TEI documents in the grid

Andrea Zielinski
Institut für Deutsche Sprache Mannheim, Germany

Wolfgang Pempe
Saphor GmbH, Germany

Peter Gietz, Martin Haase and Stefan Funk
DAASI International GmbH, Germany

Christian Simon
Institut für Deutsche Sprache Mannheim, Germany

Abstract

This article describes the life cycle of a TEI Document within TextGrid, an eHumanities platform for scholarly text processing, in which structured search is based on the TEI framework and metadata with restricted values. A workbench is provided that offers tools for handling TEI documents, *TextGridLab*, making it easier to annotate, process, search, and persistently store new digitized texts. The digitization and annotation of the Campe dictionary[1] serves as a first test bed. The overall framework of **TextGrid** is very generic and can handle different types of text (literary editions, linguistic corpora, lexica) as well as heterogeneous data formats (plain text, XML/TEI, images). In fact, the TextGrid repository, *TextGridRep*, is designed as a digital virtual library over federated archives, where humanities projects are invited to participate. Sharing of data is enabled by means of a grid-based architecture. Specifically the middleware includes most of the treatment of authorization, search, and file management. TextGrid is entirely based on open source software including Eclipse[2] and Globus Toolkit.

Correspondence:
Andrea Zielinski, FIZ Karlsruhe, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany.
E-mail:
andrea.zielinski@fiz-karlsruhe.de

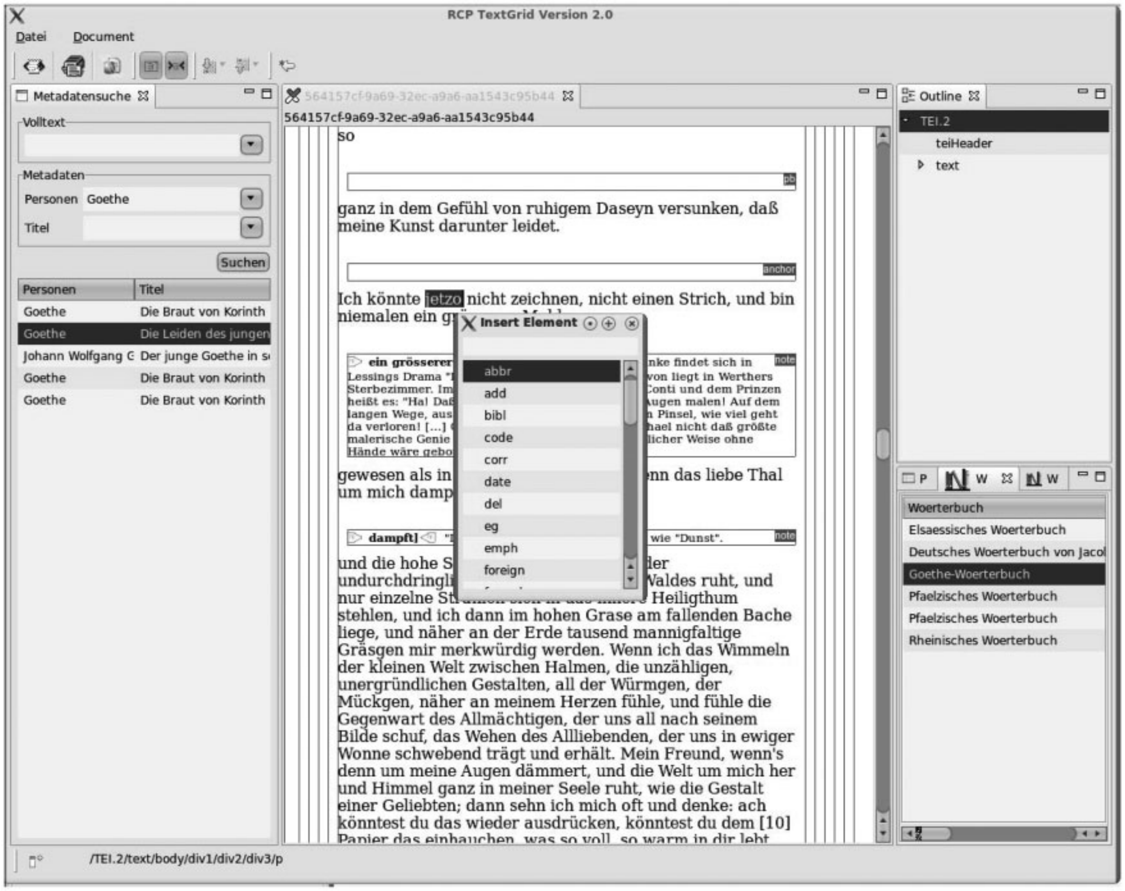
1 Introduction

This article focuses on the grid-based infrastructure for the collaborative editing, annotation, search, analysis, and publication of specialist texts, which is being developed by the TextGrid project.[3] As part of the German D-Grid[4] initiative for a three year period starting from February 2006, TextGrid is creating a community grid for the Humanities.

Coordinated by the Göttingen State and University Library, five institutional partners (Technical University Darmstadt; Institute for the German Language, Mannheim; University of Trier; University of Applied Sciences, Worms; University of Würzburg), and two small commercial companies (DAASI International, Tübingen and Saphor, Tübingen), TextGrid aims to create a *Virtual Research Library* providing the necessary tools, infrastructure, and the initial data repositories for an Open Source and Open Access environment using TEI markup.

TextGrid is committed to the advancement of e-Humanities[5] and is connected with international activities in this respect, such as DARIAH[6] and CLARIN.[7]

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012



Personen	Titel
Goethe	Die Braut von Korinth
Goethe	Die Leiden des jungen
Johann Wolfgang G	Der junge Goethe in s
Goethe	Die Braut von Korinth
Goethe	Die Braut von Korinth

564157cf-9a69-32ec-a9a6-aa1543c95b44

so

ganz in dem Gefühl von ruhigem Daseyn versunken, daß meine Kunst darunter leidet.

Ich könnte jetzo nicht zeichnen, nicht einen Strich, und bin niemalen ein g

ein grösserer ... nke findet sich in Lessings Drama "... von liegt in Werthers Sterbezimmer. Im ... Conti und dem Prinzen heißt es: "Ha! Daß ... Augen malen! Auf dem langen Wege, aus ... n Pinsel, wie viel geht da verloren! [...] ... hael nicht daß größte malerische Genie ... licher Weise ohne Hände wäre gebo

gewesen als in ... nn das liebe Thal um mich damp

dampft] "... wie "Dunst".

und die hohe S ... der undurchdringli ... Waldes ruht, und nur einzelne St ... Heiligthum stehlen, und ich dann im hohen Grase am fallenden Bache liege, und näher an der Erde tausend mannigfaltige Gräsgen mir merkwürdig werden. Wenn ich das Wimmeln der kleinen Welt zwischen Halmen, die unzähligen, unergründlichen Gestalten, all der Würmgen, der Mückgen, näher an meinem Herzen fühle, und fühle die Gegenwart des Allmächtigen, der uns all nach seinem Bilde schuf, das Wehen des Allliebenden, der uns in ewiger Wonne schwebend trägt und erhält. Mein Freund, wenn's denn um meine Augen dämmert, und die Welt um mich her und Himmel ganz in meiner Seele ruht, wie die Gestalt einer Geliebten; dann sehn ich mich oft und denke: ach könntest du das wieder ausdrücken, könntest du dem [10] Papier das einhauchen, was so voll, so warm in dir lebt

/TEI.2/text/body/div1/div2/div3/p

Fig. 1 Screenshot of the prototypical user interface. From left to right: Query interface, XML editor, interface to the Trier dictionary network (Wörterbuchnetz, http://www.woerterbuchnetz.de)

1.1 The life cycle of a TEI document

In the following sections the possibilities of the infrastructure and the general architecture are explained by means of the description of the life cycle of an TEI document, making its way from the creation (or modification) in the GUI front-end through the layers of the TextGrid architecture to the data repositories from where it is accessible for the various aspects of scholarly work.

2 Creating and Editing Content

One of the main objectives of TextGrid is to keep things as simple as possible. Thus, all actions can be performed using the eclipse-based GUI, TextGridLab (Fig. 1). Besides the client-plug-ins for the various streaming tools, the Lab provides plug-ins for the interactive TextGrid tools: project manager, project browser (a resource-navigation tool), interface for the management and collection of bibliographic data, query interface—and the XML-Editor, the central element. It allows the creation and modification of content and provides several useful features such as real-time validation, different display modes (plain text, document structure, WYSIWYM[8]), a link editor, and other useful features. In the case of creating a new (TEI-)XML file, the user has to save it as an object to the grid. Because a TextGrid object needs metadata, a further

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

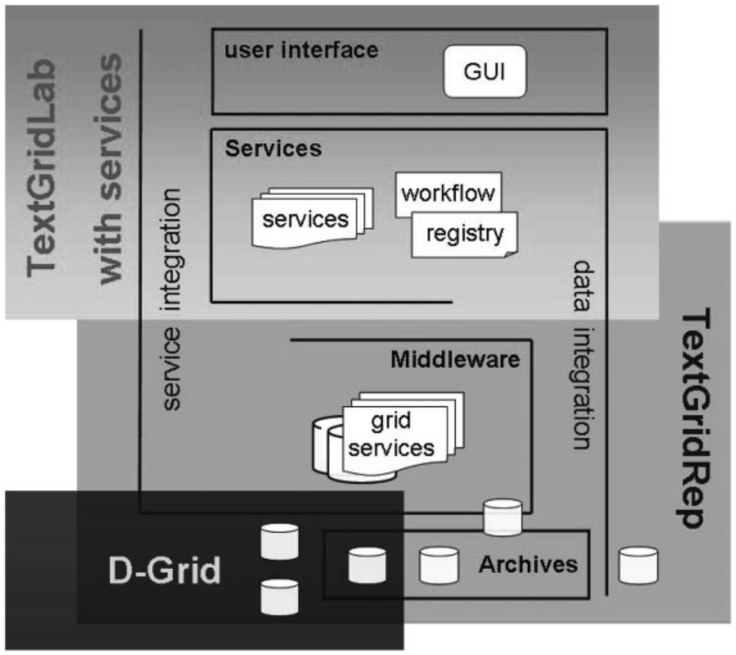


Fig. 2 The TextGrid architecture

tool, the metadata editor—displaying the project defaults—enables the user to type in the necessary information. The project management tools enable the user to set reading and writing permission—a crucial feature in a collaborative environment.

3 The Way into the Grid

While it is possible to use the TextGridLab offline on a local computer (e.g. when sitting in the train), its full power will be unleashed when moving onto the net. Then the Lab turns into a client that can access online text processing tools and the whole of the TextGrid middleware. The various middleware utilities allow for collaborative projects, controlled dissemination and publication of texts or other objects, efficient search over metadata, and safe and long-term storage in the grid.

3.1 The TextGrid architecture

The TextGrid infrastructure is a multilayered system created with the motivation to hide the complex grid infrastructure (Fig. 2), from the scholars and to make it possible to integrate external services with TextGrid tools. Basically in this service-oriented architecture (SOA), there are three layers: the user interface, a services layer with tools for textual analysis and text processing, and the TextGrid middleware, which itself includes multiple layers.

As an interface to the user, the TextGridLab has recently been developed. It integrates all interactive tools as well as interfaces to the single services/tools and to selected functions of the middleware such as the authentication and authorization module TG-auth or the metadata database module TG-search. This client platform communicates with other parts of the system via standard SOA protocols like SOAP/WSDL and REST.

3.2 The TextGrid middleware architecture and the TextGrid utilities

As already mentioned, the middleware is divided into several utilities that enables the integration of grids. Grid computing has been established in the sciences to seamlessly give access to massively

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

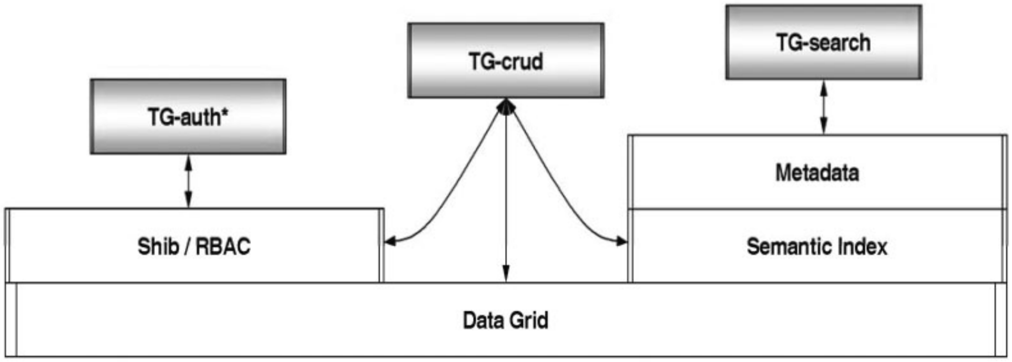


Fig. 3 TextGrid utilities

distributed resources like computing resources (CPUs) and storage resources (hard disks) hiding the complex infrastructure via a service interface just as the simple power outlet hides the complex power grid infrastructure. TextGrid has set up a storage grid infrastructure by deploying Globus Toolkit 4.[9] The TextGrid middleware currently accesses the Globus Toolkit via the JavaGAT,[10] which in future will be replaced by the SAGA API currently standardized at the Open Grid Forum.[11] Since the evolving standards for grid computing like Web Service Resource Framework (WSRF) are not easy to deploy in programs, partly due to the lack of production ready program libraries, TextGrid decided to encapsulate this grid infrastructure by its own middleware layer that is accessible via standard Web Service technology which is easier to deploy. The TextGrid middleware consists of a number of such utilities (Fig. 3), mainly:

(1) *TG-crud* is an interface for creating, reading, updating, and deleting TextGrid objects (which in most cases are TEI encoded documents). This utility creates and updates replicated resources in the storage grid as well as stores the metadata in a central XML database (eXist[12]), in the access control database of TG-auth and optionally in a RDF-based relation database (Sesame[13]). It also converts the TEI documents into the TextGrid baseline encoding (Section 5.2) using XSLT adaptors (see below) and stores them in an eXist database for an efficient structural search.

(2) *TG-search*—enabling semantic search and text retrieval.

(3) *TG-auth* encapsulates the authentication and authorization framework which integrates into the German Shibboleth federation DFN-AAI that provides users with the possibility to sign in with their own local campus account. The authorization is implemented by a Web Service framework based on Role Based Access Control (RBAC, 2004) with the notion of projects and different roles within a project. The data is stored in an openLDAP[14] database.

(4) *TG-log* is a central logging service.

(5) *Replica management* (see Section 3.3).

3.3 Replica management

There are two main reasons for using automated and configurable Replica Management in storage and service grids like TextGrid. First, the objects are stored in different locations to avoid losing the data by being corrupted or accidentally deleted. The second reason is to have certain large data objects available in different locations, possibly also at a distance from each other, so the user automatically can get the nearest copy, since network performance and fault tolerance are of interest here.

The Globus Toolkit provides a replication API[15] that is capable of the aforementioned features. This higher-level Globus Toolkit Replica Management API uses functions for registration, copying and publishing of files, and collections. An underlying

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

replica catalogue is used to provide mappings between logical names for files and collections and their storage locations.

So in the end, the user just tells the system to create a TextGrid object (via the TG-crud service), and the file will be stored in the Grid, metadata included. TextGrid Replica Management will then take care of possible storage locations and the number of replicas to store.

3.4 The service layer

The TextGrid service layer consists of a number of streaming tools or streaming components of interactive tools. These tools and components live in a completely Web Service-based environment. They are accessed by the client applications (Eclipse GUI or simple stand alone clients) via SOAP/WSDL or REST and act themselves as Web Service clients to the TG utilities (like TG-crud, see above). This architecture provides the highest possible flexibility in terms of extensibility as well as in terms of the use of programming languages.

Web Service frameworks are available for many programming languages—so if a person or institution wishes to make his/her text processing tool available to the TextGrid community and the workflow engine, the first step is to implement a Web Service wrapper for the tool and deploy it on a public server (or one of TextGrid's). The next steps are to apply for registration in the TextGrid service registry and to provide a client plug-in for the Eclipse GUI so that the tool is accessible for humans (GUI) and machines (service registry) alike.

But what about the grid? Reading from and writing to the grid via TG-crud requires authentication and authorization. To make the native TextGrid tools accessible for unregistered users, the tools usually provide two Web Service interfaces—one for reading and writing files from and to the grid, with URIs (and authentication information) as SOAP parameters, and one for reading and writing data directly from and to the SOAP message, without involvement of the TextGrid middleware.

The next (optional) step of the integration of external tools into TextGrid is to add support for TG-auth and TG-crud: The Web Service wrapper has to transfer the authentication information to TG-crud. Before this can be done, an agreement with the TextGrid Consortium should be assigned regarding the maintenance and support for the tool in question (for short descriptions of some of the native TextGrid tools, see Section 4.2).

4 Working with the Data

Once in the grid, the new or updated TextGrid object is accessible for the query interface (see below), collaborative editing, annotation, analysis, etc. Furthermore, the textual data can be processed by a variety of streaming tools that may be combined to user- or project-specific workflows using the workflow editor (Fig. 4).

4.1 Collaborative working

Now all the project members are able to work with the data. What a staff member can see and which documents he or she is allowed to read or to modify depends on the specific permissions which are usually set by the project administrator. The permissions are defined in terms of project-specific roles (default roles are: project leader, administrator, user, and observer). Working on the data is possible from any working site connected to the internet—thus we can also speak of 'distributed working'. Furthermore, it is possible to check out a local copy of the data (which is locked for other writing access) for working offline.

4.2 Using TextGrid tools

As mentioned above, the tools of the service layer are accessible via Web Service protocols. Therefore, they can be used with small, self-made Web Service clients. Since the native TextGrid tools are Open Source, many of them (tokenizer, streaming editor, lemmatizer, and others) can be downloaded and locally used as command-line tools. Using the TextGridLab-GUI, however, is much more comfortable and offers more possibilities:

- easy-to-use configuration and input wizards
- working directly on data in the grid
- preview of the results
- combining tools to workflows (see below)
- modification of input data with the XML editor

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

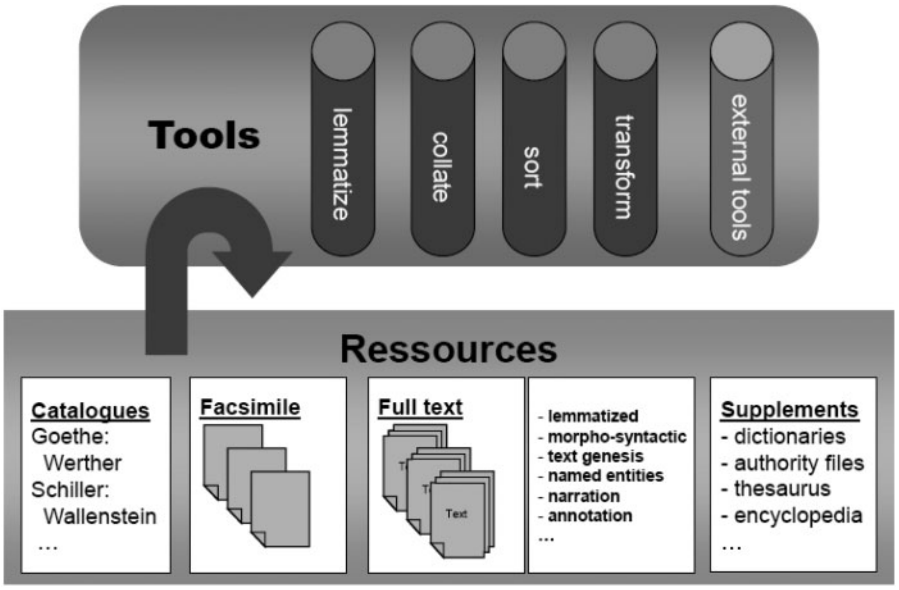


Fig. 4 Working with the data

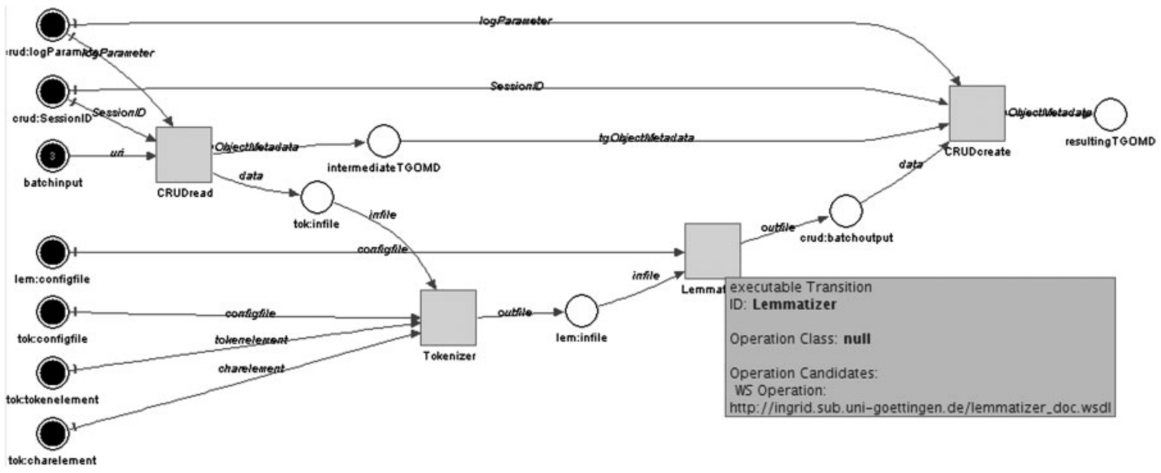
Below some of the tools are listed with a short description[16]:

- *Tokenizer:* Tagging of words according to Unicode Standard Annex #29[17]—and of punctuation marks. The names of the (word- and character-) elements, lists of predefined tokens (e.g. abbreviations, proper names) and regular expressions for markup of special, pattern-based token types (e.g. date) can be specified in the tool-configuration.
- *Lemmatizer:* Morpho-syntactic analyses of New High German words, covering inflection and word formation based on the SFST tools[18] and SMOR (Schmid *et al.*, 2004). Free dictionaries, e.g. Adelung (1789), have been integrated into SMOR.
- *Sorting tool:* Sorting of XML data or selected elements/parts according to cultural conventions (via selection of the corresponding locale) and international standards.[19] It is also possible to define individual sorting-keys.
- *Streaming editor:* Rule-based text-transformation. It can also be used for text processing based on XSLT stylesheets.
- *Collation tool:* Collation of two or more TEI documents. The results are documented in an XML format. The collation tool is mainly used in interactive mode.

4.3 Workflow

TextGrid services from the service layer and, more generally, any third party Web Service can be orchestrated to form complex workflows to automate recurring tasks. For example, editors might want to repeat the steps *tokenizeTEI - lemmatizeTEI - transformToHTML* every time they finish editing another section in order to see a preview. Technically, this is facilitated by a workflow editor where the user interactively assembles tasks into a workflow and a workflow enactor where workflows are deployed and executed. The former is a GUI and part of the TextGrid client, whereas the latter is located in the Service Layer just like any other service, albeit more complex in its interaction with the GUI. Important parts of the workflow editor GUI are: a registry of available services, a registry of workflows à la myExperiment,[20] a canvas for editing workflows (see Fig. 5), an interface to specify inputs

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012



executable Transition
ID: **Lemmatizer**

Operation Class: **null**

Operation Candidates:
WS Operation:
http://ingrid.sub.uni-goettingen.de/lemmatizer_doc.wsdl

Fig. 5 Graphical representation of a user-defined workflow

to a workflow, and a monitoring view for already deployed workflows. Authentication and logging configurations are hidden from the user but will be inserted when deploying a workflow to the enactor, which for TextGrid is the Taverna[21] Engine.

5 Searching the Data—Query Interface and Text Retrieval

TextGrid objects (as well as their equivalents in the metadata or structure database) are accessible via the query interface.

The design of the retrieval component has been motivated by two different search scenarios:

(1) The scholar searches the TextGrid repository with respect to a certain research question. As the specific encoding is unknown, the baseline encoded TEI documents (Section 5.2) are likely to deliver a hit. Accordingly, the query will be restricted to these elements and attributes only.
(2) The expert user wants to explore a project-specific TEI document in detail. Therefore, a powerful query language and access to all elements and attributes of the original is required.

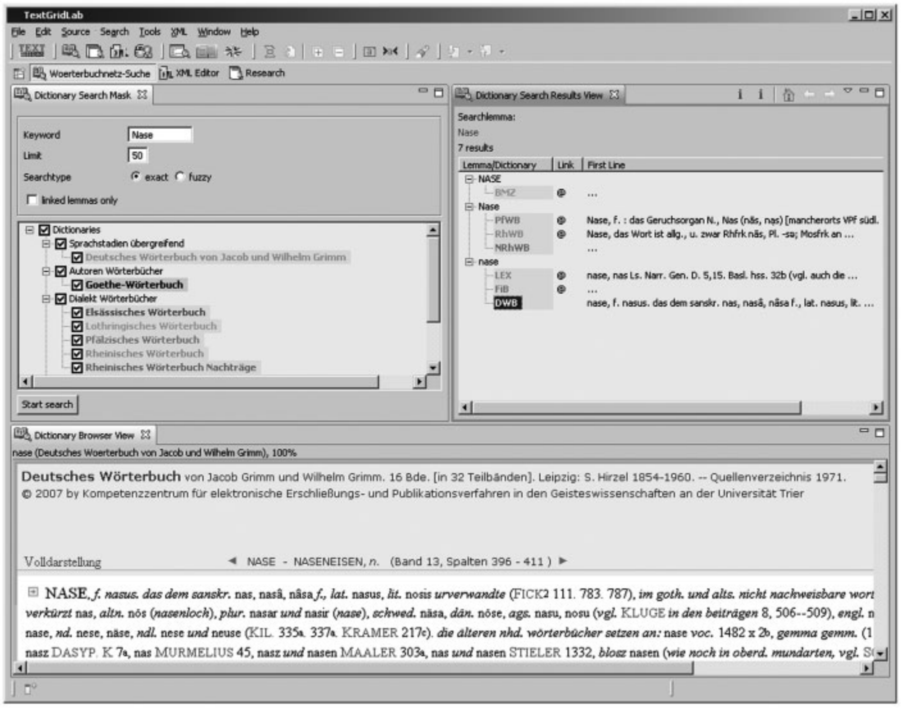
For item (1) above, TextGrid offers an intelligent query interface that hides some of the complexity of the XML structure. For query formulation, the user can choose between a command-line interface with a special query language or standard template queries that only need to be complemented with the desired values. The user can search within the TextGrid internal databases as well as in federated archives outside of TextGrid via the metadata database. An example of a GUI in a federated search scenario is shown in Fig. 6. In this case, the headword 'Bank' is searched for in a selected group of historical dictionaries (~Campe, Grimm, Goethe) forming part of the dictionary network. A query expansion via a thesaurus (OpenThesaurus) can be activated, here adding the headword 'Bankhaus' to the search. The GUI is adapted to the structural encoding of the searchable dictionaries and can be customized according to the needs of the user.

For item (2) above, TextGrid provides an interface to XQuery 1.0, which is a standardized query language recommended by the W3C. An example query will be shown in Section 5.4.

5.1 TextGrid's multi-purpose query system for philologists, lexicographers, and historians

Interestingly, the structure-based approach to retrieval is very generic and fulfils the requirements of different communities that might pose quite different research questions—e.g. historians' searches are typically content-oriented; linguists generally like to

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012



Keyword: Nase
Limit: 50
Searchtype: [✓ exact] fuzzy
linked lemmas only

- ✓ Dictionaries
  - ✓ Sprachstadien übergreifend
    - ✓ Deutsches Wörterbuch von Jacob und Wilhelm Grimm
  - ✓ Autoren Wörterbücher
    - ✓ Goethe-Wörterbuch
  - ✓ Dialekt Wörterbücher
    - ✓ Elsässisches Wörterbuch
    - ✓ Lothringisches Wörterbuch
    - ✓ Pfälzisches Wörterbuch
    - ✓ Rheinisches Wörterbuch
    - ✓ Rheinisches Wörterbuch Nachträge

Searchlemma:
Nase
7 results

Lemma/Dictionary	Link	First Line
NASE		
BMZ	@	...
Nase		
PfWB	@	Nase, f. : das Geruchsorgan N., Nas (nās, nạs) [mancherorts VPf südl.
RhWB	@	Nase, das Wort ist allg., u. zwar Rhfrk nās, Pl. -sa; Mosfrk an ...
NRhWB		...
nase		
LEX	@	nase, nas Ls. Narr. Gen. D. 5,15. Basl. hss. 32b (vgl. auch die ...
FiB	@	...
DWB		nase, f. nasus. das dem sanskr. nas, nasâ, nâsa f., lat. nasus, lit. ...

nase (Deutsches Woerterbuch von Jacob und Wilhelm Grimm), 100%

Deutsches Wörterbuch von Jacob Grimm und Wilhelm Grimm. 16 Bde. [in 32 Teilbänden]. Leipzig: S. Hirzel 1854-1960. -- Quellenverzeichnis 1971.
© 2007 by Kompetenzzentrum für elektronische Erschließungs- und Publikationsverfahren in den Geisteswissenschaften an der Universität Trier

Volldarstellung ◄ NASE - NASENEISEN, *n.* (Band 13, Spalten 396 - 411) ►

NASE, *f. nasus. das dem sanskr.* nas, nasâ, nâsa *f., lat.* nasus, *lit.* nosis *urverwandte* (FICK2 111. 783. 787), *im goth. und alts. nicht nachweisbare wort*
verkürzt nas, *altn.* nōs (*nasenloch*), *plur.* nasar *und* nasir (*nase*), *schwed.* näsa, *dän.* nöse, *ags.* nasu, nosu (*vgl.* KLUGE *in den beiträgen* 8, 506--509), *engl.* n
nase, *nd.* nese, näse, *ndl.* nese *und* neuse (KIL. 335a. 337a. KRAMER 217c). *die älteren nhd. wörterbücher setzen an:* nase *voc.* 1482 x 2b, *gemma gemm.* (1
nasz DASYP. K 7a, nas MURMELIUS 45, nasz *und* nasen MAALER 303a, nas *und* nasen STIELER 1332, *blosz* nasen (*wie noch in oberd. mundarten, vgl.* S

Fig. 6 Part of the interface of the Trier Wörterbuchnetz

find a particular word or phrase, and search is thus more structure-oriented; in philological editions, the layout structure is of great importance.

Common to all disciplines is that retrieval can relate to the primary text, the metadata, and/or the annotations. In all cases, the search facility can be integrated into a workbench that supports the user's workflow. The workflow of a lexicographer is shown in Fig. 7 as a prototypical example.

Retrieval is achieved in the following steps[22]: First, a suitable corpus sample is compiled on the basis of the metadata; then, the query with a search pattern is set up; next, the system presents a keyword-in-context (KWIC) analysis. More elaborate systems also present linguistic annotations, lists of collocations, or co-occurrences. The next step, usually done manually, is to identify distinct meanings of the word.

The main differences regard the presentation of the retrieval results: For some disciplines, links to editorial comments and passages of the transcribed text or registers with lists of persons, location names, and chronological events might have to be included.

5.2 TextGrid baseline encoding

A specific schema, called baseline encoding, is a key innovation in TextGrid developed together with other humanities computing projects encoding different varieties of text like dictionaries, drama, letters, critical editions, and language corpora. It has been designed in order to gain optimal retrieval performance across all TEI documents in TextGrid.

It should be clear that structured search is the optimal retrieval strategy for previously encoded text. Unstructured retrieval should be only a fallback solution. The best results can be obtained when the user is familiar with the structure and encoding of the document. In fact, all properties encoded can also be queried. So, the basic challenge is how to

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

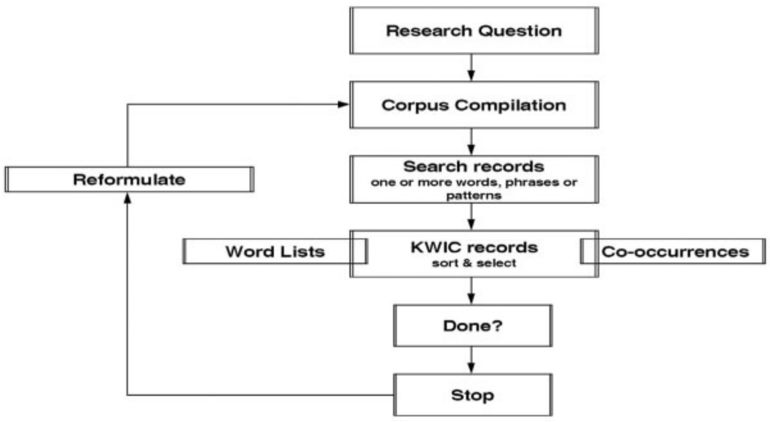


Fig. 7 Prototypical search workflow in TextGrid

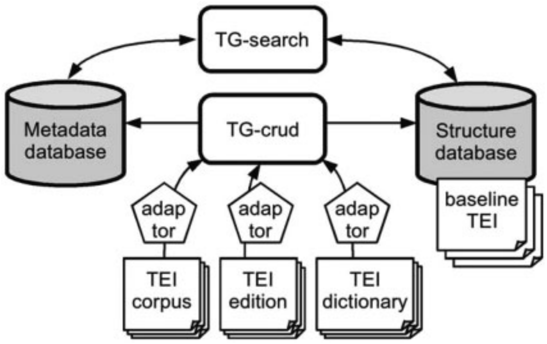


Fig. 8 The TextGrid baseline encoding

assure that researchers not familiar with the TEI-encoded text can benefit from the mark-up.

To this aim, different project encoding schemes of varying granularity are mapped onto the TextGrid baseline encoding (where only a limited set of approximately 50 TEI P5 elements are allowed). This version is generated dynamically from the original format via *adaptors*, as can be seen in Fig. 8.

Some adaptors are already provided in TextGrid that describe how various elements can be mapped onto each other (e.g. different encodings of the same text property are generally conflated). Particularly, the TEI header is used to extract metadata semi-automatically.[23] A description of the Baseline Encoding is available in Jannidis and Vitt (2008).

It is, of course, also possible to map non-TEI texts onto the TextGrid baseline encoding without investing too much annotation effort: Only those elements/attributes that are relevant to answer the most common research questions need to be encoded. Both versions of TEI documents are stored in a native XML database in the grid. This solution assures the full flexibility as provided by TEI and, yet, makes the implementation of a search algorithm much easier.

5.3 Search scenarios and TextGrid baseline encoding

Departing from typical search scenarios within different communities in the humanities, TextGrid has defined a special 'TEI baseline encoding'.

Although we concentrated on philologists and lexicographers so far, to illustrate a content-oriented search scenario and to show that the main ideas apply here as well, we claim that historians searching on historical manuscripts could profit from the use of a minimal set of TEI-elements for enhanced retrieval as well.

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

For searching across the entire TextGridRep, the following metadata set—partially derivable from the <teiHeader>—acts as a first search filter:

- Obligatory Metadata: *agent* (author, contributor, editor, illustrator, translator, providing Institution), *title*, *date*, *language*, *type* (prose, lyric, drama, letter, etc.), *owner*, *rights*
- Optional Metadata: *keywords*, *markupclass* (edition, corpus, etc.)

5.3.1 Use cases

Scholars in linguistics or lexicography often like to find corpus evidence for a selected lemma. This requires an exact match and an exhaustive listing of all linguistic expressions found. This is illustrated as follows.[24]

Research question (Examples)	
Find all sentences (divisions, paragraphs) in which the words (lemmas) 'fire' and 'burn' co-occur (e.g. in letters of German writers of the 18th century).	*Structural tags:* <*div*> (text division), <*p*> (paragraph), <*s*> (s-unit) *Content tags:* <*w*> (word), <*c*> (character), <*f*> (feature)
Find all citations by the author 'Opitz' in the dictionary 'Campe'.	*Structural tags:* <*entry*> (dictionary entry), <*cit*> (cited quotation), <*quote*> (quotation), <*bibl*> (bibliographic citation) *Content tags:* <*author*> (name of author in a bibliographic citation)

Scholars **in philology** are also interested in the layout or editorial comments, as illustrated below.

Research question (Examples)	
Find all headlines including the word 'Rose' in the work of 'Goethe', or *Find all revisions of lyrical texts that were made by the project leader within the last year.*	*Structural tags:* <*div*> (text division), <*p*> (paragraph), <*s*> (s-unit) *Layout tags:* <*lg*> (line group), <*l*> (line), <*cb*> (column break), <*lb*> (line break), <*pb*> (page break), <*milestone*> (text boundary), <*hi rend*> (highlighted) *Content tags (including editorial comments):* <*add*> (insertion), <*sic*> (text revision), <*del*> (deletion), <*orig*> (original form), <*c*> (character), <*w*> (word),

Scholars **in history** are most interested in the content of a text. A typical example is shown below.

Research question (Example)	
What questions were asked by Ambassador Kurusu in conversations with US government representatives between July and December 1941.[25]	*Structural tags:* <*div*> (text division), <*p*> (paragraph), <*s*> (s-unit), <*head*> (head) *Content tags:* <*place*> (place name), <*person*> (name of a person), <*org*> (organization or institution), <*name*> (proper noun), <*placeRef*> (indirect reference to a place), <*personRef*> (indirect reference to a person), <*orgRef*> (indirect reference to an organization or institution) <*date*> (date), <*time*> (time)

So far, little effort has been invested in the semantic tagging of historical resources.[26] It is clear, though, that people's names or places can give relevant clues to the content. While Named Entity recognition is already well established in the life science, only few initiatives augment historical resource data by means of TEI at present (cf. Paradis, 1994; Clough *et al.*, 2002; Ide and Woolner, 2004).

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

5.3.2 *Summing up*

Although different humanities disciplines require different information—varying from structure, layout to content—a restricted set of TEI elements for each community suffices to gain enhanced retrieval results in all cases. Sometimes, the required base tag sets even overlap (Fig. 9). Tags can be shared easily if one primary text level is chosen that all annotations can refer to.

A number of tools and lexical resources, e.g. historical lexica from 'MHD Wörterbuch Netz' (Rapp and Burch, 2007) and thesaurus resources like openthesaurus,[27] are integrated into TextGrid to facilitate the annotation of German texts.

Lou Burnard, Assistant Director of the Oxford University Computing Services, explains that mark-up makes 'explicit (to a machine) what is implicit (to a person),' adds 'value by supplying multiple annotations,' and facilitates 're-use of the same material in different formats, in different contexts and for different users' (Burnard, 2004).

In this respect, we claim that TextGrid exploits the full strength of the TEI markup.

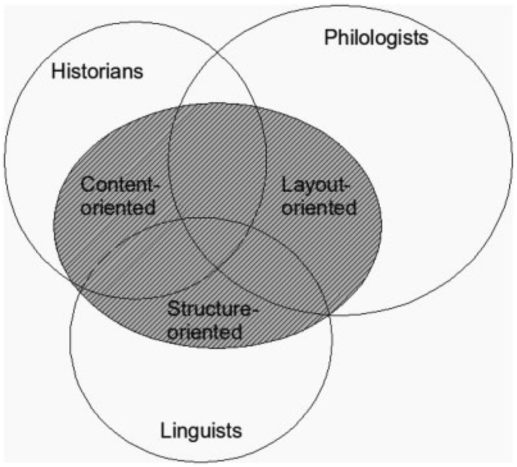


Fig. 9 Focus of (re)search in TextGrid

5.4 XQuery interface

For (b), the exploration of a project-specific TEI document in detail, a very up-to-date implementation of the Content tags XQuery 1.0 standard (W3C, 2007) is offered through the eXist database engine. As has been shown by Cassidy (2002), XQuery is a powerful query language to retrieve XML/TEI-based data. Like XPath, it offers the possibility of directly accessing XML elements and their attributes in a document. But unlike XPath, XQuery incorporates the expressive power of FLWOR-statements (for, let, where, order, return). Additionally, the query results can easily undergo further processing, since eXist offers a broad coverage of standards such as XHTML, XSLT, Javascript, etc. An example query is shown in Figure 10.

6 Publishing the Data

When the edition project or the work on a text is finished, the data may be published in different ways.

For the purpose of web publishing, TextGrid provides a framework that supports projects in creating their own website. This framework consists of templates and tools for retrieving texts, graphics,

QUERY: Find in the entire repository all word tuples of determiners and nouns (marked as <w ana="#pos_det"> or <w ana="#pos_n">) in a sentence (marked as <s>).

XQuery-statement:

```
for $a in //s/w, $b in //s/w[.>>$a][position()=1]
        where
        contains($a/@ana,"#pos_det") and
        contains($b/@ana,"#pos_n")
return
<output>
        {$a}
        {$b}
</output>
```

Fig. 10 Example of a query of the expert user

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

etc. from the TextGrid repositories and transforms these data for adequate presentation in the World Wide Web.

For print publishing, a group of TextGrid partners is developing an XML-enabled typesetting module meeting the requirements of complex editions with multiple critical apparatuses including multilingual and multidirectional texts.

7 Long-Term Storage

Long Term Storage is mainly an administrative feature, but it also depends on some technical issues such as persistent identifiers, data migration, and metadata-maintenance. Terms of usage will be available in spring 2009.

8 Conclusions

This article has described the life cycle of a TEI document within the e-Humanities platform TextGrid and the single building blocks of the system that are loosely coupled in a service-oriented architecture and leveraging a storage grid. As such, TextGrid is the first system that integrates all the features needed for scholarly text processing in a distributed grid environment. The architecture based on Web Services provides for the possibility of easily integrating external tools available in the various research communities—in philology, linguistics, lexicography etc. It has been shown how the TextGrid baseline encoding enables different search scenarios in these communities.

Acknowledgements

Thanks to the members of the TextGrid-Project for valuable input.

Funding

TextGrid is partially funded by the German Federal Ministry of Education and Research (BMBF) under the D-Grid initiative by agreement 07TG01A-H.

References

Burnard, L. (2004) Digital Texts with XML and the TEI. Presentation. *Text Encoding Initiative*, <http://www.tei-c.org/Talks/OUCS/2004-02/One/teixml-one.pdf> (accessed 10 March 2009).

Cassidy, S. (2002). XQuery as an Annotation Query Language: A Use Case Analysis. *Proceedings of LREC2002: Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, Spain, 29-31 May, 2002*. Volume VI. Paris: European Language Resources Association, pp. 2055–2060.

Clough, P., Gaizauskas, R., and Piao S.L. (2002). Building and annotating a corpus for the study of journalistic text reuse. *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-02)*, 29–31 May 2002. Volume V. Los Palmas de Gran Canaria, Spain, pp. 1678–1691.

Gietz, P., Aschenbrenner, A., Büdenbender, S., Jannidis, F, Küster, M.W., Ludwig, C., Pempe, W., Vitt, T., Wegstein, W., Zielinski, A. (2006). TextGrid and eHumanities. *Proceedings Second IEEE International Conference on e-Science and Grid Computing* (e-Science'06), p. 133.

Ide, N. and Woolner, D. (2004). Exploiting Semantic Web Technologies for Intelligent Access to Historical. *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-04)*, Lisbon, Portugal.

Jannidis, F. and Vitt, T. (2008). Markup in TextGrid. In Opas-Hänninen, L.L., Jokelainen, M., Juuso, I., and Seppänen, T. (eds), *Digital Humanities 2008. Book of Abstracts*. Oulu: University of Oulu, pp. 138–139.

Paradis, F. (1994). A model for structured textual documents. Technical *Report, FERMI 4-94* http://www.dcs.gla.ac.uk/fermi/tech_reports/(accessed 10 March 2009).

Rapp, A. and Burch, T. (2007). *Geschichte* im Netz: Praxis, Chancen, Visionen. *Das Wörterbuch-Netz: Verfahren – Methoden – Perspektiven*. Band 10, 2007.

RBAC (2004). *Role Based Access Control*, American National Standard, ANSI INCITS 359-2004.

Schmid, H., Fitschen, A., and Heid, U. (2004). SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection. *Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, pp. 1263–1266.

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012

W3C. (2007). *XQuery 1.0: An XML Query Language*. W3C Recommendation 23 January 2007. <http://www.w3.org/TR/xquery/> (accessed 10 March 2009).

Notes

1 Joachim Heinrich Campe, Wörterbuch der Deutschen Sprache, Braunschweig 1807–1811.

2 An open development framework which comprises a *Rich Client Platform* for developing graphical interfaces (http://www.eclipse.org).

3 See http://www.textgrid.de. TextGrid is partially funded by the German Federal Ministry of Education and Research (BMBF) under the D-Grid initiative by agreement 07TG01A-H. Responsibility for the contents of this publication rests with its authors.

4 See http://www.d-grid.de.

5 For TextGrid and e-Humanities, see (Gietz *et al*., 2006).

6 See http://www.dariah.eu

7 See http://www.clarin.eu

8 WYSIWYM = **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **M**ean.

9 See http://www.globus.org/toolkit/.

10 See http://www.gridlab.org/WorkPackages/wp-1.

11 See http://forge.ogf.org/sf/projects/saga-core-wg/.

12 See http://exist.sourceforge.net/.

13 See http://www.openrdf.org.

14 See http://www.openldap.org/

15 See http://www.globus.org/toolkit/docs/2.4/datagrid/deliverables/ReplicaManagementService.pdf

16 Comprehensive descriptions of the tools are available in the TextGrid Reports 2.1. (http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid_Report_2_1.pdf) and 2.2. (http://www.textgrid.de/fileadmin/TextGrid/reports/TextGrid-R2.2_ToolsII.pdf), English documentation will follow as soon as possible.

17 http://www.unicode.org/reports/tr29/tr29-9.html.

18 http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/SFST.html

19 DIN 5007, NFZ44-001 (AFNOR), TK 34.1 (SIS) und ENV 13710 (CEN).

20 See http://www.myexperiment.com for a Web 2.0 style repository of Taverna workflows.

21 See http://taverna.sourceforge.net.

22 Prominent Corpus Query Systems are Cosmas II (https://cosmas2.ids-mannheim.de/cosmas2-web/) and theIMS Corpus Workbench (IMS Stuttgart, 1998) (http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/)

23 TextGrid aims towards mappings of its metadata to standard formats to increase the exposure of its valuable assets. Among those formats may be MODS and MARC (internationally acknowledged standards maintained by the US Library of Congress) for transfer to (library) catalogues, METS (Metadata Encoding and Transmission Standard) for packaging objects, and Dublin Core.

24 For reasons of space, we cannot present the full baseline encoding here. Further baseline elements for dictionaries, e.g. would be <*form*> (lemma), <*sense*> (sense), <*gramGrp*> (morphological information), <*pos*> (Part of Speech).

25 The example is drawn from (Ide and Woolner, 2004).

26 Generally, for this task a classical IR system is used because search engines are quite robust and can even cope with heterogeneous documents. Some elaborate systems provide enhanced structure-aware ranking mechanisms. Yet, for scientific research on historical documents, semantic tagging could improve retrieval effectiveness tremendously: The professional user knows what he/she is searching for, e.g. the search paradoxon ('The need to describe that which you do not know in order to find it') is not the limiting factor.

27 See http://www.openthesaurus.de/

Downloaded from http://llc.oxfordjournals.org/ at UB Leipzig-Zwst Geistes - Und on January 17, 2012