

TextGrid – the Community-Grid for the Arts and Humanities



www.TextGrid.de

The TextGrid Infrastructure Guidelines for Cooperation Partners

Version 1.1

Table of Contents

About these Guidelines	3
The Project	3
Motivation	3
Collaborative Approach	4
Design Principles	4
TextGridLab—the Toolkit	4
Graphical User Interface.....	4
TextGrid Tools	5
TextGridRep—Data Handling and Data Preservation	7
Use of TextGrid.....	7
Cooperation Scenarios	7
Appendix: TextGrid Technology	10
Overview.....	10
GUI.....	10
Tools / Services.....	10
Middleware and Grid Integration.....	10

About these Guidelines

This publication is intended for potential cooperation partners that want to use the research infrastructure developed by TextGrid or participate in its further development. It offers an overview of the TextGrid project, its infrastructure, and the components of the TextGrid software in order to outline possible scenarios of cooperation. Technical details are provided in the appendix. Please note that this text intends to prepare cooperation. Contract details and the mutual responsibilities of future cooperation partners have to be negotiated separately.

The Project

TextGrid¹ is the first project in Germany to create a grid-based research infrastructure for the arts and humanities. Grid technology is gradually coming into use in the field of arts and humanities on an international level. There are a number of notable e-Science or e-Research initiatives and projects.

TextGrid is actively involved in the creation of a German e-Humanities infrastructure. It is interconnected with a project launched in January 2008 whose aim it is to draw up a master plan of an e-Humanities initiative for Germany. The project is funded by the German Research Foundation (DFG). The BMBF is also interested in a functional e-Humanities infrastructure and has, with the “Agenda 2020 for e-Humanities in Germany”, initiated discussion of a long-term roadmap for establishing a virtual research environment for the arts and humanities.

Being the only D-Grid² project in the field of arts and humanities, TextGrid represents and advocates the special needs and perspectives of the humanities in D-Grid. It is obvious that grid technologies are important for the humanities. While the World Wide Web usually provides more or less static information, the grid enables users to gain direct access to resources (computers, storage, research instruments, experiments, applications, and data). Grid infrastructures offer a number of advantages to researchers: unified access to distributed resources, virtually unlimited computing and storage capacities, and a great amount of flexibility as a consequence of the dynamic and concerted operation of networked resources. In the arts and humanities, resources are usually data and services (applications). However, CPU-intensive processes like migrating data for long-term preservation or OCR of large data sets are becoming more and more common in the humanities. Consequently, the significance of virtualized applications using access to distributed hardware resources is growing in this sector. However, long-term preservation, knowledge management, and the processing of text data are of fundamental importance in the arts and humanities.

Motivation

Text researchers analyzing the relations between language and discourse and into the complex processes in the production of literature still mostly work in local systems and project-oriented applications. Current research initiatives also lack integration with already existing text corpora, and they remain unconnected to resources such as dictionaries, lexica, secondary literature and text processing tools. This integration and interconnection, though, bears a wealth of opportunities. With its architecture and integrated tools and services that satisfy, for the time being, the specific requirements of text sciences, TextGrid is able to provide such forms of integration. The need for the installation of a grid-enabled architecture in the e-Humanities is obvious. Past and current text

¹ Funded by the German Federal Ministry of Education and Research (BMBF) as part of the D-Grid initiative, reference number 07TG01A-H. More information on TextGrid at <http://www.textgrid.de>.

² The German grid initiative, <http://www.d-grid.de>.

digitizing initiatives have already accrued a considerable data volume that exceeds hundreds of terabytes with more to be expected in the future. Grids are capable of handling these data volumes³.

Collaborative Approach

Grid structures can effectively make up for the dispersal of the community as well as the scattering of resources and tools. TextGrid provides a grid-based, modular infrastructure that lays the foundation of a collaborative virtual research environment. Any arts-and-humanities specialist can adopt TextGrid to initiate or join collaborative work groups. In its core functionality, however, TextGrid is, at this stage, focused on German or Western language edition philology and linguistics. Thus, TextGrid serves to collaboratively process, analyse, annotate, edit, and publish text data.

Design Principles

Designing the TextGrid architecture these three principles were observed:

- (1) Ease and intuitivity of use.
- (2) Simplicity of adaptation to individual needs.
- (3) Integration of extant infrastructures and tools.

In order to achieve the first goal, the graphical user interface of TextGrid has been carefully designed. Test series involving numerous specialists from the arts and humanities will be conducted in September 2008. The architecture of TextGrid has been designed to conceal complex grid processes from the user and encapsulate them in deeper service layers. Users usually are not interested in any details of data storage and replication processes. They want data to be reliably accessible, easily searchable, and securely protected against unauthorized access. The second and the third goal are achieved by the strict employment of open standards in a modular system of cross-platform open-source components. For details see appendix.

TextGridLab—the Toolkit

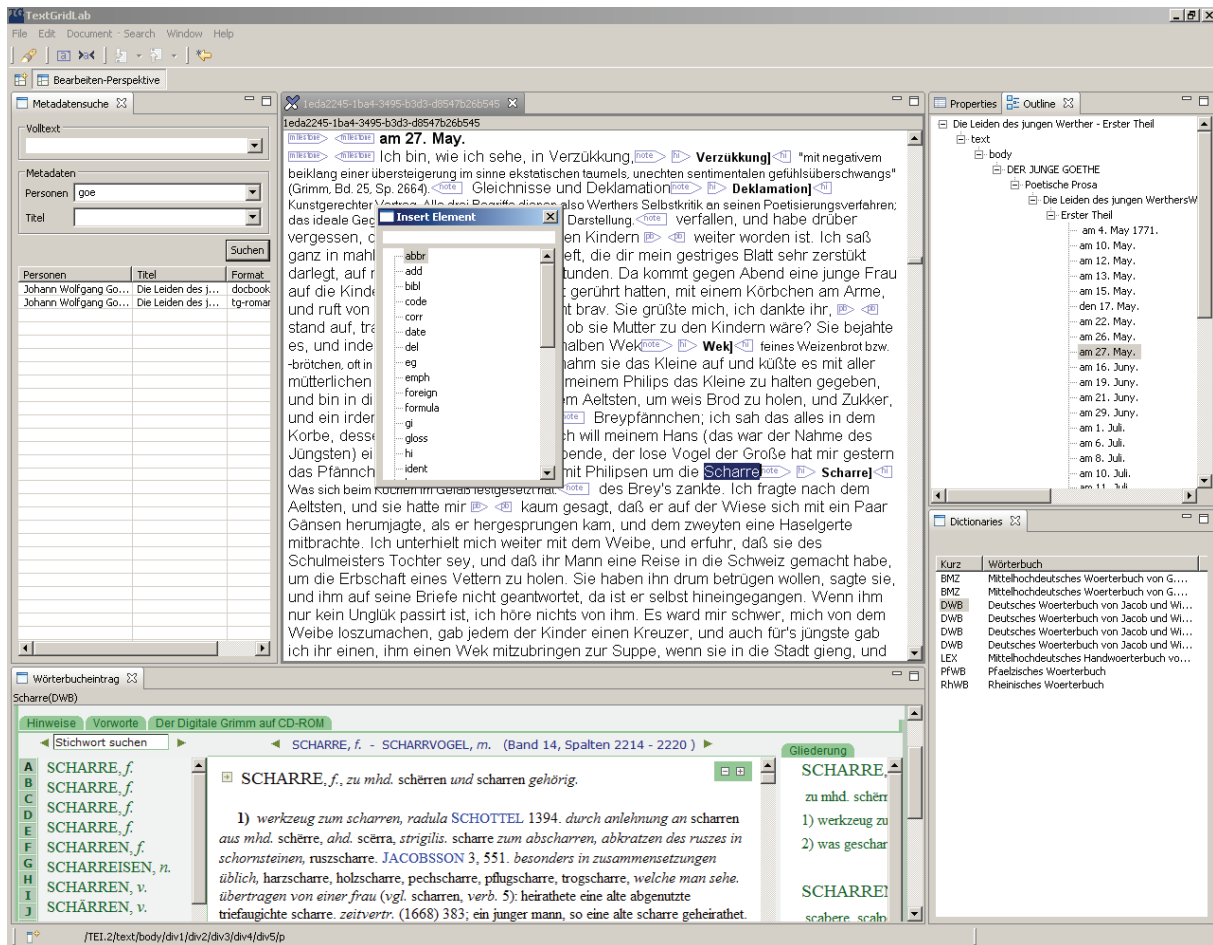
TextGridLab (Laboratory) offers the users integrated access to all TextGrid tools (see below). It is based upon an open network of open-source tools that is easily extensible and adaptable for new user groups.

Graphical User Interface

The graphical user interface of TextGrid is based on Eclipse, an open-source software development environment. A number of IT firms like Borland, IBM, Intel, Oracle, and SAP are involved in the development of Eclipse.

The central element of TextGridLab is the XML editor. Other elements like the workflow editor, the project management module, the authentication and rights management module, the searching tool, or the control elements of the specific tools are grouped around the XML editor. The grouping can be configured according to the individual needs of the respective project.

³ Important criteria: reliability, high availability, and fast access.



Prototype of the graphical user interface (from left to right: search form, XML editor, outline view, interface of the Wörterbuchnetz Trier)

TextGrid Tools

Within the funding term of the TextGrid project, the following tools⁴ have been developed as a basic TextGrid configuration:

Interactive Tools

XML Editor – The XML Editor is the editor for XML files that reside on the local computer or in the grid. Users can switch easily between a more technical view with tags and attributes and a structural view that is oriented towards the WYSIWYG display in common text processing applications.

Link Editor Text (text ↔ text) – The Link Editor Text is a helper tool for creating links in XML files. It interconnects the Search Tool with the XML Editor. With the Link Editor Text, users can generate links to elements of any TextGrid document within the current (TEI) file.

⁴ For a detailed functional description (in German) see Reports 2.1 and 2.2 at http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_2_1.pdf and http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid-R2.2_ToolsII.pdf.

Graphic Link Editor (text ↔ image) – The Graphic Link Editor supports the XML Editor to align text sequences with image sections in order to create files that contain text elements and topographic descriptions.

Workflow Editor – The Workflow Editor allows the user to configure individual automated workflows.

Search Tool – The Search Tool serves to retrieve structural data and metadata. Semantic searches based on RDF are also possible.

Collationer – The Collationer compares two or more files encoded in XML (or TEI⁵) and annotates any differences in TEI format. This tool can be run in batch mode (→ Streaming Tool).

Project Management (file and rights management module) – New projects can be created in the Project Management Module. Existing projects can be managed, i.e. project members can be deleted or added and endowed with predefined roles; access rights for TextGrid objects can be set.

Project Browser / Navigation Tool – The Project Browser is always present in TextGridLab. It gives the user easy access to all materials related to the project he is working at. Thus, the Navigator can be a starting point in TextGridLab.

Metadata Annotator – A generic tool used for adding structured metadata to TextGrid objects (texts, images etc.) by means of a configurable input form.

Bibliographic Tool – The Bibliographic Tool imports bibliographic data from existing data sets, especially library catalogue records or TEI headers of other (TextGrid) objects. The tool can be used to compile bibliographies, edit and manage them, insert them anywhere in a TextGrid file, or export them using standard formats (e.g. TEI, MODS).

Streaming Tools

Streaming tools are non-interactive tools that are run as batch processes. They do, however, usually have an interactive GUI component in order to be easily configurable.

Tokenizer – Sequences a text into a stream of logical units (tokens), i.e. words and punctuation elements, and marks them with start and stop marks. The tokens to be used can be defined in the tool configuration. The definition of predefined tokens like abbreviations, proper names, or regular expression (e.g. for calendar dates) can also be adjusted to one's needs.

Lemmatizer (morphological analysis) – The Lemmatizer analyzes word (German) forms (tokens) and returns (a) the respective lemma, i.e. the token is reduced to its basic morphologic form, (b) the part of speech, and (c) other morphologic characteristics (number, gender etc.).

Sorting Tool – With the Sorting Tool, the user can sort strings of characters according to cultural and subject-specific standards. The sort keys and sorting rules to be observed can be freely configured by the user or set to the respective national or European conventions.

Streaming Editor – The Streaming Editor enables rule-based file transformation, e.g. enrichment of potentially unstructured texts with XML structures. Input files do not have to be in XML, but can be any text format (e.g. OCR raw text), output may be XML or any other text format.

Text Publisher Web – The web publisher can be used to present project results and publications on a project website. TextGrid provides standard components for web publishing and an interface to TextGridRep (see below).

⁵ TEI = Text Encoding Initiative (<http://www.tei-c.org>), an XML standard for annotating source texts.

TextGridRep—Data Handling and Data Preservation

TextGridRep (Repository) is TextGrid's data storage component responsible for structured storage and long-term preservation of scholarly texts in the grid. At check-in, each text is automatically augmented with metadata and indexed.

Data available in the grid can be processed, managed, assigned to projects, protected with access rights, and enhanced with metadata in TextGridLab. TextGrid projects and grid objects (texts, images etc.) can be navigated with the Project Browser. Objects are accessed via their metadata and, when searching, via their text data (full text and structured search⁶).

In order to access data in TextGrid, they do not necessarily have to reside in the grid. For data integration we have conceived of a three-stage model: data access via an interface for searching and text retrieval⁷, data access via metadata imported into TextGrid, full integration of data in the TextGrid infrastructure.

Additionally, TextGrid qualifies for long-term data preservation, the policies of which are being currently developed.

Use of TextGrid

Projects can combine TextGrid services into standardized workflows⁸ and modify TextGrid applications (tools) according to their individual needs. More complex modifications of standard tools could be interesting for other projects, especially if modifications result in new functionalities or if new tools or services are developed. TextGrid strives to support all forms of use with detailed documentations and easy-to-use examples. There also will be an appropriate information infrastructure for TextGrid users (help-desk portal, discussion forum etc.).

Cooperation Scenarios

This paragraph describes in an exemplary way some possible use cases or cooperation scenarios. Cooperation can happen at different levels of the TextGrid architecture outlined in diagram below. The third column in the diagram represents further subjects in the arts and humanities, e.g. other philologies, history, or cultural studies.

- (1) A project wants to use the existing software infrastructure. The GUI of TextGridLab (red) has to be installed locally in order to access pre-configured net services (green) and process one's own locally stored data. The TextGrid middleware (blue) enables read access to public file objects.
- (2) If a project wants additional access to the file services of the TextGrid middleware (blue), i.e. read, modify, save or delete, or add metadata in the central metadata base, the project members have to authenticate against the account of their home institution or a TextGrid account. Then, a TextGrid project can be created in which the project files can be organized. The project manager can add or delete project members and assign them roles and access rights.

⁶ XML data, for instance, can be searched for using text-class specific mark-up (for the concept of TextGrid's baseline encoding see http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Kerncodierung_070615.pdf).

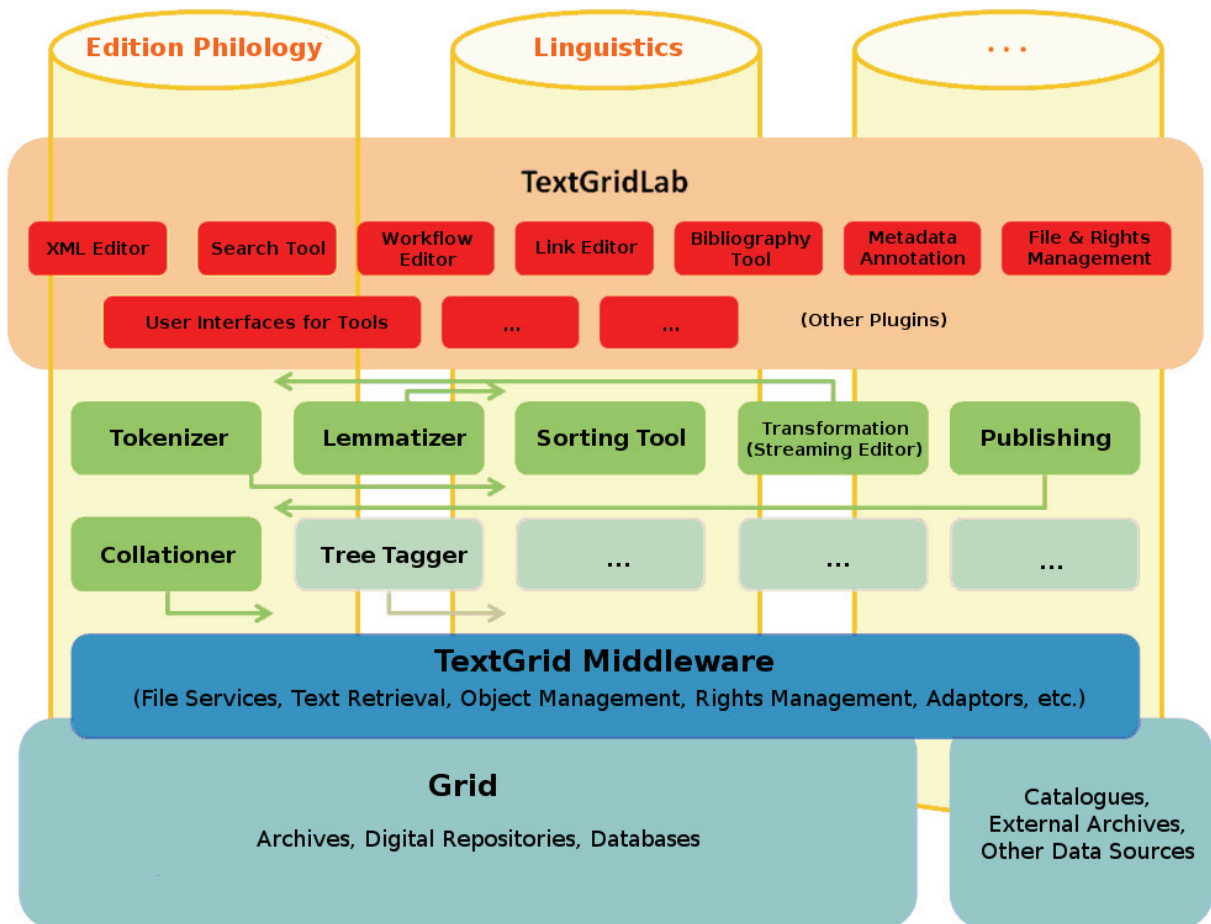
⁷ An example of data integration is the incorporation of the Trier *Wörterbuchnetz* into TextGrid via a web service interface (<http://www.woerterbuchnetz.de>).

⁸ For a survey of workflows supported by TextGrid at this stage see http://www.textgrid.de/fileadmin/TextGrid/TextGrid-Szenarien_061212.pdf.

- (3) Tools created by TextGrid projects can be integrated in the overall infrastructure and made available for general use. If a tool does not have to interact with the TextGrid middleware, it is sufficient to provide a web services interface that manages the transfer of input and output files and configuration parameters (the process level is entirely in the green area). If the tool is to be controlled directly via the GUI (i.e. not only indirectly via the Workflow Editor), an additional Eclipse plugin (red) has to be developed and integrated. If middleware services (blue) are to be used, TextGrid-specific parameters (authentication and logging information) have to be integrated in the web service.
- (4) Using a bulk upload tool or the Workflow Editor, a project can easily import an entire data archive into TextGridRep. Metadata to be added to the files have to be provided. Possibly, automatic metadata extraction from image files or metadata bases can only be accomplished through extra programming efforts. As mentioned earlier, several levels of integration with TextGrid are possible. Only full integration, however, offers all advantages of file processing and information retrieval.
- (5) An institution/organization builds its own grid infrastructure and uses the open-source TextGrid software within this structure. Cooperation with TextGrid, though not strictly necessary, is advisable as far as application development is concerned.

The TextGrid project team is glad to help, at any time, with information and support. We also accept development proposals.

For more information see project reports and publications at <http://www.textgrid.de>. Please send requests per email to info@textgrid.de.



Appendix: TextGrid Technology

Overview

TextGrid users can combine modules into project-specific workflows and replenish the existing tools with their own applications. The components of the user interface can also be combined individually. It is possible to integrate external data completely into the TextGrid storage infrastructure.

In order to facilitate the utmost level of interoperability, TextGrid employs open standards: TEI⁹ and the XML family¹⁰, RDF¹¹, SOAP¹², WSDL¹³, GSI¹⁴, WSRF¹⁵, SAML¹⁶, and LDAP¹⁷.

GUI

The graphical user interface (GUI) of *TextGridLab* is based on Eclipse¹⁸ and its framework for the development of graphical interfaces (the so-called *rich client platform*). Its components (e.g. tool-specific controls) are realized in individual modules, so-called *plugins*, that communicate via a mutual interface. The flexible *plugin* architecture allows the GUI to be configured and extended according to individual needs. *TextGridLab* is a cross-platform Java application (Windows, Linux, Mac OS X).

Tools / Services

TextGrid counts on a *service-oriented architecture* (SOA), a modular system of distributed, platform-independent *open-source* components that can be addressed as web services¹⁹. Existing applications can be integrated with the TextGrid infrastructure relatively easily by wrapping them as web services. Respective libraries exist for most programming languages. Services are accessed via internet protocols²⁰ so they can run on any server and any operating system (Windows, Linux etc.). Only the URL and the necessary parameters have to be known.

Middleware and Grid Integration

The TextGrid middleware is the mediating layer between TextGrid-specific applications, the grid, and external data sources. The interface to *TextGridLab* is modelled with utilities for data management, searching, authentication and authorization²¹, and logging. For searching and text retrieval purposes, XML data bases²² are used that store metadata and structural data redundantly (i.e. independently from

⁹ Text Encoding Initiative, <http://www.tei-c.org>.

¹⁰ For an overview see <http://www.w3.org/Talks/2002/ij-italy/slide11-0.html>.

¹¹ Resource Description Framework, <http://www.w3.org/RDF/>.

¹² XML Protocol, <http://www.w3.org/2000/xp/Group/>.

¹³ Web Services Description Language, <http://www.w3.org/2002/ws/desc/>.

¹⁴ Grid Security Infrastructure, <http://www.globus.org/security/overview.html>.

¹⁵ Web Services Resource Framework, <http://www.oasis-open.org/committees/wsrf/>.

¹⁶ Security Assertion Markup Language, <http://www.oasis-open.org/committees/security/>.

¹⁷ Lightweight Directory Access Protocol, <http://www.openldap.org/>.

¹⁸ A Java-based open-source development environment, <http://www.eclipse.org>.

¹⁹ Cf. <http://www.w3.org/2002/ws/>.

²⁰ For instance SOAP (<http://www.w3.org/2000/xp/Group/>) and REST, which is restricted to the vocabulary of the HTTP standard.

²¹ Authentication and authorization for file system processes are accomplished by means of a role-based access rights management system (RBAC, <http://csrc.nist.gov/groups/SNS/rbac/>).

²² At this stage: eXist (<http://exist.sourceforge.net>).

the metadata included in the grid-stored files). These data bases can be addressed via XQueries²³. An RDF database²⁴ facilitates semantic searches (i.e. searches for object relations).

²³ W3C standard of XML database queries, <http://www.w3.org/TR/xquery/>.

²⁴ At this stage: Sesame (<http://www.openrdf.org>).