

Usability, Zusammenfassung der bisherigen Nutzerszenarien und Identifikation der Testgruppen (R 5.3.1)

Version 30.11.12
Arbeitspaket 5.3
verantwortlicher Partner MPI WG

TextGrid

Virtuelle Forschungsumgebung für die Geisteswissenschaften



Projekt: TextGrid - Institutionalisierung einer Virtuellen Forschungsumgebung in den Geisteswissenschaften

BMBF Förderkennzeichen: 01UG1203A

Laufzeit: Juni 2012 bis Mai 2015

Dokumentstatus: final

Verfügbarkeit: öffentlich

Autoren: Klaus Thoden, MPI WG

Revisionsverlauf:

Datum	Autor	Kommentare
30.11.2012	Klaus Thoden	erste Version
21.12.2012	Klaus Thoden	zweite Version
24.01.2013	Klaus Thoden	revidierte Version
11.02.2013	Klaus Thoden	2. revidierte Version
15.02.2013	Klaus Thoden	3. revidierte Version

Inhaltsverzeichnis

1. Einleitung	4
2. Usability	4
2.1. Usability im allgemeinen.....	4
2.2. Usability in anderen VFUs.....	6
2.3. Herangehensweisen und Begriffsdefinitionen.....	6
2.3.1. Cognitive Walkthrough.....	6
2.3.2. Thinking Aloud.....	6
2.3.3. Softwarebasierte Evaluation.....	7
2.3.4. GUI Workshops.....	7
2.4. Erwartungen an die Software.....	7
2.4.1. Aus Nutzersicht.....	7
2.4.2. Aus Entwicklersicht.....	9
2.5. Usability in Textgrid.....	10
2.5.1. Vorarbeiten in TGI und TGII.....	10
3. Nutzerszenarien	11
3.1. TGII – Fragebögen.....	11
3.1.1. Auswertung der Fragebögen.....	11
3.2. Neue Benutzungsszenarien.....	12
3.2.1. Publikation.....	12
3.2.2. Annotationen.....	12
3.2.3. Dezentrale Kollaboration.....	13
3.2.4. Offlinearbeit.....	13
3.2.5. Navigation von XML-Texten im Browser.....	13
3.2.6. Visualisierung und diagrammatische Darstellung.....	13
3.2.7. Kommunikation.....	14
3.2.8. Communitybuilding.....	14
3.2.9. Konvertierung von Texten nach XML.....	14
4. Identifikation der Testgruppen	14
4.1. Methodisches Vorgehen.....	15
4.1.1. Fragebogen.....	15
4.1.2. Beobachtung bei der Arbeit – Cognitive Walkthrough/ Thinking Aloud.....	15
4.1.3. Interviews.....	15

1. Einleitung

Dies ist der Bericht aus dem Arbeitspaket 5 (Betrieb und Softwarepflege Laboratory), Unterabschnitt 3 (Usability) aus dem Projekt TextGrid III. In diesem Unterabschnitt geht es um die systematische Untersuchung, wie TextGrid in der wissenschaftlichen Praxis eingesetzt wird, vor allem in Bezug auf Nutzergruppen, die vorher noch nicht mit TextGrid gearbeitet haben. Wichtig ist dabei auch, ein breites Spektrum an geisteswissenschaftlichen Disziplinen abzudecken.

Der Bericht teilt sich in drei Teile auf. Zunächst wird ein Überblick über den Begriff Usability gegeben und wie Usability ermittelt werden kann. Im zweiten Teil werden in einem kleinen Rückblick die bisherigen Nutzerszenarien beschrieben, und es wird ein Ausblick darauf gegeben, was für neue Nutzerszenarien denkbar wären. Der dritte Teil beschreibt die Testvorhaben und die Methoden, die zur Ermittlung der Usability angewandt werden sollen.

2. Usability

Die Untersuchung von Usability und die Abschätzung ihres Erfolges ist wichtig für ein Projekt wie TextGrid im Übergang in den Dauerbetrieb von zentraler Bedeutung; Ziel der dritten Projektphase (2012-2015) ist u.a. die Erschließung neuer Nutzergruppen. Gerade für neue Nutzer ist die reibunglose und intuitive Bedienung der Software und Dienste ein entscheidendes Erfolgskriterium.

Methodisch gesehen wird zuerst ein Vorgehen definiert, nach dem die Usability ermittelt werden soll, bevor es exemplarisch auf TextGrid angewandt wird. Dies hat zwei Gründe: zum einen soll nicht der Eindruck erweckt werden, die Tests würden von vornherein auf TextGrid maßgeschneidert werden, um gute Ergebnisse zu erzielen. Zum anderen soll die Studie im Prinzip dazu geeignet sein, auch auf andere VFUs angewandt zu werden.

2.1. Usability im allgemeinen

Usability, zu deutsch auch Gebrauchstauglichkeit, ist durch die Norm DIN/ISO 9241 standardisiert, die sich im allgemeinen mit der Interaktion zwischen Mensch und Maschine beschäftigt. Ein kurzer Abriß der dort festgehaltenen Regeln ist zum Beispiel auf der Seite kommdesign.de¹ zu finden.

Daneben gibt es viele Ansätze und Leitfäden, vor allem in Hinblick auf die Gestaltung von Webseiten und Programmoberflächen. Exemplarisch sollen die "Ten Usability Heuristics" von Jakob Nielsen² betrachtet werden. Die Auflistung folgt der Veröffentlichung von Nielsen³ :

Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

¹ <http://kommdesign.de/texte/din.htm>

² http://www.useit.com/papers/heuristic/heuristic_list.html

³ Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), [Usability Inspection Methods](#), John Wiley & Sons, New York, NY

Match between system and the real world: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention: Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall: Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use: Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Eine solche Liste an Kriterien könnte als Leitfaden zur Beobachtung von Usability in TextGrid benutzt werden. Gerade, weil sie zu einem Grad abstrakt ist, könnte man daraus gute Ansätze für eine Testreihe ableiten.

2.2. Usability in anderen VFUs

Die britische Organisation JISC⁴ hat 2010 eine großangelegte Studie zu virtuellen Forschungsumgebungen veröffentlicht.⁵ Darin werden die Bestrebungen verschiedener nationaler Forschungseinrichtungen dokumentiert und anhand von Kriterien vorgestellt. Für Deutschland sind darin als Projektträger die DFG und als Fallstudien eSciDoc und TextGrid vertreten.

Im allgemeinen Teil werden Interviews mit einzelnen Projektmitarbeitern ausgewertet. Dabei wird auch Usability angesprochen. Als größtes Problem wird hier die Tatsache bezeichnet, daß die VFU, die von Forschern genutzt werden soll, so einfach wie möglich zu bedienen sein muß, da es im Prinzip nicht darum geht, die Bedienung einer neuen Software zu lernen, sondern die Forschung zu erleichtern und neue Ansätze zu unterstützen.

Was die deutschen Fallstudien angeht, sind sowohl für TextGrid als auch für eSciDoc jeweils im Rahmen von Workshops Studien zu Usability durchgeführt worden. Im Falle von eSciDoc bezieht sich dies aber auf ein Frontend dazu, nämlich PubMan⁶. Auf die Studien in TextGrid wird weiter unten, in Abschnitt 2.5.1, eingegangen.

2.3. Herangehensweisen und Begriffsdefinitionen

Es gibt verschiedene Methoden, die Gebrauchstauglichkeit zu ermitteln. Die hier vorgestellten sind aus dem MPDL-Colab-Wiki⁷ entnommen, und geben einen Überblick über gängige Verfahren.

2.3.1. Cognitive Walkthrough

Beim Cognitive Walkthrough erhält die Testperson eine klar definierte Aufgabe, die innerhalb eines Programmes ausgeführt werden soll. Dabei werden auch die Befehle und Arbeitsschritte, die zum Lösen der Aufgabe nötig sind, einzeln aufgeführt. Die Testperson ist daraufhin angehalten, beim Lösen der Aufgabe genau zu protokollieren, was für Hindernisse aufgetreten sind.

Im Vorfeld des 1.0-Release des TextGridLab im Frühjahr 2011 wurde eine solche Evaluation durchgeführt.

2.3.2. Thinking Aloud

Bei dieser Methode sollen die Anwender Aufgaben in einem Programm lösen und ihre Arbeitsschritte dabei laut denkend kommentieren. Das Eingreifen des Interviewers ist dabei nicht vorgesehen. Dieser dokumentiert das Vorgehen und stellt gegebenenfalls Fragen, um das Lautdenken zu forcieren.

⁴ <http://www.jisc.ac.uk>

⁵ <http://www.jisc.ac.uk/media/documents/publications/vrelandscapereport.pdf>, Zusammenfassung auf <http://www.jisc.ac.uk/publications/reports/2010/vrelandscapestudy.aspx>

⁶ http://colab.mpdl.mpg.de/mediawiki/PubMan_Days_2008

⁷ http://colab.mpdl.mpg.de/mediawiki/User_Interface_Evaluation

2.3.3. Softwarebasierte Evaluation

Bei der softwarebasierten Evaluation geht es darum, die Benutzung eines Programmes am Computer selbst zu erfassen, dies kann beispielsweise dadurch geschehen, daß die Bewegungen des Mauszeigers aufgenommen und später durch eine sogenannte "Heat Map" visualisiert werden können.

2.3.4. GUI Workshops

In GUI Workshops werden Anwender gebeten, sich mit dem Design von Benutzeroberflächen auseinanderzusetzen. Dabei geht es um die Entwicklung neuer oder die Verbesserung bestehender Oberflächen. Dies kann durch das Aufmalen einer gewünschten Anordnung von Elementen auf dem Bildschirm geschehen.

2.4. Erwartungen an die Software

TextGrid beabsichtigt die Erweiterung seiner Nutzerzahl. Vor diesem Hintergrund stellt dieser Abschnitt Überlegungen vor, die das TextGridLab noch attraktiver für neue Nutzergruppen machen kann. An dieser Stelle sind zwei verschiedene Perspektiven angebracht, erstens aus der Sicht der Nutzer, zweitens aus der Sicht der Entwickler. Es sollte sich daraus ein Zusammenspiel ergeben, das sich positiv auf die weitere Entwicklung des TextGridLab auswirkt.

2.4.1. Aus Nutzersicht

Mehrere Aspekte müssen beachtet werden, wenn es darum geht, wie die Nutzer mit dem TextGridLab effizient arbeiten sollen. Im folgenden sollen zwei Herangehensweisen aus der Nutzersicht beschrieben werden. Dabei geht es einerseits darum zu definieren, wer die Zielgruppe des TextGridLab ist und deren Anwendungsvorhaben zu definieren, andererseits müssen Anwendungsszenarien beschrieben werden, die perspektivisch durch Software und Dienste abgedeckt werden sollen.

Definition der Zielgruppe – Entgegenkommen in den Voraussetzungen

Es sollte Klarheit darüber bestehen, welche Nutzergruppen mit der Software arbeiten oder dies vorhaben. Bei der anzusprechenden Gruppe – Geisteswissenschaftler – handelt es sich um eine heterogene Community verschiedener wissenschaftlicher Disziplinen mit mitunter sehr unterschiedlichen Herangehensweisen und Standards. Sofern die für die jeweilige Nutzergruppe spezifischen Aufgaben auf einen gemeinsamen Nenner heruntergebrochen werden können, muß ihr eine Arbeitsumgebung vorliegen, mit der sie ihre Aufgaben besser als zuvor umsetzen kann, und die ihren Anforderungen gerecht wird. Es muß gezeigt werden, daß das TextGridLab neue Wege eröffnet, um schneller und effizienter arbeiten zu können. Ist es ein System, mit dem Ergebnisse erzielt werden können, die vorher nur mit Mühe erreicht werden konnten?

Entscheidend ist dabei auch der technische Hintergrund, den die verschiedenen Nutzer mitbringen. Es ist nicht auszuschließen, daß diese, was die Bearbeitung von Texten angeht, erheblich von herkömmlichen Textverarbeitungssystemen wie Microsoft

Word geprägt sind. Um diesen Nutzern dann ein Wiedererkennungsgefühl zu geben, könnten sich beispielsweise die Interfaces zur Textauszeichnung an Textverarbeitungssystemen orientieren, sowohl für typographische Hervorhebungen, als auch für Kennzeichnungen von fachspezifischen Entitäten. Die Nutzer müssen vertraute Konzepte wiederfinden.

Das TextGridLab erfüllt nur dann sinnvolle Aufgaben, wenn es die Möglichkeit gibt, Aufgaben im Vergleich zu ähnlichen Programmen schneller und effizienter durchzuführen. Es muß also ein Mehrwert geboten werden, der zudem schnell ersichtlich ist und auch entsprechend beworben und durch Demos belegt werden kann. Wenig technikaffinen Wissenschaftlern, denen es primär um Forschen und die Arbeit mit Quellen geht, ist es womöglich schwer vermittelbar, warum sie ein neues Programm lernen müssen, was ihnen Zeit von ihrer eigentlichen Arbeit abzieht.

Es bleibt also festzuhalten, daß eine Festlegung auf eine Zielgruppe – und damit auf den Zweck des TextGridLab – unbedingt stattzufinden hat. Aber auch die Form von digitalen Editionen kann in den unterschiedlichen geisteswissenschaftlichen Disziplinen variieren.

Nutzerszenarien

Nutzerszenarien können helfen, den Sinn und Zweck einer Software genauer zu definieren. Wenn ein Werkzeug gemeinsam von Entwicklern und den zukünftigen Anwendern spezifiziert wird, besteht die Möglichkeit, daß die Entwicklungszeit verkürzt wird und Mißverständnisse vermieden werden. Vor allem kann genau auf die Bedürfnisse der Anwender eingegangen werden und das Produkt an das Arbeitsverhalten angepaßt werden.

Dafür müssen konkrete Szenarien definiert werden, welche Arbeiten im TextGridLab überhaupt erledigt werden können. Diese sollten sich wiederum in den Anforderungen, die die Nutzer haben, widerspiegeln.

In Abschnitt 3.2 wird auf neue Nutzerszenarien eingegangen, die zur Gewinnung neuer Nutzergruppen beitragen könnten. Gerade bei neu zu entwickelnden Komponenten könnten dann Usability-Aspekte schon während der Entwicklung berücksichtigt werden.

Während des Softwarebetriebs sollte eine ständige Rückmeldung an die Entwickler möglich sein. Derzeit wird dies über ein Bugtrackingsystem JIRA⁸ realisiert, worüber Fehler, aber auch Feature Requests und Verbesserungsvorschläge eingereicht werden können. Dieser formalisierte Weg hat vor allem einen hohen Nutzen für die Entwickler. Für die Nutzer bedeutet dies aber einen hohen Grad an Disziplin, dieses System richtig zu nutzen. Hilfreich wäre dafür auch eine Kennzeichnung von Pflichtfeldern.

2.4.2. Aus Entwicklersicht

Eingeleitet werden soll dieser Abschnitt mit einem Zitat von der Homepage von TextGrid⁹:

⁸ <https://develop.sub.uni-goettingen.de/jira/browse/TG>

⁹ <http://www.textgrid.de>

Open-Source-Werkzeuge und -Services unterstützen GeisteswissenschaftlerInnen im gesamten Forschungsprozess – insbesondere beim Erstellen digitaler Editionen.

Diese Zusammenfassung sollte als Produktvision angesehen werden, die als Leitfaden für das Programm gelten sollten. Man könnte die Vision auch noch genauer durch die Beantwortung der folgenden Fragen spezifizieren:

1. Was ist das TextGridLab?
2. Für wen ist das TextGridLab geeignet?

Genau so wichtig sind aber auch die Umkehrungen dieser beiden Fragen:

3. Was ist das TextGridLab **nicht**?
4. Für wen ist das TextGridLab **nicht** geeignet?

Diese Fragen sollten bei der Weiterentwicklung im Hinterkopf behalten werden. Beispielsweise sollte darauf hingewiesen sein, daß es kein einfacher Texteditor ist¹⁰. Vielmehr sollte darauf hingewiesen sein, daß es sich dabei um eine Virtuelle Forschungs-umgebung handelt, deren Anwendung eine nicht unerhebliche Lernkurve aufweist (in dem Sinne, daß sie auf den ersten Blick nicht selbsterklärend ist). Werden Software und Dienste jedoch beherrscht, ist damit ein effizientes Arbeiten möglich.

Gut wäre außerdem, darauf zu achten, daß das Lab in einer Vielzahl von Eigenschaften sich als aus einem Guß präsentiert: Sind die verschiedenen Werkzeuge gut integriert? Gibt es ein allgemeines Bedienkonzept, das sich auf alle Teilbereiche übertragen läßt? Da für die Zukunft geplant ist, eine Entwicklercommunity außerhalb des Kern-Entwicklerteams aufzubauen, wäre es hilfreich, wenn es Richtlinien gäbe, die bei der Erstellung von neuen Komponenten einzuhalten sind. Ein entsprechendes Dokument ist auf der Grundlage der Eclipse User Interface Guidelines¹¹ schon begonnen worden.

Das TextGridLab ist als Eclipse Rich Client Platform konzipiert worden, und dies bringt eine Reihe von Implikationen mit sich. Von seiner Ausrichtung her ist Eclipse eine Plattform für Softwareentwickler und daher auch für diese Aufgabe – dem Entwickeln von Software – bestens geeignet, effizient und intuitiv bedienbar. Es gibt eine große Menge an Projekten, die ebenfalls auf diesem Konzept aufbauen¹², wobei aber die meisten wiederum Programmierumgebungen sind.

Softwareentwickler stellen jedoch andere Ansprüche an die Benutzeroberfläche als Bearbeiter von Volltexten. Beispielsweise bietet Eclipse eine Vielzahl an separaten Fenstern (*Perspective*, *View* etc.), die dem vorgesehenen Zielpublikum vorerst fremd sein werden. Entsprechende Fenster gibt es in Textverarbeitungsprogrammen nicht. Eine interessante Frage ist daher, ob der Spagat gelingen kann, in einer solchen Um-

¹⁰ Dies wäre vergleichbar damit, in Programmen wie Gimp oder Photoshop kleine Veränderungen an Urlaubsphotos vornehmen zu wollen. Es ist möglich, aber im Grunde sind die Programme dafür nicht gedacht.

¹¹ <http://www.eclipse.org/articles/Article-UI-Guidelines/Contents.html>

¹² http://en.wikipedia.org/w/index.php?title=List_of_Eclipse-based_software&oldid=530113224

gebung ein Tool herzustellen, das für eine ganz andere Nutzergruppe gedacht ist, und diese an ein neues Oberflächenkonzept heranzuführen.

Es gibt mindestens ein Beispiel, das zeigt, wie man ein erfolgreiches Produkt auf Eclipse aufbauen kann: den XML-Editor Oxygen¹³. Dieser ist auch als Plugin verfügbar und damit kann damit auch in das TextGridLab integriert werden. Die Interaktion mit anderen Tools im TextGridLab ist aber derzeit noch begrenzt.¹⁴

2.5. Usability in Textgrid

2.5.1. Vorarbeiten in TGI und TGII

Der Abschlußbericht von TextGrid II¹⁵ beinhaltet die folgenden Punkte zur Usability:

TGII, AP2: Das TextGrid Laboratory

Ein Schwerpunkt dieses Arbeitspaketes war die Verbesserung der allgemeinen Usability im TextGridLab. Für die Implementation einiger Funktionen wurde dafür im März 2010 ein Programmiersprint veranstaltet, der die Verbesserung einiger Bedienelemente zum Thema hatte. In dessen Vorbereitung wurde eine Liste an Kritikpunkten von Nutzern zusammengestellt.

TGII, AP4: Coaching/ Community-Building

In Milestone 4.1.1 wurden die schon in TGI begonnen Arbeiten zu Usability weitergeführt. Ein besonderes Augenmerk lag dabei auf der Beschreibung von Nutzerszenarien. Es wurden auch weitere Bedarfserhebungen und Usability-Evaluationen durchgeführt, die im folgenden durch AP6 umgesetzt wurden.

TGII, AP6: Fachwissenschaftliche Nachhaltigkeit

Während der Projektlaufzeit fanden in mehrere TextGrid-Testtage statt, in denen das TextGridLab getestet wurde. Die Ergebnisse daraus wurden über das Bugtrackingsystem JIRA an die Entwickler weitergegeben. Die drei Reports aus AP6¹⁶ geben einen guten Überblick darüber, was in der Folgezeit davon umgesetzt werden konnte.

3. Nutzerszenarien

Dieser Abschnitt gibt einen kurzen Überblick über die bisherigen Nutzerszenarien, wie sie anhand eines Fragebogens auf einem Nutzertreffen erhoben wurden, sowie einen Ausblick auf kommende Entwicklungen.

13 <http://oxygenxml.com/>

14 <https://dev2.dariah.eu/wiki/display/TextGrid/Interaction+of+oXygen+with+Other+Component>

15 <http://www.textgrid.de/fileadmin/berichte-2/abschlussbericht-2.pdf>

16 <http://www.textgrid.de/fileadmin/berichte-2/report-6-0-1.pdf>,

<http://www.textgrid.de/fileadmin/berichte-2/report-6-0-2.pdf>,

<http://www.textgrid.de/fileadmin/berichte-2/report-6-0-3.pdf>

3.1. TGII – Fragebögen

Im Zuge des Nutzertreffens am 22. und 23. Februar 2012 an der TU Darmstadt wurden Fragebögen an die anwesenden Projekte Nutzer aus Projektzusammenhängen verteilt, die ein gutes Bild darüber geben, wie die derzeitige Nutzung aussieht. Daraus läßt sich gut ablesen, in welchen Fachdisziplinen Interesse an TextGrid besteht. Dies beeinflußt dann wiederum die etwaige Entwicklung der Werkzeuge und natürlich die Bedienung dieser¹⁷. Es läßt sich aber ebenso daraus ablesen, mit welchen Erfahrungen, Kompetenzen, Anforderungen und Technik-Affinität die Nutzer an das TextGridLab herangehen und welche Affinität sie beispielsweise zu XML-Editoren mitbringen.

Laut TGII-Abschlußbericht (Seiten 4-5) wurden während TGII die Nutzergruppen von Editionsphilologie und Linguistik ausgeweitet auf Musik-, Kunst- und Altertumswissenschaften sowie Judaistik. Zu erwähnen ist hier noch das Projekt TEXTvire¹⁸ am King's College, London, das versucht hat, eine eigene Instanz von TextGrid aufzusetzen.

3.1.1. Auswertung der Fragebögen

Die Fragebögen spiegeln wieder, welchen Schwerpunkt die schon laufenden oder derzeit anlaufenden Projekte haben:

- Linguistische Analyse von Texten
- Erstellung von Indizes und Registern
- Eintragen von Normdaten
- Vergleichende Analyse von Texten
- Arbeit mit Wörterbuchtools und Anreicherung dieser
- Elektronische Edition von Texten
- Verbindung von Text mit Bildern (Text-Bild-Link Editor)
- Digitalisierung von Texten, Verlinken, Kontextualisierung und Kommentierung dieser

Dies zeigt schon eine Fülle an verschiedenen Projekten, die perspektivisch das TextGridLab einsetzen wollen, darunter Blumenbach-Online¹⁹, die Digitale Archiv-Edition von Georg Greflinger²⁰ sowie das Virtuelle Skriptorium St. Matthias²¹, das schon Texte im TextGridRep veröffentlicht hat.

¹⁷ Es ist anzumerken, daß in TextGridIII keine neuen Werkzeuge entwickelt werden; statt dessen werden die bereits bestehenden Werkzeuge und Dienste optimiert und externe Tools weiter an die Nutzung im Lab angepasst.

¹⁸ <http://textvire.cerch.kcl.ac.uk>

¹⁹ <http://www.blumenbach-online.de/>

²⁰ <http://www.georggreflinger.de/>

²¹ <http://stmatthias.uni-trier.de/index.php?id=1>

3.2. Neue Benutzungsszenarien

In diesem Abschnitt werden Szenarien beschrieben, die in der Zukunft dem TextGridLab hinzugefügt werden sollen, um die Attraktivität und die Funktionalität des Systems zu vergrößern. Welche Szenarien umgesetzt werden, soll durch die Steering Group und das TextGrid-Konsortium entschieden werden.

3.2.1. Publikation

Aus dem TextGridLab heraus sollte ein Text in mehrere Formate exportiert werden können:

Zur Darstellung von Texten auf Projektseiten ist der Export in ein statisches HTML-Format am geeignetsten. Außerdem wäre damit auch die Benutzung des DFG-Viewers möglich. Diese Ausgabeoption ist am dringendsten.

Als zweiter Punkt wäre PDF für Druckausgaben auf Papier zu nennen. Möglicherweise könnte man dies auch mit einem Print-on-Demand-Anbieter (epubli²², openbookpublishers²³) für gebundene Ausgaben verbinden. Aus dem TextGridLab heraus ist der PDF-Export zukünftig mit XML-Print²⁴ möglich.

Mit dem Export in das Epub-Format könnten Texte aus dem TextGridRep auf Tablets und eBook-Readern angezeigt werden. Vorstellbar wäre diese Benutzung einer elektronischen Version eines Werkes auch in Seminaren, so daß direkt mit den elektronischen Text gearbeitet werden kann.

Die Publikation via SADE²⁵ ist derzeit schon über das entsprechende externe Plugin²⁶ möglich. Voraussetzung dafür ist eine laufende SADE-Instanz. Wenn vorhanden, kann über die Funktion "SADE Publish" das aktuelle Projekt publiziert werden. Zusammen mit der SUB Göttingen wird an der Optimierung für die TextGrid-Umgebung gearbeitet.

3.2.2. Annotationen

Annotation ist in diesem Falle so zu verstehen, daß ein Text mit weiteren Anmerkungen versehen wird, die über eine strukturelle Auszeichnung hinausgehen. Dies bezieht sich insbesondere auf intellektuelle Aussagen über den Inhalt des Textes und Verweis auf externe Ressourcen oder andere Texte im TextGridRep. Idealerweise würden diese Annotationen extern gespeichert werden und nur per ID auf die Textstelle verweisen. Damit wäre es möglich, die Annotationen in den verschiedenen Repräsentationen (im TextGridLab, im Repository, in Editionen) des Textes anzuzeigen.

Ein weiteres Desiderat dabei wäre es, diese Annotationswerkzeuge auch im TextGridLab zur Hand zu haben, so daß einerseits schon während der Texteingabe Annotationen gemacht werden können, oder andererseits beim späteren Bearbeiten

22 <http://www.epubli.de/>

23 <http://openbookpublishers.com/>

24 <http://sourceforge.net/projects/xml-print/>

25 <http://www.bbaw.de/telota/software/sade/sade-1>

26 <http://textgridlab.org/marketplace/content/1>

eines schon veröffentlichten Textes die zwischenzeitlich angefertigten Annotationen berücksichtigt werden können (wenn es zum Beispiel um Korrekturvorschläge geht).

3.2.3. Dezentrale Kollaboration

Die Echtzeitkollaboration ermöglicht das zeitgleiche, aber dezentrale Arbeiten an demselben Text innerhalb des Labs. Dies ist durch das Freischalten von Bildschirmansichten möglich, aber durchaus technisch aufwendig und möglicherweise instabil. Fraglich ist auch, ob dies eine gängige Arbeitsweise von Geisteswissenschaftlern ist.

3.2.4. Offlinearbeit

Vor allem auf Reisen kann nicht immer eine Verbindung mit dem Internet gewährleistet werden. Daher sollten grundlegende Funktionen (Zugriff auf lokal gespeicherte Daten) des TextGridLab auch ohne Internet möglich sein. Es gibt zwar derzeit die Möglichkeit, Objekte lokal zu speichern und zu einem späteren Zeitpunkt wieder in das TextGridLab zu importieren, aber dieser Vorgang muß ausdrücklich vom Benutzer durchgeführt werden. Eine spätere Synchronisation des lokalen Standes mit der Version aus dem Repositorium müßte dann eingerichtet werden. Ohne die Möglichkeit einer Offlinearbeit besteht die Gefahr, daß die Benutzer sich anderer Mittel bedienen und den Text lokal speichern und in einem anderen Programm bearbeiten, während sie offline sind.

3.2.5. Navigation von XML-Texten im Browser

Es sollte eine Möglichkeit geben, XML-Texte innerhalb eines Browsers bequem zu lesen und bei Bedarf auch kleine Korrekturen am Text vorzunehmen. Bequemes Lesen wird ermöglicht durch Rendern des Textes mit CSS und XSL sowie durch Ein- und Aufklappen des XML-Baumes. Besonders, wenn der XML-Text eine viele unterschiedliche Auszeichnungen enthält, wäre auch eine selektive Anzeige einiger Tags hilfreich. Zum Teil ist dies schon durch die WYSIWYM-Ansicht gegeben, jedoch stellt die Anpassung einer XSL-Datei an die Gegebenheiten eines bestimmten Projektes eine große Hürde dar.

3.2.6. Visualisierung und diagrammatische Darstellung

Für einige Projekte wäre die Möglichkeit, Zusammenhänge zwischen Objekten graphisch darzustellen, eine große Hilfe. Ein entsprechender Editor, der im Hintergrund auf Standards wie GraphML oder DOT²⁷ basiert, wäre hierfür und für eine eventuelle spätere Visualisierung sehr hilfreich.

Ein Anwendungsbeispiel wäre die Ordnung von Manuskripten und ihre Beziehung zueinander, in Bezug auf ihre zeitliche Abfolge (Briefe).

3.2.7. Kommunikation

Allgemeine Kommunikation zwischen Benutzern könnte via ein Chatsystem implementiert werden, was in das Lab eingebaut wird. Auf diese Weise können kleinere

²⁷ http://en.wikipedia.org/w/index.php?title=DOT_language&oldid=534787386

Textschnipsel oder URLs schnell zwischen Computern und Arbeitsplätzen ausgetauscht werden. Der Zugriff auf projektspezifische Chatrooms ist über die Benutzerverwaltung möglich.

3.2.8. Communitybuilding

Eine Social-Network-Komponente in TextGrid könnte zum Aufbau von neuen Nutzercommunities genutzt werden. Denkbar wäre hier ein Aufbau eines virtuellen Kollegenkreises um bestimmte Forschungsinteressen herum oder auch ein Empfehlungssystem in der Art von "Benutzer, die diesen Text bearbeitet/gelesen/annotiert haben, haben auch diesen Text bearbeitet/gelesen/annotiert".

3.2.9. Konvertierung von Texten nach XML

Das Datenformat der Texte, die im TextGridLab bearbeitet werden sollen, ist vermutlich äußerst heterogen (Reiner Text, PDF, Word). Darum muß versucht werden, verschiedene Workflows zu definieren, wie diese Textdokumente in ein geeignetes Format überführt werden können. Da die einzelnen Texte aber immer Besonderheiten der bisherigen Bearbeiter aufweisen können, ist hier eine vollautomatische Konvertierungslösung nicht umsetzbar.

Für den Fall, daß Text direkt im TextGridLab produziert wird (Transkription von Manuskripten), sollte hier das Baseline Encoding und die Autovervollständigung des Editors benutzt werden.

4. Identifikation der Testgruppen

Innerhalb des Max-Planck-Instituts für Wissenschaftsgeschichte wird es eine kleine Anzahl von Projekte geben, die sich in Zukunft intensiv mit dem TextGridLab und der Arbeit damit auseinandersetzen wird. Das beinhaltet die Aufarbeitung schon digitalisierter Bestände in TextGridLab (Konvertierung nach TEI-XML) und soll bis zu einem kompletten Editionsprojekt voranschreiten. Konkret geht es hier um Manuskripte von Galileo Galilei und die Korrespondenz von Moritz Schlick.

Zusätzlich dazu wird es eine Zusammenarbeit mit der Berlin-Brandenburgischen Akademie der Wissenschaften geben, die schwerpunktmäßig Editionen (Werke, Epigraphik) des Akademienprogramms sowie einige kurzlaufende Editionsprojekte aus anderen Förderzusammenhängen untersuchen werden.

4.1. Methodisches Vorgehen

Die in Abschnitt 2.1 vorgestellten Richtlinien sowie die in Abschnitt 2.3 beschriebenen Verfahren werden als Grundlage genutzt werden, um die neuen Nutzergruppen in ihren ersten Schritten mit TextGrid zu beobachten und ihre Beurteilung des Programmes, und Wünsche und Anforderungen aufzunehmen.

Dazu sollen die folgenden Vorgehen zum Einsatz kommen.

4.1.1. Fragebogen

Wie schon im Rahmen des Nutzertreffens vom 22. und 23. Februar 2012 könnten für die teilnehmenden Nutzergruppen Fragebögen erarbeitet werden. In diesem Fall würden sie aber dafür genutzt werden, verschiedene Usability-Fragen zu erörtern. Dabei könnten erste Performance- und Akzeptanzdaten gesammelt werden.

4.1.2. Beobachtung bei der Arbeit – Cognitive Walkthrough/ Thinking Aloud

Es ist außerdem ist sinnvoll, die Benutzer direkt bei der Arbeit mit TextGrid in ihrer gewohnten Umgebung zu beobachten. Dies kann einerseits bei der Teilnahme an Workshops geschehen, bei denen kollaborativ an einem Projekt gearbeitet wird. Andererseits kann dies auch am einzelnen Arbeitsplatz mit den oben (Abschnitt 2.3) beschriebenen Methoden getan werden. Auch hierzu müssten dann entsprechende Aufgaben definiert werden.

4.1.3. Interviews

Weiterhin könnte in Erwägung gezogen werden, durch Einzelinterviews die Bewertung des TextGridLabs vorzunehmen. Zu den vorangegangenen eher quantitativ ausgerichteten Methoden wäre dies noch eine Möglichkeit, ein eher qualitatives Feedback einzusammeln.

Um die daraus entstandenen Bedürfnisse, Fehlermeldungen und Wünsche an die Entwickler weiterzugeben, ist es natürlich auch sinnvoll, die Funktion des Bugtrackingsystems zu erklären und zu dessen Benutzung zu ermutigen.