Document Deposit Assistant (DDA)

Broker-Software zwischen Content-Lieferanten und Open-Access-Repositorien

Gerrit Hübbers, Jan Steinberg, Agathe Gebert und Stefan Jakowatz

Fehlende bzw. ungenügend konfigurierte Schnittstellen, mangelnde Interoperabilität zwischen Systemen sowie Formatvielfalt erschweren einen strukturierten Import von Publikationsdaten in Repositorien. Der DDA löst diese Problematik, indem er als eigenständige Webanwendung zwischen Content-Providern und Ziel-Repositorien vermittelt. Seine Datenverarbeitungs-Pipeline bezieht Daten aus Quellsystemen oder per manuellem Dateiupload, transformiert diese Daten entsprechend der Konventionen des Repositoriums und spielt sie dort ein. Zwar löst der DDA damit nicht die langfristig notwendige Standardisierung von Formaten, aber er stellt kurz- bis mittelfristig eine große Erleichterung beim Import großer Datenmengen in Repositorien dar und leistet der Open-Access-Verfügbarkeit von Forschungspublikationen Vorschub. Der Einsatz dieser Softwareapplikation ist in unterschiedlichen Kontexten denkbar. Dementsprechend ist eine Weiterentwicklung des DDA mit unterschiedlichen Partnern avisiert.

Missing or insufficiently configured interfaces, lacking interoperability between systems, as well as format variety complicate a structured import of publication data into repositories. DDA provides a solution to these problems by acting as a stand-alone web application between content providers and target repositories. DDA's data processing pipeline collects data from source systems or via manual file upload, transforms this data according to the target repository's conventions, and uploads it into the repository. Even though DDA does not provide a solution to the format standardization required in the long term, it nevertheless significantly eases large data imports into repositories in the short and medium term and thereby accelerates open access availability of research publications. As this software is also useful in different contexts, DDA's ongoing development is planned with further partners.

1. Open-Access-Content-Akquise der Repositorien: Das Problem mangelnder Interoperabilität und heterogener Formate – ein Lösungsansatz

Die Anforderungen an Forschungs- und Infrastruktureinrichtungen wachsen mit der zunehmenden Verfügbarmachung wissenschaftlicher Publikationen im Open Access ganz erheblich. Diesen neuen Anforderungen haben sich Infrastruktureinrichtungen wie GESIS[1]-Leibniz-Institut für Sozialwissenschaften früh gestellt und bieten mit disziplinären Open-Access-Repositorien wie dem Social Science Open Access Repository (SSOAR)[2] geeignete Systeme zur (Selbst-) Archivierung von Volltexten an. Im Rahmen der Akquise arbeitet SSOAR mit unterschiedlichen Content-Providern aus der Community, insbesondere mit kleinen und mittelständischen Verlagen, Forschungseinrichtungen und redaktionellen Herausgebern zusammen.[3] Der Vorteil dieser Zusammenarbeit besteht darin, dass auf diesem Wege nicht nur Einzelexemplare, sondern im besten Fall vollständige Reihen- und Zeitschriftenjahrgänge bereitgestellt werden können. Die Nachfrage von Periodika, zur Erhöhung ihrer Sichtbarkeit in Repositorien nachgewiesen zu werden, steigt in den letzten Jahren kontinuierlich. Darüber hinaus wurden vor dem Hintergrund des 2014 in Kraft getretenen Zweitveröffentlichungsrechts (ZVR) Vereinbarungen mit Bibliotheken zahlreicher sozialwissenschaftlicher Einrichtungen dahingehend getroffen, dass sie die nach §38, 4 UrhG[4] zweitveröffentlichungsfähigen Publikationen ihrer Mitarbeitenden sammeln und en gros nach SSOAR abliefern. Über diese Vereinbarungen können Fachrepositorien ihren Communities zusätzlich zu den umfassenden Zeitschriftenbeständen zunehmend neueste Forschungsliteratur im freien Zugriff zur Verfügung stellen. Der Vorteil dieser strategischen Kooperationen wird jedoch erheblich in Frage gestellt, wenn sich in der Praxis die strukturierte Integration von Metadaten und Volltexten in großen Mengen in die jeweiligen Nachweissysteme und Repositorien oftmals als undurchführbares Unterfangen darstellt.

Im Mittelpunkt gegenwärtiger Herausforderungen stehen die mangelnde Interoperabilität von Datenbanken und die unzureichende Kompatibilität von Metadatenformaten bei der Akquise und Verarbeitung von digitalen Volltexten und Metadaten zur Übernahme in vorhandene Nachweissysteme. Grund dafür sind wesentliche Hindernisse beim Datenexport bzw. -import zwischen Content-Providern einerseits und akquirierenden Open-Access-Repositorien andererseits. Fehlende bzw. nicht ausreichend konfigurierte Schnittstellen, die mangelnde Interoperabilität der

1 https://www.gesis.org/home/

2 https://www.gesis.org/ssoar/home/

3 Vgl. Bambey, Doris/Gebert, Agathe: Open-Access-Kooperationen mit Verlagen – Zwischenbilanz eines Experiments im Bereich der Erziehungswissenschaft. b.i.t.online 13 (2010) 4 386-390. https://www.b-i-t-online.de/heft/2010-04-schwerpunkt4.pdf

4 https://www.gesetze-im-internet.de/urhg/38.html

Systeme, die Formatvielfalt sowie mangelnde technische Expertise zum Aufsetzen und Betreiben einer harvestbaren Infrastruktur unter den Content-Providern erschweren einen strukturierten Ex- und Import der Daten. In der Konsequenz erfolgt das Übertragen von Metadaten und Volltexten auf einer der beiden Seiten manuell – also mit kaum vertretbarem, hohem personellen und zeitlichen Aufwand.

Ein Lösungsansatz zur Bewältigung der genannten Herausforderungen besteht darin, eine Infrastruktur zu etablieren, die zwischen Content-Providern und Repositorien geschaltet ist und eine Art Broker-Funktion[5] einnimmt, bei der die verschiedenen Datenformate vereinheitlicht und große Mengen an Metadaten über funktionierende Schnittstellen in Open-Access-Repositorien importiert werden können.

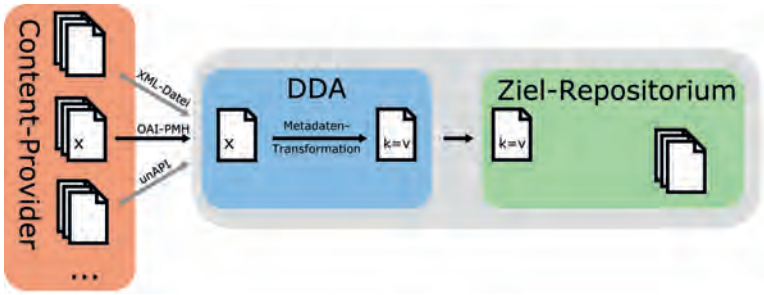

Abbildung 1: Konzeptionelle Übersicht des Brokers DDA zwischen Content-Providern und Ziel-Repositorium

Im Rahmen des von der DFG geförderten Projekts Document Deposit Assistant (DDA)[6] hat GESIS im Jahr 2015 mit dem Aufbau einer solchen Infrastruktur begonnen. Im Kontext der erfolgreichen Umsetzung des Projekts unterstützt der DDA seit 2016 den Datenimport nach SSOAR.

Dieser Beitrag zeichnet die Entwicklung des DDA von der Konzeption bis hin zu seiner Inbetriebnahme nach und benennt seine wesentlichen Funktionalitäten. Die Schwerpunkte liegen dabei auf der Darstellung des Softwareentwicklungsprozesses im Rahmen eines agilen Projektmanagements im SSOAR-Team bei GESIS sowie auf einer kritischen Einschätzung des Tools im Rahmen der Bestandserweiterung von Repositorien.

2. Konzeption eines Brokers zwischen Content-Providern und Repositorium

Die allgegenwärtige Praxis des Hochladens von Einzelpublikationen in ein Repositorium – sei es durch zuliefernde Einrichtungen oder durch die Mitarbeiterinnen und Mitarbeiter eines Repositoriums – lässt sich angesichts der zu verarbeitenden Datenmengen und vor dem Hintergrund eines effektiven Ressourcenmanagements nicht zielführend bewältigen.

Für die Anforderung, große Mengen an Metadaten und Volltexten in handhabbaren Formaten zur Verfügung zu haben, um sie über – idealerweise einheitliche – Web-Schnittstellen in die Informationsinfrastrukturen und Repositorien zu importieren, stehen seit langem Standard-Protokolle wie bspw. OAI-PMH[7], OAI-ORE[8] oder auch das SWORD-Protokoll[9] und REST-Schnittstellen[10] zur Verfügung. Diese sind auch in den gängigen OA-Repositorium-Softwarepaketen DSpace[11], Fedora[12], EPrints[13] etc. enthalten. Die Analyse des Open Access Census aus dem Jahr 2014[14] zeigt allerdings, dass allein das Vorhandensein der entsprechenden Schnittstellen noch nichts über die Qualität der übertragenen Metadaten aussagt. Der breit angelegten Analyse der deutschen Repositoriumslandschaft[15] zufolge sind bei vielen Repositorien zwar entsprechende Schnittstellen vorhanden, diese sind jedoch zumeist schlecht bis gar nicht konfiguriert. In vielen Fällen führt eine fehlende Unterstützung von Protokollen wie OAI-ORE oder SWORD sowie fehlende Compliance mit Metadaten-Qualitätsrichtlinien und -Standards wie den Empfehlungen der DINI (siehe DINI-Validator[16]), den DRIVER-Guidelines[17] oder den Richtlinien von OpenAIRE[18] zur Abwertung.[19] Vergleicht man die Situation der Repositoriums-Betreiber mit der Situation der Content-Provider (kleine und mittelständische Verlage, Forschungseinheiten, Herausgeber von Zeitschriften und Institutsreihen u.a.), so ist der Sachverhalt noch komplexer und technisch unausgereifter. Möchte ein Fachrepositorium wie SSOAR bereits strukturiert vorgehaltenen

5 Vgl. das Broker-Pattern: http://msdn.microsoft.com/en-us/library/ff648096.aspx

6 https://www.gesis.org/ssoar/home/kooperieren-mit-ssoar/projekte/

7 http://www.openarchives.org/pmh

8 http://openarchives.org/ore

9 http://swordapp.org/about

10 Beispielsweise ermöglicht DSpaces webbasierte REST-Schnittstelle, den Datenbestand des Repositoriums programmatisch abzufragen und zu verändern. https://wiki.duraspace.org/display/DSDOC6x/REST+API

11 https://duraspace.org/dspace/

12 https://duraspace.org/fedora/

13 http://www.eprints.org/uk/

14 Vierkant, Paul/Kindling, Maxi: Open-Access-Repositorien in den deutschen Bundesländern. Census on Open Access Repositories in Germany, Austria and Switzerland 2014. http://dx.doi.org/10.5281/zenodo.11608

15 http://repositoryranking.org

16 http://oanet.cms.hu-berlin.de/validator/pages/validation_dini.xhtml

17 https://wiki.surfnet.nl/display/standards/DRIVER+use+of+OAI-PMH

18 https://www.openaire.eu/guides/

19 Dies zeigt sich auch in den konkreten Zahlen des OA-Census: 99% der Repositorien unterstützen Simple Dublin Core, allerdings nur 4% Qualified Dublin Core. Andere Formate wie METS oder RDF werden ebenfalls nur von geringen 13% bzw. 7% unterstützt. Ein Harvesting von Simple Dublin Core erlaubt es aber nur, ein Minimal-Set an Metadaten zu übernehmen, da die Ausdruckskraft von Simple Dublin Core nicht ausreicht, um bspw. eine laufende Heftnummer oder ein Publikationsdatum eindeutig zu codieren.

Content nachnutzen, so findet sich dieser vorrangig in anderen Repositorien wie bspw. OAPEN[20], in institutionellen Repositorien und Systemen wie Bibsonomy[21] oder dem Datenshop der Deutschen Nationalbibliothek (DNB)[22]. Diese werden insbesondere durch kleinere Forschungseinheiten wie bspw. universitäre Sonderforschungsbereiche, das Nationale Bildungspanel (NEPS)[23] oder das Centrum für Hochschulentwicklung (CHE)[24] zur Archivierung genutzt.

Während OAPEN die Formate ONIX-XML, MARCXML, CSV sowie ein für den Import in Excel optimiertes XML ausgeben kann, speichert bspw. Bibsonomy die Publikationsdaten im BibTex-Format, die jedoch auch in den Formaten HTML, EndNote u.v.a.m. ausgeben werden können. Der Datenshop der DNB kann wiederum die Formate MARC21, MARC21-XML und RDF/XML ausliefern.

Anfragen zur Aufnahme ins Repositorium kommen zudem verstärkt von Verlagen und Redaktionen, die ihre Zeitschriften mit dem Publikations- und Verwaltungssystem Open Journal System (OJS)[25] im Netz veröffentlichen, das über eine Web-Schnittstelle für den Datenaustausch verfügt.[26]

Der überwiegende Teil der potentiellen Content-Provider hat seine Daten jedoch bestenfalls in Bibliotheks- und Katalogsystemen oder selbst programmierten, nicht standardkonformen Datenbanken ohne Web-Schnittstellen vorliegen, die die Metadaten lediglich als Datei-Exporte in Formaten wie PICA, RIS, ISBD und Excel ausgeben können. In vielen Fällen liegen die Daten und Volltexte (PDF) jedoch unstrukturiert auf den lokalen Systemen der Einrichtung vor.

Wie aufwändig, aber eben auch erfolgreich eine Infrastruktur ist, die zwischen Content-Providern und die Repositorien geschaltet wird, hat vor Jahren das von der Europäischen Union finanzierte Forschungsprojekt „Publishing and the Ecology of European Research (PEER)“[27] gezeigt. Die für dieses Projekt etablierte Infrastruktur sammelte Selbstarchivierungen und Verlagslieferungen der partizipierenden Zeitschriften großer STM-Verlage zentral in einem Repositorium, dem „PEER Depot“. Nach erfolgter Aufbereitung und Komplettierung der Metadaten wurde der Content entsprechend der mit dem Verlag vereinbarten Embargofrist und mit einer DOI ausgestattet über eine SWORD-Schnittstelle automatisiert an die jeweiligen nationalen Repositorien ausgeliefert.

20 http://www.oapen.org/home

21 https://www.bibsonomy.org

22 https://portal.dnb.de/metadataShop.htm

23 https://www.neps-data.de/

24 http://www.che.de/cms/?getObject=5&getLang=de

25 http://www.ojs-de.net

26 https://pkp.sfu.ca/wiki/index.php?title=OJS_Documentation

27 Ziel des von 2008-2012 laufenden PEER-Projekts war die Beforschung einer umfassenden Archivierung von zur Veröffentlichung akzeptierten peer-reviewed Autorenfassungen in Repositorien. Zwölf große STM-Verlage wie SAGE, Elsevier und Springer, 241 Zeitschriften und 6 Repositorien – unter anderen auch SSOAR – waren in das Projekt eingebunden. Insgesamt wurden im Untersuchungszeitraum über 53.000 Manuskripte archiviert.

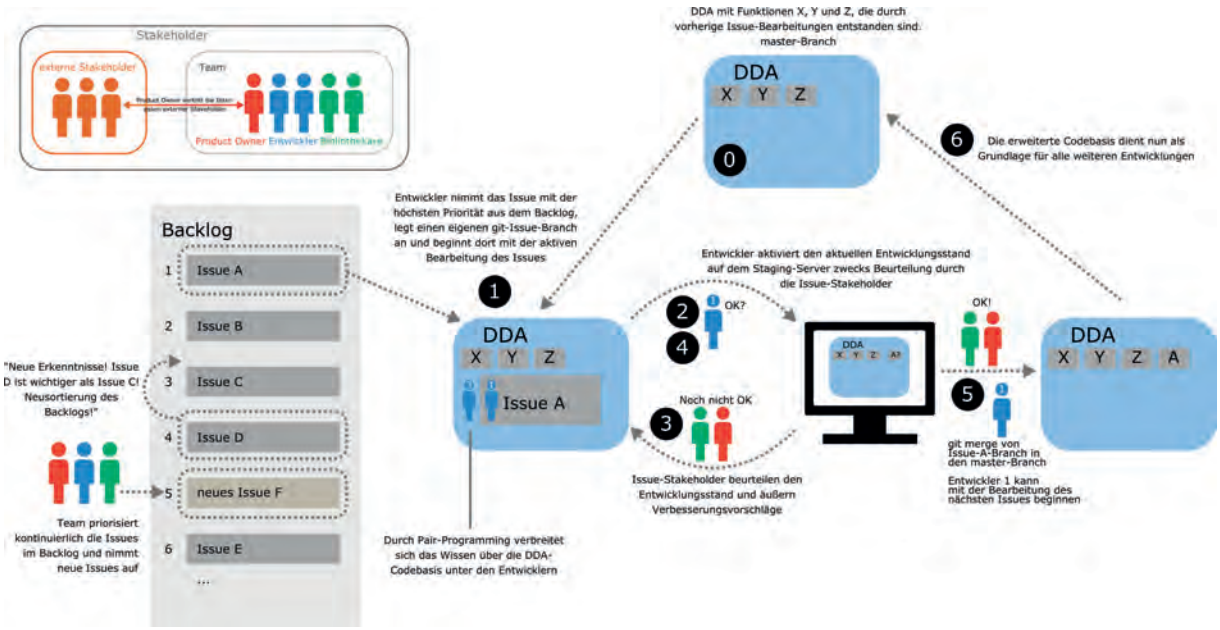

Abbildung 2: Der agile Entwicklungsprozess des DDA

Die SWORD-Schnittstellen der Version 1 und 2 zeichnen sich durch einen hohen Grad an Standardisierung aus.[28] Trotz ihrer schwergewichtigen Spezifikation fehlen diesen Versionen jedoch Funktionen, die für den zweckmäßigen Import in ein Repositorium notwendig sind. SWORD ermöglicht bspw. keine Dubletten-prüfung vor dem Import in ein Repositorium. Demgegenüber haben sich REST-Schnittstellen als leichtgewichtige Best Practices für den Datenaustausch zwischen Webdiensten etabliert.[29] Vor diesem Hintergrund wurde bei GESIS der DDA über eine erweiterte REST-Schnittstelle an das mit DSpace betriebene Repositorium SSOAR angebunden. In einer neuen Version 3[30] werden derzeit die skizzierten Mängel der SWORD-Schnittstelle aufgegriffen und optimiert, sodass eine zukünftige Anbindung des DDA auch über SWORD zweckmäßig wird.

3. Der agile Entwicklungsprozess des DDA

Zu Beginn der Entwicklung des DDA wurde ein pragmatischer Entwicklungsprozess definiert, um zeitnahe Umsetzungen von Anforderungen realisieren zu können. Für die kontinuierliche Optimierung dieser Umsetzungen wurden iterative Schleifen und die Erstellung einer ausführlichen Dokumentation etabliert, die spätere Nachnutzung sicherstellt. Der Entwicklungsprozess bedient sich an Konzepten des Scrum-Modells[31] und ermöglicht so eine agile[32] und feature-getriebene[33] Entwicklung.

Ein zentrales Element des Prozesses ist der intensive Austausch aller Projektbeteiligten, den sogenannten Stakeholdern[34]. Dazu gehören sowohl alle Personen, die an der Weiterentwicklung des Produktes interessiert sind als auch der Projektmanager/Product Owner als Vertreter von Verlagen und Institutionen, Nutzer (bspw. Bibliothekare) und Nachnutzer der Software (bspw. andere Repositorien) sowie die Softwareentwickler.

Über ein GitLab-basiertes Issue-Board[35] reichen die Stakeholder ihre Anliegen (Issues)[36] darüber ein, was die Software in einer zukünftigen Version leisten soll. Zu Beginn einer durchschnittlich zweiwöchigen Iteration wird diese Issue-To-Do-Liste (Backlog) im Rahmen eines Teammeetings priorisiert und abgearbeitet. Durch die Formulierung von und die Fokussierung auf möglichst konkrete Szenarien mit quantifizierbaren Ergebnissen, bspw. dem Harvesting aller aktuell verfügbaren Publikationen einer Datenquelle, wird der kostspieligen Entwicklung unnötiger Funktionen entgegengesteuert (YAGNI-Prinzip)[37].

Neue Software-Entwicklungen erfolgen in einem ei-

28 Daher wird die SWORD-Schnittstelle auch immer wieder (bspw. von DINI) empfohlen.

29 Vgl. Richardson, Leonard/Amundsen, Mike/Ruby, Sam: RESTful Web APIs, Sebastopol 2013 und Webber, Jim/Parastidis, Savas/Robinson, Ian: REST in Practice: Hypermedia and Systems Architecture, Sebastopol 2012. REST-Schnittstellen ermöglichen die wünschenswerte lose Kopplung verteilter Computersysteme mit allgemein akzeptierter und bekannter Semantik. Zudem ist REST programmiersprachen-unabhängig. Aus diesen Eigenschaften ergibt sich Flexibilität und Unabhängigkeit bei der Entwicklung alternativer Client- und Server-Komponenten.

30 http://swordapp.org/swordv3/

31 https://www.scrum.org/resources/what-is-scrum

32 https://de.wikipedia.org/wiki/Agile_Softwareentwicklung

33 https://de.wikipedia.org/wiki/Feature_Driven_Development

34 http://agiles-projektmanagement.org/scrum-stakeholder/

35 https://git.gesis.org/dda/dda-wizard/boards

36 Issues sind neue Features, bspw. die Anbindung einer noch nicht erschlossenen (Meta-)Datenschnittstelle oder Bugfixes, bspw. die Korrektur eines fehlerhaften Datenquellen-Harvestings.

37 https://de.wikipedia.org/wiki/YAGNI

genen git-Issue-/Feature-Branch[38]. Automatische Software-Tests (Unit- und Integrationstests) dienen als Sicherheitsnetz vor Software-Regressionen[39]. Beim Hochladen des aktuellen Stands der Entwicklung (git push) installiert eine Jenkins-Instanz[40] diese Software-Version im Rahmen einer „Build Automation mit Continuous Delivery" automatisch auf einem dedizierten Staging-Server[41]. Stakeholder können diese Software-Version nun testen und mitteilen, ob sie ihren Bedarfen entspricht oder nicht und Verbesserungsvorschläge einbringen. Jede Iteration endet mit einer Retrospektive, in der sich die Stakeholder gegenseitig über ihre Erkenntnisse der vergangenen Wochen austauschen und so voneinander über ihre Fachdomänen lernen. Direkt im Anschluss folgt die Planung der nächsten Iteration.

Ein Grund, weshalb sich bei den Stakeholdern eine hohe Zufriedenheit mit der agilen Arbeitsweise einstellt, ist die höhere Produktivität, welche ein Resultat der höheren Arbeitsqualität des Teams darstellt (bspw. durch Pair Programming, Code-Refactoring und automatisierten Tests[42]). Hinzu kommt, dass bestenfalls nur ein Issue pro Person aktiv bis zum Abschluss bearbeitet wird. So können die Entwickler ihre zugewiesenen Aufgaben schneller und störungsfreier ohne Kontextwechsel und Multitasking-Verluste abarbeiten. Einerseits bieten diese Eigenschaften zeitnah motivierende Erfolgserlebnisse, andererseits aber auch eine frühzeitige Möglichkeit zur Beurteilung und Kurskorrektur.[43]

4. Aufbau und Funktion des DDA

Als Startpunkt für die Entwicklung des DDA diente eine mit dem Scaffolding-Tool[44] JHipster[45] generierte Quelltext-Basis. JHipster erzeugt Quelltexte, die sich

38 https://en.wikipedia.org/wiki/Branching_(version_control)#Development_branch

39 Code-Änderungen können Fehler in zuvor korrekt funktionierenden Software-Komponenten verursachen. Solche Fehler nennt man Regressionen.

40 Jenkins ist eine Software, die wiederkehrende und zeitaufwändige Arbeitsschritte in der Software-Entwicklung automatisiert. Dazu gehören das Prüfen von Programmfehlern sowie das Kompilieren und Installieren auf dem Zielsystem (Continiuos Deployment) https://jenkins.io/

41 https://de.wikipedia.org/wiki/Bereitstellungsumgebung

42 Vgl. dazu Mah, Michael/ Lunt, Mike: How agile projects measure up, and what this means to you. Cutter Consortium 2008 (Agile Product & Project Management Executive Report Band 9).

43 Vgl. Cohn, Mike: Succeeding with Agile: Software Development using Scrum. Upper Saddle River, NJ 2010.

44 https://en.wikipedia.org/wiki/Scaffold_(programming)

45 https://www.jhipster.tech/

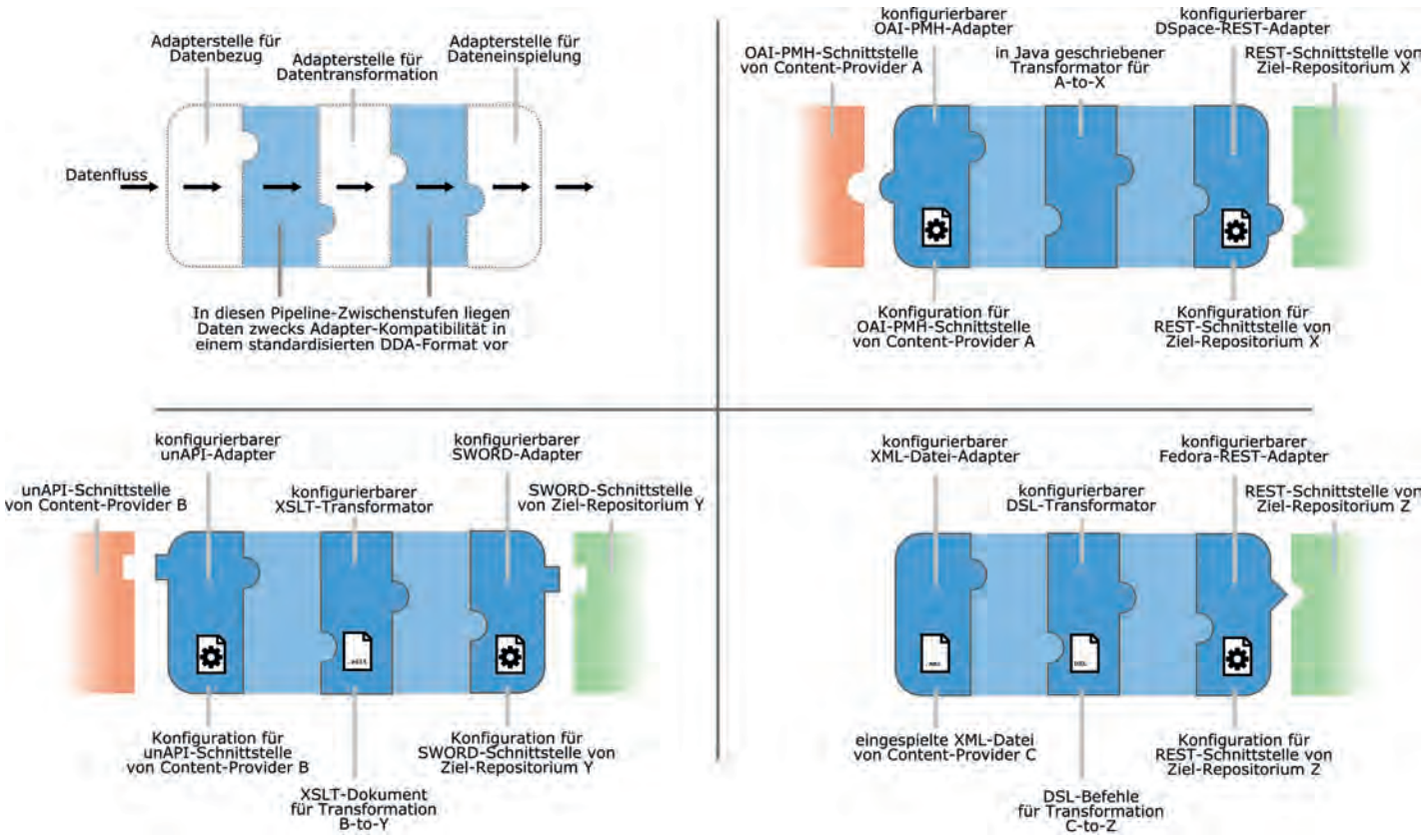

Abbildung 3: DDA ermöglicht durch seine Adapter-Architektur, Datenbezug, -transformation und -einspielung flexibel zu konfigurieren. Aufgrund der standardisierten Schnittstellen können Adapter beliebig miteinander kombiniert werden.

an aktuellen Best Practices der Webanwendung-Entwicklung orientieren: Serverseitig wird das Java-basierte Webframework Spring verwendet. Außerdem generiert JHipster eine Angular-basierte Single-Page-Webanwendung[46], die als Webbrowser-Benutzerschnittstelle dient und mit der serverseitigen Spring-Anwendung interagiert. Diese generische Software-Grundlage diente anschließend als Ausgangsbasis zur Erweiterung um die gewünschten Anforderungen an den DDA.

Der DDA fungiert als Mittler zwischen Content-Providern und dem Ziel-Repositorium und läuft als eigenständige Webanwendung unabhängig von diesen Systemen. Herzstück ist seine Datenverarbeitungs-Pipeline: Sie bezieht Daten aus Quellsystemen oder per manuellem Dateiupload, transformiert diese Daten entsprechend der Konventionen des Ziel-Repositoriums, um sie dort abschließend einzuspielen. Der Aufbau der Pipeline folgt damit einer dreistufigen Verarbeitungskette aus *Datenbezug* → *Datentransformation* → *Dateneinspielung*, die sich unter dem Namen ETL[47] in vielen anderen Softwaresystemen ebenfalls bewährt hat. Um den vielfältigen Standards und Schnittstellen der Open-Access-Landschaft zu begegnen, wurden die drei Pipeline-Stufen so entworfen, dass durch Entwicklung von Adaptern beliebige Quell- und Zielsysteme angesprochen werden können. Da der Datenaustausch zwischen den Pipeline-Stufen außerdem über definierte Schnittstellen erfolgt, können Adapter für den Datenbezug, die Transformation und den Repositoriums-Import vielfältig miteinander kombiniert werden.

Die Benutzer einer DDA-Instanz sind Mitarbeiterinnen und Mitarbeiter des Ziel-Repositoriums, die über das Fachwissen, welche Publikationen für das Repositorium geeignet sind und welche Metadaten in welcher Konvention eingespielt werden sollen, verfügen. Die DDA-Bedienung erfolgt über einen Webbrowser. Autorisierte Benutzer können hierüber neue Datenquellen hinzufügen, Metadaten-Transformationen für diese Datenquellen einrichten und aktualisieren, den aktuellen Harvesting-Stand begutachten und den Import von Publikationen in das Ziel-Repositorium initiieren. Neben Adaptern für XML-Dateien und populäre Web-Schnittstellen wie OAI-PMH existieren auch bereits Adapter für weniger gebräuchliche Datenquellen wie bspw. unAPI-Schnittstellen und Excel-Tabellen. Neben dem Bezug der Rohdaten erfüllen die Datenbezug-Adapter die Aufgabe, Metadaten in ein einheitliches Format zwecks standardisierter Weiterverarbeitung umzuwandeln.

46 https://en.wikipedia.org/wiki/Single-page_application

47 https://de.wikipedia.org/wiki/ETL-Prozess

Bsp.:

Für eine XML-Datei-Ablieferung mit dem Inhalt

```
<authors>
      <author>
           <last-name>Smith</last-name>
           <first-name>Jolene</first-name>
      </author>
</authors>
```

konvertiert DDAs XML-Datei-Datenbezug-Adapter den Metadatensatz in folgendes Format:

```
authors.author[0].last-name = "Smith"
authors.author[0].first-name = "Jolene"
```

Nachdem die Metadaten in das vom DDA verarbeitbare Format umgewandelt wurden, erfolgt durch den Einsatz von Transformationsregeln eine entsprechende Anpassung auf die Konventionen des Ziel-Repositoriums, wozu einmalig eine passende Transformationsregel pro Ablieferer hinterlegt wird. Um den Content-Providern unterstützend entgegenzukommen und die Massenablieferungen in Gang zu bringen, wurde bereits zu Beginn der Entwicklung des DDA entschieden, der Formatvielfalt individuell und mit flexibel konfigurierbaren Transformern zu begegnen. Eine positive Folge dieses Vorgehens ist, dass während der Programmierung individueller Transformationsregeln eine Software-Sammlung mit Regeln entsteht, auf die man bei der Erschließung weiterer Ablieferer-Quellen zurückgreifen kann. Mit diesen Transformationsregeln lassen sich bspw. Umformungen von Feldinhalten, Extraktionen von Details aus Zitationen, Abgleiche mit kontrollierten Vokabularlisten oder Harmonisierungen von ISBN- und ISSN-Nummern durchführen.

Um der Heterogenität von Metadatenformaten und Feldwerten zu begegnen, können Transformationsregeln auf verschiedene Weisen angelegt werden. So steht es frei, die Umformungen mit XSLT, Java oder einer eigenentwickelten, domänenspezifischen Sprache (*DSL*) für die Transformation von Metadaten zu formulieren. Beispiele für die Nutzung sowie Erläuterungen stehen für alle Varianten als dokumentierter Quelltext[48] zur Verfügung. Die Erfahrung zeigt, dass Mappings und Transformationen am effektivsten in iterativen Prozessen erarbeitet werden; dazu gehört dementsprechend auch die Möglichkeit umfassender intellektueller Tests im DDA.

Zusätzlich zu den Metadaten müssen auch die zugehörigen Volltexte ihren Weg ins Ziel-Repositorium finden. Dies geschieht über die in den Metadaten vorhandenen URLs oder Persistent Identifier (DOI, URN), die auf die entsprechenden Dokumente im Netz (HTTP, FTP) verweisen. In vielen Fällen führen diese Metadaten-Felder zuerst auf eine HTML-Landing-Page, weshalb eine Content-Resolver-Komponente[49] entwickelt wurde, die durch verschiedene Heuristiken und Strategien die Volltexte findet und diese dem zu importierenden Metadatensatz hinzufügt.

Der DDA ist in der Lage, die gewonnenen Publikationen einschließlich der Metadaten in verschiedene Repositorium-Systeme abzuliefern. Im SSOAR-Kontext entstand vor diesem Hintergrund ein Adapter zur Dateneinspielung in DSpace-5-Repositorien mit aktivierter REST-Schnittstelle. Darüber hinaus wird im Rahmen von Kooperationen mit weiteren Repositorien die Entwicklung von Adaptern zwecks Ablieferung insbesondere an Fachrepositorien angestrebt, die zumeist mit den Open-Source-Repositorien-Software DSpace, Fedora, OPUS[50], MyCoRe[51] oder EPrints betrieben werden. Zudem ist eine Anbindung über SWORD-Schnittstellen geplant.

Bei der Entwicklung des DDA wurde auf einen konsequenten Einsatz von frei verfügbaren Software-Bibliotheken geachtet. Da im Java-Umfeld keine geeignete Bibliothek zur Interaktion mit OAI-PMH-Schnittstellen existierte, entstand im DDA-Kontext der Open-Source-OAI-PMH-Harvester *ZOAI*[52]. Er kann als Java-Bibliothek in anderen Software-Projekten unabhängig vom DDA nachgenutzt werden. Weiterhin wurde die DSpace-5-REST-Schnittstelle um Endpunkte erweitert, um bspw. Suchanfragen nach Metadaten im Repositoriums-Bestand zu ermöglichen. Der DDA nutzt diese Funktion, um den Import von Dubletten zu vermeiden. Diese Erweiterung steht ebenfalls als einbindbare Software-Bibliothek für DSpace-5-Repositorien zur Verfügung[53] und kann so auch im Rahmen neuartiger REST-Client-Entwicklungen eingesetzt werden.

5. Kritische Einschätzung: Metadaten, Transformation und Standardisierung

Als eine Softwareentwicklung, die Metadaten und Volltexte harvesten, annehmen, konvertieren und ins Repositorium importieren kann, hat der DDA auf den einschlägigen Konferenzen[54] und bei den Kooperationspartnern große Aufmerksamkeit erhalten. Für

48 https://git.gesis.org/dda/dda-wizard/tree/master/src/test/java/org/gesis/dda/transformer/impl

49 https://git.gesis.org/dda/dda-wizard/blob/master/src/main/java/org/gesis/dda/publishing/domain/impl/ContentResolver.java

50 https://de.wikipedia.org/wiki/OPUS_(Dokumentenserver)

51 http://www.mycore.de/

52 https://git.gesis.org/dda/zoai

53 https://git.gesis.org/dspace/rest-additions

54 Open-Access-Tagen 2016 (Session 6, https://open-access.net/community/open-access-tage/open-access-tage-2016-muenchen/programm/) und 2017 (Tool-Marktplatz, https://open-access.net/community/open-access-tage/open-access-tage-2017-dresden/programm/toolmarktplatz/#c2940)

SSOAR wird der DDA als Infrastruktur für Massenablieferungen der Partner in das Fachrepositorium genutzt und befördert ganz entscheidend die Open-Access-Verfügbarmachung von Publikationsreihen, Mitarbeiterpublikationen und Zeitschriften. Die Anbindung des DDA sowohl an Content-Provider als auch an archivierende Repositorien wurde über unabhängige, modifizier- und erweiterbare Softwarekomponenten bzw. Adapter realisiert. Alle Softwareentwicklungen wurden ausführlich dokumentiert und stehen als Quelltext auf dem GitLab der GESIS unter https://git.gesis.org/dda/dda-wizard zur Nachnutzung zur Verfügung. Dadurch kann der DDA mit überschaubarem Aufwand an andere Systeme angebunden und in anderen Kontexten zur Datentransformation sowie für Datenexporte und -importe eingesetzt werden.

Eine bleibende Herausforderung stellt die Konvertierung der Daten in das für SSOAR notwendige Format deswegen dar, weil einzelne Content-Provider ihre Daten in zumeist sehr individuellen, von Standards abweichenden Formaten und heterogenen bibliographischen Ansetzungen übermitteln. Infolgedessen kann der Ablieferungsprozess nicht durchgängig automatisch vom Content-Provider durchgeführt werden, sodass das Fachpersonal des Ziel-Repositoriums tätig werden muss, um passende Transformationsregeln zu erstellen. Trotz dieser Einschränkungen stellt der DDA eine erhebliche Erleichterung für den Import umfangreicher Datenmengen in ein Repositorium wie SSOAR dar, zumal die angesprochenen Transformationsregeln für jeden Ablieferungspartner nur einmal entwickelt werden müssen; einmal angebunden werden Ablieferungen und Harvests vollautomatisch konvertiert und importiert. Darüber hinaus können die Transformationsregeln für die jeweiligen Datenlieferanten nicht nur hinterlegt, sondern für die Erstellung neuer Regeln nachgenutzt und in einer Bibliothek zusammengestellt werden, was die weitere Anbindung anderer Datenlieferanten deutlich erleichtert. Mit zunehmender Anzahl an Transformationsregeln in der Bibliothek sinkt die Wahrscheinlichkeit, dass Repositorien mit Metadatenformaten konfrontiert werden, für die bislang noch keine entsprechende Regel existiert. Gerade dieser doppelte Nutzungseffekt – Nachnutzung sowie Reduzierung der Heterogenität durch Erweiterung von Transformationsregeln – fand in der Repositoriums-Community eine außerordentlich positive Resonanz.

Die Anbindung einer lokalen Instanz des DDA an SSOAR zeigt beispielhaft, dass die administrativen Aufgaben eines Repositoriums beim Import umfangreicher Publikationsbestände wesentlich strukturierter ablaufen und dadurch der Arbeitsaufwand insgesamt minimiert wird. Der DDA optimiert diese Aufgaben, indem vormals wenig zusammenhängende und größtenteils manuelle Tätigkeiten in einem nunmehr integrierten Workflow entlang der dreistufigen Verarbeitungskette aus *Datenbezug* → *Datentransformation* → *Dateneinspielung* zusammengefasst und automatisiert werden. Dadurch entstehen systematisierte und deutlich schlankere Arbeitsprozesse. Letztlich wird durch die weitgehend automatisierten Verarbeitungsprozesse die Bereitstellung von gut erschlossenen und gut auffindbaren Open-Access-Publikationen ganz entscheidend beschleunigt.

Ungeachtet der großen Erleichterungen, die der DDA für die Integration von umfangreichen Datenbeständen und Volltexten in die Repositorien für einmal angebundene Content-Provider mitbringt, entsteht durch die Anbindung des DDA auch ein ambivalenter Nutzungsaspekt, der einer eingehenden Kritik unterzogen werden muss. Sowohl die Anbindung als auch die Pflege des DDA, insbesondere die kontinuierliche Hinterlegung von Datenmappings bzw. Transformationsregeln aufgrund immer neuer, zumeist proprietärer Formate, macht eine dauerhafte, professionelle informationstechnische Betreuung für Repositorien unabdingbar. Zwar erleichtert die stetig wachsende Bibliothek die Neuerstellung von Transformationsregeln, der Betrieb des DDA bleibt jedoch zumindest mittelfristig relativ aufwendig, nicht zuletzt deshalb, weil unter den Repositorien kein einheitlicher Metadatenstandard besteht.

Bilaterale Transfermodule[55], wie sie im DDA Anwendung finden, reduzieren die vorhandene Heterogenität von Metadatenformaten und setzen dabei im Wesentlichen die Prämisse von Konsistenzerhaltung und Interoperabilität um.[56] Alle Bemühungen in der Fachinformationswelt, eine weitgehende Konsistenz bzw. Standardisierung von Metadatenformaten herzustellen, werden durch die Dezentralisierung bei der Dokumenterstellung, -erschließung und -verteilung[57] konterkariert. Vor diesem Hintergrund wird das Problem mangelnder Durchsetzung von Standards durch bilaterale Transfermodule bzw. Broker wie dem DDA nicht grundsätzlich gelöst, sondern lediglich umgangen. Nichtsdestotrotz offenbart der DDA das Ausmaß der Datenvielfalt und bietet quantitative Hinweise für ge-

55 Auf Modellebene entspricht der DDA den Transfermodulen, wie sie bspw. bei der Behandlung semantischer Heterogenität entwickelt und eingesetzt wurden, vgl. Krause, Jürgen. Standardisierung von der Heterogenität her denken - zum Entwicklungsstand bilateraler Transferkomponenten für digitale Fachbibliotheken Bonn 2003 (Informationszentrum Sozialwissenschaften, IZ-Arbeitsbericht, 28). http://nbn-resolving.de/urn:nbn:de:0168-ssoar-50750-9

56 Vgl. Krause, Jürgen: Total Package Design für digitale Bibliotheken und Fachinformation, in: Hutzler, Evelinde (Hrsg.): Bibliotheken gestalten Zukunft : kooperative Wege zur digitalen Bibliothek. Dr. Friedrich Geißelmann zum 65. Geburtstag, Göttingen 2008, S. 185 ff.

57 Vgl. Krause: Standardisierung von der Heterogenität her denken, S. 7.

eignetere Formate des Datenbezugs. Die Konvertierungsregeln geben darüber hinaus wichtige Hinweise für eine notwendige Standardisierung von Metadaten. Bis solche langfristigen Harmonisierungen etabliert sind, sorgt der DDA in den belieferten Repositorien bereits jetzt für eine sehr gute Datenqualität.

6. Fazit und Ausblick: Weiterentwicklung und zukünftige Anwendungsszenarien

Parallel zu notwendigen Standardisierungen stellt der DDA kurz- bis mittelfristig eine große Erleichterung beim Import großer Datenmengen in die Repositorien dar. Gerade Fachrepositorien können dadurch die fachlich relevanten Forschungspublikationen ihrer angebundenen Kooperationspartner – insbesondere Mitarbeiterpublikationen und Publikationsreihen – im Open Access verfügbar machen. Daher nehmen die Betreiber von SSOAR die erfolgreiche Entwicklung des Prototypen zum Anlass, den DDA mit weiteren Partnern für konkrete Anwendungsszenarien weiterzuentwickeln. Während GESIS plant, alle weiteren Entwicklungen des DDA (von den Anbindungsadaptern bis hin zu den Transformationsregeln) auf dem GESIS-GitLab zu veröffentlichen, zu verwalten und zu koordinieren, wird der DDA bei den Kooperationspartnern als jeweils eigenständige lokale Applikation für den Betrieb adaptiert. Ein lokaler DDA erleichtert die Pflege und erhöht die Nachnutzbarkeit im individuellen Kontext.

Mit dem DDA soll vor allem der Nachweis von gut erschlossenen Open-Access-Volltexten nachhaltig unterstützt werden. Als potentielle Kooperationspartner für einen Roll-Out des DDA als jeweils neue Applikation nimmt das SSOAR-Team Repositorien und Projekte in den Blick, in denen große Datenmengen verarbeitet und in Repositorien archiviert werden müssen. Hinsichtlich der Einrichtung und der Inbetriebnahme des DDA müssen Kooperationspartner die damit verbundenen und zuvor skizzierten Aufgaben ausführen, insbesondere was die Hinterlegung von Transformationsregeln im DDA betrifft. Vor diesem Hintergrund bietet sich bspw. eine Zusammenarbeit mit Betreibern etablierter Fachrepositorien wie ZB MED – Informationszentrum Lebenswissenschaften[58] an, die über das Fachrepositorium Lebenswissenschaften (FRL)[59] die Open-Access-Publikationen von bislang über 150 relevanten Wissenschaftseinrichtungen im In- und Ausland nachweisen möchte.[60]

Eine weitere Überlegung für einen erfolgversprechenden Einsatz des DDA sieht das Team in einer Zusammenarbeit[61] mit dem von der DFG geförderten Projekt DeepGreen[62], das sich der Aufgabe stellt, „wissenschaftliche Veröffentlichungen, sofern lizenzrechtlich erlaubt, automatisiert nach Ablauf der Embargofristen [über Repositorien] Open Access verfügbar [zu] machen“[63]. Dazu werden über eine Datendrehscheibe, die Publikationsdaten teilnehmender Institute und Verlage vorhält, auf der Grundlage der Allianz- und Nationallizenzen und deren Open-Access-Klauseln[64] lizenzrechtlich zweitverwertbare Publikationen bei den Verlagen identifiziert, zusammen mit den Metadaten eingesammelt und an die entsprechenden Repositorien geliefert.[65] Der DDA könnte in diesem Rahmen als Broker dienen, über den Metadaten der Verlage und Institute auf die von den Repositorien benötigten Formate konvertiert werden.

58 https://www.zbmed.de/

59 https://www.publisso.de/open-access-publizieren/repositorien/fachrepositorium-lebenswissenschaften/

60 Zum Zeitpunkt der Entstehung dieses Artikels verfolgen ZB MED und GESIS konkrete Kooperationsabsichten, die momentan noch keiner formalen Vereinbarung folgen.

61 Auch hier gibt es zum Zeitpunkt der Entstehung dieses Artikels Kooperationsgespräche, jedoch noch ohne formale Ergebnisse.

62 https://deepgreen.kobv.de/de/deepgreen/

63 Ebda.

64 https://www.nationallizenzen.de/, https://www.nationallizenzen.de/ueber-nationallizenzen/allianz-lizenzen-2011-ff.

65 Aufgrund der ausgehandelten Bestimmungen der Allianz- und Nationallizenzen dürfen aufgrund der OA-Klausel zweitverwertbare Mitarbeiterpublikationen im institutseigenen Repositorium archiviert werden. Es wäre zu klären, unter welchen Voraussetzungen ein Nachweis auch in Fachrepositorien möglich ist.

Alle AutorInnen sind
Wissenschaftliche MitarbeiterInnen bei:
GESIS-Leibniz-Institut für Sozialwissenschaften
Team Open Access, Abt. Wissenstransfer
Unter Sachsenhausen 6-8, 50667 Köln

Dipl.-Ing. Gerrit Hübbers
Chef-Entwickler des DDA
gerrit.huebbers@gesis.org

Dipl.-Bibl. Jan Steinberg (M.A. LIS)
Wissenschaftlicher Mitarbeiter und Softwareentwickler
jan.steinberg@gesis.org

Dr. Agathe Gebert
Leitung des Teams. Seit 2009 mit dem Aufbau von Open-Access-Repositorien pedocs (DIPF) und SSOAR (GESIS) beschäftigt.
agathe.gebert@gesis.org

Dipl.-Soz. Stefan Jakowatz
Wissenschaftlicher Mitarbeiter
stefan.jakowatz@gesis.org