

SADE-Dokumentation und Referenzinstanz

Publikationsframework für TEI-Daten

Version 09.09.2014

Arbeitspaket 4.4

Verantwortlicher Partner: SUB Göttingen

TextGrid

Virtuelle Forschungsumgebung in den Geisteswissenschaften

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Projekt: TextGrid – Institutionalisierung einer Virtuellen
Forschungsumgebung in den Geisteswissenschaften

BMBF Förderkennzeichen: 01UG1203A

Laufzeit: Juni 2012 bis Mai 2015

Dokumentstatus: final

Verfügbarkeit: öffentlich

Autoren: Göbel, Mathias; SUB
Veentjer, Ubbo; SUB

INHALT

1	Entwicklungshistorie.....	1
2	SADE und TextGrid.....	1
2.1	Umgebung (Technical Requirements).....	2
3	Allgemeine Komponenten.....	2
3.1	eXist-db	2
3.1.1	Template Engine.....	2
3.1.2	SADE.....	3
3.2	Digilib.....	3
4	TextGrid-spezifische Komponenten	4
4.1	tg-client.....	4
4.2	tg-connect	5
4.3	tg-menu.....	5
4.4	digilib-Proxy.....	5
4.5	SADE Module	6
4.5.1	Navigation	6
4.5.2	Viewer	7
4.5.3	Facettensuche	7
5	Installation	8
5.1	Server.....	8
5.1.1	Download	8
5.1.2	Starten.....	8
5.2	TextGrid Laboratory Plugin (Client).....	9
5.2.1	Konfiguration des Plugins	9
6	TextGrid SADE Publish	10
7	Publizieren von Objekten und Aggregationen aus dem TextGrid Lab	10
8	Konfiguration der SADE-Instanz	12
8.1	Config.xml.....	12
8.2	Menüführung	13

8.2.1	Menü oben.....	13
8.2.2	Datenstruktur (Sidebar).....	13
9	Zukünftige Anbindungsmöglichkeit.....	13
9.1	DARIAH-eXist-Hosting.....	13
10	Referenzinstanz	14

1 Entwicklungshistorie

SADE steht als Akronym für "Skalierbare Architektur für digitale Editionen" und ist ein Programmpaket, welches viele Komponenten umfasst, die zum Entwickeln und Darstellen einer digitalen Edition besonders geeignet sind. Der Fokus liegt dabei auf der Präsentation von XML-Dokumenten, die den Richtlinien der Text Encoding Initiative (TEI) folgen. Ursprünglich an der Berlin-Brandenburgischen Akademie der Wissenschaften (BBAW) im Rahmen der Digitalisierungsinitiative "The Electronic Life Of The Academy" (TELOTA) entwickelt, wird diese Software seit 2012 von einer größeren Gemeinde weiterentwickelt. An dieser beteiligen sich unter anderen die BBAW, die Österreichische Akademie der Wissenschaften (ÖAW; im Rahmen des Projektes CLARIN: Common Language Resources and Technology Infrastructure), das Max-Planck-Institut für Wissenschaftsgeschichte (MPI-WG), das Cologne Center for eHumanities (CCeH) und die Niedersächsische Staats- und Universitätsbibliothek Göttingen (SUB; im Rahmen von TextGrid).

Ferner bezeichnet sich eine Programmkomponente innerhalb des Paketes als SADE. Diese Komponente steuert den gesamten Output. Es handelt sich dabei um eine Anwendung, die in XQuery geschrieben ist und in der XML-Datenbank aktiv ist. Demnach ist die XML-Datenbank "eXist-db" auch wichtigster Bestandteil des Programmpaketes.

2 SADE und TextGrid

Mittels SADE soll eine Publikationsplattform geschaffen werden, die es den mit TextGrid arbeitenden Projekten ermöglicht, eigene Webseiten mit ihren erarbeiteten Daten zu füllen und diese auch selbst gestalten zu können. Zudem soll in diesem möglichst generisch funktionierenden System auch bereits eine Suchfunktion und Menüführung zur Verfügung stehen.

Zur Integration in das TextGrid Laboratory existiert ein Plug-in, welches Nutzer über den Marketplace in ihre Labinstallation integrieren können. Dieses Plug-in stellt die Verbindung zu einer SADE-Instanz her, die als Publikationsplattform dient und zu Testzwecken auf dem eigenen Computer laufen kann, aber auch bereits auf einem im Internet zugänglichen Server installiert sein kann. Beide Programmteile, Plug-in und Serverkomponente, sind Teil der Entwicklung, die hier beschrieben und dokumentiert wird.

2.1 Umgebung (Technical Requirements)

Das gesamte Programmpaket funktioniert als Webapplikation, die ihren eigenen Servlet-Container (Jetty) mitbringt. Damit reduzieren sich die technischen Anforderungen auf ein x86-System mit Java-Laufzeitumgebung (JRE).

3 Allgemeine Komponenten

3.1 eXist-db

EXist ist eine XML-Datenbank, das heißt es können XML-Dokumente abgelegt und mit XML-Technologien wie XQuery und XSLT abgefragt und weiterverarbeitet werden. Bei größeren Datenmengen lässt sich die Suche durch das Anlegen eines Index beschleunigen. Die aktuelle Entwicklerversion von eXist bietet einen neugestalteten Range-Index. Mit diesem lassen sich schnellere Suchen in einzelnen XML-Feldern oder Attributen realisieren. Von dieser Entwicklung kann insbesondere die facettierte Suche profitieren, deshalb wurde in diesem Projekt die bei GitHub im Quelltext vorliegende aktuelle Entwicklerversion von eXist genutzt. Eine der kommenden SADE-Versionen wird dann auf der Version 2.2 basieren und sich laufenden Aktualisierungen der stabilen Releases annehmen. EXist bringt nicht nur die Datenbank und die Unterstützung der XML-Technologien mit, es wurden auch bereits einige relevante Programme geschrieben, deren Integration sinnvoll ist. Diese sind zumeist über ein eigenes öffentliches Repository verfügbar. Ein dort zur Verfügung gestellter Beispieldatensatz umfasst die Werke William Shakespeares und zeigt schon exemplarisch die Nutzung mit TEI-Dokumenten.

3.1.1 Template Engine

Die Template Engine von SADE ist ein Fork einer frühen Version der eXist Template Engine. Teilweise wurden Änderungen der Template Engine aus neueren eXist Versionen in die SADE Template Engine zurückportiert. Die Template Engine eröffnet die Möglichkeit, SADE Module zu schreiben die unabhängig von CSS und Webseitendesign mit verschiedenen Darstellungen funktionieren. So gibt es z.B. für die facettierte Suche ein Bootstrap Template, es wären aber auch Templates mit anderen CSS Frameworks möglich, die Funktionalität des Moduls bleibt in allen Darstellungen gleich. Zur Demonstration wurde ein professionelles Beispieltemplate in die Referenzinstanz integriert.

3.1.2 SADE

SADE als Software innerhalb der eXist-db nutzt diese Template Engine und XAR-Module. XAR Module sind Pakete, in denen XML-Ressourcen wie XQuery-Skripte, XSLT-Stylesheets und XML zusammengefasst werden und sich so gebündelt als Apps oder zur Funktionalitätserweiterung in eXist installieren lassen. Das in eXist genutzte XAR-Format ist eine modifizierte Version des eXPath package-Formates. Die Distribution dieser Pakete kann als Einzel-Upload oder über Paket-Repositories erfolgen.

Die Nutzung der eXist Template Engine stellt die Unabhängigkeit solcher XAR-Module vom Layout der dargestellten Webseite sicher. In einer Konfigurationsdatei lassen sich Module, Templates und Pfade zu XML-Daten festlegen und so eigene Portale mit den jeweils benötigten Komponenten und einem individuellem Design zusammenstellen. Damit bleiben auch innerhalb der Datenbank Daten, Datenverarbeitung und Visualisierung voneinander getrennt.

Komponenten, die in digitalen Editionen in der Regel benötigt werden - zum Beispiel die Nutzung von TEI-Stylesheets, eine Suche und die Navigation - gehören zum Kern des Projektes und können so gemeinsam weiterentwickelt werden. Der Austausch eigener Entwicklungen, wie beispielsweise eine Zeitleiste oder eine Landkarten-basierte Visualisierung, können als XAR-Module der Community zur Verfügung gestellt werden.

Im TextGrid-Kontext gibt es derzeit zwei Projekte, die frühere Versionen von SADE in modifizierter Form einsetzen: Blumenbach-Online und die digitale Edition der Notizbücher Theodor Fontanes.

3.2 Digilib

Digilib ist eine serverseitige Software zur Bereitstellung und Manipulation von Bildern. Mit dieser ist es unter anderem möglich, Bilder in unterschiedlichen Skalierungen und Formaten oder auch nur einzelne Bildausschnitte vom Server abzurufen. Es gibt bei digilib einen Service, der die Operationen auf den Bildern vornimmt und eine REST-API bereitstellt sowie einen webbasierten Client, der die Nutzerinteraktion mit den vom Service angebotenen Bildern und Bildbearbeitungsfunktionen erlaubt.

Digilib ist schon seit der zweiten Förderphase in TextGrid verfügbar, die Integration wurde 2013, im Rahmen eines von der Mellon-Foundation geförderten Projektes zur Integration von IIIF in TextGrid, stark verbessert, wodurch auch eine Optimierung der Geschwindigkeit erreicht werden konnte. Für im TextGrid gehostete Bilder kann in Zukunft der TextGrid-seitige digilib-Service

anstelle einer SADE-integrierten Version zu verwendet werden. Diese Lösung hat einige Vorteile: So können in TextGrid eingespielte Digitalisate direkt in dem Format, in dem sie den Scan-Prozess verlassen – dem TIFF-Format – abgelegt werden. Eine Vorhaltung der Bilder im JPEG-Format, wie es bei der SADE-integrierten Version für die Publikation unerlässlich wäre, ist nun nicht mehr nötig, es können automatisch vorskalierte und komprimierte Version mit digilib bereitgestellt werden. Die Bilder selbst brauchen bei der Publikation aus dem TextGridLab nicht zur SADE-Installation kopiert zu werden. Bei der Ausgabe von Dokumenten, die Bilder enthalten, werden die Objekte anhand der TextGrid-URI von digilib als JPEG in einer zweckmäßigen Auflösung (100px für Thumbnails, 1500px für die Ansicht von Seitenscans) abgerufen. So können in Zukunft Geschwindigkeitsoptimierungen zentral über den TextGrid-Service vorgenommen werden, die einzelnen SADE-Server werden von der Bildumrechnung entlastet. Die Unabhängigkeit von SADE von dem integrierten digilib vereinfacht die Nutzung des DARIAH-eXist-Hosting, da dadurch für eine funktionierende Installation nur eine eXist-Datenbank vorhanden sein muss.

4 TextGrid-spezifische Komponenten

Die nachfolgend aufgeführten Komponenten funktionieren als Module in der eXist-Datenbank, sie können genauso aber auch ohne die Hauptkomponente SADE genutzt werden. Weiterhin folgen noch die Erweiterungen von SADE selbst in einem anschließenden Abschnitt. Alle Quellcodes sind unter github.com/ubbo/ frei verfügbar.

4.1 tg-client

Die meisten Komponenten von TextGrid lassen sich per REST/SOAP-Schnittstelle oder Webinterface ansteuern. Das XQuery-Skript stellt all diese Funktionen in der Datenbank zur Verfügung. So kann man aus allen anderen Modulen oder ggf. auch von der Weboberfläche aus auf Authentifizierungsdienste und mehr zugreifen. Es bietet derzeit Anknüpfungen an:

- die SPARQL-Schnittstelle von tg-search
- die von tg-crud bereitgestellten Metadaten-Objekte
- die von tg-crud bereitgestellten Daten-Objekte
- den Authentifizierungsdienst tg-auth (inkl. einer zwischengespeicherten Session-ID, um unnötige Last bei tg-auth zu vermeiden)

und es bietet zudem eine Funktion, um die URI-Prefixe "textgrid:" aus den Objektreferenzen zu entfernen. Damit stehen die Kernservices von TextGrid innerhalb der Datenbank zur Verfügung und können in andere Module eingebunden werden.

4.2 tg-connect

Dieses Skript wird bei der Dokumentenübertragung genutzt. Es ist für das Holen und Ablegen der Dokumente verantwortlich und bedient sich intensiv der von tg-client bereitgestellten Funktionen. Es greift dabei die aus dem TextGrid Laboratory vom Publish-Tool SADE (TextGridLab Plug-in) übertragenen Informationen ab und verarbeitet diese, überträgt also die Dokumente aus dem TextGrid Repository (public und/oder non-public) in diese SADE-Instanz. Abschließend wird das Skript zum Aufbau der Menüführung abgerufen.

4.3 tg-menu

Mit dem Modul tg-menu wird die aus dem TextGrid Laboratory innerhalb des Navigators zu sehende hierarchische Struktur aufgebaut. Diese wird dann als Menü in die digitale Präsentationsoberfläche eingebaut. Dabei wird weiterhin auf die Trennung von Daten und Layout geachtet. Primär wird ein XML-Dokument erzeugt, welches die hierarchische Struktur entsprechend des Datenmodells nachempfendet (wie es auch im Navigator innerhalb des Labs geschieht) und in der dem Projekt zugehörigen Kollektion abgelegt. Um schließlich konform mit dem jeweiligen Template, dem Layout der digitalen Edition, zu sein, ist ein weiterer Verarbeitungsschritt nötig: ein XSLT, welches in der Projekt-Kollektion abgelegt sein muss und den Namen des gewählten Templates tragen sollte, wird automatisch aufgerufen. Dadurch können flexible Layouts und Designs eingesetzt werden.

4.4 digilib-Proxy

Bei der ursprünglichen Version der TextGrid Sade-Publish-Komponente wurden die Bilder beim Publizieren im lokalen digilib-Verzeichnis der SADE-Installation abgelegt. Daher brauchte die Authentifizierung bei TextGrid nur beim Publizieren der Bilder vorgenommen werden. In Zukunft kann der TextGrid digilib-Service für die Bilddarstellung genutzt werden, was einer Authentifizierung bei jeder Anzeige von Bildern bedarf. Dies gilt für den Fall, dass ein Rechtemanagement für die Bildabfrage nötig ist, beispielsweise bei nur für die betreffende Edition freigegebenen Bilddokumenten. Daher wurde in XQuery ein Proxy implementiert, ein Skript, welches einen gültigen TextGrid-Account nutzen kann und diese Abfragen koordiniert. Anfragen an

digilib aus dem Portal werden an den Proxy gestellt, dieser meldet einen hinterlegten TextGrid Nutzeraccount am TextGrid Repository an, bezieht eine SessionID und damit die Berechtigung, das angeforderte Bild abrufen zu dürfen. Das Bild wird an den Endnutzer weitergegeben, die zugehörige Session-ID bleibt aber versteckt.

Exkurs: Funktionsaccounts

Schon jetzt werden von den beiden genannten Projekten sogenannte Funktionsaccounts benutzt, deren Daten in der Konfiguration von SADE angegeben werden müssen. Dabei handelt es sich um TextGrid-Accounts, denen in den betreffenden Projekten lediglich Beobachter-Status zukommt. Derzeit kann jeder TextGrid Nutzeraccount TextGrid-Projekte erstellen und Daten publizieren. Da für alle TextGrid Accounts ein realer Nutzer hinterlegt ist, ist hier die Haftung, Verantwortlichkeit etc. geklärt. Ein im SADE-Projektportal hinterlegter Nutzeraccount könnte aber potentiell allen Nutzern, die erweiterte Zugangsrechte innerhalb des Portals haben, bekannt werden. Hier bedarf es eines noch stärker beschränkten Accounttypes. Das sind Nutzeraccounts, die explizit nur Leserechte an Projekten, denen sie zugeordnet sind, haben können.

4.5 SADE Module

Es wurden drei neue Module für SADE programmiert: eines für die Navigation, ein Viewer und eine facettierte Suche. Alle drei Module nutzen die Template Engine.

4.5.1 Navigation

Um die Inhalte des SADE-Portals in einer eigenen Menüführung auffindbar zu machen, wurde das Navigationsmodul entwickelt. Mit diesem Modul ist es möglich, in einer XML-Datei die Menüstruktur des Portals festzulegen. Einzelne Links werden mit einem Namen und einem Verweis versehen und auch externe Links sind möglich. Für dieses Modul wurde eine Nutzerdokumentation erstellt. Die Portal-Navigation kann in einer einfachen XML-Struktur beschrieben werden.

Beispiel - XML Konfiguration und die resultierende Menüansicht (im Bootstrap Template):

```
<navigation>
  <item label="Text" link="content.html?id=text.md"/>
  <submenu label="Links">
    <item label="textgrid.de" link="http://textgrid.de"/>
  </submenu>
</navigation>
```

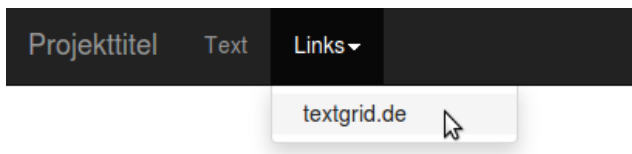


Abbildung 1: Mit dem Navigationsmodul erstelltes Menü

Derzeit können an dieser Stelle noch keine Untermenüs verschachtelt werden.

4.5.2 Viewer

Es wurde ein Viewer-Modul programmiert, welches die Datei-Endung prüft und zwischen HTML-, Markdown- und XML-Dokumenten unterscheidet. Für diese drei Dokumenttypen sind Anweisungen hinterlegt, die jeweils zu einer HTML-Ausgabe führen. Im Falle eines XML-Dokuments wird noch geprüft, ob es sich um ein mit dem in TextGrid integrierten Text-Bild-Link-Editor erstelltes Dokument handelt, oder eine andere Struktur vorliegt, bei der dann entsprechend des Namespaces ein passendes XSLT-Stylesheet ausgewählt wird. So ist es mit demselben Viewer beispielsweise möglich Markdown, TEI und Blumenbach-Objekt-XML darzustellen. Der Viewer unterstützt auch die seitenweise Ausgabe von TEI-Dokumenten, wobei keine im Dokument zuvor stehenden Elemente berücksichtigt werden und somit auch potentiell semantisch übergreifende leere Elemente vernachlässigt werden.

Die Markdown-Darstellung wurde integriert, um einzelne Portaltex te, die allerdings derzeit nicht durchsuchbar sind, schnell und einfach zu erstellen. Die Hilfe- und Tutorialdokumentationen der TextGrid-SADE-Referenzinstallation sind in Markdown verfasst.

4.5.3 Facettensuche

Für die Suche in publizierten XML-Daten wurde ein SADE-Modul geschrieben, welches die Suche durchführt und entsprechend der Konfiguration Facetten bildet. In der Konfiguration lassen sich für die einzelnen zu bildenden Facetten XPath-Ausdrücke angeben. Facetten lassen sich in der Webseite per Klick auf (-) aus dem Such-Ergebnis ausschließen. Mit Klick auf die Facette lässt sich das Ergebnis auf diese Facette einschränken. Das Modul ist derzeit via Eingabefeld in der Navigationsleiste anzusteuern.

5 Installation

5.1 Server

5.1.1 Download

Dieser Report bezieht sich auf die im TextGrid Wiki bereitgestellte SADE-Version latest-stable.tar.gz (SADE-Server).

Diese Software befindet sich noch in Entwicklung. Der tagesaktuelle *build* kann unter <http://dev2.dariah.eu/jenkins/view/TextGrid/job/SADE/lastSuccessfulBuild/artifact/dist-utils/antbuild/build/sade-textgrid.tar.gz> abgerufen werden. Windows-Nutzer benötigen ggf. ein Programm, um den komprimierten Ordner entpacken zu können.

5.1.2 Starten

Nach dem Download des komprimierten Archives muss dieses entpackt werden. Anschließend kann man im nun entpackten Verzeichnis den Ordner "bin" finden. Hier stehen für die verschiedenen Betriebssysteme Startprogramme bereit. Unter Windows öffnet man die Dateien mit der Erweiterung ".bat", unter Linux und Mac jene mit ".sh".

Dateiname	Funktion
backup	erstellt ein Backup der Datenbank
client	Start Script for eXist interactive client
run	mit Startoptionen für Linked Java Object Builder und YourKit Java Profiler mit Launcher und Tray-Icon
server	Start Script for Jetty + eXist
shutdown	Stop Jetty + eXist
startup	Start Script for Jetty + eXist mit Option "Java debugging via JDWP on port 4000 in Server mode" (auskommentiert) ohne Launcher und Tray-Icon

Bei Installationen mit graphischen Interface (zum Beispiel Windows oder Linux/Mac auf dem eigenen PC) ist "run.sh" (Linux/Mac) bzw. "run.bat" (Windows) zu bevorzugen, da es einen Launcher mitbringt, im Hintergrund startet und via Tray-Icon aufgerufen oder beendet werden kann.

Zur Konfiguration eines entfernten Servers bietet die Dokumentation von eXist eine Anleitung zum autostarten, zu finden unter <http://exist-db.org/exist/apps/doc/advanced-installation.xml>.

Standardmäßig läuft die Kommunikation zur Instanz über den Port 8080. Will man den für http und somit für Internetseiten gängigen Port 80 benutzen, muss man die Datei „conf.xml“ bearbeiten und dort den entsprechenden Eintrag vornehmen. Port 80 kann für eingehende Anfragen an Computer oft nur von einem privilegierten Nutzer geöffnet werden. Entsprechend muss das Programm als Administrator/Super User gestartet werden. Die folgend beschriebenen Konfigurationen müssen dann auch entsprechend angepasst werden.

5.2 TextGrid Laboratory Plugin (Client)

Die Installation eines Plugins wird im TextGrid Benutzerhandbuch eingehend beschrieben und ist abrufbar unter <https://dev2.dariah.eu/wiki/display/TextGrid/Installing+Additional+Tools>. In diesem Fall muss man das Plugin „SADE Publish Tool“ auswählen und installieren. Anschließend bekommt man ein neues Icon in der Symbolleiste angezeigt, im Kontextmenü der im Navigator angezeigten Objekte erscheint die Option „Publish to SADE“ und Nutzerinnen können in den Einstellungen die Konfiguration des Plugins vornehmen.

5.2.1 Konfiguration des Plugins

Um das Plugin nutzen zu können, muss ein Endpunkt in Form einer URL angegeben werden, die zu einem gestarteten SADE-Server führt. Im Fall einer lokalen Installation auf dem eigenen PC lautet diese URL: <http://localhost:8080/exist/apps/textgrid-connect/publish/textgrid/>
Als „Authorized User“ gibt man „admin“ an und das Passwortfeld kann in der Standardkonfiguration leer bleiben.

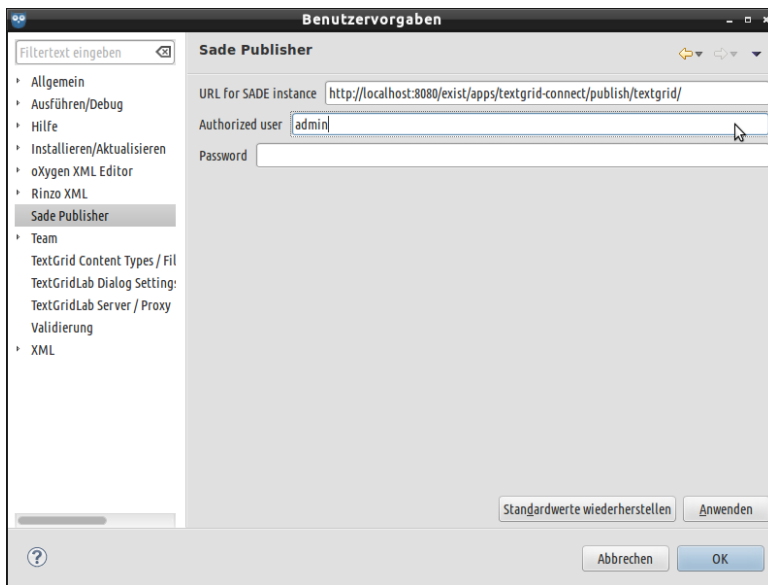


Abbildung 2: Konfiguration des SADE Publish-Tools

6 TextGrid SADE Publish

TextGrid bietet zur besseren Integration von SADE ein eigenes SADE-seitiges Publikationsmodul. Dieses ist in JavaScript programmiert und wird im integrierten Webbrowser des TextGridLabs angezeigt. Es dient dazu, im Navigator des TextGridLabs ausgewählte Objekte zur SADE Installation zu übertragen. Dieses ursprünglich in Java implementierte Modul wurde in XQuery neu geschrieben. Die Verfügbarkeit dieses Moduls in XQuery hat den Vorteil, dass es auch ohne Admin-Rechte auf dem Server direkt in ein beliebiges eXist als XAR-Paket deployed werden kann, was unter anderem bei der Nutzung des DARIAH-eXist-Hostings von Vorteil ist. Obendrein besteht so ein einfacher Zugriff auf die Nutzerauthentifizierung von eXist. Im Zuge der Neu-Implementierung wurde die Oberfläche des Moduls mit Bootstrap neu gestaltet.

7 Publizieren von Objekten und Aggregationen aus dem TextGrid Lab

Sind Server und Plugin installiert, kann mit der Publikation von Dokumenten, Bilder und sonstigen Daten begonnen werden. Man startet das im Laboratory konfigurierte Plugin über das Menü „Werkzeuge“ und den dort gelisteten Punkt „SADE Publish“.

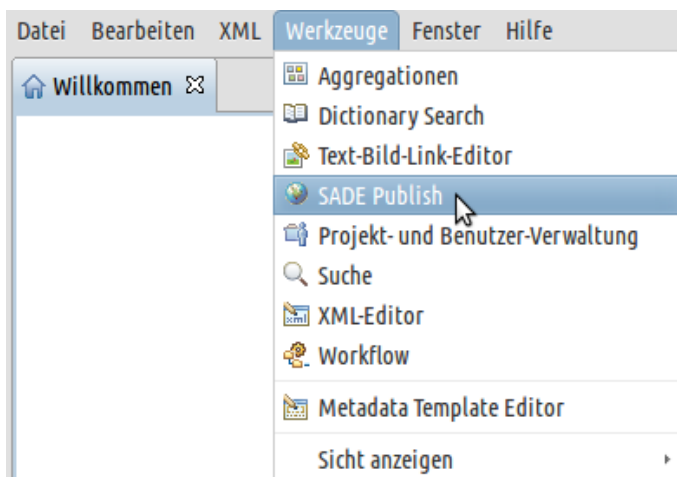


Abbildung 3: Starten des SADE-Plugins

Im TextGridLab wird der Navigator benutzt, um die zu übertragenden Daten auszuwählen.

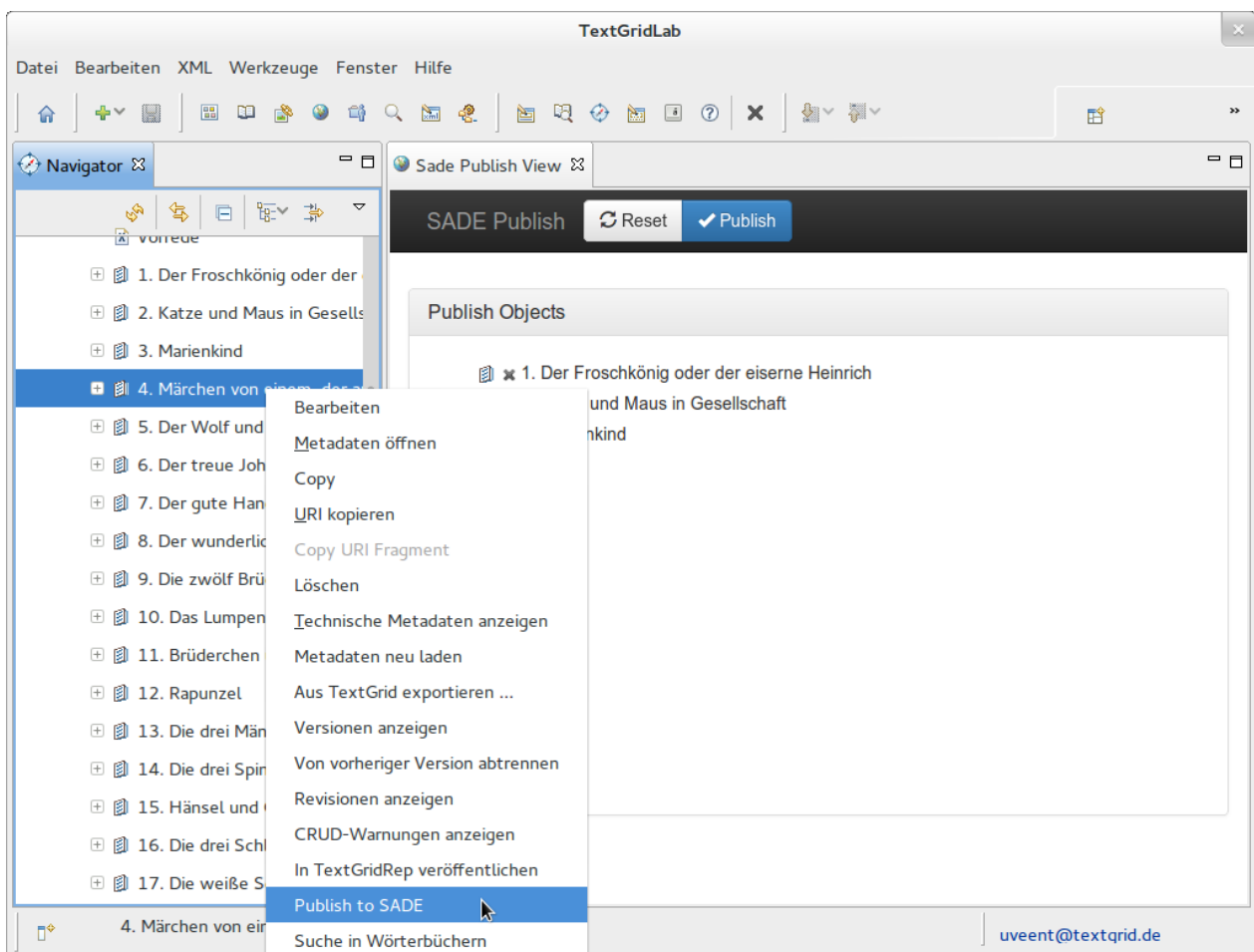


Abbildung 4: SADE Publish-Tool und Kontextmenü des Navigators

Auch Aggregationen (Kollektionen, Editionen und Aggregationen selbst) können ausgewählt werden und deren kompletter Inhalt wird kopiert, das Skript arbeitet also rekursiv. Im Kontextmenü

zu den Objekten (zu erreichen via rechter Maustaste) wählt man „Publish to SADE“, unter Linux und Mac kann man auch *drag and drop* benutzen.

Anschließend wählt man im Publish-Tool den Button „Publish“ an und startet damit den Übertragungsvorgang. Es werden die XML-Dokumente (z. Bsp. TEI-Dokumente, aber auch Objekte, die mit dem Text-Bild-Link-Editor erstellt wurden) und auch die zugehörigen Metadaten übertragen. Wählt man Bilder, die im SADE-Portal dargestellt werden sollen, werden nur die Metadaten übertragen und in der daraus resultierenden Menüstruktur wird ein Link zum Bild erstellt.

Der Publikationsvorgang ist erfolgreich abgeschlossen, wenn ein grünes Häkchen hinter den aufgelisteten Objekten erscheint. Der Prozess kann bei umfassenderen Datensätzen einige Minuten in Anspruch nehmen.

8 Konfiguration der SADE-Instanz

Um innerhalb der SADE-Instanz Änderungen an den Dokumenten und der Konfiguration vornehmen zu können, kann das von eXist-db mitgelieferte Tool „eXide“ genutzt werden. Man erreicht es über die URL <http://localhost:8080/exist/apps/eXide/index.html>. Dort steht ein Editor zur Verfügung, man kann den Button „OPEN“ nutzen um in dem sich dann öffnenden Dialogfenster die Datenbank zu durchsuchen und betreffende Dokumente zu öffnen.

Wie man der konfigurierten URL (<http://localhost:8080/exist/apps/textgrid-connect/publish/textgrid/>, siehe 5.2.1) entnehmen kann, werden die Daten hier in ein Projekt „textgrid“ abgelegt. Der zuletzt stehende Part deutet darauf hin. (An dieser Stelle kann die URL auch abgeändert werden und ein eigener Ordner verwendet werden.) Demnach sind die Dokumente in der Datenbank nun unterhalb des Hauptverzeichnisses „sade-projects“ in einer Kollektion mit dem Titel „textgrid“ abgelegt. Dort finden sich weitere Dateien, die für das Layout zuständig sind und auch weitere Inhalte generieren. Im XML-Dokument „config.xml“ werden alle nötigen Konfigurationen hinterlegt.

8.1 Config.xml

Derzeit muss lediglich für die Nutzung des Digilib-Proxy die folgende Minimalkonfiguration vorgenommen werden:

```
<param key="textgrid.user">[OBSERVER-ACCOUNT-NAME]</param>  
<param key="textgrid.password">[OBSERVER-ACCOUNT-PASSWORD]</param>
```


Alle weiteren Angaben sind optional anpassbar.

8.2 Menüführung

Generell handelt es sich bei dem integrierten Template um eine beispielhafte Umsetzung eines Bootstrap-Templates. An dieser Stelle kann ein beliebiges Template eingefügt werden, was lediglich kleinerer Anpassungsarbeiten bedarf.

8.2.1 Menü oben

Das Menü in der Navigationsleiste oben wird über das bereits beschriebene Navigations-Modul in SADE erstellt.

8.2.2 Datenstruktur (Sidebar)

Das Datenmodell von TextGrid und die Struktur des Navigators aus dem TextGridLab werden komplett übernommen. In einer Sidebar werden die Titel aller übertragenen Dokumente angezeigt und die Inhalte sind von dort aus per Link erreichbar. Entscheidend hierfür sind zwei am Ende des Publikationsvorganges aufgerufenen Funktionen. Das erste dient der weiteren Trennung von Daten und Layout und erzeugt zunächst eine XML-Datei, ähnlich der des Navigation-Modules. Hier werden die hierarchische Struktur abgebildet und die notwendigen Metadaten zusammengefasst. Dazu gehören Objekttyp, Titel und die um das Präfix bereinigte TextGrid URI samt Revisionsnummer. Diese Struktur kann über ein simples XSLT schließlich konform zum verwendeten Template gemacht werden und in einem neuen Dokument abgelegt werden, welches ins Template integriert wird.

9 Zukünftige Anbindungsmöglichkeit

9.1 DARIAH-eXist-Hosting

Im Rahmen des Projektes DARIAH-DE werden, derzeit noch im Testbetrieb, die Einrichtung und das Hosting von eXist-Datenbanken angeboten. Diese können von Geisteswissenschaftlern genutzt werden, die dadurch den Vorteil haben, sich nicht selbst um die Hardware, das Betriebssystem, den Internetzugang usw. kümmern zu müssen.

Um ein Hosting von SADE bei DARIAH zu vereinfachen, wurde im Rahmen dieses Projektes darauf geachtet, dass die hier programmierte Software innerhalb einer eXist-Installation als XAR-Paket installiert (und aktualisiert) werden kann.

10 Referenzinstanz

Derzeit ist eine Referenzinstanz zu besichtigen unter

<http://141.5.100.153/exist/apps/SADE/textgrid/index.html>. Dorthin wurden Beispieldaten aus dem TextGrid Repository übertragen. Ein mittels Template Engine integriertes Layout, welches nur durch die Sidebar (tg-menu) ergänzt wird, wird angewendet.

Um eigene Inhalte einzuspielen und die Funktionen zu testen, können sich Interessenten unter der Adresse <http://141.5.100.153/exist/apps/SADE/textgrid/sign-up.html> einen eigenen Bereich einrichten. Dabei wird ein neuer Nutzer in der Datenbank eingerichtet. Lese- und Schreibzugriffe beschränken sich auf diesen exklusiven Bereich. Um die Datenbankstruktur zu erkunden und Änderungen und Entwicklungen einzupflegen, kann man das oben beschriebene „eXide“ benutzen. Erreichbar ist es unter <http://141.5.100.153/exist/apps/eXide/>. Für den Login werden die im Formular angegebenen und per Mail erhaltenen Daten benötigt. Außerdem sind in der E-Mail die Konfigurationsparameter für das TextGridLab Plug-in angegeben.