

# TextGrid: eScholarship und vernetzte Angebote

Marc Wilhelm Küster\*, Christoph Ludwig\* and Andreas Aschenbrenner†

\* University of Applied Sciences Worms, Erenburgerstr. 19, 67549 Worms, Germany, Email: {kuester|ludwig}@fh-worms.de

† Goettingen State and University Library, Papendiek 14, 37073 Goettingen, Germany, Email: {aschenbrenner}@sub.uni-goettingen.de

**Abstract**—Das TextGrid-Konsortium bringt acht über ganz Deutschland verteilte Institutionen aus dem akademischen und kommerziellen Sektor zusammen, um eine *modulare Plattform für verteilte und kooperative wissenschaftliche Textdatenverarbeitung* zu erarbeiten. Unter Einsatz der Grid-Infrastruktur führt es verschiedene Datenstrukturen aus unterschiedlichen Projekten in einem virtuellen Korpus zusammen, das als ganzes oder in ausgewählten Teilen durchsucht und analysiert werden kann.

Als dezidiert auf Kollaboration auch über die Grenzen von TextGrid selbst ausgerichtete Plattform verändert sie auch den Arbeitsalltag des geisteswissenschaftlichen Forschers, weg vom Bild des weitgehend isoliert arbeitenden Philologen hin zur Teamarbeit an Daten und Werkzeugen gleichermaßen. Sie bildet aber auch auf Informatikseite ein Beispiel für ein komplexes, lose gekoppeltes Digitales Ökosystem mit Spielern aus vielen Disziplinen und Ländern.

## I. DIE LEHREN DES BENEDICTUS LEVITA

In Vielem hat sich das Ideal philologisch-kritischen Arbeitens seit den legendären Editionen des Aldo Manuzio kaum verändert. Ein Philologe erfasst (zumindest im Idealfall) alle erhaltenen Manuskripte eines ihn interessierenden Werkes und identifiziert die Abhängigkeiten, die zwischen ihnen existieren mögen (Stemma), wobei er oft im Prozess neue Manuskripte entdeckt und das Stemma entsprechend modifiziert. Er dokumentiert alle oder doch zumindest die seiner Einschätzung nach wichtigsten textuellen Abweichungen der als wesentlich erkannten Manuskripte untereinander und, je nach Schule, etabliert davon ausgehend einen idealen Autorentext, dem im Druck die Abweichungen, üblicherweise in Form eines kritischen Apparats beigegeben werden. Dabei bezieht er möglicherweise bereits existierende Editionen des Textes in seine Überlegungen ein. Meistens fügt der Herausgeber noch einen oder mehrere Erklärungsapparate hinzu, in denen dunkle Passagen erläutert, wenig bekannte Personen oder Orte vorgestellt und (je nach Zielgruppe) ungewöhnliche Wörter erklärt werden. Regelmäßig werden zumindest Personen und Orte auch über ein oder mehrere Register verschlagwortet.

Selbst wenn viele der großen Editionsprojekte wie die *Monumenta Germaniae Historica (MGH)* über die Jahrzehnte oder gar Jahrhunderte ihres Bestehens sehr zahlreiche Editoren eingebunden haben, so ist doch zumindest im Normalfall ein einziger Wissenschaftler für die kritische Edition eines gegebenen Werkes alleinverantwortlich. Dieser Prozess war und ist extrem zeitaufwändig und in großem Maße abhängig von der Person des Editors. Eine kritische Edition zu erstellen

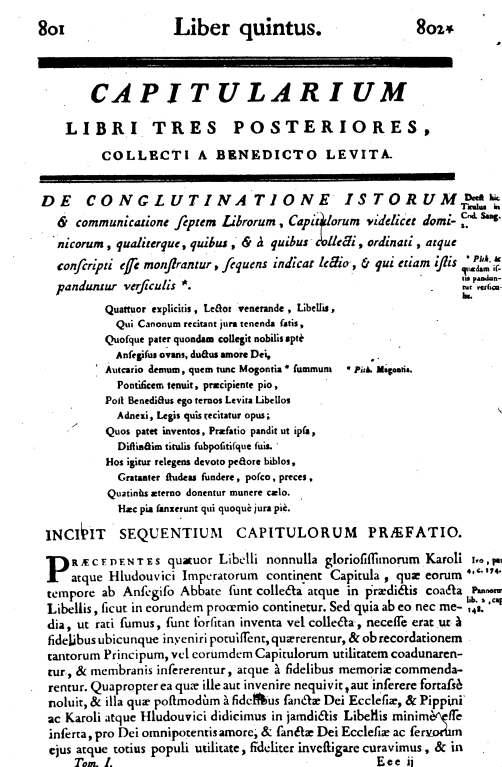


Fig. 1. Beginn der falschen Kapitularien des Benedictus Levita in der Edition von Étienne Baluze in der Ausgabe von Pierre de Chiniac [1]

dauert regelmäßig viele Jahre, nicht selten Jahrzehnte, während derer bereits erreichte Zwischenergebnisse niemandem zur Verfügung stehen. Nicht selten erlebt ein Herausgeber die Vollendung seiner Arbeit nicht einmal, und ein Nachfolger muss wieder bei Null anfangen.

Nicht anders erging es bei der Edition der falschen Kapitularien des Benedictus Levita. Diese umfangreiche Fälschung von Urteilen und anderen Entscheidungen aus der Mitte des neunten Jahrhunderts führte viele unterschiedliche Quellen in drei Büchern und vier *additiones* zusammen. Über Jahrhunderte haben die Kapitularien tatsächlich die europäische Rechtsgeschichte beeinflusst.

Obwohl die falschen Kapitularien also eine rechtshistorische Quelle erster Güte sind, wurden sie in ihrer Gesamtheit zum letzten Mal von Étienne Baluze (1630-1718) im Jahr 1677

ediert (Figur 1). Obwohl diese Edition längst selbst ein historisches Dokument ist, gilt sie zusammen mit ihren verschiedenen Nachdrucken immer noch als Referenzedition. Seither sind, wie [2] berichtet, alle Versuche einer neuen Edition fehlgeschlagen. Allein im zwanzigsten Jahrhundert haben vier prominente Forscher ihr Glück an dem Werk probiert, aber drei davon, darunter der eminente Berliner Rechtshistoriker Emil Seckel (1864-1924), starben, ohne ihre Arbeit zu vollenden. Ihre Nachfolger mussten jeweils weitgehend wieder bei Null anfangen.

Auch losgelöst vom konkreten Fall kann uns Benedictus Levita einiges zu den tieferen Ursachen dieser wiederholten Fehlschläge lehren, hier illustriert an einem Auszug aus dem Nachruf für Emil Seckel:

Denn er [Emil Seckel] machte alles selbst; er gleich die von seinen Mitarbeitern eingetragenen Kollationen auf das genaueste nach und untersuchte jede Lesart und hat, was man kaum glauben möchte, den ganzen Text des Benedictus samt dem größten Teile des Apparats propria manu sauber und gleichmäßig ins Reine geschrieben. [...] aber mit den zeitfressenden Methoden des ehemaligen freien Privatgelehrten konnte der überlastete Universitätsprofessor nicht zum Ziel kommen. Dies war die Tragik in Seckels wissenschaftlicher Produktion, dass er sich nur schwer zu entschließen vermochte, seine Kräfte zu konzentrieren [...]. Und so ist eingetreten, was ich bei der ersten Kunde seiner schweren Erkrankung sogleich fürchtete: die Ausgabe, auf die er mehr als zwei Jahrzehnte gründlichster Arbeit verwendet hat, ist nicht zum Abschluss gelangt [3], S. 160

Erst das vierte Team um Gerhard Schmitz, das zu den frühen Nutzern auch Web-basierter Kollaboration gehört, dokumentiert Zwischenstände für die Allgemeinheit und wird dieses Projekt hoffentlich zu einem glücklichen Ende führen.

Benedictus Levita ist kein extremer Einzelfall. Um willkürlich noch ein weiteres Beispiel herauszupicken: Werner Peek hat an seinen *Griechischen Versinschriften* über einen Zeitraum von sechs Jahrzehnten gearbeitet. "Er [Peek] war enorm fleißig und hat von 1930 bis zu seinem Tod, also mehr als 60 Jahre lang, unermüdlich gearbeitet, das ganze Mittelmeergebiet bereist, Abklatsche und Photos angefertigt, neues Material erfaßt, die Inschriften selber mit Augen gesehen und neue Lesungen gewonnen – und ist nicht zum Ziel gekommen; sein magnum opus ist eine Forschungsruine geblieben"[4] – und sechzig Lebens- und Arbeitsjahre in guten Teilen verloren.

## II. EINE VISION KOLLABORATIVEN ARBEITENS

Ein solches, auf den weitgehend isoliert arbeitenden Philologien abgestelltes Modell wissenschaftlichen Edierens kann nicht befriedigen. — es verschwendet nicht nur ganz direkt in hohem Maß wertvolle Ressourcen, sondern enthält auch der gesamten wissenschaftlichen Gemeinschaft oft über Jahrzehnte Zwischenstände vor, die für einen weiteren Fortschritt essentiell wären. Das gilt umso mehr, als diese Methode der kritischen Edition ganz auf ein Leitmedium, den Druck,

ausgerichtet ist. Ein Buch, einmal gedruckt, lässt sich nur schwer verbessern (es ist selten, dass eine kritische Edition in absehbarer Zeit den Absatz erzielt, der eine weitere Auflage ermöglicht) und hat naturgemäß Platzbeschränkungen, einmal durch die Dimensionen der einzelnen Druckseite, dann durch eine maximale Seitenzahl für die Edition, die sowohl drucktechnisch als auch ökonomisch machbar ist. Auch hier beeinflusst das Medium stark die wissenschaftliche Arbeitsweise [5].

Die Beschränkungen des Druckmediums existieren heute natürlich nur noch begrenzt. Damit wird ein kooperatives und kollaboratives Wissenschaftsmodell möglich, in dem Wissenschaftler gleichzeitig weiterhin die volle Kontrolle über ihre Daten behalten und Zwischenergebnissen sukzessiv publizieren und mögliche Fehler schnell verbessern können. Die Zwischenergebnisse sind viel früher für Kollegen verfügbar und bleiben auch später zitierbar. Die Arbeitsschritte, die von den Zwischenständen zum Endergebnis führen, werden transparent und in sich selbst wichtige Projektergebnisse.

Zusätzlich zur schnelleren Publikation muss auch die synchrone Kooperation über räumliche Grenzen hinweg erleichtert werden. Geographisch dislozierte Teams müssen zusammen an einem Projekt arbeiten können.

### A. TextGrid

Als *modulare Plattform für verteilte und kooperative wissenschaftliche Textdatenverarbeitung* ist TextGrid dem kooperativen und kollaborativen Wissenschaftsmodell in den Philologien verpflichtet. Unter Einsatz der Grid-Infrastruktur führt es verschiedene Datenstrukturen aus unterschiedlichen Projekten in einem virtuellen Korpus zusammen, das als ganzes oder in ausgewählten Teilen durchsucht und analysiert werden kann. Dabei sprechen wir von großen Datenmengen: Die über die letzten Jahre und Jahrzehnte allein in Deutschland erfassten Texte und Digitalisate gehen in viele Terrabyte, verteilt über viele Institutionen. Anders als etwa in der Teilchenphysik liegt die Hauptherausforderung aber nicht in der schiereren Datenmenge, sondern in ihrer Heterogenität und in der Vielzahl der Projekte, die sie bereitstellen. Die Daten müssen dazu möglichst einheitlich beschrieben werden und ihr konkreter Speicherort muss für den Nutzer transparent sein, wenn die Vision eines virtuellen Korpus mit Leben gefüllt werden kann.

Neben die Daten bieten die teilnehmenden Organisationen dezidierte Dienste, die über wohldefinierte Schnittstellen in einen nutzerdefinierten Work flow eingebunden werden können, um existierende Textdaten auszuwerten und um neue Daten manuell oder (semi-)automatisch zu strukturieren. Weitere Dienste können leicht eingebunden werden, auch über die Grenzen von TextGrid im engen Sinne hinaus.

### B. TextGrid als Projekt

Die vom BMBF auf drei Jahre geförderte Arbeit an TextGrid begann im Februar 2006 in einem aus acht quer über Deutschland verteilten Organisationen aus: die Staats- und Universitätsbibliothek Göttingen als Projektleiter, die Technische Universität Darmstadt, das Institut für Deutsche Sprache (IDS)

in Mannheim, die Universität Trier und die Fachhochschule Worms als akademische sowie DAASI International und Saphor GmbH als kommerzielle Partner. Die Organisationen zusammen decken verschiedenste textwissenschaftliche Disziplinen ab: kritische Editionen, Lexikographie, Korpuslinguistik und Informatik. Für TextGrid II planen wir, die Breite der abgedeckten Fächer nochmals beträchtlich zu erweitern.

TextGrid ist eines der sechs Community Grid Projekte der ersten Stunde in der D-Grid-Initiative, dem strategischen Projekt für eine nationale Grid-Infrastruktur in Deutschland. D-Grid bildet den grundlegenden organisatorischen Rahmen, in dem sich fachbezogene Grids bewegen. Als strategische Initiative für eine nationale deutsche Grid-Infrastruktur ist D-Grid selbst nicht auf bestimmte Disziplinen festgelegt, sondern unterstützt Projekte von Hochenergiephysik über Ingenieurwissenschaften, Geographie und Medizin bis hin zur Meteorologie – also oftmals Projekte aus Bereichen, die als “klassische” Gridnutzer gelten können. TextGrid als einziges geisteswissenschaftliches Projekt war zumindest am Anfang ein gewisser Außenseiter, aus dem aber mittlerweile ein respektierter Partner geworden ist, der selbstbewusst die spezifischen Anforderungen der Textwissenschaften wie Langzeitarchivierung und Datenföderation einbringt.

### C. TextGrid als Digitales Ökosystem

Im Licht des Gesagten sieht TextGrid sich als Digitales Ökosystem, sehr bewusst offen für Interaktion mit anderen Systemen aus den eHumanities und darüber hinaus, aber auch mit verteilten Daten verschiedenster Provenienz und, last but not least, den Nutzern selbst.

Nachdem das Konzept des Digitalen Ökosystems relativ neu ist, lohnt es sich, es kurz vorzustellen. Ein Digitales Ökosystem ist ein “open, loosely coupled, domain clustered, demand-driven, self-organising collaborative environment[s], where each species is proactive and responsive for its own benefit or profit” [6]. Auch wenn diese Definition sehr bewusst in Analogie zur biologischen Begrifflichkeit gewählt wurde, so ist ein Digitales Ökosystem doch primär ein Konzept der Informatik, in der Realisierung bezogen auf das “semantic web, ontology based knowledge sharing, self organizing intelligent agents, trust and recommender systems, web services and platform based systems” (a.a.O.), notwendig mit besonderem Blick auf zwei Charakteristika, Offenheit und Bereitschaft zum permanenten Wandel. Es ist, wie wir in [7] und [8] argumentieren, ein Paradigma, in dem Dienste und Daten gleichermaßen essentielle Teile sind, die sich gegenseitig bedingen und zusammen mit den Nutzern einen Lebensraum bilden.

## III. ARCHITEKTUR

Die Forderung nach Offenheit und Kollaboration gekoppelt mit der Notwendigkeit, technisch oft wenig versierte Nutzer einzubinden, bestimmen die Architektur von TextGrid.

Dies beginnt, wie in Figur 2 illustriert, bei der TextGrid Middleware selbst. So baut TextGrid auf Grid-Technologie, genau auf Globus Toolkit 4 (GT4) auf. Gleichzeitig ist es aktuell das einzige der großen eHumanities-Projekte weltweit,

das diesen Weg geht. Viele andere wie z. B. TAPoR (Text Analysis Portal for Research) [11], Bricks (Building Resources for Integrated Cultural Knowledge Services) [12], [13] und Gate (General Architecture for Text Engineering) [14], [15], [16] in seinen neuesten Inkarnationen verwenden Plain Vanilla SOAP Web Services. Ähnliches gilt für viele Repositoriumsprojekte.

Selbst wenn auch GT4 weitgehend, wenn auch nicht durchgehend Web Services zur internen Kommunikation verwendet, so doch im Kontext des Web Services Resource Framework (WSRF) [17], das zusätzlich zu den Basisstandards SOAP [18] und WSDL [19] weitere Standards wie WS-Addressing [20] nutzt. Somit verkapselt die TextGrid-API alle diese Besonderheiten der gridbasierten Middleware und interagiert mit den anderen Ebenen des Systems über reine Web Service Aufrufe.

Die eigentlichen Dienste, zu denen unten mehr zu sagen sein wird, offerieren dementsprechend auch ein rein SOAP-basiertes Interface, das problemlos auch von vollkommen Grid-agnostischen Workflows heraus angesprochen werden kann. Umgekehrt können Nicht-Grid-Dienste ohne Schwierigkeiten in TextGrid-spezifische Arbeitsabläufe integriert werden.

Für die Nutzeroberfläche (GUI) haben wir uns bewusst für einen Eclipse-basierten Fat Client entschieden, der es einerseits ermöglicht, auch Offline mit Texten zu arbeiten und die Daten später mit dem Grid zu synchronisieren, andererseits die Endnutzer bei deren etablierten, eher durch Office-Pakete geprägten GUI-Gepflogenheiten abholt. Damit haben wir bislang gute Erfahrungen gemacht.

Diese Entscheidung prädatiert allerdings Entwicklungen wie Google Gears [21] und GWT [22], die es zumindest perspektivisch ermöglichen können, diese Vorteile auch mit rein Web-basierten GUIs zu realisieren.

### A. Daten und Datenformate

Daten, speziell textuelle Daten sind das Herzstück aller Philologien. Es war daher von Anfang an klar, dass zuverlässige Strategien für Datenstrukturen und Datenspeicherung für den Projekterfolg essentiell sein würden. Fünf Anforderungen haben sich dabei als besonders wichtig herauskristallisiert:

- 1) Langzeitverfügbarkeit: die Daten bleiben garantiert erhalten
- 2) Referenzierbarkeit: alle Ressourcen werden unabhängig von ihrem tatsächlichen Speicherort stabil identifiziert
- 3) Zitierbarkeit: alle einmal publizierten Textversionen bleiben weiterhin sichtbar
- 4) Flexibel: Projekten steht es frei, ihre eigenen XML-Schemata zu verwenden
- 5) Transparenz: der Speicherort ist für den Nutzer transparent (einschließlich des transparenten Zugriffs auf Ressourcen bei Inhaltsanbietern, entsprechende Rechte vorausgesetzt)

1) *Datenformate*: Die Arbeit mit Textdaten im TextGrid Ökosystem und darüber hinaus bringt viele Herausforderungen mit sich. Selbst wenn wir uns auf XML-Dateien beschränken, so verwenden Projekte doch typischerweise ihre eigenen XML-Schemata. Selbst wenn viele kritische Editionen sich an den Richtlinien der *Text Encoding Initiative (TEI)*

— aktuell meistens noch TEI P4 [23], aber auch noch TEI P3 [24] oder bereits TEI P5 [25] — orientieren, so nutzen viele auch ihre ganz eigenen Formate. TEI lässt außerdem eine große Bandbreite an möglichen Auszeichnungen zu, um den unterschiedlichen Anforderungen philologischen Arbeitsprojekte und Forscher gerecht zu werden. Dies ist einerseits unvermeidlich — die konkreten Bedürfnisse der verschiedenen Disziplinen, Projekte und Forscher sind zu unterschiedlich —, verhindern aber andererseits komplexere Abfragen über Korpusgrenzen.

Als Antwort auf diesen Zwiespalt hat TextGrid das Konzept der Kernkodierung entwickelt (Figur 3), mit der einerseits auch über Korpusgrenzen hinweg inhaltliche Abfragen wie “welche Orte zitiert Goethe im Phaedrus”, “welche französischen Wörter werden in der Literatur des 19. Jahrhunderts besonders häufig genutzt” oder “welche Wörterbücher enthalten das Lemma ‘Schaf’” ermöglicht werden, andererseits aber jedes Projekt seine ihm eigene Auszeichnungsstruktur behalten kann.

Die Kernkodierung ist eine kleine, wohldefinierte Untermenge von TEI P5, die für die projektübergreifende Suche herangezogen wird.

2) *Datenspeicherung und Abfrage*: Über projektspezifische Adaptoren wird zusätzlich zum originalkodierte Text beim Import und bei jeder Speicherung eine kernkodierte Version angelegt (s. Figur 4). Diese Transformation ist fast immer verlustbehaftet. Projektspezifische Dienste, die um die Spezifika der Originalkodierung wissen, können aber jederzeit auf diese rekurren. Außerdem wird vom kernkodierte XML auch auf die korrespondierenden Passagen der Ursprungsdatei verlinkt, so dass beide Versionen auch im Sinne einer *Resource Oriented Architecture (ROA)* [26] als Repräsentationen derselben Ressource begriffen werden können.

Eine entsprechende Strategie verwendet TextGrid auch für Metadaten, die für die projektübergreifende Sicht auf ein minimales, auf Dublin Core [27] aufbauendes Metadaten-set abgebildet werden. Die reicheren Originalmetadaten stehen aber parallel weiter zur Verfügung.

Die in TextGrid verwendeten Interoperabilitätsstrategien lassen sich auch auf andere heterogene Daten anwenden, um wenigstens eine partielle semantische Interoperabilität [28] über Organisationsgrenzen hinweg zu ermöglichen.

## B. Dienste

Wie erwähnt, verwendet die Dienstebene von TextGrid ein reines Web-Service-Protokoll, alle Grid-Spezifika werden von der TextGrid-API verkapselt, so dass auch interessante Dienste aus anderen eHumanities-Ökosystemen eingebunden werden können. Im klassischen Sinn des Unix-Werkzeugkasten entspricht jeder Dienst dabei möglichst einer atomaren Funktionalität wie etwa dem Sortieren von Listen, dem Vergleich von Dateien, der Lemmatisierung von Wörtern usw.

Die Dienste lassen sich dabei in den meisten Fällen sowohl programmatisch ansprechen als auch über ihre Eclipse-basierte GUI nutzen. Eine Workflow-Engine erlaubt es, Dienste zu komplexen Arbeitsabläufen zu orchestrieren.

Mit der Veröffentlichung der Betaversion lässt sich TextGrid auch von Endnutzern verwenden. Integriert im sogenannten TextGridLab, der Eclipse-basierten TextGrid-GUI, finden sich neben interaktiven Komponenten Views auf die verschiedenen Dienste (s. Figur 5).

Im gegebenen Beispiel ist Goethes *Leiden des jungen Werther* im graphischen Modus des integrierten XML-Editors geöffnet. Der Metadateneditor links zeigt die entsprechenden Metadaten, während rechts eine integrierte Recherche im Wörterbuchnetz läuft.

### A. TextGridLab

Das TextGridLab ist für den Nutzer die primäre Schnittstelle zu TextGrid. Sie bietet neben einem validierenden XML-Editor einen Metadateneditor, der über einen Templateeditor auf eigene Metadaten-schemata angepasst werden kann, sowie einen Bild-Link-Editor (in der Endphase der Entwicklung).

### B. Dienste

An konkreten Diensten unterstützt TextGrid momentan folgende Leistungen:

- Wörterbuchdienst: parallele Suche durch Allgemeine, Fach- und Dialektwörterbücher
- Lemmatizer: Lemmatisierung und Wortformanalyse hochdeutscher Wörter der Gegenwartssprache
- Tokenizer: Aufteilung des Textes in individuelle Wörter
- Kollationierer: Vergleich von mehreren Textzeugen. Der Kollationierer befindet sich noch im Frühstadium und wird in Kooperation mit Peter Robinson (Collage / [29]) weiterentwickelt
- Sortieren: alphabetische Anordnung von Listen entsprechend bestehender Locales oder freidefinierter Regeln

### C. Middleware

Die Funktionalität der Middleware bleibt für den Nutzer weitgehend transparent. CRUD — Create, Update, Delete — als zentraler Teil der TextGrid-API steht hinter den entsprechenden Aktionen der GUI (Datei öffnen / Speichern / Löschen) und sorgt dafür, dass die Dateien im Grid aus Nutzersicht genauso angesprochen werden können wie lokale Daten. Über den Recherchedienst kann der Nutzer gezielt nach Texten recherchieren, sei es über eine Google-ähnliche Volltextsuche, über die Kernkodierung oder vermittels der entsprechenden Metadaten (“alle von Goethe zwischen 1780 und 1785 geschriebenen Texte” oder “alle zum ersten Band des Campe-Wörterbuch gehörenden Bilddaten”).

Am ehesten direkt sichtbar ist die Middleware als Garant von Authentifizierung und Autorisierung der Nutzer gegen shiboleth. Daten werden dabei im TextGridLab in Projekte mit berechtigten Mitarbeitern gegliedert, um die problemlose Kollaboration in Forschungsteams zu ermöglichen und gleichzeitig die Vertraulichkeit noch unveröffentlichter Texte zu garantieren. Sie können, entsprechende Berechtigungen vorausgesetzt, auch von anderen Projekten aus verlinkt werden. Publierte Daten sind in der jeweiligen publizierten Version

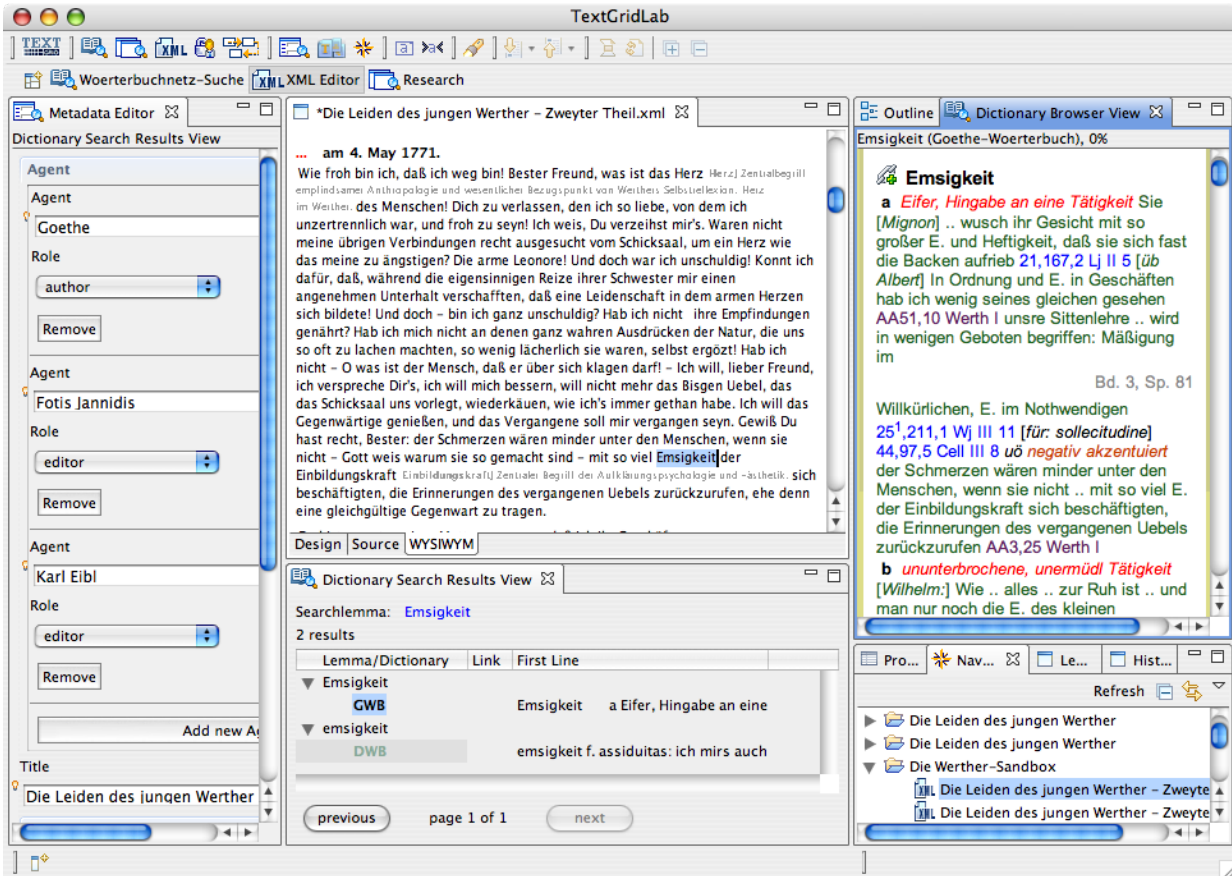


Fig. 5. Screenshot des TextGridLab in Aktion

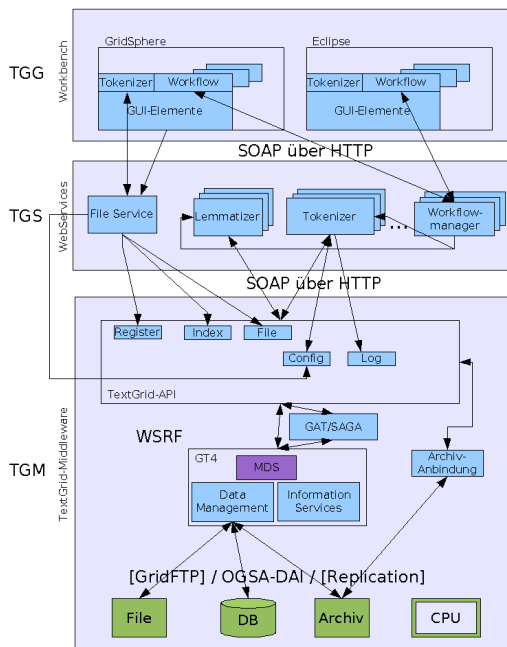


Fig. 2. Textgrid-Architektur (aus [9], s. auch [10])

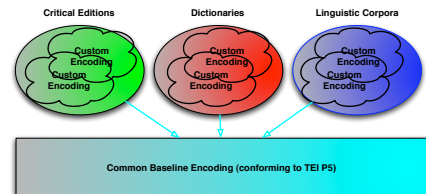


Fig. 3. Idee der Kernkodierung (aus [7], s. auch [9])

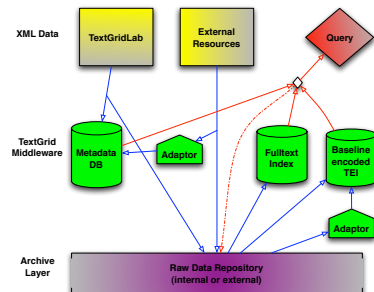


Fig. 4. Datenhaltung und -verwaltung in TextGrid (aus [7], s. auch [9])

insgesamt in TextGrid sichtbar und können herangezogen werden.

## V. LEBEN IM TEXTGRID-ÖKOSYSTEM

Nachdem wir in den vorherigen Abschnitten einen Blick auf die abiotischen (Daten) und biotischen (Dienste) Bewohner des TextGrid-Ökosystem geworfen haben, fehlt noch der Blick auf dessen menschliche Bewohner. Denn wenn der Nutzer in der Modellierung von Ökosystemen auch oftmals geradezu übersehen wird, so wäre dies besonders im TextGrid-Szenario fatal. Die *raison d'être* von TextGrid ist schließlich, in einem tendenziell wenig technik-affinen Umfeld nicht nur neue Werkzeuge bereitzustellen, sondern einen neuen, kollaborativen Arbeitsstil zu unterstützen. Dabei sind drei Untersysteme für den Erfolg besonders wichtig:

- 1) Endnutzer: die Philologen, die TextGrid für ihre Forschung einsetzen
- 2) Inhaltsanbieter: Institutionen — Archive, Bibliotheken und Verlage, aber auch Endnutzer, insofern sie ihre Zwischenergebnisse publizieren
- 3) Software-Entwickler: das Team, das die Softwareplattform selbst weiterentwickeln und neue Dienste konzipieren und realisieren

Nur im Zusammenspiel dieser drei, sich oft überlappenden Gruppen können wir sicherstellen, dass wir in TextGrid auch weiterhin die richtigen Schlüsse aus den Lehren des Benedictus Levita ziehen.

## VI. DANKSAGUNGEN

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 07TG01A-H und als Teil von D-Grid gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

## REFERENCES

- [1] B. Levita, *Capitularia Regum Francorum*, nova editio auctior ac emendatior ad fidem autographi baluzii qui de novo textum purgavit ed., Étienne Baluze and P. de Chiniac, Eds. Paris: Quillau, 1780.
- [2] G. Schmitz and C. Radl, "Die Edition der Kapitulariensammlung des Benedictus Levita," <http://www.zdv.uni-tuebingen.de/tustep/prot/prot841-benlev.html>, Feb 2002.
- [3] P. Kehr, "Emil Seckel. Ein Nachruf," *Neues Archiv der Gesellschaft für ältere deutsche Geschichtskunde*, pp. 158–180, 1926. [Online]. Available: <http://www.mgh-bibliothek.de/cgi-bin/nada/na46.pl?seite=160>
- [4] R. Merkelbach, "Nochmals Inscriptiones Graecae," *Zeitschrift für Papyrologie und Epigraphik*, vol. 122, p. 293–299, 1998.
- [5] M. McLuhan, *Understanding media*. Routledge, 1964.
- [6] T. Dillon and E. Damiani. (2007) 2<sup>nd</sup> Call for Papers for IEEE-DEST 2008. [Online]. Available: <http://www.ieee-dest.curtin.edu.au/2008/CFP.pdf>
- [7] C. Ludwig and M. W. Küster, "Digital ecosystems of ehumanities resources and services," in *Digital Ecosystems and Technologies, 2008. DEST 2008. 2nd IEEE International Conference on*. IEEE, 2008, pp. 476–481.
- [8] M. W. Küster, M.-S. Li, P. Malo, and C. Ludwig, "E2ESU — stable foundations for digital business ecosystems," in *Digital Ecosystems and Technologies, 2008. DEST 2008. 2nd IEEE International Conference on*. IEEE, 2008, pp. 77–82.

- [9] A. Aschenbrenner, P. Gietz, M. Haase, F. Knoll, C. Ludwig, W. Pompe, M. Sosto, and T. Vitt, "Textgrid architektur," Niedersächsische Saats- und Universitätsbibliothek, Göttingen und DAASI International GmbH, Tech. Rep., 2007. [Online]. Available: [http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid.Report\\_3.2.pdf](http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid.Report_3.2.pdf)
- [10] P. Gietz, A. Aschenbrenner, S. Büdenbender, F. Jannidis, M. W. Küster, C. Ludwig, W. Pompe, T. Vitt, W. Wegstein, and A. Zielinski, "Textgrid and ehumanities," in *Proc. of Second IEEE International Conference on e-Science, December 4-6, 2006, Amsterdam*. IEEE CS Press, 2006.
- [11] TAPoR, "Tapor: text analysis portal for research," <http://tapor.humanities.mcmaster.ca/home.html>, Nov 2006.
- [12] T. Risse, P. Knežević, C. Meghini, R. Hecht, and F. Basile, "The BRICKS infrastructure — and overview," in *8th International Conference EVA 2005 Moscow*. EVA, 2005.
- [13] BRICKS Project, "Bricks — building resources for integrated cultural knowledge services," <http://www.brickcommunity.org/>, Nov 2006.
- [14] H. Cunningham, Y. Wilks, and R. J. Gaizauskas, "Gate: a general architecture for text engineering," in *Proceedings of the 16th conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1996, pp. 1057–1060.
- [15] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "Gate: an architecture for development of robust hlt applications," in *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2001, pp. 168–175.
- [16] GATE Project, "GATE, a general architecture for text engineering," <http://gate.ac.uk/>, Nov 2006.
- [17] OASIS, "Web services resource 1.2," OASIS, OASIS standard, 4 2006.
- [18] W3C, "Soap version 1.2," W3C, W3C Candidate Recommendation, Jun 2003.
- [19] —, "Web Services Description Language (WSDL) 1.1," W3C, W3C note, Mar 2001.
- [20] W3C, "Web service addressing 1.0 – core," W3C, W3C Recommendation, May 2006.
- [21] Google. (2009) Gears. improving your web browser. [Online]. Available: <http://gears.google.com/>
- [22] —. (2009) Google web toolkit. [Online]. Available: <http://code.google.com/webtoolkit/>
- [23] C. M. Sperberg-McQueen and L. Burnard, Eds., *Guidelines for Text Encoding and Interchange (P4)*. Humanities Computing Unit, University of Oxford, 2002.
- [24] —, *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. Chicago, Oxford: Text Encoding Initiative, 1994. [Online]. Available: <http://www.tei-c.org/Vault/GL/teip3doc.tar.gz>
- [25] L. Burnard and S. Bauman, "TEI P5: Guidelines for Electronic Text Encoding and Interchange," 10 2007. [Online]. Available: <http://www.tei-c.org/release/doc/tei-p5-doc/html/>
- [26] L. Richardson and S. Ruby, *RESTful web services*. Farnham: O'Reilly, 2007.
- [27] *The Dublin Core Metadata Element Set : an American national standard*. Bethesda, MD: NISO Press, 2001.
- [28] IDABC, *European Interoperability Framework for Pan-European eGovernment Services*. Luxembourg: European Communities, 2004.
- [29] P. M. W. Robinson, "The collation and textual criticism of icelandic manuscripts (1): Collation," *LLC*, vol. 4, no. 2, pp. 99–105, 1989.