ftg.work%2Bxml&filter=project.value%3ADigitale+Bibliothek

[125] Suche nach XML-Dateien in der Digitalen Bibliothek des TextGrid Repositorys. https://textgridrep.org/search?filter=format:text%2Fxml&filter=project.value%3ADigitale+Bibliothek

[126] Vgl. Betz (2015), S. 233ff.

Weitere 547 Texte aus den Bereichen Märchen, Geschichte, Kulturgeschichte, Kunst, Musik, Naturwissenschaften, Philosophie, Soziologie sowie Nachschlagewerke werden, sobald sie erschlossen und aufbereitet wurden, ebenfalls zugänglich sein.

Ein Beispiel einer wissenschaftlichen Untersuchung, die mit einer Auswahl der Texte der Digitalen Bibliothek von TextGrid durchgeführt wurde und noch wird, ist die Netzwerkanalyse der *dlina* Arbeitsgruppe[127]. Die interinstitutionelle Arbeitsgruppe aus Literaturwissenschaftlern und Informatikern untersuchte ein Korpus von 1.435 meist deutschsprachigen Dramen unter anderem auf das „Erkenntnisversprechen strukturanalytischer Ansätze"[128]. Von den 1.435 Dramen wurden 690 aus der Digitalen Bibliothek des TextGrid Repositorys zur Untersuchung herangezogen, viele Texte wurden für die Analyse aufbereitet.[129]

Zur Optimierung von Korpus und Workflow und somit zur Verbesserung der strukturellen Qualität der für die Untersuchung genutzten Texte wurde das Social Editing-Tool *Play(s)*[130] als mobile App entwickelt. Mit dieser wird spielerisch die geisteswissenschaftliche Community animiert, gemeinsam an der Qualität der Auszeichnung der Dramen zu arbeiten.[131]

3.2 Use Case #2 – Theodor Fontane: Notizbücher

Die Virtuelle Forschungsumgebung TextGrid ermöglicht ein kooperatives Arbeiten, das zudem ortsunabhängig ist und Dienste und Werkzeuge für die Bearbeitung von digitalen Editionen bietet. Die *genetisch-kritische und kommentierte Hybrid-Edition der Notizbücher von Theodor Fontane* wird von der Deutschen Forschungsgemeinschaft[132] (DFG) gefördert und entsteht an der Theodor-Fontane-Arbeitsstelle der Universität Göttingen und an der Niedersächsischen Staats- und Universitätsbibliothek Göttingen.[133] Theodor Fontane führte die 67 Notizbücher von 1859 bis 1880. Sie gelten als „(…) das letzte, noch unveröffentlichte größere Textkorpus des Autors"[134]. Die Notizbücher enthalten unter anderem Tagebuchaufzeichnungen, poetische Pläne, Vortragsmitschriften, Buchexzerpte, Zeichnungen und Notizen. Auch Alltägliches ist enthalten, beispielsweise Abfahrtszeiten von Zügen und To-do-Listen.[135]

Das Editionskonzept des Projekts stellt, im Gegensatz zu Einzelpublikationen, die bisher eher inhaltlich orientiert waren, das „(…) Medium Notizbuch mit seinen materialen Eigenschaften – der Papierqualität, dem Format, dem Nach- und Nebeneinander von beschrifteten und unbeschrifteten Seiten, den mehrschichtig beschrifteten Seiten"[136] und weiteren Eigenschaften mehr[137] in den Mittelpunkt. Die Inhalte der Notizbücher werden ermittelt, transkribiert, kodiert, kommentiert und veröffentlicht.

[127] Vgl. dlina Workgroup (2018). https://dlina.github.io
[128] Vgl. Fischer; Kampkaspar und Trilcke (2015), S. 2.
[129] Vgl. ebd., S. 12f.
[130] Vgl. Göbel (2018a). https://github.com/mathias-goebel/mobile-plays
[131] Vgl. Trilcke u. a. (2016), S. 5/2. https://dlina.github.io/presentations/2016-leipzig/#/5/2
[132] Vgl. DFG (2018). http://www.dfg.de
[133] Vgl. Radecke (2015), S. 39. und Theodor Fontane: Notizbücher (2018). https://fontane-nb.dariah.eu
[134] Ebd., S. 86.
[135] Vgl. Theodor Fontane: Notizbücher. (2018). https://fontane-nb.dariah.eu/content.html?id=ueber_das_projekt.md
[136] Radecke; Göbel und Söring (2013), S. 90.
[137] Vgl. ebd., S. 90ff.
[138] Quelle: https://fontane-nb.dariah.eu/edition.html?id=/xml/data/16b00.xml&page=1r

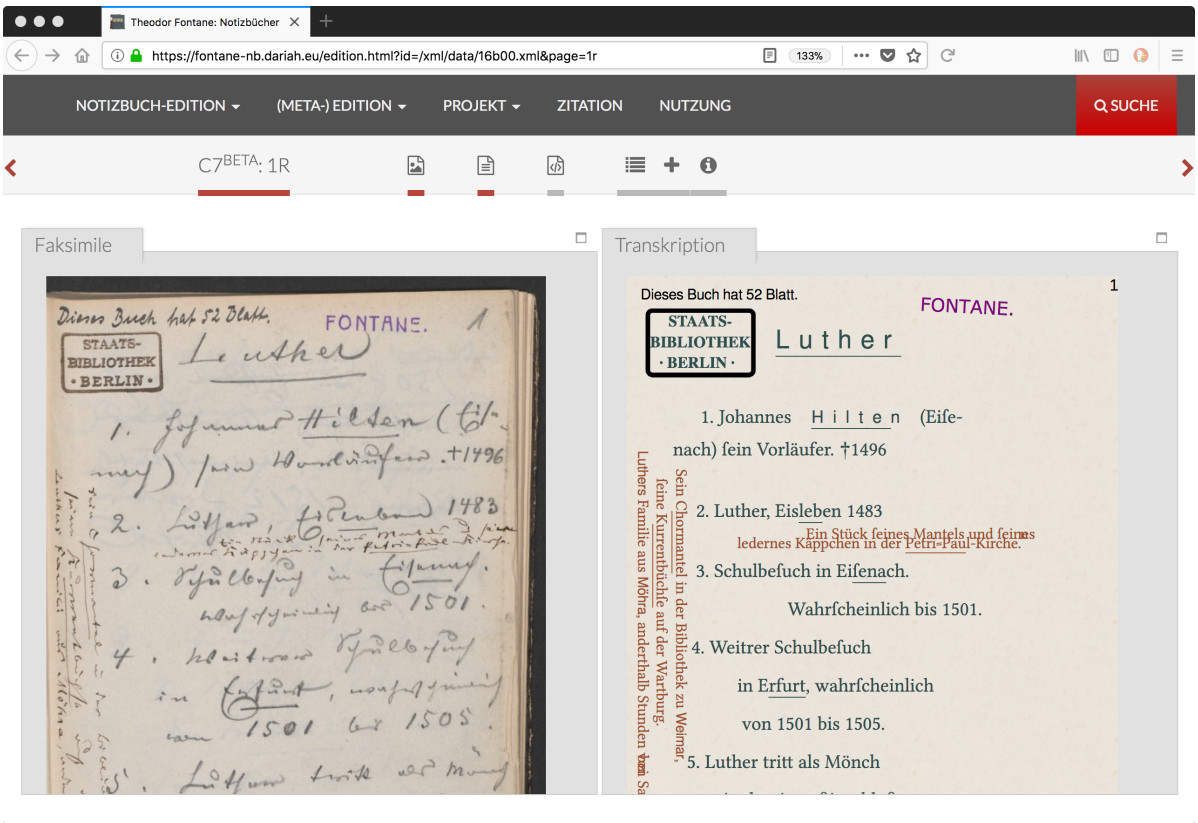

Abbildung 9: Sicht auf Faksimile und Transkription der Beta-Version von Notizbuch C7 im Fontane-Notizbuch-Portal (Screenshot)[138]

Ein Teil der Hybrid-Edition, die *digitale Edition im Fontane-Notizbuch-Portal*, veranschaulicht die Materialität der Notizbücher und bietet konfigurierbare Sichten auf das Faksimile, den historisch-kritisch edierten Text sowie die diplomatische Transkription.[139] Als Beispiel ist in Abbildung 9 die Sicht auf eine Seite von Faksimile und Transkription einer bereits veröffentlichten Beta-Version von Notizbuch C7 dargestellt. Der zweite Teil der Edition, die *Buch-Edition*, umfasst eine historisch-kritische Textfassung samt Apparat und Kommentaren.

Der projektspezifische Workflow im Fontane-Projekt ist in Abbildung 10 auf Seite 34 dargestellt. Die Virtuelle Forschungsumgebung TextGrid eignet sich in hohem Maße dazu, den gesamten Forschungsprozess zu begleiten und stellt die Infrastruktur sowohl für eine Aufbereitung des Materials im TextGridLab als auch für die Veröffentlichung der Faksimiles im TextGrid Repository zur Verfügung.[140] Der Workflow besteht in der Hauptsache aus den Arbeitsbereichen *Input*, *Prozessierung* und *Output*:

1. Bereitstellung und Erschließung des Materials (Input)
2. Aufbereitung des Materials im TextGridLab (Prozessierung)
3. Verschiedene digitale und analoge Publikationsmöglichkeiten (Output)

[139] Vgl. Theodor Fontane: Notizbücher. (2018). https://fontane-nb.dariah.eu/content.html?id=ueber_das_projekt.md
[140] Vgl. Radecke; Göbel und Söring (2013), S. 99.

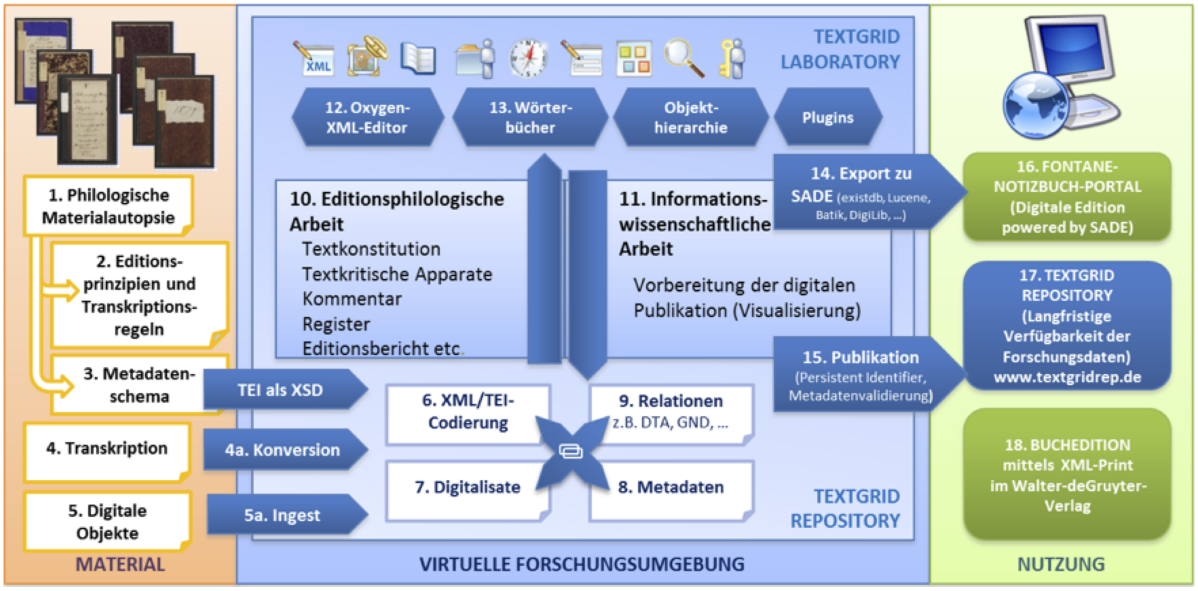

Abbildung 10: Workflow im Fontane-Notizbuch-Projekt; Grafik erstellt vom TextGrid-Team, ergänzt und bearbeitet von Gabriele Radecke, Martin de la Iglesia und Mathias Göbel[141]

Der erste Arbeitsbereich (Input) wird als Materialautopsie in Radecke (2015) detailliert beschrieben und betrifft TextGrid nicht direkt. Die Bearbeitung des Materials im TextGridLab ist im zweiten Arbeitsbereich (Prozessierung) angesiedelt, hierhin wurden die Faksimiles sowie die edierten Texte importiert und stehen zur weiteren Bearbeitung bereit. Hierzu wird ein XML-Editor sowie der Text-Bild-Link-Editor des TextGridLab genutzt, ebenso die Visualisierung einer Vorschau über eine XSL-Transformation[142], die das TextGridLab anbietet.[143] Für den dritten Arbeitsbereich (Output) steht zum einen das TextGrid Repository für die nachhaltige Speicherung der Faksimiles, von denen bereits alle 5429 publiziert wurden, bereit.[144] Zum anderen werden die TEI/XML-Dateien über eine SADE-Instanz[145] für die Visualisierung gespeichert, aufbereitet und das HTML über XQuery[146] vorprozessiert, so dass eine performante Darstellung der Faksimiles, Transkriptionen und TEI/XML-Dateien im Fontane-Notizbuch-Portal erfolgen kann.

3.3 Use Case #3 – Virtuelles Skriptorium St. Matthias

Der Bestand der mittelalterlichen Trierer Bibliothek der Benediktinerabtei St. Matthias umfasst Einträge zu 526 Kodizes mit 3.942 Inhalten[147], diese sind auf 25 Standorte verteilt. Das ebenfalls von der Deutschen Forschungsgemeinschaft (DFG) geförderte Digitalisierungsprojekt *Virtuelles Skriptorium St. Matthias* hat diese digitalisiert und der Öffentlichkeit zugänglich gemacht. Dadurch können Wissenschaftler

[141] Quelle: Radecke (2015), S. 40.
[142] Vgl. XSLT (2018). https://www.w3.org/TR/xslt
[143] Vgl. TextGridLab Nutzerhandbuch 2.0 (2015c). https://wiki.de.dariah.eu/display/TextGrid/Vorschau-Ansicht
[144] Suche nach Dateityp `image/jpeg` im Projekt Fontane Notizbücher. https://textgridrep.org/search?filter=format:image%2Fjpeg&filter=project.value%3AFontane+Notizbücher
[145] Vgl. Göbel (2018b)
[146] Vgl. XQuery (2018). https://www.w3.org/TR/xquery
[147] Vgl. Virtuelles Skriptorium St. Matthias (2018a). http://www.stmatthias.uni-trier.de/index.php?l=n&s=hilfe

unterschiedlichster Fachdisziplinen ortsungebunden und mit Hilfe digitaler Werkzeuge an dieser rekonstruierten Bibliothek forschen.[148] Es soll außerdem die Möglichkeit gegeben werden, die „Bibliothek von St. Matthias als gewachsene Institution zu begreifen"[149]. Das Projekt wurde von der Universität Trier, der Technischen Universität Darmstadt und der Stadtbibliothek Trier durchgeführt – in Zusammenarbeit mit der Bibliothek des Bischöflichen Priesterseminars Trier, dem Trier Center for Digital Humanities der Universität Trier, dem Karlsruher Institut für Technologie (KIT), DARIAH-DE und TextGrid.[150]

Mit der Gründung eines Benediktinerklosters in den Jahren zwischen 970 und 980 wurde der Grundstein für die Bibliothek des Klosters St. Matthias gelegt. Die älteste dort vorhandene Handschrift wird in das Entstehungsjahr 719 n. Chr. datiert, im Jahr 1125 wird das Skriptorium St. Matthias erstmals schriftlich erwähnt. Neben Handschriften werden im Jahr 1530 auch Frühdrucke in einer Abschrift des Bibliothekskatalogs aufgeführt. Heute sind alle nachgewiesenen Handschriften der ehemaligen Klosterbibliothek auf Standorte in der ganzen Welt verteilt, da nach der Besatzung Triers durch französische Revolutionstruppen alle Trierer Klöster aufgelöst und alle Abteibibliotheken beschlagnahmt und zum Teil veräußert wurden.[151] Die Stadtbibliothek Trier übernahm danach den größten Teil der Sammlung der Abtei St. Matthias.

Die entstandenen digitalen Daten des Virtuellen Skriptoriums St. Matthias sind an der Universität Trier – im Rahmen des Projekts DARIAH-DE[153] – und im TextGrid Repository gespeichert. So ist eine nachhaltige Aufbewahrung gewährleistet. Die Digitalisate des Virtuellen Skriptoriums werden über eine Zusammenstellung von Inhalten aus verschiedenen Handschriftenkatalogen erschlossen, sie liegen in den Dateiformaten TIFF, JPEG und PDF vor. So können sie zum einen mit den Werkzeugen des TextGridLab und zum anderen mit Methoden der eHumanities maschinell erschlossen werden. Die Metadaten zu den Inhalten des Virtuellen Skriptoriums sind in einer Datenbank gespeichert und können über XSL-Transformationen nach METS und XML/TEI exportiert werden.[154]

Für den Import der Digitalisate und der zugehörigen Metadaten in das TextGrid Repository wurde TG-import mit der Policy `dfgviewermets_import` genutzt. Damit sind alle Digitalisate samt deskriptiven und strukturellen Metadaten im TextGrid Repository zugänglich[155] Die METS-Dateien wurden ebenso eingespielt wie alle dort referenzierten Digitalisate. Die im TextGrid Repository publizierte METS-Datei kann im DFG-Viewer angezeigt werden (siehe Abbildung 11 auf Seite 36). Abbildung 12 auf Seite 37 zeigt die Sammelhandschrift im Mirador-Viewer[156], die dazu ein aus der METS-Datei generiertes

[148] Vgl. Virtuelles Skriptorium St. Matthias (2017). http://stmatthias.uni-trier.de

[149] DARIAH-DE (2018e). https://de.dariah.eu/virtuelles-skriptorium

[150] Vgl. Virtuelles Skriptorium St. Matthias (2017). http://stmatthias.uni-trier.de

[151] Vgl. Virtuelles Skriptorium St. Matthias (2018b). http://stmatthias.uni-trier.de/?l=n&s=bibliothek

[152] Quelle: http://dfg-viewer.de/show/?tx_dlf%5Bpage%5D=14&tx_dlf%5Bdouble%5D=1&tx_dlf%5Bid%5D=https://hdl.handle.net/11378/0000-0009-A235-4@data&tx_dlf%5Bpagegrid%5D=0&cHash=320378649736304e0cfa37fb8633825f

[153] Vgl. Vanscheidt; Rapp und Tonne (2012)

[154] Vgl. DARIAH-DE (2018e). https://de.dariah.eu/virtuelles-skriptorium

[155] Suche nach Editionen des Virtuellen Skriptoriums St. Matthias im TextGrid Repository (möglicherweise sind noch nicht alle Objekte final publiziert, weswegen diese Suche erst nach Anmeldung im TextGridRep und Einstellung von *Ergebnisse aus Sandbox anzeigen* alle Objekte anzeigt). https://textgridrep.org/search?filter=format:text%2Ftg.edition%2Btg.aggregation%2Bxml&filter=project.value%3AVirtuelles+Skriptorium+St.+Matthias

[156] Vgl. mirador (2018). http://projectmirador.org

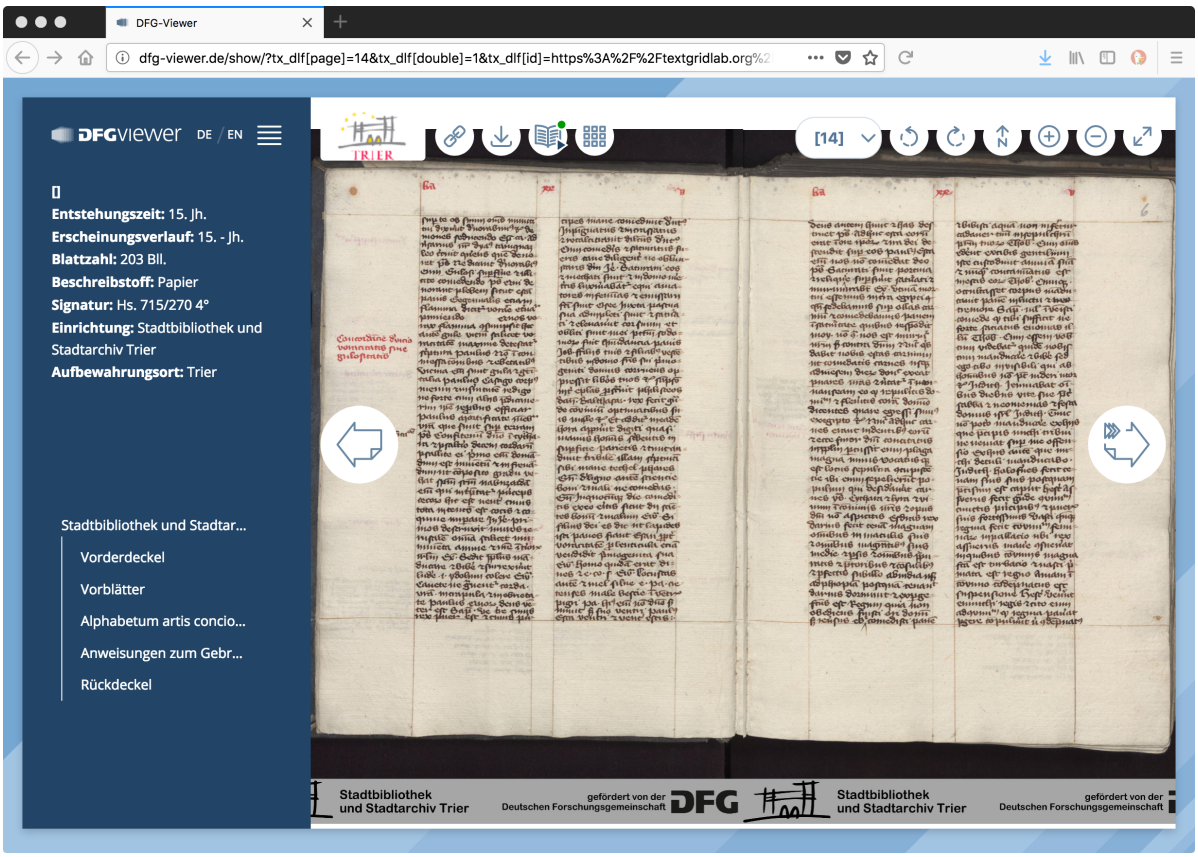

Abbildung 11: Sammelhandschrift Stadtbibliothek und Stadtarchiv Trier, Hs. 715/270 4° im DFG-Viewer (Screenshot)[152]

IIIF-Manifest[157] nutzt.[158]

3.4 Use Case #4 – Publizieren aus dem TextGrid Laboratory: Die Sandbox

Wie schon in Kapitel 2.5 auf Seite 25 vorgestellt, gibt es zwei grundsätzlich verschiedene Publikationsprozesse für das TextGrid Repository. Der erste publiziert Daten, die im TextGridLab vorliegen, direkt in das TextGrid Repository. Der zweite bearbeitet auf Seiten der Nutzerin vorliegende Daten mit Hilfe des Tools TG-import und importiert diese am Ende des Prozesses zunächst in die Sandbox des TextGrid Repository. Die Funktion dieser Sandbox wird im Folgenden erklärt und wäre für das TextGridLab eine sinnvolle Erweiterung.

Bei beiden Prozessen kann vor der tatsächlichen Publikation ein sogenannter *dryRun* vorgenommen werden, bei dem alle Prozessierungsmodule durchlaufen und alle Tests durchgeführt werden, jedoch nicht wirklich endgültige Modifikationen vorgenommen werden. So kann im Vorfeld getestet werden, ob die Daten den Anforderungen entsprechen und genutzte Dienste erreichbar sind. Funktionale Dinge wie

[157] Vgl. IIIF (2018). http://iiif.io
[158] Vgl. https://textgridlab.org/1.0/iiif/manifests/textgrid:35482.0/manifest.json
[159] Quelle: https://textgridlab.org/1.0/iiif/mirador/?json=521319

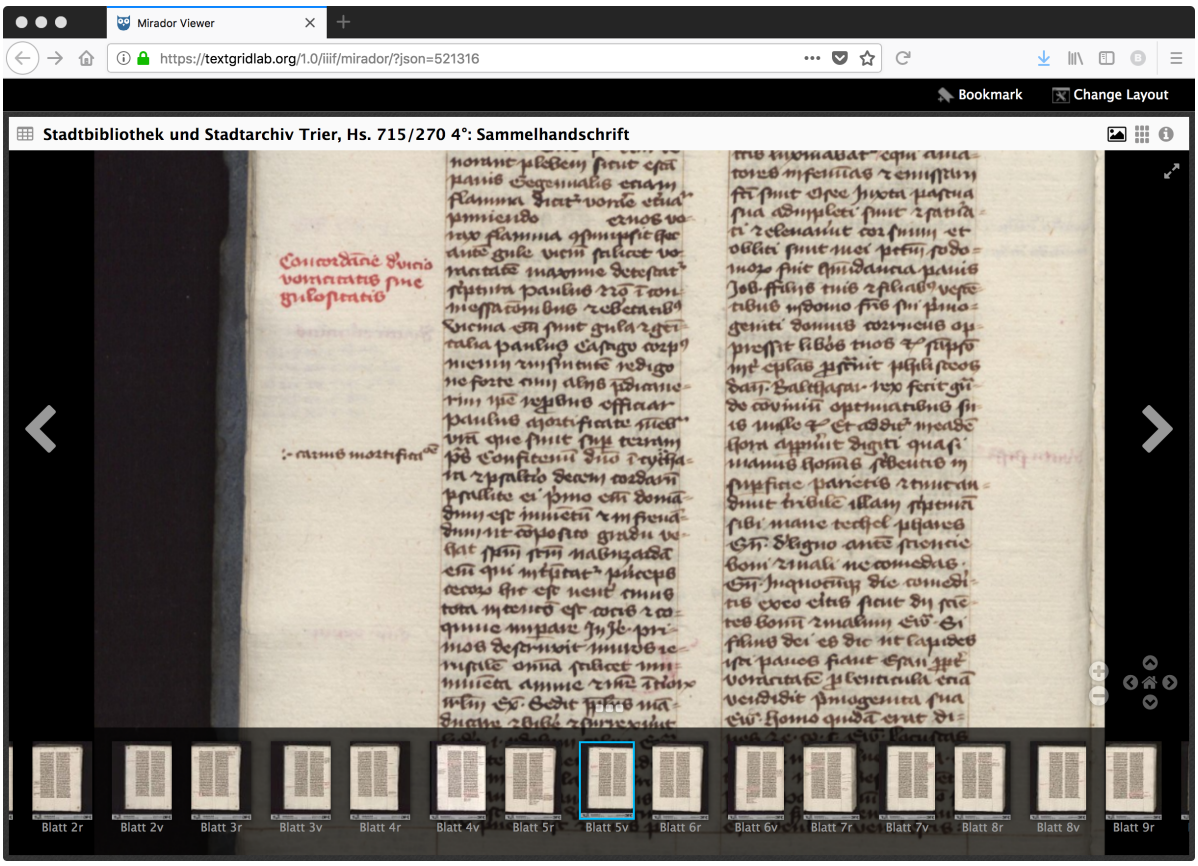

Abbildung 12: Sammelhandschrift Stadtbibliothek und Stadtarchiv Trier, Hs. 715/270 4° im Mirador-Viewer (Screenshot)[159]

beispielsweise umgeschriebene lokalen Dateipfade in TextGrid-URIs (TG-import) oder umgeschriebene TextGrid-URIs in ePIC-Handle-PIDs (TG-import und TG-publish) können nicht immer vorhergesehen werden.

Eine *Publikation aus dem TextGridLab* heraus ist aufgrund ihres Workflows nicht umkehrbar. Nach der Publikation sind die Daten öffentlich zugänglich, haben einen PID und können nicht mehr gelöscht werden. Die Index- und Relationsdaten werden in die öffentlichen Suchindizes kopiert und die Daten- und Metadatendateien in den statischen TextGrid-Storage verschoben, die Daten werden im TG-auth als `isPublic` registriert. Durch die Kopiervorgänge der Index- und Relationsdaten sind publizierte Daten noch indiziert und können im Navigator und über die Suche noch gefunden, und außerdem noch gelesen und angezeigt werden (TG-crud non-public kann auch öffentliche Daten lesen). Eine Kontrolle der publizierten Daten ist also nur *vor* der Publikation möglich, wenn sich die Daten noch nicht im finalen Zustand befinden, *nach* der Publikation ist eine Korrektur nicht mehr möglich.

Bei der *Publikation per TG-import* werden ähnliche Arbeitsschritte wie bei der Publikation aus dem TextGridLab heraus durchgeführt, nur werden die Daten auf Seiten der Nutzerin vorbereitet und dann direkt in das TextGrid Repository eingespielt. Jedes Objekt bekommt ebenfalls einen PID, alle Index- und Relationsdaten werden in die öffentlichen Suchindizes, die Daten- und Metadatendateien in den statischen TextGrid-Storage geschrieben, jedoch werden die Daten TG-auth nur als `nearlyPublished`

gemeldet. Alle publizierten Daten befinden sich nun zunächst in der sogenannten *Sandbox*. Sie haben alle Eigenschaften von publizierten Daten wie oben genannt, die Daten befinden sich in ihrem finalen Zustand. Nun können sie in der Sandbox *nach* der eigentlichen Publikation in Ruhe kontrolliert werden – auch kooperativ mit anderen Forscherinnen, die keinen Zugang zu den Daten im TextGridLab hatten. Die Daten sind öffentlich zugänglich, allerdings noch nicht über die öffentliche Suche auffindbar, beispielsweise im TextGrid Repository-Browser. Nach eingehender Prüfung können die Daten nun entweder *final publiziert* werden, TG-auth registriert die Daten als `isPublic`. Bei nicht zufriedenstellender Publikation können alle Daten auch wieder komplett aus der Sandbox *gelöscht* werden. Ein erneuter Publikationsvorgang kann – nach erfolgter Korrektur – jederzeit durchgeführt werden.

Genauso wie die Sandbox eine sinnvolle Erweiterung für das TextGridLab ist, ergeben sich aus den Use Cases ebenfalls sinnvolle und wünschenswerte Verbesserungen für die jeweiligen Szenarien. Diese Anforderungen werden im nächsten Kapitel in Kategorien zusammengefasst und analysiert.

4 Neue Anforderungen und Anforderungsanalyse

4.1 Anforderungen an elektronische Publikationen

In Schirmbacher und Müller (2009) werden Anforderungen an Publikationen als Grundlage für den Publikationsprozess bzw. den Publikationskreislauf angeführt. Davon ausgehend wird im Folgenden auf die speziellen Anforderungen an elektronische Publikationen eingegangen. Einige der genannten Aspekte greifen ineinander und können nicht immer streng voneinander getrennt werden. Weiterhin spielen alle folgenden Punkte speziell auch bei der Zertifizierung von digitalen Langzeitarchiven bzw. Repositorien eine große Rolle, um die Erfüllung der Anforderungen zu prüfen und zu dokumentieren. So kann sichergestellt werden, dass durch Anwendung von Methoden der digitalen Langzeitarchivierung und der Datenkuration die Gefahr eines Datenverlusts in zertifizierten Repositorien so gering wie möglich ist.[160]

4.1.1 Zugänglichkeit

Wissenschaftliche Publikationen sollen für die Wissenschaft zugänglich sein und das gesamte Publikationswesen allen Wissenschaftlern zur Verfügung stehen. Die wissenschaftlichen Bibliotheken sind verantwortlich für die Beschaffung, Erschließung und Bereitstellung der Publikationen.[161]

Alle diese Aspekte gelten genauso für elektronische Publikationen. An dieser Stelle kommen die institutionellen digitalen Repositorien ins Spiel: Die elektronischen Publikationen, die in einem solchen Repositorium gespeichert sind, müssen ebenso kuratiert und gepflegt werden wie die Bücher in den Regalen einer Bibliothek. Es muss sichergestellt werden, dass eine einmal veröffentlichte Publikation dauerhaft zugreifbar, d. h. online verfügbar ist und auch vorgehalten wird. Hier spielen auch Rechenzentren eine Rolle, die im Storage-Backend die Daten mehrfach und idealerweise an verschiedenen Orten replizierend vorhalten sowie die Speichermedien überwachen und aktuell halten.

Für das TextGrid Repository ergibt sich daraus die Anforderung an sichere Speicherung und dauerhafte Kuratierung der Inhalte. Diese sind bereits weitestgehend erfüllt durch die institutionelle Pflege durch DARIAH-DE und werden mit der Zertifizierung mit dem CoreTrustSeal dokumentiert und bestätigt.

4.1.2 Nachhaltigkeit

Aus dem „(…) etablierte(n) System der Bezugnahme auf bereits erschienene Publikationen mittels Zitationen“[162] ergibt sich, dass sich wissenschaftliche Publikationen aufeinander beziehen und sich gegenseitig referenzieren. Sie werden weiterhin dazu verwendet, Forschungsergebnisse zu verifizieren oder zu falsifizieren. Für eine nachhaltige Zitationspraxis ist es daher wichtig, dass publizierte Werke nachhaltig verfügbar bleiben und sich außerdem ihre Inhalte nicht ändern.[163]

[160] Vgl. Engelhardt; Funk und Veentjer (2013), S. 7ff.
[161] Vgl. Schirmbacher und Müller (2009), S. 9.
[162] Ebd.
[163] Vgl. ebd.

Auch für elektronische Publikationen gilt die Anforderung der Nachhaltigkeit, was auch spezielle Anforderungen an die Technik stellt. Eine einmal publizierte elektronische Ressource darf aus Gründen der nachhaltigen Referenzierung nicht verändert werden, was bei elektronischen Medien ungleich schwieriger zu gewährleisten ist als bei nicht elektronischen. Absichtlich oder versehentlich alle Ausgaben eines Buches zu ändern oder alle Kopien einer DVD zu manipulieren ist sicherlich sehr viel aufwendiger (und ab einem bestimmten Verbreitungsgrad auch kaum mehr möglich), als eine elektronische Publikation in Form einer Datei auf einem Server – sei es nun versehentlich oder vorsätzlich – zu löschen oder auszutauschen.

Zur Nachhaltigkeit gehört auch die Garantie bzw. die Möglichkeit der Kontrolle, dass es sich bei einer elektronischen Publikation auch genau um die zitierte handelt, und diese nicht geändert wurde. Dies kann durch die Erstellung und Dokumentation von Checksummen sichergestellt werden, die einmal bei der Ablage bzw. beim Import der zu publizierenden Datei erstellt und als Metadatum verzeichnet werden.[164] Nun kann zum einen das Repositorium regelmäßig prüfen, ob sich die Checksumme einer Datei geändert hat (Kuration) und zum anderen auch der zitierende Nutzer die Datei herunterladen, seinerseits die Checksumme berechnen und prüfen, ob seine berechnete Checksumme der vom Repositorium angegebenen entspricht.

Dieser Aspekt ist ebenfalls durch das TextGrid Repository abgedeckt. TG-crud berechnet Checksummen für jede Datei und überträgt diese in die TextGrid-Metadaten und außerdem in die Metadaten des Handle-PID. Die Checksummen werden regelmäßig geprüft, mit denen der gespeicherten Datei verglichen. Im Fall einer Abweichung muss eine Prüfung stattfinden.

4.1.3 Nachvollziehbarkeit

Der gesamte Publikationsprozess soll kontrollierbar dokumentiert sein. Hierzu gehören die Erfassung und Speicherung relevanter Metadaten sowie die Möglichkeit der eindeutigen und dauerhaften Identifizierbarkeit einer Publikation. Die Primärdaten, auf die sich eine Publikation bezieht und auf denen sie aufbaut, sollten für eine umfassende Nachvollziehbarkeit ebenfalls publiziert werden.[165]

Relevante Metadaten, beispielsweise Publikationsdatum, Autor etc. sollen sinnvollerweise mit Hilfe von standardisierten und dokumentierten Metadatenschemata dokumentiert werden. Nach Funk (2014) gehört die eindeutige und dauerhafte Identifizierbarkeit einer elektronischen Ressource ebenso zum Publikationsprozess wie die Forderung nach der Veröffentlichung der Primärdaten, auf die sich eine Publikation bezieht und auf denen sie aufbaut.[166]

Für die eindeutige und dauerhafte Identifizierbarkeit einer Publikation können *Persistente Identifikatoren* (PIDs) wie etwa DataCite-DOIs[167], ePIC-Handles[168] oder auch URNs[169] genutzt werden. Ein PID besteht aus einer eindeutigen Zeichenkette, die dauerhaft an eine elektronische Ressource gebunden wird.

[164] Vgl. Engelhardt; Funk und Veentjer (2013), S. 7ff.
[165] Vgl. Funk (2014), S. 10.
[166] Vgl. ebd.
[167] Vgl. DataCite (2018a). https://datacite.org und DataCite (2018b). https://datacite.org/dois.html
[168] Vgl. ePIC PID Consortium (2018a). http://www.pidconsortium.eu
[169] Vgl. Wikipedia (2018e). https://de.wikipedia.org/wiki/Uniform_Resource_Name

Anhand dieser Zeichenkette kann dann mit Hilfe von sogenannten Resolvern, also Diensten, die die Zuordnung von Zeichenkette und Ressource kennen und speichern, auf den Speicherort der Ressource weitergeleitet bzw. aufgelöst werden. So kann sich z. B. die URL eines Dokuments beliebig ändern – und sofern die Zuordnung angepasst wird, zeigt der PID immer noch (oder wieder) auf die gewünschte Ressource. Hinter solchen PIDs stehen institutionelle Organisationen, die die dauerhafte Auflösung der PIDs garantieren und für die Aktualität der Bindung von PID zu Speicherort Sorge tragen.

Während des Imports und der Publikation in das TextGrid Repository werden relevante Metadaten über das TextGrid-Metadatenschema dem Objekt zugeordnet und mit diesem gespeichert. Das TextGrid-Metadatenschema basiert auf standardisierten Metadatenschemata und nutzt die Elemente nach deren Vorgaben.[170] Außerdem wird während der Publikation jedem Objekt ein ePIC-Handle-PID zugewiesen. Das TextGrid Repository garantiert eine langfristige Pflege der Zugänglichkeit der Inhalte des Repositoriums über diesen PID.

Anforderung 1: Die Auszeichnung der Objekte mit DataCite-DOIs – zusätzlich zu den ePIC PIDs – ist eine sinnvolle Erweiterung. DOIs sind als Persistente Identifikatoren in der wissenschaftlichen Community sehr viel weiter verbreitet, die Unterstützung und Pflege der technischen Infrastruktur sind durch die International DOI Foundation[171] (IDF) und deren Zertifizierung mit ISO 26324[172] garantiert. Die SUB Göttingen[173] und die GWDG[174] sind beide über DataCite Mitglied bei der IDF und können so DOIs als Persistente Identifikatoren für ihre Repositorien generieren.

4.1.4 Authentizität

Eine Publikation ist dann authentisch, wenn die als Verfasser genannte Person und ihr Urheber ein und dieselbe Person sind. Für die Richtigkeit dieser Metadaten ist normalerweise der Herausgeber verantwortlich.[175]

Eine Prüfung der Authentizität kann hier auf die gleiche Weise wie für nicht-elektronische Publikationen geschehen. Für den elektronischen Publikationsprozess kann u. U. eine Prüfung automatisiert werden, weil beispielsweise die veröffentlichende Wissenschaftlerin sich während des Publikationsprozesses authentifiziert hat und so ihre Identität bestätigt wurde. Die Betreiber des Repositoriums müssen dann entweder der Wissenschaftlerin vertrauen, dass sie nur authentische Werke publiziert oder die Authentizität händisch prüfen. Weiterhin kann das Repositorium natürlich einem Vertrauensmissbrauch entgegenwirken, indem Authentizität in den Nutzungsbedingungen gefordert wird und im Falle eines Verstoßes Maßnahmen seitens des Betreibers ergriffen werden.

Beispielsweise könnte eine Publikation im Laufe des Publikationsprozesses dem Autor in der Form zugeordnet werden, dass er dem Repositorium in geeigneter Form erlaubt, die Publikation einer autorisierten

[170] Vgl. TextGrid (2017b). https://textgridlab.org/schema/textgrid-metadata_2010.xsd
[171] Vgl. IDF – International DOI Foundation (2018). https://www.doi.org
[172] Vgl. International Organization for Standardization (2017). https://www.iso.org/standard/43506.html
[173] Vgl. SUB Göttingen (2018). https://www.sub.uni-goettingen.de
[174] Vgl. GWDG (2018). https://www.gwdg.de
[175] Vgl. Schirmbacher und Müller (2009), S. 9f.

Liste seiner Publikationen hinzuzufügen, beispielsweise den über seine ORCID ID[176] verzeichneten. Um den Autor eindeutig zu bestimmen, empfiehlt es sich, Normdatenverzeichnisse wie die Gemeinsame Normdatei (GND)[177] zu nutzen und die entsprechenden Identifikatoren in den Metadaten zu verzeichnen.

Nur bei TextGrid angemeldete Wissenschaftlerinnen können Daten im TextGrid Repository publizieren. Die TextGrid Terms of Use[178] regeln die Voraussetzungen für die Nutzung der TextGrid-VFU. Die Inhalte der publizierten Daten können nicht geprüft werden, die Authentizität der Autorin ist durch die Anmeldung über die DARIAH AAI sichergestellt.

Anforderung 2: Eine weitere neue Anforderung an die Authentizität des Forschers ist die Verknüpfung des TextGrid-Accounts mit der ORCID ID des Autors. So könnte die Publikation mit ihrer Referenz (PID) als Publikation dieses Forschers automatisch bei ORCID eingetragen werden, sollte er dies wünschen.

4.1.5 Qualitätssicherung

Die Sicherung der Qualität einer Publikation findet generell nach der Annahme eines Manuskripts und vor dessen tatsächlicher Veröffentlichung statt. So soll sichergestellt werden, dass die Publikation einem gewissen Qualitätsstandard genügt.[179] Ein Verfahren zur Qualitätssicherung, das im Wissenschaftsbetrieb von großer Bedeutung ist, ist die *Peer Review*[180], bei der wissenschaftliche Texte mittels Bewertung durch unabhängige Gutachter beurteilt werden.

Mit der Qualitätssicherung verhält es sich ebenso wie mit der Authentizität: Entweder ein Repositorium erlaubt jedem Wissenschaftler beliebige Daten zu veröffentlichen, oder es gibt eine Qualitätskontrolle seitens des Repositoriums. Ist jedoch die Authentizität von Autor und Herausgeber (sollte es sich beim Publizierenden nicht um den Autor handeln) sichergestellt, kann schon die Angst vor einem möglichen Verlust der Reputation des Autors im Falle der Publikation einer qualitativ minderwertigen Arbeit dafür sorgen, dass der Autor eine möglichst große Sorgfalt beim Erstellen seiner Arbeiten walten lässt. Eine automatisierte inhaltliche Qualitätssicherung von elektronischen Publikationen ist, zumindest derzeit, kaum möglich.

Es gibt lediglich diverse Möglichkeiten, die technische Qualität zu beurteilen, zum Beispiel durch Vorgabe und Validierung des angelieferten Dateiformats. Dies ist für eine Kuratierung der Daten wichtig. Sollen alle Dateien eines obsolet werdenden Formats im Zuge einer Formatmigration in ein aktuelles Format umgewandelt werden, kann dies nur – wenn überhaupt – bei Dateien garantiert werden, sofern sie der Formatspezifikation entsprechen.

Anforderung 3: Bei der Publikation wird momentan noch nicht die Validität des Dateiformats der Dateien geprüft. Eine neue Anforderung an das TextGrid Repository und dessen Publikationsworkflow ist die Validierung des Dateiformats: Es soll festgestellt werden, welches Format eine Datei hat, welche Version eines Dateiformats vorliegt und ob die Datei nach der Formatspezifikation valide ist.

[176] Vgl. ORCID (2018). https://orcid.org
[177] Vgl. GND (2018). http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html
[178] Vgl. TextGrid Terms of Use (2016). https://textgrid.de/terms-of-use
[179] Vgl. Schirmbacher und Müller (2009), S. 10.
[180] Vgl. Wikipedia (2018f). https://de.wikipedia.org/wiki/Peer-Review

4.1.6 Bewertung

Wie die Arbeit von anderen Wissenschaftlerinnen bewertet wird, welche Auswirkung und Bedeutung sie hat und welchen Einfluss die Arbeit auf die Reputation des Autors hat, stellt sich erst nach der Veröffentlichung heraus.[181]

Eine Bewertung durch andere Wissenschaftler und der Einfluss der Arbeit auf die Reputation der Autorin kann bei elektronischen Publikationen z. B. anhand von Download-Statistiken ermittelt werden.[182] Auch mit Hilfe von Zitationsdatenbanken kann eine Bewertung – inzwischen auch automatisiert – stattfinden.[183]

Anforderung 4: Metriken und Download-Statistiken können bisher vom TextGrid Repository nicht automatisiert geliefert werden. Es sollte eine automatisierte Extraktion von Metriken auf Anforderung entweder der Nutzerinnen oder der Administratoren möglich sein.

4.1.7 Geschwindigkeit

Der Zeitraum, in dem eine Publikation den Publikationsprozess durchläuft, sollte aus wissenschaftlicher Sicht möglichst kurz sein. Oft steht dem jedoch die Dauer der verschiedenen Schritte des Publikationsworkflows entgegen, beispielsweise der Qualitätssicherung oder diverser technischer Arbeitsschritte.[184]

Die Dauer des Publikationsprozesses kann bei elektronischen Publikationen sehr kurz sein, sofern es sich um das Importieren oder Einspielen von Publikationen in ein Repositorium handelt. Sind im Publikationsprozess keine Arbeitsschritte enthalten, die eine inhaltliche Prüfung durch Mitarbeiterinnen bzw. händische Freischaltung beinhalten, können alle weiteren Arbeitsschritte automatisiert ablaufen und die Publikation kann praktisch jederzeit und sofort erfolgen. Persistente Identifikatoren können ebenso generiert und vergeben werden, so dass nach der Online-Publikation eine sofortige Zitation der Arbeit sowie der sofortige Zugriff darauf möglich ist.

Anforderung 5: Als sehr allgemeine und programmatische Anforderung kann die Erhöhung der zeitliche Performanz des Publikationsworkflows des TextGrid Repositorys gefordert werden, die mit hoher Wahrscheinlichkeit verbessert werden kann.

4.1.8 Vollständigkeit

Die Vollständigkeit einer Publikation bezieht sich nicht direkt auf das Werk selbst, sondern auf die Vollständigkeit der Metadaten desselben. In Schirmbacher und Müller (2009)[185] wird hauptsächlich unterschieden zwischen beschreibenden Metadaten, verwaltungstechnischen Metadaten, strukturellen Metadaten, technischen Metadaten und Archivierungsmetadaten.

[181] Vgl. Schirmbacher und Müller (2009), S. 10.
[182] Vgl. Funk (2014), S. 10.
[183] Vgl. Wikipedia (2018g). https://de.wikipedia.org/wiki/Zitationsdatenbank
[184] Vgl. Schirmbacher und Müller (2009), S. 10.
[185] Vgl. ebd.

Auch das trifft direkt auf elektronische Publikationen zu. Laut Funk (2014) sind gerade im elektronischen Bereich Metadaten für die formale Vollständigkeit und zur inhaltlichen Beschreibung einer Publikation unerlässlich, insbesondere, um die Vorteile elektronischer Publikationen nutzen zu können, wie in Stäcker (2013)[186] angeführt.

Die beschreibenden *deskriptiven Metadaten* sind inhaltlicher Natur, sie werden im Allgemeinen von der Autorin vergeben, da sie ihre Arbeit am besten beschreiben kann. Dies trifft sicherlich auch auf die *strukturellen Metadaten* zu, die die Struktur der Arbeit sowie Verknüpfungen zu anderen Arbeiten beschreiben. Die verwaltungstechnischen *administrativen Metadaten* werden meist vom Repositorium erzeugt. *Technische Metadaten* können aus der Arbeit selbst, bzw. der Datei extrahiert werden, samt Dateiformat, Formatversion und Validität. Archivierungsmetadaten oder auch *Herkunftsmetadaten* können entweder zum Zeitpunkt der Publikation bereits existieren und mit der Arbeit mitgeliefert werden oder auch erst beim Publikationsprozess erzeugt oder ergänzt werden.

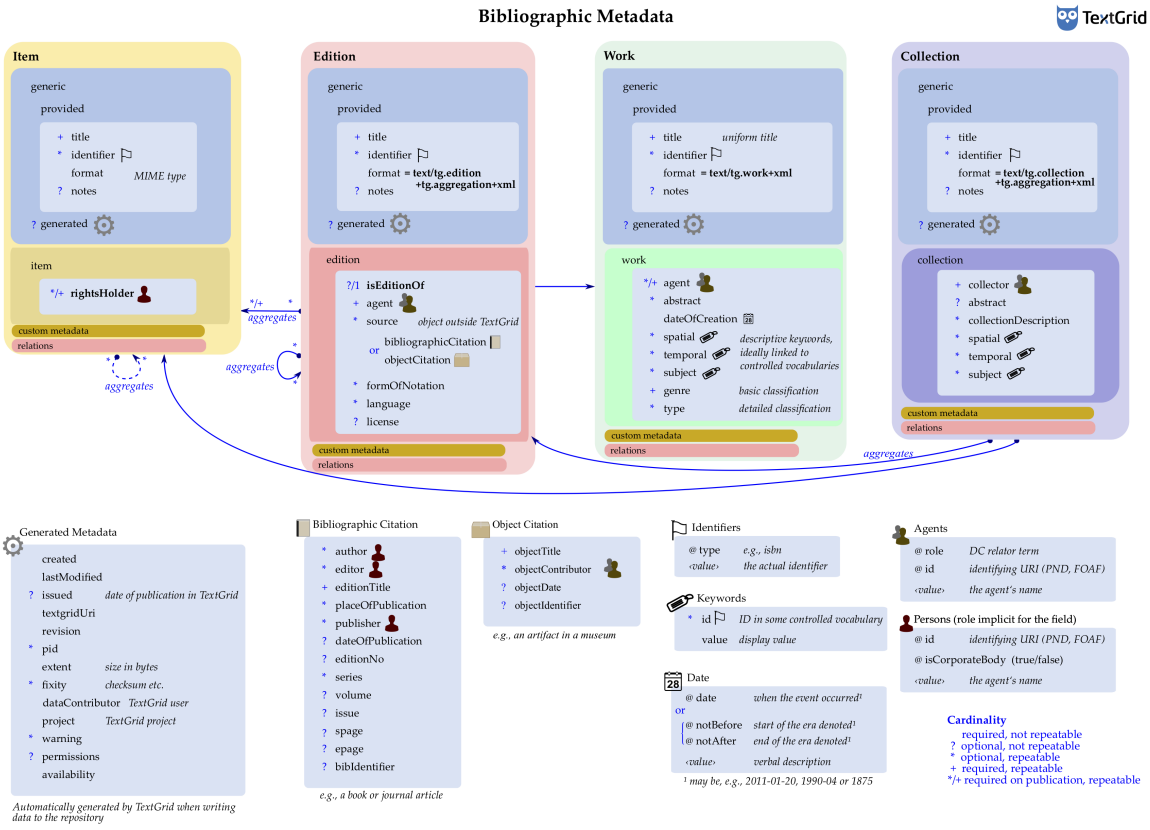

Abbildung 13: Visualisierung des TextGrid-Metadatenschemas[187]

Das TextGrid-Metadatenschema erwartet für zu publizierende Editionen oder Kollektionen bereits einen Satz von zwingend erforderlichen Metadaten. Ein Publikationsvorgang kann nur erfolgen, wenn diese angegeben werden. Aus Abbildung 13 wird ersichtlich, welche Metadaten für TextGrid-Objekte, die für die Publikation relevant sind – *Edition*, *Collection*, *Work* und *Item* –, erwartet werden.

[186] Vgl. Stäcker (2013), 29ff.

[187] Quelle: https://wiki.de.dariah.eu/download/attachments/12189756/Metadata-Cheatsheet.pdf?api=v2

4.2 Anforderungen an digitale Forschungsdaten

Im Kontext von DARIAH-DE sollten digitale Forschungsdaten einige minimale Anforderungen erfüllen, um DARIAH-DE Dienste sinnvoll nutzen zu können, zu denen auch das TextGrid Repository gehört. Dies sind unter anderem[188] (viele der Anforderungen wurden bereits in Kapitel 4.1 zu elektronischen Publikationen beschrieben):

- Validität
- Vertrauenswürdigkeit und Dokumentation des Entstehungs- und Erhebungskontextes
- Maschinenlesbarkeit und somit Prozessierbarkeit
- Referenzierbarkeit mit Angaben der Urheberschaft und zu rechtlichen Informationen hinsichtlich ihrer weiteren Verwendung durch Dritte

Diese Punkte können auch generell als Anforderungen für Forschungsdaten angenommen werden, die für die Nachnutzung oder auch den Nachweis im Rahmen einer Forschungsarbeit veröffentlicht werden sollen.

Alle diese Anforderungen wurden bereits im vorigen Kapitel adressiert, mit Ausnahme der Angaben zur Urheberschaft. Während des Publikationsprozesses des TextGrid Repositorys werden erforderliche Metadatenangaben geprüft. Für jede Datei – also für jedes TextGrid-Objekt – muss ein Metadatenfeld `rightsOwner` ausgefüllt und für jede Edition muss eine Lizenzangabe vorhanden sein, die die Nachnutzung sowie die Urheberschaft regelt.

4.3 Anforderungen an den TextGrid-Publikationsworkflow

Die Anforderungen, die bereits in Funk (2014) als Verbesserungen für den TextGrid Publikationsworkflow qualifiziert wurden, sollen hier noch einmal untersucht werden. Diese wurden gleichfalls in die Kategorien der grundlegenden Anforderungen an Publikationen als Grundlage für den Publikationsprozess von Schirmbacher und Müller (2009) aufgeteilt.

Anforderung 6: Zur Verbesserung der *Zugänglichkeit* könnten die TextGrid-Publikationen noch automatisiert in den Bibliothekskatalog aufgenommen werden.[189]

Anforderung 7: Die *Nachhaltigkeit* kann verbessert werden, wenn schon bei der Publikation der Daten technische Metadaten extrahiert werden[190], beispielsweise mit dem *File Information Toolset*[191] (FITS). Diese Metadaten helfen, zukünftig Formatmigrationen zu planen und durchzuführen, wenn unter Umständen auch erst langfristig, vor allem, wenn man die Definition von *Long Term* im Zusammenhang mit der digitalen Langzeitarchivierung sieht, wie sie im OAIS[192] definiert wird:

[188] Vgl. DARIAH-DE (2018a). https://de.dariah.eu/weiterfuhrende-informationen
[189] Funk (2014), S. 20.
[190] Vgl. ebd., S. 21.
[191] Vgl. FITS (2018). https://projects.iq.harvard.edu/fits
[192] Vgl. The Consultative Committee for Space Data Systems (2012). https://public.ccsds.org/pubs/650x0m2.pdf

> „Long Term is long enough to be concerned with the impacts of changing technologies, including support for new media and data formats, or with a changing user community. Long Term may extend indefinitely."[193]

Sinnvollerweise werden auch die technischen Metadaten aller bereits publizierten Daten (nachträglich) extrahiert und die Metadaten komplettiert. Nur so kann später eine Migration aller Daten durchgeführt werden.

Anforderung 8: Im Rahmen der *Qualitätssicherung* könnte die Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal (CTS) – ehemals Dataseal of Approval (DSA) – gefordert werden. Eine Beantragung der Zertifizierung ist bereits im Rahmen des Projektes DARIAH-DE in Vorbereitung und soll noch bis Fertigstellung dieser Arbeit abgeschlossen sein.[194]

Anforderung 9: Alle Module des TextGrid-Publikationsworkflows (TG-import und TG-publish) werden auf ihre *Robustheit und Fehlerunanfälligkeit* untersucht. Es soll sichergestellt sein, dass aufgrund von temporären Fehlern abgebrochene Publikationsvorgänge fortgesetzt werden können.[195]

Anforderung 10: Zur Verbesserung der *Intuitivität und Verständlichkeit* – obwohl bereits zufriedenstellend implementiert – könnten sowohl die Publish-GUI des TextGrid Lab als auch die Client-Server-Kommunikation des Publikationsprozesses von TextGridLab und TG-publish verbessert werden.[196]

4.4 Anforderungen an die Module der kopal Library for Retrieval and Ingest

Ebenfalls aus Funk (2014) sind einige Anforderungen an die koLibRI-Module von TG-publish und TG-import entnommen. Die kopal Library for Retrieval and Ingest wurde ursprünglich im Projekt kopal[197] entwickelt. In TextGrid wurden bisher neben Verbesserungen der Basisklassen hauptsächlich die Module TG-publish und TG-import implementiert[198].

Anforderung 11: Die Zusammenführung von TG-publish und TG-import-Modulen ist ein wichtiger Aspekt für die Pflege und Wartbarkeit der Import- und Publish-Komponenten, da beide ähnliche Module mit teilweise denselben Algorithmen verwenden.[199]

Anforderung 12: Beim Importieren von Daten mittels TG-import wird das Mapping von lokalen Dateipfaden zu TextGrid-URIs und das Mapping von TextGrid-URIs zu Handle-PIDs jeweils in eine Datei geschrieben. Das Format dieser Mappings sollte dem Format der Mapping-Dateien angepasst werden, die das TextGridLab beim Importieren schreibt.[200]

---

[193] The Consultative Committee for Space Data Systems (2012), S. 1-1.
[194] Vgl. Funk (2014), S. 21.
[195] Vgl. ebd., S. 22.
[196] Vgl. ebd., S. 23.
[197] Vgl. kopal (2018). http://kopal.langzeitarchivierung.de
[198] TG-publish umfasst die Module *textgrid-publish-api*, *kolibri-publish-service*, *kolibri-tgpublish-client* und *kolibri-tgpublish-service*. TG-import ist mit dem Modul *kolibri-addon-textgrid-import* implementiert.
[199] Vgl. Funk (2014), S. 22.
[200] Vgl. ebd.

Anforderung 13: Das Modul zur Anforderung der Handle-PIDs muss noch an die Version 2 des PID-Services der GWDG angepasst werden. Das betrifft den TextGrid PID-Wrapper-Service TG-pid, der direkt auf den PID-Service der GWDG zugreift sowie auf die TG-import- und TG-publish-Module, die von TG-pid bedient werden.[201]

Anforderung 14: Eine prinzipiell nur kosmetische Operation ist das Verschieben der TextGrid koLibRI-Module vom Namespace *de.langzeitarchivierung.kolibri* in den TextGrid-Namespace *info.textgrid.middleware.kolibri*.[202]

4.5 Anforderungen aus den Use Cases

4.5.1 Use Case #1 – Die Digitale Bibliothek bei TextGrid

Neben der Publikation weiterer Teile der Digitalen Bibliothek von TextGrid soll es auch eine Aktualisierung der Texte des Literaturordners geben. Hierfür soll die Möglichkeit der Revisionierung von Objekten in TextGrid genutzt werden. Eine Revision[203] eines TextGrid-Objekts ist prinzipiell eine nachfolgende Version einer Datei. Eine TextGrid Base-URI (`textgrid:164pv`) verweist immer auf die letzte existierende Revision. Ein Merkmale einer neuen Revision ist die aus der vorgehenden Version erzeugte TextGrid-URI (`textgrid:164pv.1` folgt beispielsweise auf `textgrid:164pv.0` – die Ziffern nach dem Punkt der URI werden einfach hochgezählt). Ein Beispiel findet sich im TextGrid Repository im Projekt der Fontane Notizbücher unter http://textgridrep.org/textgrid:164pv.1.

Anforderung 15: Eine Erzeugung von Revisionen direkt per TG-import (koLibRI) im TextGridRep soll möglichst mit der Aktualisierung des Literaturordners der Digitalen Bibliothek erfolgen. Es muss hier geprüft werden, inwieweit eine solche Revisionierung bereits im betroffenen Modul *kolibri-addon-textgrid-import* implementiert ist. Falls dies (noch) nicht wie gewünscht möglich ist, ist eine Implementierung anzuraten.

Anforderung 16: Es ist nicht nur der Import wichtig für eine erfolgreiche Revisionierung von Objekten im TextGrid Repository, sondern auch eine adäquate Darstellung derselben auf `textgridrep.org`. Momentan werden dort alle existierenden Revisionen eines Objekts angezeigt, es soll jedoch idealerweise nur die letzte Revision angezeigt werden. Weiterhin muss auch die Suche nach vorgehenden Revisionen möglich sein. Weiterhin ist es wünschenswert, eine Möglichkeit zum Vergleich von verschiedenen Revisionen zur Verfügung zu haben, zum Beispiel mit der CollateX[204] Service-Instanz von DARIAH-DE[205].

Anforderung 17: Die Lemmata von Wörterbüchern, die als einzelne Dateien importiert wurden, sollen einzeln ausgeliefert werden können. Hierzu muss die Index-Datenbank insoweit angepasst werden, dass einzelne Lemmata aus dem eigentlichen Objekt – der gespeicherten Datei – extrahiert und einzeln an die Klienten ausgegeben werden können.

---

[201] Vgl. ebd.
[202] Vgl. ebd.
[203] Vgl. TextGrid Wiki (2018). https://wiki.de.dariah.eu/display/TextGrid/Using+Revisions
[204] Vgl. The Interedition Development Group (2017). https://collatex.net
[205] Vgl. DARIAH-DE (2018f). http://collatex.dariah.eu/collatex/

4.5.2 Use Case #2 – Theodor Fontane: Notizbücher

Die Faksimiles der Fontane Notizbücher wurden zunächst zur Arbeit mit dem Text-Bild-Link-Editor im TextGridLab verfügbar gemacht und später über die TG-publish-API publiziert. Bei einigen Bildern wurde nach der Publikation festgestellt, dass die Bild-Dateien aktualisiert werden sollen, da die publizierten Versionen fehlerhaft waren. Da publizierte Daten jedoch nicht mehr änderbar sind, bleibt die Alternative, eine neue Revision der Faksimiles zu publizieren und diese über die Suche mittels TG-search adäquat anzuzeigen, denn es soll per Grundeinstellung immer nur die neueste Revision eines Objekts gefunden werden, alle älteren Revisionen nur mit optionalem Suchfilter (siehe Anforderung 16).

Anforderung 18: Da aus dem TextGridLab heraus nur Kollektionen und Editionen publiziert werden können – und nicht einzelne Dateien –, muss ein Weg gefunden werden, wie die Faksimiles, oder generell einzelne Dateien, aktualisiert und publiziert werden können. Der Workflow sollte so einfach und verständlich wie möglich und intuitiv aus dem TextGridLab nutzbar sein.

4.5.3 Use Case #3 – Virtuelles Skriptorium St. Matthias

Nach Bearbeitung der ersten ca. 300 Handschriften mit jeweils einigen hundert Faksimiles und der Fertigstellung ihrer METS-Dateien wurden diese über TG-import mit der Policy `dfgviewermets_import` publiziert. Bei diesem Import wurden einige Schwachstellen des DFG-Viewer-METS-Imports deutlich, die hier als Anforderungen formuliert werden. Da mittlerweile alle Handschriften komplett ediert und die METS-Dateien finalisiert worden sind – und es diverse Anpassungen seit dem ersten Import gab –, wurde beschlossen, alle Publikationen des ersten Imports wieder zu löschen, um die neuen Versionen komplett erneut einzuspielen, denn die zunächst in die Sandbox publizierten Handschriften waren noch nicht final publiziert worden. Dabei wurde festgestellt, dass das Löschen mit der Policy `delete_import` durchaus komfortabler konfigurierbar sein könnte.

Anforderung 19: Die Workflow-Konfiguration für die Import-, Publish- und Delete-Funktionen sollen komfortabler gestaltet werden. Bisher ist eine Konfiguration beispielsweise der zu löschenden Objekte nur über die beim Import angelegten Mapping-Dateien möglich. Hat man diese Datei vom Import nicht aufbewahrt, ist das Löschen oder ein finales Publizieren aus der Sandbox schwer möglich. Es wäre eine erhebliche Vereinfachung, könnte einfach über die `ProjectID` ein TextGrid-Projekt oder über eine TextGrid-URI eine Kollektion samt aller dort enthaltenen Objekte adressiert werden.

Anforderung 20: Die Generierung der Handle-PIDs dauert sehr lange, momentan ca. 5 Sekunden pro PID. Eine Optimierung der Performanz für die PID-Erstellung wäre wünschenswert.

Anforderung 21: Die METS-Dateien sollen vor dem Importieren auf XML-Validität geprüft werden, dazu muss die METS-Bibliothek von TG-import (bzw. koLibRI) von Version 1.4 auf Version 1.7 aktualisiert und eine Validierung implementiert werden.

Anforderung 22: Auch die Metadaten der einzuspielenden Objekte sollen wie bei TG-publish vor dem Import auf verpflichtende Metadatenfelder geprüft werden. Eine serverseitige Prüfung ist hier sinnvoll, so dass der Nutzer von TG-import nicht einfach diesen Arbeitsschritt des Imports aus der Policy entfernt (und damit das Modul praktisch aus dem Workflow nimmt).

4.5.4 Use Case #4 – Publizieren aus dem TextGrid Laboratory: Die Sandbox

Die Sandbox ist eine im Rahmen der Anforderungsanalyse gewünschte Erweiterung für das TextGridLab. Nach dem *dryRun* können die zur Veröffentlichung vorgesehen Daten publiziert werden und ausführlich vor dem Indizieren gemeinsam mit anderen Wissenschaftlerinnen geprüft werden. Nach der Kontrolle werden die Daten dann endgültig publiziert oder wieder gelöscht.

Anforderung 23: Die Publish-GUI des TextGridLab sowie der betroffene Dienst TG-publish sollen für eine Nutzung der Sandbox angepasst werden. Dabei ist besonderes Augenmerk auf die Kommunikation von Klient (TG-lab) und Server (TG-publish) sowie die nutzerfreundliche und intuitive Bedienung der GUI zu richten.

4.6 Analyse

Aus den oben behandelten Anforderungen an elektronische Publikationen, digitale Forschungsdaten, den TextGrid-Publikationsworkflow und die koLibRI-Module sowie aus den vier behandelten Use Cases wurden 23 Aspekte als Anforderungen identifiziert. Diese sollen den Publikationsworkflow des TextGrid Repositorys bzw. die den Workflow ausführende Software erweitern und verbessern. Diese Anforderungen können in die vier Kategorien *Workflow*, *Zertifizierung & Data Curation*, *Software* und *Visualisierung* eingeteilt werden:

Workflow Der Kategorie Workflow werden im Folgenden diejenigen Anforderungen zugewiesen, für die ein vergleichsweise großer Eingriff in den Publikationsworkflow nötig ist bzw. neue Module für die Import- und Publish-Komponenten des TextGrid Repositorys implementiert werden müssen und die keinen Einfluss auf bereits publizierte Daten haben.

Zertifizierung & Data Curation Dieser Kategorie werden die Anforderungen im Zusammenhang mit der Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal zugeordnet sowie die Anforderungen, die der Kuration der Inhalte des TextGrid Repositorys dienen und unter Umständen auch für die Objekte durchgeführt werden müssen, die bereits publiziert wurden.

Software Unter der Kategorie Software werden alle Anforderungen zusammengefasst, die direkt die Softwarekomponenten des Publikationsworkflows betreffen und das Nutzungserlebnis (User Experience)[206] eher nicht beeinflussen. Dies betrifft sowohl allgemeine Verbesserungen der Softwarequalität als auch Konfigurationsoptionen und interne Aspekte.

Visualisierung schließlich umfasst alle Anforderungen, die die Darstellung der Daten im TextGrid Repository adressieren.

In der folgenden Tabelle werden die vier Kategorien und die ihnen zugeordneten Anforderungen zusammengeführt, auf deren Basis im nachfolgenden Kapitel ein Konzept erstellt wird, in dem Lösungsvorschläge für die identifizierten Anforderungen erarbeitet werden.

[206] Vgl. Wikipedia (2018h). https://de.wikipedia.org/wiki/User_Experience

Nr.	Beschreibung
	Workflow
2	Verknüpfung des TextGrid-Accounts mit der ORCID ID der Autorin und automatische Eintragung der Publikation bei ORCID.
3	Validierung des Dateiformats aller eingespielten Dateien vor der Publikation.
6	Aufnehmen der TextGrid-Publikationen in den Bibliothekskatalog.
10	Verbesserung der Client-Server-Kommunikation des Publikationsprozesses von TextGridLab und TG-publish.
15	Publikation von Revisionen direkt per TG-import im TextGridRep.
18	Ermöglichen der Revisionierung einzelner publizierter Objekte im TextGridLab.
21	Validieren der METS-Dateien auf XML-Validität (mindestens METS Version 1.7) vor dem Import bzw. Publizieren.
22	Validieren der verpflichtenden Metadaten aller einzuspielenden Objekte vor der Publikation.
23	Die Publish-GUI des TextGridLab sowie der betroffene Dienst TG-publish sollen für eine Nutzung der Sandbox angepasst werden.
	Zertifizierung & Data Curation
1	Auszeichnung aller TextGrid-Objekte mit DataCite-DOIs.
4	Automatisierte Extraktion von Metriken auf Anforderung der Nutzer oder der Administratorinnen.
7	Extraktion von technischen Metadaten während der Publikation.
8	Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal.
	Software
5	Erhöhung der Performanz des Publikationsworkflows des TextGrid Repositorys.
9	Robustheit und Fehlerunanfälligkeit der Module des TextGrid-Publikationsworkflows (TG-import und TG-publish).
11	Zusammenführen von TG-publish und TG-import-Modulen.
12	Anpassen des Formats der TG-import Mapping-Dateien an das Format des TextGridLab.
13	Anpassung des Moduls zur Anforderung der Handle-PIDs an die Version 2 des PID-Services der GWDG.
14	Verschieben der TextGrid koLibRI-Module vom Namespace *de.langzeitarchivierung.kolibri* in den TextGrid-Namespace *info.textgrid.middleware.kolibri*.
19	Workflow-Konfiguration für die Import-, Publish- und Delete-Funktionen komfortabler gestalten.
20	Optimierung der Performanz der Handle-PID-Erstellung.
	Visualisierung
16	Adäquate Darstellung von Revisionen im TextGrid Repository (textgridrep.org bzw. TG-search).
17	Darstellung von einzelnen Lemmata aus Wörterbüchern, die als einzelne Dateien importiert wurden, Anpassung der Index-Datenbank.

5 Konzept

Aufgrund der Natur der identifizierten Anforderungen sowie des modularen Aufbaus der Komponenten des Publikationsworkflows können nahezu alle Anforderungen für sich betrachtet und einzeln umgesetzt werden. Es bestehen nur sehr wenige Abhängigkeiten zueinander, so dass nur im Einzelfall auf diese eingegangen wird. Das Konzept für die Implementierung wird aus diesem Grund für jede Anforderung einzeln beschrieben. Bevorzugt für die Implementierung ausgewählt werden in der Regel die Anforderungen aus den Use Cases sowie diejenigen, die im Zuge des Zertifizierungsprozesses mit dem CoreTrustSeal implementiert werden sollen.

5.1 Workflow

Anforderung 2 Verknüpfung des TextGrid-Accounts mit der ORCID ID der Autorin und automatische Eintragung der Publikation bei ORCID

Betroffene Dienste/Module:

- TG-publish#`AddPublicationToORCID` (neu)
- TG-import#`AddPublicationToORCID` (neu)

Um Arbeiten während des Publikationsprozesses als Publikation dem ORCID-Account einer Nutzerin hinzuzufügen, muss sich das TextGrid Repository als Mitglied bei ORCID anmelden und einen speziellen *Workflow für Repository Systems*[207] integrieren. Diese Workflow-Erweiterung wird in einem neuen Modul der TG-publish- und TG-import-Dienste angesiedelt und an geeigneter Stelle in den existierenden Publikations-Workflow integriert. Da die Kontrolle über die Anmeldung am ORCID-Service aus Sicherheitsgründen serverseitig implementiert werden muss (und nicht von TG-import auf dem Rechner des Nutzers ausgeführt werden darf), kann die Anmeldung am ORCID-Service sowie die Eintragung selbst serverseitig durchgeführt und nach Nutzer-Authentifizierung (SessionID) eine solche Service-Methode aufgerufen werden.

Anforderung 3 Validierung des Dateiformats aller eingespielten Dateien vor der Publikation

Betroffene Dienste/Module:

- TG-crud#`CREATE`
- TG-crud#`MOVEPUBLIC`

Das Untersuchen von Dateien hinsichtlich Formatspezifikation (Version und Validität) kann während des Publikationsvorgangs in das TextGrid Repository geschehen. Sinnvollerweise wird die Validität einer Datei direkt nach ihrem Upload geprüft, so dass beim Scheitern der Erkennung bzw. Validierung ein Fehler gemeldet werden kann[208]. Es ist sinnvoll, diese Anforderung in der `CREATE`- bzw. `MOVEPUBLIC`-Methode

[207] Vgl. ORCID Member Support Center (2018). https://members.orcid.org/api/workflow/repository

[208] ...sofern die Korrektheit des Dateiformats als eine zwingend zu erfüllende Anforderung angesehen wird. Diese Funktion wird am besten konfigurierbar implementiert.

des TG-crud zu implementieren, damit sowohl TG-publish als auch TG-import diese Funktionalität nutzen können.

Anforderung 6 Aufnehmen der TextGrid-Publikationen in den Bibliothekskatalog

Betroffene Dienste/Module:

- TG-publish#`AddMetadataToOPAC` (neu)
- TG-import#`AddMetadataToOPAC` (neu)

Zunächst muss geklärt werden, welche Voraussetzungen für einen Import in ein Bibliothekssystem seitens der Bibliothek vorliegen. Abgesehen von bürokratischen und inhaltlichen Aspekten müssen auch technische Fragen geklärt werden. Sicherlich müssen erforderliche Metadaten für einen solchen Import vorliegen und es muss geklärt werden, ob diese seitens des TextGrid Repositorys vorliegen. Wie bei Anforderung 2 muss dieses Modul teilweise serverseitig implementiert werden.

Anforderung 10 Verbesserung der Client-Server-Kommunikation des Publikationsprozesses von TextGridLab und TG-publish

Betroffene Dienste/Module:

- TextGridLab (Publish-GUI)
- TG-publish#`publish()`
- TG-publish#`getStatus()`

Für diese allgemeine Anforderung muss zunächst untersucht werden, wieso und unter welchen Umständen die Client-Server-Kommunikation zwischen der Publish-GUI des TextGridLab und dem TG-publish-Service nicht korrekt funktioniert. Die Methoden des TG-publish-Dienstes[209] sind dokumentiert und scheinen serverseitig korrekt zu funktionieren. Zunächst sollten die JUnit-Tests des Services auf Korrektheit geprüft werden, danach kann das Publish-Modul des TextGridLab untersucht werden.

Anforderung 15 Publikation von Revisionen direkt per TG-import im TextGridRep

Betroffene Dienste/Module:

- TG-import#`SubmitFiles`

Der Dienst TG-crud kann per API-Methode `CREATE` Revisionen anlegen, dies ist im TextGridLab schon möglich und wird von vielen Projekten für die kooperative Arbeit genutzt. Prinzipiell kann die Methode `CREATE` mit den Flag `createRevision=true` auch von TG-import aufgerufen werden, um neue Revisionen anzulegen. Dies muss getestet werden und evtl. muss der Workflow von TG-import angepasst oder erweitert werden, damit ein Erzeugen von Revisionen auch per TG-import möglich ist.

[209] Vgl. TextGrid Publish (2017). http://textgridlab.org/doc/services/submodules/kolibri/kolibri-tgpublish-service/docs/index.html

Anforderung 18 Ermöglichen der Revisionierung einzelner publizierter Objekte im TextGridLab

Betroffene Dienste/Module:

- TextGridLab (TG-lab GUI)
- TG-crud#`UPDATE`

Eine Revisionierung von nicht publizierten Objekten ist im TextGridLab möglich (siehe Anforderung 18). Eine Revisionierung von bereits publizierten Objekten zieht – nach Publikation der einzelnen Datei – lediglich eine Anpassung der übergeordneten Strukturen (Edition, Kollektion, Aggregation) nach sich. Nach Rechtsklick auf ein publiziertes Objekt und Erzeugen der neuen Revision müssen lediglich die (neuen) TextGrid-URIs des revisionierten Objekts in die übergeordneten Objekte eingetragen werden. Dafür sind weder im TextGridLab noch bei TG-crud die entsprechenden Methoden implementiert. Denkbar wäre ein TG-crud#`UPDATE` mit Flag `createRevision=true`, das übergeordnete Objekte automatisch mit aktualisiert.

Anforderung 21 Validieren der METS-Dateien auf XML-Validität (mindestens METS Version 1.7) vor dem Import bzw. Publizieren

Betroffene Dienste/Module:

- TG-import#`ProcessDfgViewerMets` mit Policy `dfgviewermets_import`

Das erste Modul, das mit der Policy `dfgviewermets_import` ausgeführt wird, ist das Modul `ProcessDfgViewerMets`. Dieses liest die angegebene METS-Datei und startet den Importprozess. Die METS-Datei kann gleich zu Beginn dieses Moduls validiert werden. Da ein XML-Reader eingesetzt wird, muss theoretisch nur die Validierung eingeschaltet werden. Da jedoch koLibRI noch mit der METS-Version 1.4 arbeitet, muss diese zunächst aktualisiert werden. Die METS-Dateien des Projekts St. Matthias liegen in Version 1.7 vor, also ist ein Update von koLibRI auf METS 1.7 sinnvoll.

Anforderung 22 Validieren der verpflichtenden Metadaten aller einzuspielenden Objekte vor der Publikation

Betroffene Dienste/Module:

- TG-import#`PublishCheck` mit Policies `aggregation_import`, `dfgviewermets_import` und `complete_import` (neu)
- TG-publish (Anpassungen)

Genau wie bei TG-publish#`PublishCheck` soll bei einer Publikation über TG-import das Vorhandensein der erforderlichen Metadaten geprüft werden. Sinnvollerweise wird dies serverseitig geprüft, so dass der Nutzer das entsprechende Modul nicht einfach aus dem Workflow entfernen kann. Es wäre jedoch schon eine Verbesserung, wenn die Prüfung per TG-import-Modul geprüft würde. Unter Umständen kann das Modul `PublishCheck` von TG-publish direkt nachgenutzt werden.

Anforderung 23 Die Publish-GUI des TextGridLab sowie der betroffene Dienst TG-publish sollen für eine Nutzung der Sandbox angepasst werden

Betroffene Dienste/Module:

- TextGridLab (Publish-GUI)
- TG-publish#`publish()`

Diese Anforderung muss ausführlich diskutiert werden, bevor sie implementiert wird. Im Fall von TG-import liegen alle zu publizierenden Dateien auf der Festplatte der Nutzerin vor und werden während des Publikationsprozesses nicht geändert. Aus diesem Grund kann, wenn gewünscht, einfach der komplette Import wieder aus der Sandbox gelöscht werden. Die Daten, die aus dem TextGridLab publiziert werden sollen, werden während der Publikation direkt durch TG-publish bearbeitet und am Ende des Prozesses publiziert. Ein Löschen der publizierten Dateien würde diese komplett entfernen, ein nochmaliger Publikationsvorgang wäre nicht möglich. Hier muss ein anderer Ansatz gefunden werden. Möglicherweise sind Revisionen eine Lösung: Es könnten alle zu publizierenden Objekte als Revision publiziert werden. So bleiben die ursprünglichen Objekte erhalten und könnten nach der Löschung aus der Sandbox erneut publiziert werden.

5.2 Zertifizierung & Data Curation

Anforderung 1 Auszeichnung aller TextGrid-Objekte mit DataCite-DOIs

Betroffene Dienste/Module:

- TG-pid#`getTextgridDOI()` (abgeleitet aus `getDariahDoi`)
- TG-import#`getDoisAndRewrite` (neu)
- TG-publish#`getDois` (neu)

Zum einen muss der TG-pid-Service um eine Methode für die Erzeugung von DOIs für das TextGrid Repository erweitert werden, es können Methoden der DOI-Erzeugung für das DARIAH-DE Repository nachgenutzt werden. Die Methode `getTextgridDOI()` wird dann von TG-import- und TG-publish-Modulen angesprochen. Zum anderen sollten für alle existierenden Inhalte des TextGrid Repositorys – diese sind bereits mit ePIC-Handles ausgezeichnet – die DOIs nachträglich vergeben werden.

Anforderung 4 Automatisierte Extraktion von Metriken auf Anforderung der Nutzer oder der Administratorinnen

Betroffene Dienste/Module:

- TG-publish

Nachdem diskutiert wurde, welche Metriken bereitgestellt werden sollen, können diese aus den internen Datenbanken des TextGrid Repositorys abgefragt werden. Je nach dem, für wen die Metriken generiert werden sollen, können diese entweder für die Administratoren – möglicherweise per Skript – oder für die Nutzerinnen von TG-publish erstellt und angefordert werden.

Anforderung 7 Extraktion von technischen Metadaten während der Publikation

Betroffene Dienste/Module:

- TG-crud#`CREATE`
- TG-crud#`MOVEPUBLIC`

Technische Metadaten zu erheben macht dann Sinn, wenn digitale Objekte dauerhaft aufbewahrt und nicht mehr verändert werden sollen. Technische Metadaten sollten die Eigenschaften eines digitalen Objekts enthalten, die für eine spätere Formatmigration nach jetzigem Stand der Technik (bzw. der LZA-Forschung) nötig sind. Technische Metadaten können automatisiert aus digitalen Objekten extrahiert werden und werden gemeinsam mit dem digitalen Objekt aufbewahrt. Es macht Sinn, die technischen Metadaten dann zu generieren (bzw. zu extrahieren), solange das digitale Objekt selbst bzw. dessen Dateiformat noch in Benutzung und hinreichend bekannt ist. Es empfehlen sich für die Langzeitarchivierung Dateiformate, deren Formatspezifikationen offen und frei für die Allgemeinheit zugänglich und auch hinreichend dokumentiert sind.

Die Extraktion technischer Metadaten wird sinnvollerweise bei denselben Diensten angesiedelt wie *Anforderung 3: Validierung des Dateiformats aller eingespielten Dateien vor der Publikation*. Sehr wahrscheinlich wird die Extraktion mit demselben Dienst wie die Validierung des Dateiformats implementiert. Es bieten sich Tools wie beispielsweise JHOVE, JOVE2 oder FITS an. Da die Version von JHOVE in der koLibRI-Implementierung (Modul `MetadataGenerator` von TG-import und TG-publish) nicht mehr aktuell ist, wird eine erneute Evaluierung von Extraktionstools empfohlen.

Anforderung 8 Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal

Betroffene Dienste/Module:

- Dokumentation von Workflows, Technik und Schnittstellen

Bei der Evaluierung der Richtlinien des CoreTrustSeal seitens DARIAH-DE wurde als Ergebnis festgestellt, dass die technischen Rahmenbedingungen des TextGrid Repositorys für eine Zertifizierung nahezu komplett erfüllt sind. Es fehlt lediglich an der öffentlichen Dokumentation von Workflows, Technik und Schnittstellen. Im Rahmen der Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal wurden im Rahmen dieser Arbeit die verschiedenen Publikationsworkflows des Repositorys dokumentiert (siehe Kapitel 2.3 auf Seite 23). Diese und weitere Dokumentation werden ebenfalls im Rahmen der Zertifizierung veröffentlicht.

5.3 Software

Anforderung 5 Erhöhung der Performanz des Publikationsworkflows des TextGrid Repositorys

Betroffene Dienste/Module:

- alle

Eine sehr allgemeine Anforderung ist die Prüfung der Module des TextGrid Repositorys auf die Möglichkeit der Erhöhung der Performanz. Sobald es Hinweise auf Verbesserungsmöglichkeiten gibt, werden diese untersucht und umgesetzt.

Anforderung 9 Robustheit und Fehlerunanfälligkeit der Module des TextGrid-Publikationsworkflows (TG-import und TG-publish)

Betroffene Dienste/Module:

- TG-import
- TG-publish

Auch die Robustheit der TG-import- und TG-publish-Module soll erhöht sowie die Anfälligkeit der Module für Fehler reduziert werden. Hier können im Laufe der Entwicklungsarbeiten auffallende Probleme und auftretende Fehler sinnvoll behandelt werden. Allgemeine Verbesserungen (beispielsweise in generischen Modulen und Methoden) können implementiert werden.

Anforderung 11 Zusammenführen von TG-publish und TG-import-Modulen

Betroffene Dienste/Module:

- TG-import
- TG-publish

Da die Workflows der beiden Dienste TG-import und TG-publish in unterschiedlichen Kontexten sehr ähnliche Aufgaben bearbeiten, ist eine Zusammenführung von Methoden dann sinnvoll, wenn Quellcode für eine Aufgabe mehrfach vorliegt bzw. an verschiedenen Orten implementiert ist. Der Pflegeaufwand der Software wird verringert und die Struktur der Software wird übersichtlicher.

Anforderung 12 Anpassen des Formats der TG-import Mapping-Dateien an das Format des TextGrid-Lab

Betroffene Dienste/Module:

- TG-import

Für jeden Import- und Publikationsvorgang erzeugt TG-import zwei Mapping-Dateien: In der ersten ist das Mapping von lokalen Dateipfaden zu TextGrid-URIs verzeichnet, in der zweiten – sofern Handle-PIDs erzeugt werden – das Mapping von TextGrid-URIs zu Handle-PIDs. Momentan werden diese als einfache Textdateien gespeichert und gelesen. Da das von koLibRI genutzte Modul für das Umschreiben der Dateipfade, URIs und PIDs auch im TextGridLab genutzt wird, und das Schreiben von Mapping-Dateien im XML-Format nativ ermöglicht, bietet es sich an, das Format dieses Mappings ebenfalls in koLibRI zu nutzen. So sind die Mapping-Dateien des TextGridLab Im- und Exports zu denen von TG-import kompatibel.

Beispiel einer URI-Mapping Text-Datei (TextGrid-URI ⟶ lokaler Dateipfad):

```
textgrid:bx5.0    file:/Users/fugu/work/asterix.xml
textgrid:bx6.0    file:/Users/fugu/work/obelix.xml
textgrid:bx7.0    file:/Users/fugu/work/idefix.xml
```

Beispiel einer PID-Mapping Text-Datei (Handle-PID ⟶ TextGrid-URI):

```
hdl:21.T11991/0000-0007-5047-F    textgrid:bx5.0
hdl:21.T11991/0000-0007-4C28-8    textgrid:bx6.0
hdl:21.T11991/0000-0007-4E66-0    textgrid:bx7.0
```

Anforderung 13 Anpassung des Moduls zur Anforderung der Handle-PIDs an die Version 2 des PID-Services der GWDG

Betroffene Dienste/Module:

- TG-pid

Das Modul von TG-import und TG-publish, das die ePIC-Handle-PIDs für die Objekte des TextGrid Repositorys vom GWDG Handle-Service anfordert, soll an die Version 2 der Handle-API[210] angepasst werden. Mittelfristig wird nur noch die neue API unterstützt werden, eine Anpassung ist notwendig.

Anforderung 14 Verschieben der TextGrid koLibRI-Module vom Namespace *de.langzeitarchivierung.kolibri* in den TextGrid-Namespace *info.textgrid.middleware.kolibri*

Betroffene Dienste/Module:

- Alle Komponenten der koLibRI-Module *kolibri-addon-textgrid-import*, *kolibri-tgpublish-service* und *kolibri-tgpublish-client*

Eine eher kosmetische Anforderung ist die Umbenennung der Namespaces der genannten Module in den TextGrid-Namespace. Da eine Umbenennung alle Java-Klassen betrifft und ebenso alle die genannten Module aufrufenden Dienste, hätte eine Umbenennung weitreichende Folgen für viele abhängige Komponenten (unter Umständen auch außerhalb von TextGrid). Eine solche Maßnahme sollte umfangreich evaluiert werden.

[210] Vgl. ePIC PID Consortium (2018b). http://doc.pidconsortium.eu/guides/overview

Anforderung 19 Workflow-Konfiguration für die Import-, Publish- und Delete-Funktionen komfortabler gestalten

Betroffene Dienste/Module:

- TG-import

Teilweise können TG-import-Policies nur mit der Angabe von z. B. Pfaden zu Mapping-Dateien ausgeführt werden, welche dann über die dort enthaltenen Objekte iterieren. Eine Erweiterung der Konfigurationsmöglichkeiten der Policies `publish_import`, `continue_import` oder `delete_import` unter Angabe einer TextGrid-URI oder einer ProjectID als Startpunkt ist wünschenswert, so dass einzelne Objekte oder Gruppen von Objekten auch ohne die Mapping-Dateien angesprochen werden können. Für eine Implementierung können die HTTP-Client-Module[211] für die verschiedenen TextGrid-Dienste genutzt werden.

Anforderung 20 Optimierung der Performanz der Handle-PID-Erstellung

Betroffene Dienste/Module:

- TG-pid
- GWDG PID-Service

Es soll hier untersucht werden, ob die Generierung von ePIC-Handle-PIDs beschleunigt werden kann. Momentan bekommt der TG-pid-Service ca. 0,2 PIDs pro Sekunde vom GWDG-Handle-Service (siehe hier auch *Anforderung 5: Erhöhung der Performanz des Publikationsworkflows des TextGrid Repositorys*). Zunächst sollte der Dienst TG-pid geprüft werden, in einem nächsten Schritt kann der GWDG PID-Service auf Performanz getestet werden.

5.4 Visualisierung

Anforderung 16 Adäquate Darstellung von Revisionen im TextGrid Repository (textgridrep.org bzw. TG-search)

Betroffene Dienste/Module:

- TextGrid Repository-Browser `textgridrep.org`
- TextGridLab Navigator (nur für publizierte Objekte)
- TG-search

[211] Vgl. TextGrid Common (2018). https://projects.gwdg.de/projects/common/repository/revisions/master/textgrid-clients

Revisionen von TextGrid-Objekten wurden in vorigen Kapiteln bereits ausführlich besprochen. Prinzipiell soll nur die neueste Revision eines Objekts dargestellt und über die Suche gefunden werden, mit der Möglichkeit, auch ältere Revisionen zu finden, anzuschauen und mit anderen Revisionen des selben Objekts zu vergleichen. Nach dieser Spezifikation sollen revisionierte Objekte in den diversen Darstellungen präsentiert werden. Die Anzeige von sehr vielen Revisionen eines Objekts in einer Suchanzeige ist wenig zielführend und verwirrend. TG-search muss eine Methode implementieren, die eine Abfrage nach neuesten Revisionen unterstützt und diese Methode muss von betroffenen Diensten angesprochen, die Ergebnisse entsprechend angezeigt und eine Möglichkeit der Filterung angeboten werden – beispielsweise eine Suche in allen Revisionen etc.

Anforderung 17 Darstellung von einzelnen Lemmata aus Wörterbüchern, die als einzelne Dateien importiert wurden, Anpassung der Index-Datenbank

Betroffene Dienste/Module:

- TG-Search
- TG-aggregator
- ElasticSearch

Wörterbücher sind nicht immer in einzelne Dateien pro Lemma organisiert, es gibt auch solche, die für alle Lemmata eines Buchstaben eine Datei vorhalten. In der Suche sollen dennoch nicht alle Lemmata des Buchstaben „A" gefunden und angezeigt werden, sondern beispielsweise nur das vom Nutzer gesuchte Lemma „Aalsalat". Zu diesem Zweck soll es möglich sein, aus einer Datei für einen einzelnen Buchstaben nur das gesuchte Lemma zu liefern. Dies könnte in ElasticSearch implementiert werden oder auch über den TextGrid-Aggregator[212], das Export- und Konversions-Tool des TextGrid Repository, das einzelne Objekte und Aggregationen in verschiedenen Formaten auszuliefern vermag.

Die Implementierung der einzelnen Anforderungen wird im nächsten Kapitel anhand der festgelegten Kategorien diskutiert und – sofern sinnvoll möglich – auch durchgeführt und dokumentiert.

[212] Vgl. TextGrid Aggregator (2018). http://www.textgridlab.org/doc/services/submodules/aggregator/docs/index.html

6 Implementierung

Nach der Anforderungsanalyse und der Erstellung eines Konzepts für die identifizierten Anforderungen an den TextGrid Publikationsworkflow folgt die Beschreibung der Implementierung der einzelnen Anforderungen, die auch hier nach den vier Kategorien *Workflow*, *Zertifizierung & Data Curation*, *Software* und *Visualisierung* sortiert wurden. Die Anforderungen aus den Use Cases wurden bevorzugt für die Implementierung ausgewählt, ebenso wie allgemeine Verbesserungen, die auch für die generische Nutzung des TextGrid Repositorys die Qualität der involvierten Publikationsprozesse erhöhen.

6.1 Workflow

Anforderung 2 Verknüpfung des TextGrid-Accounts mit der ORCID ID der Autorin und automatische Eintragung der Publikation bei ORCID

Für die Verknüpfung des TextGrid- bzw. des DARIAH-Accounts mit der ORCID ID des Autors und der automatischen Eintragung der Publikation bei ORCID ist eine Implementierung für die nächste Version des DARIAH-DE Repositorys im Frühjahr 2018 geplant. Bei der Implementierung wird eine Adaption für den TextGrid-Publikationsworkflow mit eingeplant, so dass diese anschließend für das TextGrid Repository ebenfalls implementiert werden kann.

Anforderung 3 Validierung des Dateiformats aller eingespielten Dateien vor der Publikation

Technische Metadaten zu extrahieren – siehe *Anforderung 7: Extraktion von technischen Metadaten während der Publikation* – funktioniert nur dann zielführend, sofern das Dateiformat des zu untersuchenden Objekts zum einen bestimmt werden kann und zum anderen es valide im Sinne der Formatspezifikation ist. Es wäre denkbar, je nach dem, wie restriktiv die Qualität der Daten im Repository geprüft werden soll, vor der Publikation zu testen, ob die zu publizierenden Dateien ihrer Formatspezifikation entsprechen und nur valide Daten anzunehmen.[213] In dieser Arbeit wird zunächst das Format der digitalen Objekte bestimmt, ihre Validität geprüft und die Ergebnisse festgehalten.

Die in *Anforderung 7* extrahierten technischen Metadaten enthalten für erkannte Dateiformate die Formatversion sowie die Validität. Ein für `well-formed`, aber nicht für `valide` befundenes XML-Dokument enthält beispielsweise einen Abschnitt wie:

```
<filestatus>
    <well-formed toolname="Jhove" toolversion="1.16" status="SINGLE_RESULT">
        true
    </well-formed>
    <valid toolname="Jhove" toolversion="1.16" status="SINGLE_RESULT">
        false
    </valid>
    <message toolname="Jhove" toolversion="1.16" status="SINGLE_RESULT">
```

[213] Die Frage, ob eine solche Restriktion sinnvoll ist für ein Repository, bzw. für die Qualität der Daten desselben, bleibt in dieser Arbeit offen und darf gerne bei Bedarf an anderer Stelle diskutiert werden.

```
9           cvc-elt.1: Cannot find the declaration of element 'rdf:RDF'.
10          Line = 1, Column = 391
11      </message>
12  </filestatus>
```

Anhand dieser Angaben könnte die Validität einer zu publizierenden Datei geprüft und entsprechend darauf reagiert werden. Näheres zur Implementierung unter *Anforderung 7* auf Seite 65.

Anforderung 6 Aufnehmen der TextGrid-Publikationen in den Bibliothekskatalog

Das Extrahieren von Metadaten aus dem Bibliothekskatalog der SUB Göttingen für Digitalisate des Göttinger Digitalisierungszentrums[214] (GDZ) wurde bereits im Projekt kopal implementiert.[215] Ein Import von Metadaten in den Göttinger Universitätskatalog[216] (GUK), der heute regelmäßig aus der Datenbank des Gemeinsamen Bibliotheksverbundes[217] (GBV) aktualisiert wird, wäre technisch vermutlich ohne großen Aufwand möglich. Eine Integration der Publikationen des TextGrid Repositorys in den Göttinger Universitätskatalog ist wünschenswert. Dies müsste mit den am Prozess Beteiligten Parteien diskutiert werden, dazu gehören der GBV, die SUB Göttingen und TextGrid.

Anforderung 10 Verbesserung der Client-Server-Kommunikation des Publikationsprozesses von TextGridLab und TG-publish

Diese Anforderung beinhaltet zunächst die Untersuchung der Client-Server-Kommunikation zwischen TextGridLab (Publish-GUI) und dem TG-publish-Service. Da der TG-publish-Service scheinbar zufriedenstellend seine Aufgaben zu erfüllen scheint, die Antworten des Dienstes TG-publish in dieser Arbeit mit weiteren Anforderungen nicht adressiert werden und im Rahmen dieser Arbeit auch keine Kapazitäten für eine Arbeit am TextGridLab zur Verfügung stehen, bleibt diese Anforderung zunächst bestehen. Möglicherweise wurde bei einer Aktualisierung des TG-publish-Services die API leicht geändert, weswegen es zwischen TextGridLab und TG-publish zeitweise zu Kommunikationsunwägbarkeiten kommen. Die Aufgaben im Zusammenhang mit dieser Anforderung sind:

- Tests gegen die TG-publish-API entwickeln, die die zufriedenstellende Arbeitsweise bestätigen
- Untersuchen, ob es zwischen TextGridLab und TG-publish weiterhin zu Problemen kommt
- Evtl. Probleme an Service oder TextGridLab beheben

Anforderung 15 Publikation von Revisionen direkt per TG-import im TextGridRep

Die Publikation von Revisionen mit TG-import wurde erfolgreich implementiert und getestet. Die erforderlichen Erweiterungen wurden für die Policy `complete_import` entwickelt, so dass sie im Rahmen einer Revisionierung der Digitalen Bibliothek von TextGrid genutzt werden können. Die Anpassungen wurden inzwischen mit der Entwicklungsversion – inzwischen 7.0.2-SNAPSHOT – zusammengeführt und

[214] Vgl. Göttinger Digitalisierungszentrum (2018). https://gdz.sub.uni-goettingen.de
[215] Vgl. koLibRI (2018b). https://projects.gwdg.de/projects/kolibri/repository/revisions/master/entry/kolibri-base/src/main/java/de/langzeitarchivierung/kolibri/actionmodule/sub/GetPicaDmdForPpnFromOpac.java
[216] Vgl. Göttinger Universitätskatalog (2018). https://opac.sub.uni-goettingen.de
[217] Vgl. Verbundzentrale des GBV (2018). https://www.gbv.de

können ab der Version des Moduls *kolibri-addon-textgrid-import* Version 6.7.0-SNAPSHOT[218] genutzt werden. Alle Anpassungen werden in der kommenden Release der TextGrid Repository-Middleware (geplant für Frühjahr 2018) enthalten sein. Die zugehörige Dokumentation ist bereits mit Hinweis auf den SNAPSHOT vorhanden.[219] Einige Aspekte müssen bei der Konfiguration beachtet werden:

- Bei einem Revisions-Import müssen die TextGrid-URIs aller Objekte anstatt der lokalen Dateipfade vorliegen, die TextGrid-URIs müssen eine Revisionsangabe haben, z. B. `textgrid:vqmz.1`.
- Nur bereits existierende TextGrid-Objekte können per `complete_import` revisioniert werden.
- Für jedes revisionierte TextGrid-Objekt wird eine neue Handle-PID generiert und in die Metadaten eingetragen.

Eine Anleitung für die Revisionierung der Texte von Johanna Spyri mit Hilfe von TG-import findet sich in Anhang 9.2 auf Seite 73.

Anforderung 18 Ermöglichen der Revisionierung einzelner publizierter Objekte im TextGridLab

Für diese Anforderung sollte zukünftig eine intuitive Lösung im TextGridLab implementiert werden. Da aber momentan keine Revisionierung einzelner Objekte vom TextGridLab aus möglich ist, soll hier ein Workaround beschrieben werden. Prinzipiell muss einfach eine Kollektion oder Edition mit publiziert werden, da aus dem TextGridLab heraus nur Kollektionen und Editionen publiziert werden können. Als Beispiel dient hier ein Bild als Einzelobjekt:

1. Das Bild wird revisioniert.
    - Zunächst wird das Bild aus dem TextGridLab exportiert.
    - Die erforderlichen Änderungen werden am Bild vorgenommen.
    - Beim Import des geänderten Bildes wird die Option *Import as new Revision* in der Import-Perspektive ausgewählt. Als Projekt wird nun *[Projektname] (new Revision)* ausgewählt und importiert. Das Bild liegt nun als neue Revision vor.
2. Die Kollektion oder Edition wird revisioniert.
    - Die publizierte Kollektion oder Edition wird per *Open with → Text Editor* geöffnet.
    - Alle vorhandenen Base-URIs (z. B. `textgrid:vqmz`) in allen `ore:aggregates` Tags wird das Attribut `rdf:resource` händisch an die neue Revisions-URI (z. B. `textgrid:vqmz.1`) abgepasst. Bei dem revisionierten Bild sollte unbedingt auf die alten und neuen Revisionsnummern geachtet werden!
    - Die Kollektion oder Edition wird gespeichert unter *File → Save as new Revision*.

[218] Vgl. koLibRI (2017b). https://projects.gwdg.de/projects/kolibri/repository/kolibri-addon-textgrid-import?rev=6.7.0-SNAPSHOT&tag=6.7.0-SNAPSHOT

[219] Vgl. TextGrid Import (2017a). Complete Import Configuration. http://textgridlab.org/doc/services/submodules/kolibri/kolibri-addon-textgrid-import/docs/index.html

3. Publizieren aller revisionierten Objekte.

    - Im Kontextmenü der Kollektion oder Edition wird *Publish in TextGridRep* gewählt.
    - Die Publish-GUI wird für die Publikation genutzt, zunächst wird der dryRun mit *Proof* gestartet.
    - Bei allen nicht revisionierten Objekten wird erkannt, dass diese bereits publiziert sind.
    - Die neuen Objekte können nun publiziert werden.

Dieser Workaround ist sicherlich nicht komfortabel und auch nicht intuitiv, er ermöglicht jedoch jetzt schon die Revisionierung einzelner bereits publizierter Objekte aus dem TextGridLab heraus. Für eine komfortable und intuitive Lösung der Problematik muss die Publish-GUI des TextGridLab angepasst werden sowie alle Methoden, die mit dem Speichern von neuen Revisionen zu tun haben und auch der Aggregations-Editor.

Anforderung 21 Validieren der METS-Dateien auf XML-Validität (mindestens METS Version 1.7) vor dem Import bzw. Publizieren

Für ein Update des METS-Schemas innerhalb koLibRIs von Version 1.4 auf Version 1.7 wurde das XML-Schema von METS in Version 1.7 eingebunden. Einige wenige Methodenaufrufe von verschiedenen koLibRI-Modulen mussten angepasst werden. Danach konnte die Validierung des XML-Parsers aktiviert werden, ohne dass Fehler auftraten: Die METS-Dateien des Virtuellen Skriptoriums St. Matthias, die in METS 1.7 vorliegen, konnten korrekt auf Validität geprüft werden.

Die Korrektur liegt für die Import-Policy `dfgviewermets_import` in Modul *kolibri-addon-textgrid-import* seit Version 6.4.0[220] vor und wurde bereits mit der Produktivversion zusammengeführt.

Anforderung 22 Validieren der verpflichtenden Metadaten aller einzuspielenden Objekte vor der Publikation

Die Validierung der verpflichtenden Metadaten der einzuspielenden Objekte vor der Publikation wurde für den Dienst TG-publish deutlich verbessert. Am Modul `PublishCheck` wurden einige Optimierungen vorgenommen, die nun zum einen für eine korrekte Validierung der Metadaten sorgen und zum anderen die Konfigurierbarkeit erhöhen:

- Werke werden nun nicht mehr auf die verpflichtenden Metadaten von Items geprüft.
- Die Metadaten bereits publizierte Werke werden ebenfalls nicht mehr geprüft.
- Verpflichtende Metadaten für Werke können nun konfiguriert werden.
- Das Modul `PublishCheck` prüft nun zunächst alle Objekte auf Validität der Metadaten, erst dann werden Fehler gemeldet, nicht bereits beim ersten Fehler.

[220] Vgl. koLibRI (2017c). https://projects.gwdg.de/projects/kolibri/repository/revisions/6.4.0/kolibri-addon-textgrid-import

Diese Anforderung ist im Modul *kolibri-tgpublish-service* seit Version 6.4.0[221] implementiert und wurde ebenfalls bereits mit der Produktivversion zusammengeführt.

Diese Validierung soll ebenfalls in den Dienst TG-import integriert werden, wobei die Implementierung seitens TG-publish teilweise übernommen werden kann. Es stehen eine serverseitige Prüfung mit Einrichtung eines neuen Service-Endpunkts sowie eine Implementierung im Modul TG-Import zur Auswahl. Dieser Teil der Anforderung muss noch evaluiert werden.

Anforderung 23 Die Publish-GUI des TextGridLab sowie der betroffene Dienst TG-publish sollen für eine Nutzung der Sandbox angepasst werden

Die Implementierung der Sandbox des Dienstes TG-import kann nicht ohne Weiteres für das TextGrid-Lab übernommen werden, da die Ausgangssituationen beider Szenarien unterschiedlich sind. Die zu publizierenden Daten werden während des Publikationsprozesses auf verschiedene Art verarbeitet. TG-import kopiert die Daten vor der Bearbeitung: Die Originaldaten bleiben unberührt und können so jederzeit nochmals prozessiert werden. TG-publish verändert die zu publizierenden Daten an sich und verschiebt diese in den statischen TextGrid-Storage: Die Originaldaten existieren im Prinzip nicht mehr.

Als Lösung dieses Dilemmas müssten im TextGridLab Kopien der Originaldaten erstellt werden. Das könnte durch eine Revisionierung aller zu publizierenden Objekte geschehen. Diese Revisionen lägen dann in der Sandbox und könnten – ohne die Originaldaten anzurühren – auch wieder gelöscht werden. Problematisch an einer Implementierung dieser Lösung ist, dass sie nicht nur die Publish-GUI des TextGridLab und den TG-publish-Service betreffen würde, sondern auch weitere Komponenten des TextGridLab und auch der TextGrid Repository Middleware, die Revisionslogik der Middleware müsste überarbeitet werden:

- Wird beispielsweise eine Edition mit noch nicht revisionierten Objekten publiziert, bleiben alle Objekte der Revision 0 erhalten. Die Objekte in der Sandbox (und nach einer finalen Publikation) hätten alle die Revision 1.
- Der Navigator müsste die Revision 1 als Sandbox-Edition erkennen und entsprechend anzeigen.
- Wird die Edition aus der Sandbox publiziert, wird die Revision 0 nach momentanem Stand der Diskussion gar nicht mehr angezeigt, denn die aktuelle Revision ist ja die 1. Eine weitere Bearbeitung der Revision 0 als zukünftige Revision 2 ist momentan nicht möglich.
- Wird die Edition (Revision 1) aus der Sandbox wieder gelöscht, wird bei erneuter Publikation erneut eine Revision 1 angelegt oder wird diese übersprungen und eine Revision 2 erzeugt?
- Unter Umständen wird die Vermischung von publizierten und nicht publizierten Objekten problematisch.
- Das Konzept der Base-URIs und Revisions-URIs muss ebenfalls überdacht werden.

[221] Vgl. koLibRI (2017d). https://projects.gwdg.de/projects/kolibri/repository/revisions/6.4.0/kolibri-tgpublish-service

Aufgrund der hohen Komplexität der Anforderung und der Notwendigkeit eines tiefen Eingriffs in die Logik der Middleware und des TextGridLab wird die Sandbox nicht im Rahmen dieser Arbeit für das TextGridLab implementiert.

6.2 Zertifizierung & Data Curation

Anforderung 1 Auszeichnung aller TextGrid-Objekte mit DataCite-DOIs

Das TextGrid Repository und das DARIAH-DE Repository nutzen ePIC-Handle-PIDs mit dem Präfix `21.11113`. Das Suffix wird vom GWDG Handle-Service angehängt, ein TextGrid Repository Handle-PID ist `21.11113/0000-0000-0021-A`[222]. Seit der Release des DARIAH-DE Repositorys werden dort DataCite-DOIs als Persistente Identifikatoren für die inhaltliche Referenzierung der Inhalte verwendet, die ePIC-Handles werden dort für administrative Zwecke genutzt. Für die DataCite-DOIs nutzt das DARIAH-DE Repository ebenfalls ein eigenes Präfix: `10.20375`. Im Gegensatz zum Handle kann das Suffix für DOIs jedoch innerhalb des Präfix-Namensraums frei gewählt werden, solange es in diesem eindeutig ist. Die DOI wird aus dem DOI-Präfix und dem Handle-Suffix gebildet, weshalb sie im Namensraum von TextGrid und DARIAH-DE Repositorys eindeutig ist. Für das TextGrid Repository können die DOIs aus den vorhandenen Handle-PIDs gebildet werden. Die oben aufgeführte TextGrid Repository Handle-PID würde somit die DOI `10.20375/0000-0000-0021-A` bekommen.

Der TextGrid PID-Service, dessen Code für die Erzeugung von Persistenten Identifikatoren für beide Repositorien genutzt wird, erzeugt bereits DOIs für das DARIAH-DE Repository. Eine Erweiterung für TextGrid-DOIs kann mit wenig Aufwand implementiert werden.

Ein wichtiger Schritt zur Erhaltung der Datenkonsistenz ist jedoch die Nachführung der bereits mit Handle-PIDs publizierten Objekte. Diese sollen ebenfalls mit DOIs ausgezeichnet werden. Es gibt hier zwei Dinge zu beachten: Erstens muss geprüft werden, ob die für die publizierten TextGrid-Objekte erhobenen Metadaten für das DataCite-Metadatenschema[223] ausreichend sind. Zweitens muss die DOI in die TextGrid-Metadaten eingetragen werden (Erzeugung einer neuen Revision). Die Auszeichnung der Inhalte des TextGrid Repositorys mit DOIs ist geplant und wird im Rahmen eines Data-Curation-Prozesses wahrscheinlich noch dieses Jahr durchgeführt.

Anforderung 4 Automatisierte Extraktion von Metriken auf Anforderung der Nutzer oder der Administratorinnen

Diese Anforderung wird im Rahmen dieser Arbeit nicht implementiert.

Anforderung 7 Extraktion von technischen Metadaten während der Publikation

Wie in *Anforderung 3* bereits festgestellt, gehört zur Extraktion technischer Metadaten eine Formatbestimmung sowie eine Validitätsprüfung der digitalen Objekte. Es bietet sich an, auch aus Gründen der Performanz, eine solche Untersuchung erst bei der Publikation durchzuführen. Bis zum jetzigen

[222] Aufgelöst werden kann dieser PID über https://hdl.handle.net/21.11113/0000-0000-0021-A, die Handle-Metadaten können aufgerufen werden über https://hdl.handle.net/21.11113/0000-0000-0021-A?noredirect

[223] Vgl. DataCite (2018c). https://schema.datacite.org

Zeitpunkt erhält ein Objekt im TextGridRep eine Checksumme sowie einen Persistenten Identifikator, technische Metadaten werden bisher nicht extrahiert. Es gibt bereits eine Reihe von Tools, die in der Lage sind, technische Metadaten im Sinne der LZA zu extrahieren, beispielsweise JHOVE oder JHOVE2[224]. Ein weiterer Dienst ist das File Information Toolset (FITS), der gegenüber anderen den Vorteil hat, dass es zum einen sehr aktuell ist, eine große Nutzerbasis hat und intensiv gepflegt wird. Zum anderen nutzt das FITS selbst eine große Zahl von weiteren Tools für die Extraktion der technischen Metadaten und bietet die Möglichkeit, die zu publizierenden Dateien von dieser Vielzahl von Tools prüfen zu lassen. Hierbei wird vom FITS sowohl untersucht, ob die verschiedenen Tools zu unterschiedlichen Ergebnissen für Formatbestimmung und Validitätsprüfung kommen, als auch eine zusammenfassende XML-Datei im FITS-Schema[225] erstellt.

Im Rahmen dieser Arbeit wurde der FITS-Webservices[226] in die TextGrid-Middleware integriert, der aus den im TextGrid Repository publizierten Objekten – in unserem Fall Dateien –, technische Metadaten extrahiert und diese FITS XML-Datei in dasselbe Verzeichnis speichert, in dem die Objekte selbst sowie die zugehörigen TextGrid Metadaten-Dateien liegen. Die Extraktion wird zunächst für alle Objekte erfolgen, die neu publiziert werden: Aus dem TextGridLab heraus spricht TG-publish die `CREATE`-Methode des TG-crud-Services an, die den TextGrid FITS-Service anspricht, TG-import ruft ebenfalls TG-crud auf, und dieser erzeugt in der Methode `MOVEPUBLIC` die technischen Metadaten über den FITS-Service. Es kann für spätere Metriken hilfreich sein, bestimmte Metadaten in die ElasticSearch-Datenbank des TextGrid Repositorys zu schreiben, um einen Überblick über die Dateiformate aller publizierten Objekte und die Validität derselben zu bekommen (siehe *Anforderung 4: Automatisierte Extraktion von Metriken auf Anforderung der Nutzer oder der Administratorinnen*).

Eine Beispieldatei technischer Metadaten, die von FITS für eine XML-Datei extrahiert wurde, zeigt, welche Metadaten von welchen Tools identifiziert wurden und dass die untersuchte XML-Datei zwar für `well-formed`, jedoch nicht für `valide` befunden wurde (siehe Anhang 9.3.4, Seite 82).[227]

In TG-crud Version 8.0.7-SNAPSHOT[228] ist die FITS-Integration implementiert und bereits auf dem Entwicklungssystem deployed und wird in der nächsten Release der TextGrid-Middleware in den Produktivbetrieb übernommen.

Anforderung 8 Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal

Die Beantragung der Zertifizierung des TextGrid Repositorys mit dem CoreTrustSeal überschneidet sich mit der Abgabe dieser Arbeit und wird bis dahin wahrscheinlich nicht abgeschlossen sein. Die Zertifizierung selbst ist nicht Teil der Arbeit.

[224] Vgl. JHOVE2 (2018). https://bitbucket.org/jhove2/main/wiki/Home
[225] Vgl. Harvard Library (2018). http://hul.harvard.edu/ois/xml/xsd/fits/fits_output.xsd
[226] Vgl. FITSServlet (2018). https://github.com/harvard-lts/FITSservlet
[227] Vgl. https://dev.textgridlab.org/1.0/tgcrud-public/rest/textgrid:2vr39.0/tech
[228] Vgl. TG-crud (2018). https://projects.gwdg.de/projects/tg-crud/repository?rev=8.0.7-SNAPSHOT&branch=master&tag=8.0.7-SNAPSHOT

Anforderung 5 Erhöhung der Performanz des Publikationsworkflows des TextGrid Repositorys

Diese Anforderung wurde nicht dediziert in dieser Arbeit bearbeitet.

Anforderung 9 Robustheit und Fehlerunanfälligkeit der Module des TextGrid-Publikationsworkflows (TG-import und TG-publish)

Diese Anforderung wurde nicht dediziert in dieser Arbeit bearbeitet.

Anforderung 11 Zusammenführen von TG-publish und TG-import-Modulen

Diese Anforderung wurde nicht dediziert in dieser Arbeit bearbeitet.

Anforderung 12 Anpassen des Formats der TG-import Mapping-Dateien an das Format des TextGrid-Lab

Vor der Anpassung wurden die lokalen Dateipfade zu den TextGrid-URIs über einfache Text-Dateien gemapped, genauso wie das Mapping von TextGrid-URIs zu Handle-PIDs (siehe Kapitel 5, Seite 56). Da das Rewrite-Mapping-Modul des TextGridLab-Imports jedoch auch von TG-import genutzt wird, und das Modul bereits eine Serialisierung der IMEX-Dateien (Import/Export) bietet, wurde diese in TG-import eingebunden und genutzt.

Beispiel einer URI-Mapping IMEX-Datei (TextGrid-URI ⟶ lokaler Dateipfad):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<importSpec xmlns="http://textgrid.info/import">
    <importObject textgrid-uri="textgrid:bx5.0"
        local-data="file:/Users/fugu/work/asterix.xml" rewrite-method="xml"/>
    <importObject textgrid-uri="textgrid:bx6.0"
        local-data="file:/Users/fugu/work/obelix.xml" rewrite-method="xml"/>
    <importObject textgrid-uri="textgrid:bx7.0"
    local-data="file:/Users/fugu/work/idefix.xml" rewrite-method="xml"/>
</importSpec>
```

Beispiel einer PID-Mapping IMEX-Datei (Handle-PID ⟶ TextGrid-URI):

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<importSpec xmlns="http://textgrid.info/import">
    <importObject textgrid-uri="hdl:21.T11991/0000-0007-5047-F"
        local-data="textgrid:bx5.0" rewrite-method="xml"/>
    <importObject textgrid-uri="hdl:21.T11991/0000-0007-4C28-8"
        local-data="textgrid:bx6.0" rewrite-method="xml"/>
    <importObject textgrid-uri="hdl:21.T11991/0000-0007-4E66-0"
        local-data="textgrid:bx7.0" rewrite-method="xml"/>
</importSpec>
```

Anforderung 13 Anpassung des Moduls zur Anforderung der Handle-PIDs an die Version 2 des PID-Services der GWDG

Der TextGrid TG-pid-Service wurde in Release-Version 2.4.0 an die ePIC PID-Service API 2.0 angepasst.[229]

Anforderung 14 Verschieben der TextGrid koLibRI-Module vom Namespace *de.langzeitarchivierung.kolibri* in den TextGrid-Namespace *info.textgrid.middleware.kolibri*

Ein Ändern des Namespaces der koLibRI-Module wird im Rahmen dieser Arbeit für nicht sinnvoll erachtet, da es für zu viele abhängige Dienste bzw. Module unbestimmte Auswirkungen haben würde. Weiterhin besteht kein dringender Grund für die Umbenennung, abgesehen von der Kosmetik.

Anforderung 19 Workflow-Konfiguration für die Import-, Publish- und Delete-Funktionen komfortabler gestalten

Für die Import-, Publish- und Delete-Policies kann durch Nutzung der TextGrid HTTP-Clients – hauptsächlich durch Nutzung von Anfragen an TG-search und TG-auth* – mit vier Optionen auf die zu bearbeitenden Objekte zugegriffen werden:

- Nutzung der URI-Mapping-Datei: `file:./folders/temp/1470065621459_data_URI.imex`
- Nutzung der PID-Mapping-Datei: `file:./folders/temp/1470065621459_data_PID.imex`
- Nutzung einer TextGrid-URI: `textgrid:12345.0`
- Nutzung einer TextGrid ProjectID: `project:TGPR-f1867520-4a53-9ced-9da5-503762ba0f61`

Diese Anforderung wurde im Rahmen der Arbeiten an der Policy `dfgviewermets_import` bearbeitet.[230]

Anforderung 20 Optimierung der Performanz der Handle-PID-Erstellung

Die Performanz der Handle-PID-Erstellung wurde im Rahmen dieser Arbeit um ca. den Faktor zehn erhöht. Eine bisher nicht benötigte Option des GWDG PID-Services konnte abgeschaltet werden, die vom TG-pid Service-Wrapper bisher genutzt wurde. Es werden nun ca. zwei PIDs pro Sekunde erzeugt. Diese spezielle Option `CHECK_PIDS_FIRST` testet – sofern aktiviert – direkt in der PID-Datenbank, ob für eine angegebene URL bereits eine PID erzeugt wurde. Dieses Rückwärts-Tracking ist kostspielig im Rahmen der Abfrage und wird für die TextGrid PID-Generierung nicht benötigt.

[229] Vgl. TG-pid (2018). https://projects.gwdg.de/projects/tg-pid/repository?rev=2.4.0&branch=master&tag=2.4.0

[230] Vgl. TextGrid Import (2017a). DFG Viewer METS Import Configuration. http://textgridlab.org/doc/services/submodules/kolibri/kolibri-addon-textgrid-import/docs/index.html

6.4 Visualisierung

Anforderung 16 Adäquate Darstellung von Revisionen im TextGrid Repository (textgridrep.org bzw. TG-search)

Sobald die ersten Revisionen im großen Stil eingespielt werden – beispielsweise die Digitalen Bibliothek von TextGrid – sollte diese Anforderung erneut geprüft werden. Im Rahmen dieser Arbeit wurde sie nicht implementiert.

Anforderung 17 Darstellung von einzelnen Lemmata aus Wörterbüchern, die als einzelne Dateien importiert wurden, Anpassung der Index-Datenbank

Auch diese Anforderung wird erst bei Notwendigkeit neu evaluiert. Seitens ElasticSearch und TextGrid Aggregator wurden bereits einige Untersuchungen vorgenommen, die jedoch ebenfalls hier nicht behandelt werden.

Während der Implementierungsphase dieser Arbeit wurde ein Großteil der Anforderungen bearbeitet und davon einige für die Use Cases wichtige Anforderungen vollständig implementiert. Nicht behandelt wurden einige allgemeine Anforderungen aus der Kategorie *Software* sowie die aus Kategorie *Visualisierung*, da diese eher die GUI betreffen und nicht Middleware und Publikationsworkflow des TextGrid Repositorys.

7 Verwandte Arbeiten

Das TextGrid Repository wurde als Virtuelle Forschungsumgebung Anfang des neuen Jahrtausends für Geisteswissenschaftlerinnen entwickelt, weil keine vergleichbaren elektronischen Werkzeuge für kooperatives Arbeiten in den eHumanities verfügbar waren. Zwar existierten zu diesem Zeitpunkt bereits Repository-Softwareprodukte, beispielsweise D-Space[231] und Fedora[232], auf der man hätte aufbauen können, aber die Anforderungen waren zu spezifisch – beispielsweise im Bereich der XML-Verarbeitung bzw. -Suche. Es gab auch Werkzeuge für die philologische Datenverarbeitung wie beispielsweise TUSTEP, aus deren Nutzerkreis sich schließlich auch die Idee für TextGrid entwickelte.[233] Diese Tools waren jedoch weder „gridifiziert", noch für die kooperative Arbeit mit mehreren Personen entwickelt worden, noch wurden sie in diese Richtungen weiter entwickelt. Das TextGrid Repositorium wurde im Sommer 2011 in Betrieb genommen, die Version 2.0 folgte im Mai 2012.[234]

Inzwischen gibt es weitere Repositorien, die ähnliche Angebote für die sichere Archivierung und die Publikation von Daten anbieten. Im Jahr 2013 wurde Zenodo[235] released, ein Repositorium, das vom Projekt OpenAIRE in Zusammenarbeit mit dem CERN entwickelt wurde.[236] Weiterhin gibt es Repositorien als sogenannte *Dark Archives*, die – oft auf kommerzieller Basis – Langzeitarchivierungs-Dienste zur Verfügung stellen, jedoch explizit nur auf die Archivierung und meistens nicht auf die Präsentation und Verbreitung der beinhalteten Daten fokussiert sind.

Als spezifische Forschungsumgebung für die Geisteswissenschaften wurde TextGrid im Rahmen dieser Arbeit explizit ausgewählt, um Analysen der bestehenden Publikationsworkflows durchzuführen und diese zu erweitern.

[231] Vgl. DSPACE (2018). https://dspace.org
[232] Vgl. Fedora (2018). http://fedorarepository.org
[233] Vgl. Wegstein; Rapp und Jannidis (2015). S. 30ff.
[234] Vgl. Funk; Veentjer und Vitt (2013), S. 277.
[235] Vgl. Zenodo (2018a). https://zenodo.org
[236] Vgl. Zenodo (2018b). http://about.zenodo.org/

8 Schluss und Ausblick

Als das Projekt TextGrid vor fast zwölf Jahren startete, war das Angebot an Repositorien oder gar Virtuellen Forschungsumgebungen noch nicht sonderlich groß bzw. für die Geisteswissenschaften eigentlich nicht existent. In den Jahren nach der Jahrtausendwende arbeiteten und forschten die Naturwissenschaften bereits mit großen Datenmengen – beispielsweise aus Elektronenbeschleunigern, astronomischen Großgeräten und anderen Daten, die maschinell erstellt wurden –, und nutzten für ihre Datenspeicherung und -prozessierung *DAS GRID*[237]. In den Geisteswissenschaften wurde zu diesem Zeitpunkt zwar ebenfalls bereits mit und an großen Korpora gearbeitet, jedoch meist auf einzelnen Rechnern mit Satz- und Analyseprogrammen und nicht kooperativ und gemeinsam an Forschungsumgebungen für digitale Daten. Die Unterschiede lagen also nicht nur in den Datenmengen und -formaten, sondern in der Form und Methodik, wie Forschung betrieben wurde, und zugleich auch in der wissenschaftlichen Praxis: Die Naturwissenschaften, die eher in Gruppen an einzelnen Forschungsthemen arbeiteten und die Geisteswissenschaften, in denen oftmals Forschung durch Einzelforscherinnen und kleine Forschungsgruppen betrieben wurde. Die digitale Transformation und zugleich die elektronische Verfügbarkeit von Daten und Werkzeugen – nicht zuletzt unterstützt durch sich verändernde Rahmenbedingungen der Forschungsförderer in Deutschland – haben dazu geführt, dass seit der Jahrtausendwende auch in den Geisteswissenschaften verstärkt digital und kooperativ Forschung betrieben wird. Dabei haben insbesondere die in den letzten Jahren entwickelten digitalen Werkzeuge zum Forschungsprozess in großem Maße beigetragen.

Als weitere Entwicklung im Zuge der digitalen Transformation hat sich das Publikationswesen in der Form verändert, dass es – auch für Einzel- und auch Hobbywissenschaftler – nun sehr viel einfacher möglich ist, ihre Ergebnisse und Forschungsdaten zu publizieren und auch anderen Forscherinnen zur Verfügung zu stellen, und dies in einer Form, die auch eine langfristige Verfügbarkeit und Referenzierbarkeit garantiert. Open Access, Open Science und ein neues Selbstverständnis der Bibliotheken tragen maßgeblich dazu bei, dass auch zukünftige Entwicklungen im Bereich der Publikationen und wissenschaftlichen Repositorien in diesem Sinne weiter entwickelt werden. Das Verhältnis zum Publizieren hat sich ebenfalls verändert, in dem ein Kulturwandel des elektronischen Publizierens gemeinsam mit der digitalen Transformation eigeleitet wurde.

Als Antwort auf die Fragen aus der Einleitung kann gesagt werden, dass digitale Publikationsworkflows permanent an die Anforderungen der Wissenschaft angepasst werden müssen, da sich Anforderungen und Forschungsfragen stetig ändern. Ein Publikationsworkflow muss im Allgemeinen – und unter Umständen für die Geisteswissenschaften im Spezifischen – intuitiv zu bedienen und gut dokumentiert sein. Er muss allerdings auch für Wissenschaftlerinnen mit profunden Kenntnissen aus der Informationstechnologie oder der Informatik sinnvolle Schnittstellen zu Daten und Publikationen geben, so dass diese auch für spezielle Forschungsfragen und -techniken zur Verfügung stehen.

Die vorhandenen Workflows aus der Virtuellen Forschungsumgebung TextGrid sind geeignet, die Anforderungen für digitale Publikationen aus den Geisteswissenschaften zu bedienen, auch diese müssen jedoch immer wieder anhand der Use Cases geprüft, angepasst und notfalls korrigiert werden, was im

[237] Vgl. Wikipedia (2018i). https://de.wikipedia.org/wiki/Grid-Computing und Wikipedia (2018j). https://de.wikipedia.org/wiki/D-Grid

Rahmen dieser Arbeit zu einem Teil erreicht wurde. Die nötigen technischen Erweiterungen wurden ebenfalls teilweise umgesetzt – beispielsweise Vorbereitungen für Langzeitarchivierungs-Strategien durch Extraktion von technischen Metadaten –, und auch hier müssen die Entwicklungen weiterhin beobachtet und angepasst werden. In dieser Arbeit wurden die Anforderungen aus den Use Cases analysiert und implementiert, andere allgemeine Anforderungen erfüllt und einige aufgezeigt, die noch erforscht und diskutiert und die im Rahmen von zukünftigen Forschungsvorhaben aufgegriffen werden müssen.

Das TextGrid Repository ist durch die im Rahmen dieser Arbeit durchgeführten Arbeiten und Entwicklungen einen weiteren Schritt in Richtung Langzeitarchivierung und Datenbewahrung gegangen. Die Zertifizierung mit dem CTS wurde in der Arbeit thematisiert, Publikationsprozesse und Workflows wurden dafür dokumentiert und der Zertifizierungsprozess wird in den nächsten Wochen oder sogar Tagen angestoßen. Der Import wurde analysiert und verbessert, und es hat sich gezeigt, dass es auch hier nicht ausreicht, ab und zu mal ein paar Zeilen Code zu schreiben. Es muss geplant werden, Prozesse müssen re-analysiert werden, so dass diese Prozesse und auch die Software stetig an neue Anforderungen angepasst werden können.

Ein Repositorium kann Daten für lange Zeit vorhalten und sicherlich mit guter Planung auch Inhalte über die Lebensdauer von obsoleten Datenformaten hinweg mit LZA-Strategien bewahren. Wichtig ist für die Zukunft, dass die Daten nicht nur bewahrt, sondern auch genutzt werden können. Der wichtigste Punkt nach einer sicheren Aufbewahrung (PIDs, Storage, Zertifizierung etc.) ist es, die schwer erarbeiteten Forschungsdaten zu nutzen, neue Ergebnisse damit zu erzielen, sie in anderen Kontexten zu verwenden und sie in einem anderen Licht zu sehen.

So scheint letztlich ein Schulterschluss zwischen Informationstechnologie und den Geisteswissenschaften gut gelungen zu sein. Die Bibliotheken nehmen ihre neue Verantwortung ernst, forschen und unterstützen die Forschung an Forschungsumgebungen und neuen Publikationsformen. Abgeschlossen ist diese Aufgabe jedoch noch nicht, die Kommunikation zwischen IT und den Geisteswissenschaften muss weiterhin gefördert werden, so dass die Informatik, die Informations- und Bibliothekswissenschaften und die Geisteswissenschaften sich weiter aneinander annähern und miteinander kommunizieren und forschen können.

9 Anhang

9.1 Entwicklungsumgebung

Zur Entwicklung der neuen Funktionalitäten wurde mit dem Administrationsprogramm Puppet[238] eine lokale virtuelle Maschine mit einer kompletten TextGrid Middleware aufgesetzt (siehe https://gitlab.gwdg.de/dariah-de-puppet/dariah_de_puppet). Sowohl TG-import (koLibRI) als auch das TextGridLab kann mit dieser lokalen TextGrid-Entwicklungsinstanz arbeiten und Nutzer können sich dort anmelden.

Mit Hilfe des Puppet-Moduls https://github.com/DARIAH-DE/puppetmodule-dhrep mit Scope `textgrid` werden alle TextGrid-spezifischen Dienste einrichtet und die VM korrekt konfiguriert. Ein Entwicklungs-Branch `master_fu` wurde eingerichtet: https://projects.gwdg.de/projects/tg-crud/repository?utf8=%E2%9C%93&rev=master_fu&branch=master_fu. Dieser wurde nach ausgiebigen Tests mit dem Master-Branch zusammengeführt und so wurden alle Neuerungen in die produktive TextGrid-Middleware eingebracht.

Die genutzten TG-crud-Pakete *tgcrud-webapp* und *tgcrud-webapp-public* sowie die TG-import- und TG-publish-Pakete *kolibri-addon-textgrid-import* und *kolibri-tgpublish-service* wurden lokal als Debian-Package (.deb) mittels Maven gebaut und dann zu Testzwecken auf der lokalen virtuellen Maschine installiert.

Alle implementierten Dienste und Module wurden nach erfolgreichen lokalen Tests auf die TextGrid-Testinstanz `dev.textgridlab.org` (`textgrid-esx1.gwdg.de`) deployed und nach ebenso erfolgreichen Tests dort zuletzt auch auf die Produktivinstanz `textgridlab.org` (`textgrid-esx2.gwdg.de`). Die API-Dokumentation befindet sich auf beiden Systemen jeweils unter https://dev.textgridlab.org/doc/services/ bzw. https://textgridlab.org/doc/services.

9.2 Anleitung für die Revisionierung der Texte von Johanna Spyri

Diese Kurzanleitung geht von einem installierten TextGridLab in Version 3.2 sowie der Einstellung des TextGrid Konfigurations-Services (*Preferences → TextGrid Server / Proxy → Configuration Service URL*) auf den TextGrid Repository Entwicklungsserver https://dev.textgridlab.org/1.0/confserv aus.

1. Die Kollektion mit dem Namen der Autorin Johanna Spyri werden per Rechtsklick aus dem TextGrid Repository in ein eigenes Projekt kopiert und beliebig verändert.
2. Die Edition aus dem neuen Projekt mit den veränderten Dateien werden aus dem TG-lab *mit* Export-Mapping exportiert. Die TextGrid-Metadaten, die aus dem TG-lab exportiert wurden, sind in Quellcode 9.3.1 auf Seite 74 aufgeführt.[239].

[238] Vgl. puppet (2018). https://puppet.com

[239] Entfernt wurden lediglich die Teile, die für den Import nicht relevant sind, dies betrifft z. B. alle Daten aus dem `<generated>` Tag, denn diese werden bei einem `CREATE` von TG-crud neu generiert.

3. Importieren der neuerlich exportierten Daten als Revision. Für das Publizieren in das Textgrid Repository wird TG-import mit der Policy `complete_import` genutzt. Die Konfigurationsdatei ist im Quellcode 9.3.3 auf Seite 76 abgedruckt[240].

    - Die Metadaten XML-Datei für den Revisions-Import – nötig ist hier eigentlich nur die TextGrid-URI im `<generated>` Teil – findet sich in Quellcode 9.3.2 auf Seite 75.
    - Alle Aggregations-Objekte (Editionen, Kollektionen und Aggregationen) *müssen* nun für einen Import mit Revisionierung als Referenzen TextGrid-URIs enthalten anstelle der lokalen Dateipfade.
    - Die vorhandenen PIDs in den Metadaten müssen vor dem Re-Import entfernt werden, sonst bleiben die Verweise in den Metadaten erhalten und die neue PID wird als weiterer Identifikator der alten hinzugefügt. Dies ist über die Konfigurationsdatei konfigurierbar.

4. Nach Ausführung und erfolgreicher Publikation sind nun alle revisionierten Objekte in der Sandbox des TextGrid Repository zu finden.
5. Bei erfolgreicher Revisionierung und Publikation in die Sandbox und abschließender Prüfung kann die Revision mit der Policy `publish_import` final publiziert werden.

9.3 Quellcode und Konfiguration

9.3.1 Metadaten für den initialen Import

```
<object xmlns="http://textgrid.info/namespaces/metadata/core/2010">
    <generic>
        <provided>
            <title>Heidis Lehr- und Wanderjahre</title>
            <format>text/tg.edition+tg.aggregation+xml</format>
            <notes>Erstdruck (anonym): Gotha (Justus Perthes) 1880.</notes>
        </provided>
    </generic>
    <edition>
        <isEditionOf>../../Heidis_Lehr-_und_Wanderjahre.2v9m7.0.work</isEditionOf>
        <agent role="author" id="">Spyri, Johanna</agent>
        <source>
            <bibliographicCitation>
                <author id="pnd:118616455">Spyri, Johanna</author>
                <editionTitle>
                    Johanna Spyri: Heidis Lehr- und Wanderjahre, Zürich: Diogenes,
                    1978.
                </editionTitle>
```

[240] Alle unnötigen Modulkonfigurationen sind der Übersichtlichkeit halber entfernt worden.

```
                <placeOfPublication>
                    <value>Zürich</value>
                </placeOfPublication>
                <dateOfPublication date="1978"></dateOfPublication>
                <spage>11</spage>
            </bibliographicCitation>
        </source>
        <license
            licenseUri="http://creativecommons.org/licenses/by/3.0/de/legalcode">
            Der annotierte Datenbestand der Digitalen Bibliothek inklusive
            Metadaten sowie davon einzeln zugängliche Teile sind eine
            Abwandlung des Datenbestandes von www.editura.de durch TextGrid und
            werden unter der Lizenz Creative Commons Namensnennung 3.0
            Deutschland Lizenz (by-Nennung TextGrid) veröffentlicht. Die Lizenz
            bezieht sich nicht auf die der Annotation zu Grunde liegenden
            allgemeinfreien Texte (Siehe auch Punkt 2 der Lizenzbestimmungen).
        </license>
    </edition>
</object>
```

9.3.2 Metadaten für den Revisions-Import

```
<object xmlns="http://textgrid.info/namespaces/metadata/core/2010">
    <generic>
        <provided>
            <title>Heidis Lehr- und Wanderjahre</title>
            <format>text/tg.edition+tg.aggregation+xml</format>
            <notes>Erstdruck (anonym): Gotha (Justus Perthes) 1880.</notes>
        </provided>
        <generated>
            <created>2017-03-19T01:49:05.307+01:00</created>
            <lastModified>2017-03-19T01:49:05.307+01:00</lastModified>
            <textgridUri extRef="">textgrid:2v9pd.0</textgridUri>
            <revision>0</revision>
            <extent>527</extent>
            <fixity>
                <messageDigestAlgorithm>md5</messageDigestAlgorithm>
                <messageDigest>0e00eb98c2ae7f8742fef527231f9896</messageDigest>
                <messageDigestOriginator>
                    crud-base 7.5.0-SNAPSHOT
                </messageDigestOriginator>
            </fixity>
            <dataContributor>stefan.funk@textgrid.de</dataContributor>
```

```
            <project id="TGPR-6052a148-79c9-ed4f-a357-4f54e0d6d4a1">
                Fugus Publish Testprojekt 2
            </project>
            <permissions>delegate publish delete read write</permissions>
        </generated>
    </generic>
    <edition>
        <isEditionOf>textgrid:2v9p5.0</isEditionOf>
        <agent role="author" id="">Spyri, Johanna</agent>
        <source>
            <bibliographicCitation>
                <author id="pnd:118616455">Spyri, Johanna</author>
                <editionTitle>
                    Johanna Spyri: Heidis Lehr- und Wanderjahre, Zürich: Diogenes,
                    1978.
                </editionTitle>
                <placeOfPublication>
                    <value>Zürich</value>
                </placeOfPublication>
                <dateOfPublication date="1978"></dateOfPublication>
                <spage>11</spage>
            </bibliographicCitation>
        </source>
        <license licenseUri="http://creativecommons.org/licenses/by/3.0/de/legalcode">
            Der annotierte Datenbestand der Digitalen Bibliothek inklusive
            Metadaten sowie davon einzeln zugängliche Teile sind eine Abwandlung
            des Datenbestandes von www.editura.de durch TextGrid und werden
            unter der Lizenz Creative Commons Namensnennung 3.0 Deutschland
            Lizenz (by-Nennung TextGrid) veröffentlicht. Die Lizenz bezieht sich
            nicht auf die der Annotation zu Grunde liegenden allgemeinfreien
            Texte (Siehe auch Punkt 2 der Lizenzbestimmungen).</license>
    </edition>
    <relations>
        <isDerivedFrom>textgrid:vqn3.0</isDerivedFrom>
    </relations>
</object>
```

9.3.3 Konfigurationsdatei für TG-import mit Policy `complete_import`

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="koLibRI-config" xmlns:koLibRI="koLibRI-config"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
    "koLibRI-config http://kopal.langzeitarchivierung.de/schema/koLibRI/config.xsd">
```

```
    <common>
        <!-- THE LOG LEVEL. DO EDIT AS YOU LIKE! -->
        <property>
            <field>logLevel</field>
            <value>INFO</value>
        </property>

        <!-- EDIT COMMON TEXTGRID USER SETTINGS BELOW -->
        <property>
            <field>defaultPolicyName</field>
            <value>complete_import</value>
        </property>
        <property>
            <field>hotfolderDir</field>
            <value>./folders/hotfolder-db/importHeidi-reprevisions/</value>
        </property>
        <property>
            <field>projectId</field>
            <value>TGPR-372fe6dc-57f2-6cd4-01b5-2c4bbefcfd3c</value>
        </property>
        <property>
            <field>rbacSessionId</field>
            <value>***</value>
        </property>
        <!-- EDIT COMMON TEXTGRID USER SETTINGS ABOVE -->

        <!-- EDIT COMMON TEXTGRID VALUES BELOW -->
        <property>
            <field>tgcrudServerUrl</field>
            <value>
                https://dev.textgridlab.org/1.0/tgcrud-public/TGCrudService?wsdl
            </value>
        </property>
        <property>
            <field>tgauthServerUrl</field>
            <value>https://dev.textgridlab.org/1.0/tgauth/tgextra.php</value>
        </property>
        <property>
            <field>tgsearchServerUrl</field>
            <value>https://dev.textgridlab.org/1.0/tgsearch-public</value>
        </property>
        <property>
            <field>sesameServerUrl</field>
            <value>https://dev.textgridlab.org/1.0/triplestore/textgrid-public</value>
```

```
        </property>
        <property>
            <field>tgpublishServerUrl</field>
            <value>https://dev.textgridlab.org/1.0/tgpublish/</value>
        </property>
        <property>
            <field>tgpidServerUrl</field>
            <value>https://dev.textgridlab.org/1.0/tgpid/</value>
        </property>
        <property>
            <field>logParameter</field>
            <value/>
        </property>
        <property>
            <field>createNewRevisions</field>
            <value>true</value>
        </property>
        <property>
            <field>ignoredUrlPrefixes</field>
            <value>http://</value>
            <value>www.</value>
            <value>file:/</value>
        </property>
        <!-- EDIT COMMON TEXTGRID VALUES ABOVE -->

        <!-- FIXED COMMON TEXTGRID VALUES BELOW (DO NOT EDIT!) -->
        <property>
            <field>uriPrefix</field>
            <value>textgrid</value>
        </property>
        <!-- FIXED COMMON TEXTGRID VALUES ABOVE (DO NOT EDIT!) -->

        <!-- MORE SETTINGS BELOW (DO ONLY EDIT IF YOU KNOW WHAT YOU DO!) -->
        <property>
            <field>defaultProcessStarter</field>
            <value>MonitorHotfolder</value>
        </property>
        <property>
            <field>policyFile</field>
            <value>./config/policies.xml</value>
        </property>
        <property>
            <field>logfileDir</field>
            <value>./folders/log/</value>
```

```
        </property>
        <property>
            <field>maxFiles</field>
            <value>5000</value>
        </property>
        <property>
            <field>maxNumberOfThreads</field>
            <value>1</value>
        </property>
    </common>
    <modules>
        <class name="processstarter.MonitorHotfolder">
            <!-- Commonly defined values: defaultPolicyName, hotfolderDir -->
            <property>
                <field>readDirectoriesOnly</field>
                <value>false</value>
            </property>
            <property>
                <field>recheckHotfolder</field>
                <value>false</value>
            </property>
            <property>
                <field>ignoreHiddenFiles</field>
                <value>true</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.TextgridMetadataProcessor">
            <!-- Commonly defined values: metadataSuffix, aggregationSuffix,
            jhoveMetadataSuffix -->
            <property>
                <field>textgridMetadataTemplate</field>
                <value>./config/textgrid_metadata_template.xml</value>
            </property>
            <property>
                <field>useUofTechnicalMetadata</field>
                <value>false</value>
            </property>
            <property>
                <field>omitFileSuffix</field>
                <value>true</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.GetUris">
            <!-- Commonly defined value: metadataSuffix, createNewRevisions -->
```

```
            <property>
                <field>urisForFolders</field>
                <value>false</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.CreateAggregations">
            <!-- Commonly defined values: metadataSuffix, aggregationSuffix,
            referenceSuffix, workSuffix -->
            <property>
                <field>useBaseUrisInAggregations</field>
                <value>false</value>
            </property>
            <property>
                <field>sortAlphabetically</field>
                <value>true</value>
            </property>
            <property>
                <field>omitWorkReferences</field>
                <value>true</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.RenameAndRewrite">
            <!-- Commonly defined values: uriPrefix, metadataSuffix,
            aggregationSuffix, editionSuffix, collectionSuffix, xmlSuffix,
            ignoredUrlPrefixes -->
            <property>
                <field>jhoveMetadataSuffix</field>
                <value>.jhove</value>
            </property>
            <property>
                <field>rewritePrefix</field>
                <value>tmp.</value>
            </property>
            <property>
                <field>useBaseUrisInAggregations</field>
                <value>false</value>
            </property>
            <property>
                <field>otherTgsearchServerUrl</field>
                <value>https://dev.textgridlab.org/1.0/tgsearch</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.GetPidsAndRewrite">
            <!-- Commonly defined values: metadataSuffix, uriPrefix, rbacSessionId,
```

```
            logParameter, ignoredUrlPrefixes, tgPidServerUrl -->
            <property>
                <field>getPids</field>
                <value>true</value>
            </property>
            <property>
                <field>amountOfPidsAtOnce</field>
                <value>23</value>
            </property>
            <property>
                <field>rewriteMetadataUris</field>
                <value>false</value>
            </property>
            <property>
                <field>rewriteAggregationUris</field>
                <value>false</value>
            </property>
            <property>
                <field>rewriteTeiUris</field>
                <value>true</value>
            </property>
            <property>
                <field>rewriteXsdUris</field>
                <value>true</value>
            </property>
            <property>
                <field>rewriteLinkEditorUris</field>
                <value>true</value>
            </property>
            <property>
                <field>pidResolverPrefix</field>
                <value>https://dev.textgridlab.org/</value>
            </property>
            <property>
                <field>removeOldPids</field>
                <value>true</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.SubmitFiles">
            <!-- Commonly defined values: uriPrefix, tgcrudServerUrl, rbacSessionId,
            projectId, metadataSuffix, createNewRevisions -->
            <property>
                <field>deleteIfIngested</field>
                <value>true</value>
```

```
            </property>
            <property>
                <field>storeResponseMetadata</field>
                <value>true</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.DeleteFiles">
            <!-- Commonly defined values: rbacSessionId, logParameter,
            tgsearchServiceUrl, tgcrudServiceUrl -->
            <property>
                <field>objectUri</field>
                <value>textgrid:12345.0</value>
            </property>
            <property>
                <field>dryrun</field>
                <value>false</value>
            </property>
        </class>
        <class name="actionmodule.textgrid.PublishFiles">
            <!-- Commonly defined values: rbacSessionId, logParameter,
            tgpublishServerUrl, sesameServerUrl -->
            <property>
                <field>objectUri</field>
                <value>textgrid:12345.0</value>
            </property>
            <property>
                <field>dryrun</field>
                <value>false</value>
            </property>
        </class>
    </modules>
</config>
```

9.3.4 Beispiel technischer Metadaten – extrahiert vom FITS auf dem TextGrid Entwicklungssystem

```
<?xml version="1.0" encoding="UTF-8"?>
<fits xmlns="http://hul.harvard.edu/ois/xml/ns/fits/fits_output"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://hul.harvard.edu/ois/xml/ns/fits/fits_output
    http://hul.harvard.edu/ois/xml/xsd/fits/fits_output.xsd"
    version="1.1.0" timestamp="1/25/18 4:48 PM">
    <identification>
```

```
        <identity format="Extensible Markup Language" mimetype="text/xml"
            toolname="FITS" toolversion="1.1.0">
            <tool toolname="Droid" toolversion="6.1.5"/>
            <tool toolname="Jhove" toolversion="1.16"/>
            <tool toolname="file utility" toolversion="5.14"/>
            <tool toolname="Exiftool" toolversion="10.00"/>
            <tool toolname="ffident" toolversion="0.2"/>
            <tool toolname="Tika" toolversion="1.10"/>
            <version toolname="Droid" toolversion="6.1.5">1.0</version>
            <externalIdentifier toolname="Droid" toolversion="6.1.5" type="puid">
                fmt/101
            </externalIdentifier>
        </identity>
    </identification>
    <fileinfo>
        <size toolname="Jhove" toolversion="1.16">581</size>
        <filepath toolname="OIS File Information" toolversion="0.2"
            status="SINGLE_RESULT">/media/isilon/textgrid-esx1/public/productive
            /pairtree_root/te/xt/gr/id/+2/vr/39/,0/textgrid+2vr39,0
        </filepath>
        <filename toolname="OIS File Information" toolversion="0.2"
            status="SINGLE_RESULT">textgrid+2vr39,0
        </filename>
        <md5checksum toolname="OIS File Information" toolversion="0.2"
            status="SINGLE_RESULT">9517157852e4fc5547d85981ac69e188</md5checksum>
        <fslastmodified toolname="OIS File Information" toolversion="0.2"
            status="SINGLE_RESULT">1516895306000</fslastmodified>
    </fileinfo>
    <filestatus>
        <well-formed toolname="Jhove" toolversion="1.16" status="SINGLE_RESULT">
            true
        </well-formed>
        <valid toolname="Jhove" toolversion="1.16" status="SINGLE_RESULT">
            false
        </valid>
        <message toolname="Jhove" toolversion="1.16" status="SINGLE_RESULT">
            cvc-elt.1: Cannot find the declaration of element 'rdf:RDF'.
            Line = 1, Column = 391
        </message>
    </filestatus>
    <metadata>
        <text>
            <charset toolname="Jhove" toolversion="1.16" status="SINGLE_RESULT">
                UTF-8
```

```
            </charset>
            <markupBasis toolname="Jhove" toolversion="1.16" status="CONFLICT">
                XML
            </markupBasis>
            <markupBasis toolname="Exiftool" toolversion="10.00" status="CONFLICT">
                XMP
            </markupBasis>
            <markupBasisVersion toolname="Jhove" toolversion="1.16"
                status="SINGLE_RESULT">1.0</markupBasisVersion>
            <standard>
                <textMD:textMD xmlns:textMD="info:lc/xmls/textMD-v3">
                    <textMD:character_info>
                        <textMD:charset>UTF-8</textMD:charset>
                    </textMD:character_info>
                    <textMD:markup_basis version="1.0">XML</textMD:markup_basis>
                </textMD:textMD>
            </standard>
        </text>
    </metadata>
    <statistics fitsExecutionTime="124">
        <tool toolname="MediaInfo" toolversion="0.7.75" status="did not run"/>
        <tool toolname="OIS Audio Information" toolversion="0.1"
            status="did not run"/>
        <tool toolname="ADL Tool" toolversion="0.1" status="did not run"/>
        <tool toolname="VTT Tool" toolversion="0.1" status="did not run"/>
        <tool toolname="Droid" toolversion="6.1.5" executionTime="15"/>
        <tool toolname="Jhove" toolversion="1.16" executionTime="71"/>
        <tool toolname="file utility" toolversion="5.14" executionTime="71"/>
        <tool toolname="Exiftool" toolversion="10.00" executionTime="118"/>
        <tool toolname="NLNZ Metadata Extractor" toolversion="3.6GA"
            status="did not run"/>
        <tool toolname="OIS File Information" toolversion="0.2" executionTime="16"/>
        <tool toolname="OIS XML Metadata" toolversion="0.2" status="did not run"/>
        <tool toolname="ffident" toolversion="0.2" executionTime="15"/>
        <tool toolname="Tika" toolversion="1.10" executionTime="16"/>
    </statistics>
</fits>
```

Abkürzungen

AIP Archival Information Package

APA American Psychological Association

API Application Programming Interface

ASCII American Standard Code for Information Interchange

BMBF Bundesministerium für Bildung und Forschung

CRL Council on Library Resources

CRUD Create / Retrieve / Update / Delete

CTS CoreTrustSeal

DARIAH Digital Research Infrastructure for the Arts and Humanities

DFG Deutsche Forschungsgemeinschaft

DIN Deutsches Institut für Normung

DIP Dissemination Information Package

DOI Digital Object Identifier

DSA Dataseal of Approval

DC Dublin Core

EDV Elektronische Datenverarbeitung

ePIC european Persistent Identifier Consortium

GDZ Göttinger Digitalisierungszentrum

GND Gemeinsame Normdatei

GUI Graphical User Interface

GBV Gemeinsamer Bibliotheksverbund

GUK Göttinger Universitätskatalog

GWDG Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen

HDC Humanities Data Center

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

ID Identifier

IMEX Import Export

ISO International Organization for Standardization

JPEG Joint Photographic Experts Group

koLibRI kopal Library for Retrieval and Ingest

kopal Kooperativer Aufbau eines Langzeitarchivs digitaler Informationen

LZA Langzeitarchivierung

OCLC Online Computer Library Center

ORCID Open Researcher and Contributor ID

PDF Portable Document Format

PID Persistent Identifier

PND Personennamendatei

SPARQL SPARQL Protocol And RDF Query Language

SQL Structured Query Language

RDF Resource Description Framework

RDLC Research Data Lifecycle

REST Representational State Transfer

SIP Submission Information Package

TBLE Text-Bild-Link-Editor

TEI Text Encoding Initiative

TextGridLab TextGrid Laboratory

TextGridRep TextGrid Repository

TIFF Tagged Image File Format

URI Uniform Resource Identifier

URL Uniform Resource Locator

URN Uniform Resource Name

VFU Virtuelle Forschungsumgebung

VM Virtuelle Maschine

XML Extensible Markup Language

XQuery XML Query Language

XSLT Extensible Stylesheet Language Transformation

Abbildungsverzeichnis

Literatur- und Quellenverzeichnis

Letztes Abrufdatum aller aufgeführten Internet-Dokumente ist der 19. Februar 2018.

Betz, Katrin (2015): *Ein virtuelles Bücherregal: Die Digitale Bibliothek im TextGrid Repository*. In: TextGrid: Von der Community – für die Community. Neuroth, Heike et al. (Hrsg.). Glückstadt: Verlag Werner Hülsbusch. S. 229–238. https://doi.org/10.3249/webdoc-3947

Brodhun, Maximilian; Stefan E. Funk; Roman Hausner; Mathias Göbel und Ubbo Veentjer (2013): *Dokumentation und Leitfaden für den TextGrid-Import. TextGrid-Report 4.4.1*. https://textgrid.de/fileadmin/user_upload/TextGrid_R4.4.1_FINAL.pdf

CoreTrustSeal (2018): *Core Trustworthy Data Repositories*. https://www.coretrustseal.org

Creative Commons (2018): *Mehr über die Lizenzen*. https://creativecommons.org/licenses

CRL – Center for Research Libraries (2018): *Enriching Research. Expanding Possibilities. Since 1949*. http://www.crl.edu

DARIAH-DE (2017a): *Repository*. https://de.dariah.eu/repository

DARIAH-DE (2017b): *Publikator*. https://de.dariah.eu/publikator

DARIAH-DE (2018a): *Forschungsdaten im Kontext von DARIAH-DE*. https://de.dariah.eu/weiterfuhrende-informationen

DARIAH-DE (2018b): *Digitale Forschungsinfrastruktur für die Geistes- und Kulturwissenschaften*. https://de.dariah.eu

DARIAH-DE (2018c): *Dienste und Werkzeuge*. https://de.dariah.eu/list-services

DARIAH-DE (2018d): *Authentifizierungs- und Autorisierungs-Infrastruktur*. https://wiki.de.dariah.eu/display/publicde/DARIAH+AAI+Documentation

DARIAH-DE (2018e): *Virtuelles Skriptorium St. Matthias*. https://de.dariah.eu/virtuelles-skriptorium

DARIAH-DE (2018f): *CollateX Service-Instanz*. http://collatex.dariah.eu/collatex/

DARIAH-EU (2018): *DARIAH – Digital Research Infrastructure for the Arts and Humanities*. https://dariah.eu

DataCite (2018a): *Locate, identify, and cite research data with the leading global provider of DOIs for research data*. https://datacite.org

DataCite (2018b): *Assign DOIs*. https://datacite.org/dois.html

DataCite (2018c): *Metadata Schema*. https://schema.datacite.org

DFG (2018): *Deutsche Forschungsgemeinschaft*. http://www.dfg.de

DFG-Viewer (2018): *Referenzimplementierung für die Digitalisierungsstandards der Deutschen Forschungsgemeinschaft (DFG)*. https://dfg-viewer.de

DIN 31644:2012-04 (2012): *Information und Dokumentation – Kriterien für vertrauenswürdige digitale Langzeitarchive*. https://www.beuth.de/de/norm/din-31644/147058907

dlina Workgroup (2018): *Digital Humanities and Literary Studies – Network Analysis of Dramatic Texts*. https://dlina.github.io

DP4lib (2018): *koLibRI – kopal Library for Retrieval and Ingest*. http://dp4lib.langzeitarchivierung.de/index_koLibRI.php.de

DSPACE (2018): *DuraSpace Web Properties Server*. https://dspace.org

Duden (2017): *Repositorium, das*. https://www.duden.de/rechtschreibung/Repositorium

Duden (2018a): *Publikation, die*. https://www.duden.de/rechtschreibung/Publikation

Duden (2018b): *Publizierung, die*. https://www.duden.de/rechtschreibung/Publizierung

Duden (2018c): *publizieren*. https://www.duden.de/rechtschreibung/publizieren

Eclipse (2018): *Rich Client Platform*. https://wiki.eclipse.org/Rich_Client_Platform

Engelhardt, Claudia; Stefan E. Funk und Ubbo Veentjer (2013): *Ein Forschungsdatenarchiv für die Geisteswissenschaften: Konzeptionelle Überlegungen und Qualifizierungsaspekte*. Unveröffentlichter Projektbericht, Technische Hochschule Köln. 2013.

ePIC PID Consortium (2018a): *Persistent Identifiers for eResearch*. http://www.pidconsortium.eu

ePIC PID Consortium (2018b): *Overview | ePIC API*. http://doc.pidconsortium.eu/guides/overview

Ewert, Gisela und Walther Umstätter (1997): *Lehrbuch der Bibliotheksverwaltung*. Stuttgart: Hirsemann.. Stuttgart: Hirsemann.

Fedora (2018): *Fedora Repository*. http://fedorarepository.org

Fischer, Frank; Dario Kampkaspar und Peer Trilcke (2015): *Digitale Netzwerkanalyse dramatischer Texte*. http://www.gcdh.de/dhd2015-fischer-kampkaspar-trilcke-netzwerkanalyse-slides.pdf

FITS (2018): *File Information Tool Set*. https://projects.iq.harvard.edu/fits

FITSServlet (2018): *A web application to expose FITS as a service*. https://github.com/harvard-lts/FITSservlet

forschungsdaten.org (2018): *Forschungsdaten*. http://www.forschungsdaten.org/index.php/Forschungsdaten

Funk, Stefan E. (2014): *Entwicklung eines digitalen Publikationsworkflows am Beispiel der virtuellen Forschungsumgebung TextGrid*. Unveröffentlichter Projektbericht, Technische Hochschule Köln. 2014.

Funk, Stefan E.; Ubbo Veentjer und Thorsten Vitt (2013): *Digitale Werkzeuge in den digitalen Geisteswissenschaften. Die Virtuelle Forschungsumgebung TextGrid – Status quo und neue Entwicklungen*. In: Evolution

der Informationsinfrastruktur. Neuroth, Heike et al. (Hrsg.). Glückstadt: Verlag Werner Hülsbusch. 2013. S. 277–300. https://doi.org/10.3249/webdoc-39006

Gantert, Klaus (2016): *Bibliothekarisches Grundwissen. 9., vollständig neu bearbeitete und erweiterte Auflage*. Berlin/Boston: De Gruyter Saur, 2016. 493 Seiten. ISBN 978-3-11-032145-6.

GND (2018): *Gemeinsame Normdatei*. http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html

Göbel, Mathias (2018a): *mobile-plays*. https://github.com/mathias-goebel/mobile-plays

Göbel, Mathias (2018b): *Fontane Notizbücher – SADE*. https://gitlab.gwdg.de/fontane-notizbuecher/SADE

Göttinger Digitalisierungszentrum (2018): *Ein Service der SUB Göttingen*. https://gdz.sub.uni-goettingen.de

Göttinger Universitätskatalog (2018): *Niedersächsische Staats- und Universitätsbibliothek Göttingen*. https://opac.sub.uni-goettingen.de

GWDG (2018): *Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen*. https://www.gwdg.de

Harvard Library (2018): *XML schema for FITS XML*. http://hul.harvard.edu/ois/xml/xsd/fits/fits_output.xsd

Herrmann, J. Berenike und Gerhard Lauer (2017): *Das „Was-bisher-geschah" von KOLIMO. Ein Update zum Korpus der literarischen Moderne*. In: Konferenzabstracts. DHd2017 Bern – Digitale Nachhaltigkeit. 13.-18. Februar 2017. Bern. 2017. S. 107–111. http://www.dhd2017.ch/wp-content/uploads/2017/03/Abstractband_def3_M%C3%A4rz.pdf

IDF – International DOI Foundation (2018): *Digital Object Identifier System*. https://www.doi.org

IIIF (2018): *International Image Interoperability Framework*. http://iiif.io

International Organization for Standardization (2012): *ISO 16363:2012: Audit and certification of trustworthy digital repositories*. https://www.iso.org/standard/56510.html

International Organization for Standardization (2017): *ISO 26324:2012: Information and documentation – Digital object identifier system*. https://www.iso.org/standard/43506.html

JHOVE (2018): *JSTOR/Harvard Object Validation Environment*. http://jhove.openpreservation.org

JHOVE2 (2018): *The Next-Generation Architecture for Format-Aware Characterization*. https://bitbucket.org/jhove2/main/wiki/Home

Kay, Alan und Adele Goldberg (1977): *Personal Dynamic Media*. In: Computer. 10, H. 3, S. 31–41. http://www.newmediareader.com/book_samples/nmr-26-kay.pdf

Kindling, Maxi und Peter Schirmbacher (2013): *„Die digitale Forschungswelt" als Gegenstand der Forschung*. In: Information. Wissenschaft & Praxis. 64, H. 2/3, S. 127–136. https://doi.org/10.1515/iwp-2013-0017

koLibRI (2017a): *kopal Library for Retrieval and Ingest. Git Projektarchiv*. https://projects.gwdg.de/projects/kolibri/repository

koLibRI (2017b): *kopal Library for Retrieval and Ingest. Git Projektarchiv. kolibri-addon-textgrid-import. Tag 6.7-0-SNAPSHOT*. https://projects.gwdg.de/projects/kolibri/repository/kolibri-addon-textgrid-import?rev=6.7.0-SNAPSHOT&tag=6.7.0-SNAPSHOT

koLibRI (2017c): *kopal Library for Retrieval and Ingest. Git Projektarchiv. kolibri-addon-textgrid-import. Release-Tag 6.4.0*. https://projects.gwdg.de/projects/kolibri/repository/revisions/6.4.0/kolibri-addon-textgrid-import

koLibRI (2017d): *kopal Library for Retrieval and Ingest. Git Projektarchiv. kolibri-tgpublish-service. Release-Tag 6.4.0*. https://projects.gwdg.de/projects/kolibri/repository/revisions/6.4.0/kolibri-tgpublish-service

koLibRI (2018a): *kopal Library for Retrieval and Ingest. Git Projektarchiv. kolibri-addon-textgrid-import*. https://projects.gwdg.de/projects/kolibri/repository/revisions/master/kolibri-addon-textgrid-import

koLibRI (2018b): *Action Module GetPicaDmdForPpnFromOpac*. https://projects.gwdg.de/projects/kolibri/repository/revisions/master/entry/kolibri-base/src/main/java/de/langzeitarchivierung/kolibri/actionmodule/sub/GetPicaDmdForPpnFromOpac.java

kopal (2018): *Kooperativer Aufbau eines Langzeitarchivs digitaler Informationen*. http://kopal.langzeitarchivierung.de

Lauer, Gerhard (2017): *KOLIMO. A corpus of Literary Modernism for comparative analysis*. https://kolimo.uni-goettingen.de/about.html

MEI (2018): *Music Encoding Initiative*. http://music-encoding.org

METS (2018): *Metadata Encoding & Transmission Standard*. https://www.loc.gov/standards/mets

mirador (2018): *Open-source, web based, multi-window image viewing platform with the ability to zoom, display, compare and annotate images from around the world*. http://projectmirador.org

Mittler, Elmar (2007): *Open Access zwischen E-Commerce und E-Science – Beobachtungen zu Entwicklung und Stand*. In: Zeitschrift für Bibliothekswesen und Bibliographie. 4-5S. 163–169. https://doi.org/10.18452/9343

Müller, Uwe; Frank Scholze; Ursula Arning; Dörte Bange; Daniel Beucke; Thomas Hartmann; Nikola Korb; Isabella Meinecke; Heinz Pampel; Jochen Schirrwagen; Thomas Severiens; Friedrich Summann; Marco Tullney; Paul Vierkant; Michaela Voigt und Nadine Walger (2016): *DINI-Zertifikat für Open-Access-Repositorien und -Publikationsdienste 2016 [Oktober 2016]*. Deutsche Initiative für Netzwerkinformation (DINI). https://doi.org/10.18452/1503

NARA Task Force on Digital Repository Certification (2007): *Trustworthy Repositories Audit & Certification: Criteria and Checklist (TRAC)*. http://www.crl.edu/PDF/trac.pdf

nestor – Kompetenznetzwerk Langzeitarchivierung (2008): *nestor-materialien 8: Kriterienkatalog vertrauenswürdige digitale Langzeitarchive – Version 2*. nestor-Arbeitsgruppe Vertrauenswürdige Archive – Zertifizierung. http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0008-2008021802

nestor – Kompetenznetzwerk Langzeitarchivierung (2012): *nestor-materialien 16: Referenzmodell für ein Offenes Archiv-Informations-System – Deutsche Übersetzung 2.0*. nestor-Arbeitsgruppe OAIS – Übersetzung / Terminologie. http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:0008-2013082706

OCLC – Online Computer Library Center (2018): *OCLC: Worldwide, member-driven library cooperative*. https://www.oclc.org

Open Access (2018a): *Der freie Zugang zu wissenschaftlicher Information*. https://open-access.net

Open Access (2018b): *Informationsplattform Open Access: Repositorien*. https://open-access.net/informationen-zu-open-access/repositorien

ORCID (2018): *Open Researcher and Contributor ID*. https://orcid.org

ORCID Member Support Center (2018): *Workflow: Repository Systems*. https://members.orcid.org/api/workflow/repository

Oßwald, Achim; Regine Scheffel und Heike Neuroth (2012): *Langzeitarchivierung von Forschungsdaten – Einführende Überlegungen*. In: Langzeitarchivierung von Forschungsdaten – Eine Bestandsaufnahme. Neuroth, Heike et al. (Hrsg.). Boizenburg: Verlag Werner Hülsbusch. 2012. S. 13–22. http://nbn-resolving.de/urn:nbn:de:0008-2012031401

Puhl, Johanna; Peter Andorfer; Mareike Höckendorff; Stefan Schmunk; Juliane Stiller und Klaus Thoden (2015): *Diskussion und Definition eines Research Data LifeCycle für die digitalen Geisteswissenschaften*. In: DARIAH-DE Working Papers. Göttingen: DARIAH-DE. 2015. http://nbn-resolving.de/urn:nbn:de:gbv:7-dariah-2015-4-4

puppet (2018): *Get on the shortest path to better software*. https://puppet.com

Radecke, Gabriele (2015): *Materialautopsie: Überlegungen zu einer notwendigen Methode bei der Herstellung von digitalen Editionen am Beispiel der Genetisch-kritischen und kommentierten Hybrid-Edition von Theodor Fontanes Notizbüchern*. In: TextGrid: Von der Community – für die Community. Neuroth, Heike et al. (Hrsg.). Glückstadt: Verlag Werner Hülsbusch. 2015. S. 39–56. https://doi.org/10.3249/webdoc-3947

Radecke, Gabriele; Mathias Göbel und Sybille Söring (2013): *Theodor Fontanes Notizbücher. Genetisch-kritische und kommentierte Hybrid-Edition, erstellt mit der Virtuellen Forschungsumgebung TextGrid*. In: Evolution der Informationsinfrastruktur. Neuroth, Heike et al. (Hrsg.). Glückstadt: Verlag Werner Hülsbusch. 2013. S. 85–106. https://doi.org/10.3249/webdoc-39006

Rat für Informations Infrastrukturen (2017): *RfII Empfehlungen 2017: Datenschutz und Forschungsdaten*. http://www.rfii.de/?wpdmdl=2249

Riehm, Ulrich; Knud Böhle und Bernd Wingert (2004): *Elektronisches Publizieren*. In: Grundlagen der praktischen Information und Dokumentation. Band 1: Handbuch zur Einführung in die Informationswissenschaft und -praxis. 5., völlig neu gefasste Ausgabe. Kuhlen, R.; Seeger, Th.; Strauch, D. (Hrsg.). München: Saur 2004. S. 549–559.

Schirmbacher, Peter und Uwe Müller (2009): *Das wissenschaftliche Publizieren – Stand und Perspektiven*. In: cms-journal. 32S. 7–12. http://nbn-resolving.de/urn:nbn:de:kobv:11-10098123

Schmunk, Stefan und Stefan E. Funk (2016): *Das DARIAH-DE- und das TextGrid-Repositorium: Geistes- und kulturwissenschaftliche Forschungsdaten persistent und referenzierbar langzeitspeichern*. In: Bibliothek – Forschung und Praxis. 40, H. 2, S. 213–221. https://doi.org/10.1515/bfp-2016-0020

Schwerpunktinitiative Digitale Information (2011): *Definition Virtuelle Forschungsumgebung*. https://www.allianzinitiative.de/fileadmin/user_upload/www.allianzinitiative.de/2011_VRE_Definition.pdf

Stäcker, Thomas (2013): *Wie schreibt man Digital Humanities richtig. Überlegungen zum wissenschaftlichen Publizieren im digitalen Zeitalter*. In: Bibliotheksdienst. 47, H. 1, S. 24–50. https://doi.org/10.1515/bd-2013-0005

Stock, Wolfgang G. (2010): *Was ist eine Publikation? Zum Problem der Einheitenbildung in der Wissenschaftsforschung*. In: Wissenschaft und Digitale Bibliothek: Wissenschaftsforschung Jahrbuch 1998, 2. Auflage 2010. Fuchs-Kittowski, Klaus et al. (Hrsg.). Berlin: Gesellschaft für Wissenschaftsforschung. S. 239–282. http://wissenschaftsforschung.de/JB98_239-282.pdf

SUB Göttingen (2018): *Niedersächsische Staats- und Universitätsbibliothek Göttingen*. https://www.sub.uni-goettingen.de

TEI (2018): *Text Encoding Initiative*. http://www.tei-c.org/index.xml

TextGrid (2017a): *Ein Projekt und seine Geschichte*. https://textgrid.de/projekt

TextGrid (2017b): *Metadaten-Schema*. https://textgridlab.org/schema/textgrid-metadata_2010.xsd

TextGrid (2018a): *Repository*. https://textgridrep.org

TextGrid (2018b): *Nachhaltigkeit*. https://textgrid.de/nachhaltigkeit

TextGrid (2018c): *Das TextGrid und das DARIAH-DE Repository*. https://wiki.de.dariah.eu/display/publicde/Das+DARIAH-DE+Repository+und+das+TextGrid+Repository

TextGrid Aggregator (2018): *API Dokumentation*. http://www.textgridlab.org/doc/services/submodules/aggregator/docs/index.html

TextGrid Common (2018): *HTTP Clients*. https://projects.gwdg.de/projects/common/repository/revisions/master/textgrid-clients

TextGrid CRUD (2017): *TG-crud API Dokumentation*. http://textgridlab.org/doc/services/submodules/tg-crud/service/tgcrud-webapp/docs/index.html#api-documentation

TextGrid Digitale Bibliothek (2016): *Die Digitale Bibliothek bei TextGrid*. https://textgrid.de/digitale-bibliothek

TextGrid Import (2017a): *TG-import Dokumentation*. http://textgridlab.org/doc/services/submodules/kolibri/kolibri-addon-textgrid-import/docs/index.html

TextGrid Import (2017b): *TG-import Dokumentation. Import-Policies*. https://textgridlab.org/doc/services/submodules/kolibri/kolibri-addon-textgrid-import/docs/import_and_configuration.html#editing-the-config-file

TextGrid Laboratory (2018): *Download und Installation*. https://textgrid.de/download

TextGrid OAI-PMH (2018): *TG-oaipmh Dokumentation*. http://dev.textgridlab.org/doc/services/submodules/oai-pmh/docs_tgrep/index.html

TextGrid Publish (2017): *TG-publish Dokumentation*. http://textgridlab.org/doc/services/submodules/kolibri/kolibri-tgpublish-service/docs/index.html

TextGrid Repository (2017): *Explore Repository*. https://textgridrep.org/repository.html

TextGrid Search (2017): *TG-search Dokumentation*. http://textgridlab.org/doc/services/submodules/tg-search/docs/index.html

TextGrid Terms of Use (2016): *Nutzungsordnung für TextGrid*. https://textgrid.de/terms-of-use

TextGrid Text-Bild-Link-Editor (2018): *TBLE – Text-Bild-Link-Editor*. https://wiki.de.dariah.eu/display/TextGrid/Text-Bild-Link-Editor

TextGrid Wiki (2018): *Revisionen verwenden*. https://wiki.de.dariah.eu/display/TextGrid/Using+Revisions

TextGrid XML-Editor (2018): *XML-Editor*. https://wiki.de.dariah.eu/display/TextGrid/XML-Editor

TextGrid – Presseinformation (2009): *Forschungsverbund TextGrid erwirbt geisteswissenschaftliche Textsammlung*. https://www.uni-goettingen.de/de/3240.html?cid=3426

TextGridLab Nutzerhandbuch 2.0 (2015a): *Import*. https://wiki.de.dariah.eu/pages/viewpage.action?pageId=40220393

TextGridLab Nutzerhandbuch 2.0 (2015b): *Publish*. https://wiki.de.dariah.eu/pages/viewpage.action?pageId=40220493

TextGridLab Nutzerhandbuch 2.0 (2015c): *Vorschau-Ansicht*. https://wiki.de.dariah.eu/display/TextGrid/Vorschau-Ansicht

TextGridLab Nutzerhandbuch 2.0 (2018a): *Publikationswerkzeug SADE*. https://wiki.de.dariah.eu/display/TextGrid/Publikationswerkzeug+SADE

TextGridLab Nutzerhandbuch 2.0 (2018b): *Navigator*. https://wiki.de.dariah.eu/pages/viewpage.action?pageId=40220331

TextGridLab Nutzerhandbuch 2.0 (2018c): *Suche*. https://wiki.de.dariah.eu/display/TextGrid/Suche

TextGridLab Nutzerhandbuch 2.0 (2018d): *TextGrid-Objekte*. https://wiki.de.dariah.eu/display/TextGrid/TextGrid+Objects

TG-crud (2018): *Git Projektarchiv. 8.0.7-SNAPSHOT*. https://projects.gwdg.de/projects/tg-crud/repository?rev=8.0.7-SNAPSHOT&branch=master&tag=8.0.7-SNAPSHOT

TG-pid (2018): *Git Projektarchiv. 2.4.0*. https://projects.gwdg.de/projects/tg-pid/repository?rev=2.4.0&branch=master&tag=2.4.0

The Consultative Committee for Space Data Systems (2012): *Recommendation for Space Data System Practices: REFERENCE MODEL FOR AN OPEN ARCHIVAL INFORMATION SYSTEM (OAIS)*. RECOMMENDED PRACTICE CCSDS 650.0-M-2. https://public.ccsds.org/pubs/650x0m2.pdf

The Interedition Development Group (2017): *CollateX*. https://collatex.net

Theodor Fontane: Notizbücher (2018): *Digitale genetisch-kritische und kommentierte Edition, Hrsg. von Gabriele Radecke*. https://fontane-nb.dariah.eu

Theodor Fontane: Notizbücher. (2018): *Über das Projekt*. https://fontane-nb.dariah.eu/content.html?id=ueber_das_projekt.md

Trilcke, Peer; Frank Fischer; Mathias Göbel; Dario Kampkaspar und Christopher Kittel (2016): *Dramen als ›Small Worlds‹? Netzwerkdaten zur Geschichte und Typologie deutschsprachiger Dramen 1730-1930*. https://dlina.github.io/presentations/2016-leipzig/#/

Vanscheidt, Philipp; Andrea Rapp und Danah Tonne (2012): *Storage Infrastructure of the Virtual Scriptorium St. Matthias*. In: Digital Humanities 2012. Jan Christoph Meister (Hrsg.). Hamburg. 2012. S. 529–532. http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/storage-infrastructure-of-the-virtual-scriptorium-st-matthias.1.html

Verbundzentrale des GBV (2018): *Gemeinsamer Bibliotheksverbunde – Verbundzentrale*. https://www.gbv.de

Virtuelles Skriptorium St. Matthias (2017): *Der mittelalterliche Bibliotheksbestand der Trierer Abtei St. Matthias digital im Netz*. http://stmatthias.uni-trier.de

Virtuelles Skriptorium St. Matthias (2018a): *Hilfe zur Suche – Hinweise zum Bestand*. http://www.stmatthias.uni-trier.de/index.php?l=n&s=hilfe

Virtuelles Skriptorium St. Matthias (2018b): *Bibliothek*. http://stmatthias.uni-trier.de/?l=n&s=bibliothek

Waters, Donald und John Garrett (1996): *Preserving Digital Information: Report of the Task Force on Archiving of Digital Information*. http://www.clir.org/pubs/reports/pub63

Wegstein, Werner; Andrea Rapp und Fotis Jannidis (2015): *TextGrid — Eine Geschichte*. In: TextGrid: Von der Community – für die Community. Neuroth, Heike et al. (Hrsg.). Glückstadt: Verlag Werner Hülsbusch. 2015. S. 23–35. https://doi.org/10.3249/webdoc-3947

Wicherts, Jelte M.; Denny Borsboom; Judith Kats und Dylan Molenaar (2006): *The Poor Availability of Psychological Research Data for Reanalysis*. In: American Psychologist. 61, H. 7, S. 726–728. https://doi.org/10.1037/0003-066X.61.7.726

Wikipedia (2017): *Wissenschaftliche Publikation*. https://de.wikipedia.org/wiki/Wissenschaftliche_Publikation

Wikipedia (2018a): *Publikation*. https://de.wikipedia.org/wiki/Publikation

Wikipedia (2018b): *Digitaldruck*. https://de.wikipedia.org/wiki/Digitaldruck

Wikipedia (2018c): *Open Access*. https://de.wikipedia.org/wiki/Open_Access

Wikipedia (2018d): *Offene Wissenschaft*. https://de.wikipedia.org/wiki/Offene_Wissenschaft

Wikipedia (2018e): *Uniform Resource Name*. https://de.wikipedia.org/wiki/Uniform_Resource_Name

Wikipedia (2018f): *Peer Review*. https://de.wikipedia.org/wiki/Peer-Review

Wikipedia (2018g): *Zitationsdatenbank*. https://de.wikipedia.org/wiki/Zitationsdatenbank

Wikipedia (2018h): *User Experience*. https://de.wikipedia.org/wiki/User_Experience

Wikipedia (2018i): *Grid-Computing*. https://de.wikipedia.org/wiki/Grid-Computing

Wikipedia (2018j): *D-Grid-Initiative*. https://de.wikipedia.org/wiki/D-Grid

XQuery (2018): *An XML Query Language*. https://www.w3.org/TR/xquery

XSLT (2018): *Extensible Stylesheet Language Transformation*. https://www.w3.org/TR/xslt

Zeno.org (2018): *Meine Bibliothek*. http://www.zeno.org

Zenodo (2018a): *Research. Shared*. https://zenodo.org

Zenodo (2018b): *About Zenodo*. http://about.zenodo.org

danke.

achim (mr malis!). ameli (beste freundin). asterix (mr bachelor). claudia (hauptleserin!). claudio (mr coretrustseal). […] (kaffee! kaffee! kaffee!). edwin (mein pa). emil (muss einfach mit hier rein :-). frank (mr espresso). heike (mentorin der ersten stunde!). idefix (hund von obelix). johannes (mr publikator). kathleen (ms california). levilai (lekker-galette). manu (ms welli). markus (mr ich-hab-was-vergessen). martin (mr köln). martina (mitstreiterin). mathias (mr blues). max (mr oaipmh). obelix (mr bachelor). paul (mr schiebetür). peter (mr korrekteur!). robert (meine-inspiration-der-informatik switzer!). robert (bester jahresabschluss ever). ronja (töchterchen!). roswitha (meine ma). stefan (lza-leser). stefan (mr testitest!). thorsten (mr LaTeX). tibor (herr der konzepte!). tiffy (abstract queen). tim (erster berater). toni (gastgeberin). troubadix (spielen-darfst-du). ubbo (mr seafile). wolfgang (mr lammkeule). und natürlich alle, die ich vergessen habe.

changelog.

2018/02/21 – printed and uploaded to moodle and added mr blues to »danke.«
2018/02/22 – formatting of »Ingest 1« corrected in »Importworkflows in TextGrid«.
2018/02/23 – added mr köln to »danke.«
2018/02/24 – added ms california to »danke.«
2018/02/24 – »Digitale Geisteswissenschaften« corrected in keywords.
2018/06/29 – removed fugu's address from title page.