TextGrid and eHumanities

Peter Gietz[1] Andreas Aschenbrenner[2] Stefan Büdenbender[3] Fotis Jannidis[7] Marc W. Küster[5]
Christoph Ludwig[5] Wolfgang Pempe[6] Thorsten Vitt[7] Werner Wegstein[4] Andrea Zielinski[8]

[1]DAASI International, Tübingen [2]SUB Göttingen [3]Universität Trier
[4]Universität Würzburg [5]Fachhochschule Worms [6]Saphor, Tübingen
[7]Technische Universität Darmstadt [8]Institut für Deutsche Sprache, Mannheim

Abstract

TextGrid is a new Grid project in the framework of the German D-Grid initiative, with the aim to deploy Grid technologies for humanities scholars working on historical (German) texts. Its two roots, humanities computing and eScience (Grid computing used by research together with modern communication technologies), are the basis for TextGrid to provide pioneer work in eHumanities. After summarizing Humanities Computing and modern network technologies, community expectations in the fields of philological edition and other application areas are set forth, from which functional requirements such as modularity, distribution, etc. are distilled. The first version of the TextGrid architecture was designed in accordance with these requirements, and focuses on openness by standard conformance and encapsulation. It provides storage Grid services via a pure Web Services interface to dedicated Web Services tools for different aspects of text processing, analysis and retrieval. This platform aims to provide easily usable tools for scholars, but also specifies interfaces for external program developers to add functionality.

1. Introduction

TextGrid (http://www.textgrid.de), which started in late 2005, is the acronym of the humanities partners of D-Grid (http://www.d-grid.de), an initiative funded for three years by the German Ministry of Education and Research to establish a Grid infrastructure for research. Coordinated by the Göttingen State and University Library, five institutional partners (Darmstadt University of Technology; Institut für Deutsche Sprache, Mannheim; University of Trier; University of Applied Sciences, Worms; University of Würzburg) and two commercial companies (DAASI International, Tübingen and Saphor, Tübingen) aim to create a virtual research library which entails a Grid-enabled workbench that will process, analyse, annotate, edit, link and publish text data for academic research in an Open Source and Open Access environment supporting TEI markup [27].

1.1. Humanities computing and modern IT technologies

Since the beginning of computing, humanities have been using the evolving new technologies for their research. The pioneer Pater Roberto A. Busa, who in the late 1940s began using IBM mainframe computers to help create a complete word index of the works of St. Thomas Aquinas [3], was followed by a great number of scholars in different fields of humanities [26], such as philologies, linguistics, lexicography and literary studies.

From the 1960s onwards tools have been created and utilized by communities concerned with computer aided text analysis that can broadly be characterized by three types:

- very specialized programs dedicated to only one specific problem, language or project. An example for such an approach can be found in the software created by SIL International, which is dedicated to lesser known languages (see http://www.sil.org/computing/catalog/index.asp)
- general purpose tools like TUSTEP [25], which provides highly configurable interoperable building blocks of text processing and analysing tools.
- The third type of tools evolved in the 1980s: Easy to use scripting programming languages, which try to make the act of programming as easy as possible. Examples of these are Python, Ruby and the more established Perl, a general purpose scripting language with great merits in string processing and in inclusion of the pattern matching language called Regular Expressions.

It has been argued [2] that TUSTEP and Perl, although being two quite different things, are both similarly useful

for general purpose text manipulation.

Two professional societies, the Association for Computers in the Humanities and the Association for Literary and Linguistic Computing, and their journals, "Computers and the Humanities" (1966ff.) and "Literary and Linguistic Computing" (1986ff.) institutionalized this area of research. They are now gathered under the ADHO, the Alliance of Digital Humanities Organizations (http://www.digitalhumanities.org) with an annual conference. With "Humanities Computing" a term has been found that fits well for all the different computing activities in the humanities [18]. The term "Digital Humanities" [26] seems to be equivalent.

An important step in text based Humanities Computing has been the standardization of text encoding, which was developed by the TEI (http://www.tei-c.org), started within an SGML framework, now converted to XML (P4) [27, 14] and Schema (P5) [28]. TEI provides combinable tag sets for a wide range of disciplines using markup to interchange data as well as to communicate a theory about the structure of a text. So it turned out that TEI did not only provide the possibility to exchange information but also was a data description language that improves the scholar's ability to describe textual features [24].

1.2. Web services paradigm and Grid computing – The move from humanities computing to eHumanities

With the advent of the World Wide Web, based on the network protocol HTTP and on HTML (another SGML offspring), more than one revolution in the history of media took place, putting the computer network in the focus of society as an interlinked and distributed library. In humanities computing the WWW has dramatically increased the development and distribution of electronic texts. But the tools available for accessing these texts were of lesser functionality than those available on CD-ROM [31].

Grid computing [11] stands for the idea of providing computing, network, and storage resources, while the complex infrastructure is totally invisible, comparable to the power grid, where one only sees the power outlet. sharing coordinated resources and solving problems within dynamic, multi-institutional virtual organizations [12].

Grid computing has been one of the driving forces behind the term eScience, which denotes a new form of net-based scholarly work and collaboration by deploying new network technologies and infrastructures, especially Grid computing. The term eScience is often used by European public research funding bodies. In the US, the term cyberinfrastructure is typically used instead. The overall aim of eScience is to provide shared access to research facilities, mainly computational processing and data collections, across the Internet, which allows for innovative research designs and thus is prone to change the way research is done [16].

We would like to establish the term eHumanities as the equivalent of eScience in the field of humanities, thus equivalent to the term "Arts and Humanities E-research" as used by [1]. The difference to Humanities Computing is the deployment of Grid-based infrastructure and network collaboration tools sharing common resources within Virtual Organizations. Nevertheless, the connotations of an early use of the term eHumanities [8] concerned with questions of how technology affects traditional humanities disciplines is not negated here. The "e" in eHumanities thus not only stands for "electronic", but also for "enhancing, extending, and enabling" [4].

TextGrid is taking up the challenge to develop eHumanities in this sense, first of all, but not only exclusively, in the area of scholarly text processing, i.e. textual criticism which creates and uses digital texts for answering traditional and new questions with empirical methods. The introduction of collaborative methods and the delivery of standardized tools will put the field of text data processing on a new footing through the use of distributed resources. The aim is to promote academic research in a networked and interdisciplinary environment that is both mobile and virtual.

TextGrid will develop a modular platform for scholarly text processing based on Unicode character encoding, Web Services and other standards, which will make the platform open for other software developers who are invited to contribute modules. The project started in February 2006 and is now in an intense phase of prototyping and specification, defining architecture, data and process modells, etc., partly in cooperation with external partners. The current state of these discussions is represented in this paper.

2. Community and use cases – putting TextGrid into practice

TextGrid aims to serve a whole range of communities and applications in textual scholarship. The concepts and tools established for TextGrid are extensible and multiple purpose. Thus, while the initial focus of the project is on philological edition, project partners work towards the integration of requirements in linguistics and lexicography already now. Moreover, tools can be re-mixed and the platform can be extended for other application areas by the TextGrid community in the future.

2.1. Philological Edition

TextGrid supports the creation of an edition at various stages, ranging from initial collection of bibliographic resources and transcription over rich annotation, linking, and

collation up to providing integrated extensive search facilities over distinct editions.

Currently, many projects aim at large scale digitisation of historical manuscripts and prints. However, high quality scans result in huge quantities of image data. By linking the humanities to the storage Grid, TextGrid will connect the community to the resources for storing and accessing such quantities of data.

Links between those digital images of the original source and the transcribed and computer-processable text offer a variety of possibilities: This includes imposing extracts of the transcript as manuscript reading aid on the image [19], producing a printed (and more readable) version of the text resembling the original manuscript's topography, and navigation from the searchable and annotated text to the corresponding fragment of the scanned manuscript.

TextGrid tools will facilitate the tedious task of encoding these links by automating parts of this work (like the segmentation of the original image) and by integrating the linking process with the process of transcription.

A lot of scholarly work in the making of a critical edition is invested in the creation of annotations and metadata. Besides markup of persons or places which will later be used for creating indexes this on the one hand means annotations that improve the text's accessibility to the reader. On the other hand, a dominant component of critical editions is information on corrections conveying knowledge of a work's genesis and differences between various witnesses of a text. Particularly the latter step can be supported by a computing-intensive automated collation tool, which will benefit from the computing aspect of the Grid.

In printed editions, this information is typically represented in a rather compressed and, at least for the uninitiated, hard to decode form in the critical apparatus; with digital editions, more appealing representations are possible: E. g., the possibility to click through the different steps in the history of a text.

TextGrid will offer easy-to-use tools to support the editors in collaboratively creating these annotations as well as comprehensive bibliographical and structural information in a consistent way, so that they can be easily used for further electronic processing like detailed searches and generating web or print renditions.

TextGrid's goal to help join distinct scholarly editions will not only be served by supporting editors to introduce explicit links. As well, the end users will be enabled to perform integrated search and retrieval in all connected electronic texts, including the possibility to restrict the search domain using the texts' metadata – and excluding the vast amount of irrelevant results of general-purpose web search engines.

As far as the TextGrid use case is concerned, the new genetic critical edition of the works of Jean Paul, one of the leading classical authors in German literature and a prominent figure in our cultural heritage around 1800, will be used. Nevertheless, up to now a critical edition of his works (22.000 pages printed during his lifetime) and his huge literary legacy remains (40.000 manuscript pages) has never been completed. The Würzburg multimedia edition that will be used as testbed combines images of the manuscript material, transcriptions, images of all the printed text, typoscripts and the critical edition using standard information-handling techniques and TEI Markup in order to encode texts for publishing in both conventional printed and electronic form. With a volume of about 4 Terabyte of data the edition reaches the quantity necessary to test the functionality of the TextGrid infrastructure. First sample parts of the edition are offered at the Jean-Paul-Portal: www.jean-paul-portal.de.

2.2. Lexicography, Linguistics, and further Application Areas

While lexicographers and linguists of contemporary German can make use of numerous freely-available electronic language resources, text archives for historical German (from its Middle High German stage onwards) are still rare.

Therefore, one of the aims of TextGrid is to offer an integrative platform to support the compilation of a corpus of historical and contemporary German, based on a semantic Grid framework, which can be openly accessed. Moreover, intelligent services will be provided that - apart from full-text search - support enhanced access to the resources in TextGrid meeting the requirements of diverse linguistic disciplines:

- etymology - search for loanwords,
- dialectology - search for regional variants,
- morphology - search for lexemes as well as single morphemes within larger units (e.g., compounds),
- syntax - search for proper names and terminology (e.g., nominal phrases),
- semantics - search on word meanings (concepts) and semantically related words (e.g., synonyms),
- text linguistics - search for a specific text type (e.g., poems).

This aim can only be reached stepwise, applying state-of-the art technologies from computational linguistics, information retrieval and Grid computing.

First, eight historical dictionaries are integrated into TextGrid, covering a range from Middle High German to the era of Goethe, and five dialect dictionaries, covering most of West Middle German, with more to come over the next years: TextGrid will define interfaces to allow the integration of further external dictionaries and lexical resources.

Second, morphological analysis tools for different time stages are developed. When integrated into the indexing component of an information retrieval system, dif-

ferent word forms of a lexeme can be found automatically. Likewise, each token of a corpus can be enriched with morpho-syntactic information pertaining to the lemma, part-of speech, region, and language.

To achieve optimal results, the dictionaries will have to undergo further pocessing. Firstly, the word clusters generated by the original cross-references between lexical entries are expanded in terms of symmetry and transition. Secondly, new links are generated by information retrieval techniques. These identify semantic relations that were not explicitly marked in the printed work. While this is done automatically, the resulting net of references can be manually annotated, expanded, or if necessary, restricted.

This will not just enhance the services outlined above but will also enhance the usability of the dictionaries as such, which are made searchable through a standardized interface. This interface will attend both to the structural differences and different levels of annotation achieved during the process of retro-digitisation and offer a uniform set of search functions: category search for headwords (lemmata), grammatical information, and others (depending on the specific dictionary: quotes, definitions, etc.) as well as an unspecified search for plain text and, based on the original and newly inserted references, semantically related terms.

One of the main desiderata in future strategies is a lexicographer's workbench for better representing and managing lexical data of different regions and times, enriched with corpus-analysis tools that a) calculate semantic relatedness, b) generate co-occurrence matrixes, c) align corpora pertaining to different languages/dialects, and d) extract meaningful units (named entities, terminology).

A second desideratum both for historical linguists and dialectologists is the creation of a list of hyper-lemmas, that is words that share the same meaning at different times. While TextGrid in its current stage will not attempt to complete such a list, we will continually expand and enrich the word nets created for the linking of dictionaries. An editor will allow all participants to add new lexemes and to specify and annotate the established relations, making use of existing ontologies (GermaNet) and thesauri as far as possible.

A variety of other use cases and application areas are conceivable. The open and extensible architectural framework will allow for their integration by any interested party. With the emergence of an active TextGrid community, TextGrid will be a living platform that grows over time and extends to all requirements in textual scholarship and the humanities.

3. Requirements for eHumanities

From early on, computing in the humanities has had to face two often conflicting requirements. On the one hand, it must enable scholars to work on very specific research questions with specific text corpora with their often singular demands that are every bit as complex as those in other branches of eSciences. On the other hand (and unlike in many other fields), those very scholars have only rarely experienced a thorough training in computer science. This dichotomy has become more poignant as computing equipment – though not necessarily the corresponding expertise – has become ubiquitous also in the humanities.

In other words, eHumanities in general and TextGrid in particular have to work towards two goals that are sometimes difficult to reconcile:

- be easy to install and use (user interface and publication platform);
- offer flexible support (user defined workflows and data structures, extensibility and modularity).

Additionally, current research in the humanities is often team-based with team members frequently spread across the country or even the globe. Hence, any solution needs to support collaborative working methods (collaboration, distributed data, versioning, distributed modules, scalability, and security).

In the following paragraphs we shall look into these requirements in more detail, presenting examples of the state of the art where applicable.

User Interface. We do not expect the typical TextGrid user to be familiar with the many, often complicated technical issues that accompany net-centric technologies. Therefore, a graphical user interface (GUI) must hide all details that are not directly related to the philological task at hand. Given today's user expectations, having a GUI is a must, but it is also crucial for visual tasks such as image annotation and linking. Many of today's frameworks – ARCHWay [15], GATE [6] etc. – already use GUIs as a matter of course.

When working with XML data, it is most convenient to have several "views" on the data: e. g., raw XML, a view that hides tags that are currently of no interest to the user, and a WYSIWYM (what you see is what you mean) view. Many philological applications can in fact be split into well defined small steps. The user interface has to provide some means to specify such a workflow intuitively.

Publication Platform Researchers generate print editions of their works, publish them on the Web using plain HTML or with sophisticated retrieval interfaces like the one described in [5], or integrate them into text corpora that allow programmatic access through standard interfaces. TextGrid will support all of these formats. Ideally, TextGrid itself will become one large virtual text corpus whose texts can be accessed, queried and published. Virtual documents that combine textual annotations with, e. g., images (facsimiles) can exist, even though their parts are physically distributed across the network.

User Defined Workflows. As mentioned above, tasks in the humanities can often be seen as a sequence of specific,

frequently automatable steps. These steps can include, e. g., tokenizing, index building, lemmatization, structural analysis, type setting, etc. The researcher normally needs to execute these steps over and over again, which is tedious and error prone if done manually.

User Defined Data Structures. Many components of TextGrid will not impose fixed requirements on the data structures they can handle. Notably modules for tokenizing and data enhancement can handle almost arbitrary data structures. Other tools, however, such as the WYSIWYM XML editor will impose stricter preconditions on the data they understand. A streaming editor will enable conversions from and to user defined formats into the TEI [27] conforming structure preferred by many tools.

Extensibility and Modularity. No project however ambitious can anticipate and implement all functionality that users may require. It is therefore essential that TextGrid allow users to extend the system with their own modules. This emphasizes the paradigm that already TUSTEP implements and that is embodied in the concept of the Unix toolbox, namely that all functionality needs to be encapsulated in different modules that can be combined and extended arbitrarily. The system can be extended by providing or integrating new Web Services [23] and a front end GUI.

Collaboration. Similar to other eSciences, research in the humanities is increasingly collaborative. Even in a given project, experts from across the globe contribute to the eventual findings. This poses its own challenges — here Grid technology excels.

Distributed Data. The Net makes it possible to access, query, enhance, and, most notably, connect resources from arbitrary locations. This opens fascinating possibilities, yet finding the relevant resources is getting increasingly challenging. If a researcher builds upon a resource controlled by a third party, then he risks that his own work will lose its context if the third party decides to take the resource off the Net. Therefore, researchers need an environment in which resources are automatically replicated and references are transparently resolved to one of the actual copies. This ensures the referential integrity of derived works.

It becomes necessary to distribute high amounts of text and multimedia data to many physical locations while maintaining full transparency for the end user.

Versioning. Software development is consistently done with the aid of version control systems [9], that allow to reference and retrieve specific versions of source code even after the source code itself has been changed. These systems also resolve conflicting write accesses to source files. This very same requirement also applies to documents referenced and used in TextGrid. We thus need a mechanism that offers similar functionality in a Grid environment.

Distributed Modules. Systems in humanities computing have for many decades now worked with modules (cf. the example of TUSTEP [25], GATE [6, section 1.3.1]), but those modules were still glued together in one more or less fixed system that was locally installed on a single computer. In today's world, not only data is distributed, but also useful application modules are provided by many different players in the eHumanities scene. However, at present it is very difficult, if not impossible, to combine those existing modules into one application. TextGrid has to provide a uniform platform for seamlessly linking together conforming modules even if they are geographically far apart, implemented in different programming languages and on different operating systems. Thus, both data and programs can and must become a globally federated entity.

Scalability. Distributing data, applications and computing power also removes traditional borders to scalability. It is infeasible for a single computer to search a large number of text corpora – for the simple reason that this computer needs to download all data before it can perform any search on it. In a Grid, one can delegate the search to many agents that each search a manageable chunk of data; only the (presumably few) hits need to be transferred back to the machine that initiated the query.

Security. Within a Grid environment, we provide read and possibly even write access to data as well as hard- and software resources to many people we often do not know at all. We thus need a mechanism to manage access privileges on an inter-institutional basis – in Grid terminology, we need to manage Virtual Organizations (VOs). Joining a VO must be straightforward (provided that the VO permits it), it must be feasible to belong to many VOs at the same time, and the actual resource access has to be transparent from the user's perspective. It is mandatory, though, that the resource provider maintain full control over who can access which of their resources at any given time. Technologies that meet these requirements are under way [17, 30, 13].

4. TextGrid architecture

After evaluating the requirements from the use cases, it was clear that the provision of a storage Grid has a higher priority than a computing Grid. The success of the project will depend on whether distributed data will be easily accessible by the user and by the set of tools for text creation, modification, analysis and publishing. Nevertheless, these tools have to be accessible via the net and have to be interoperable and combinable by a workflow component. To make the development of these tools (and future ones provided from external programmers) as easy as possible, the project decided to use standard Web Services technologies, available in libraries for all popular programing languages and already used in Humanities Computing, e.g.,

in the projects Tapor (http://www.tapor.ca/) and DAM-LR (http://www.mpi.nl/DAM-LR/).

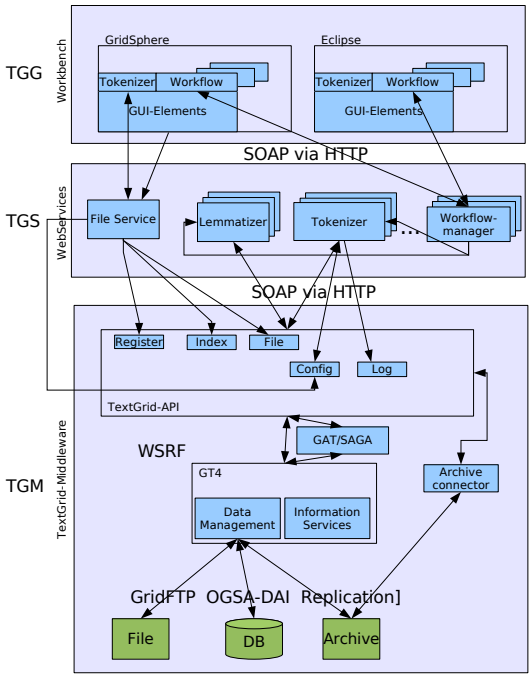

Figure 1. TextGrid Architecture Version I

Thus, a three layered architecture (see figure 1) was designed, consisting of a GUI layer (called TGG for TextGrid GUI), a rather simple service layer with the actual functionality (called TGS for TextGrid Services) and a rather complex middleware layer (called TGM for TextGrid Middleware), which provides Grid functionality to the Services. All communications between these three layers happen via simple Web Services protocols (SOAP and WSDL) over HTTP. Thus the architecture complies with the paradigm of Service Oriented Architecture (SOA) [10] where "everything is a service". In TextGrid, every non-interactive tool of the workbench, like a tokenizer, a lemmatizer, a collating tool, etc. will be a Web Service.

We call this architecture TextGrid Architecture Version I because this is just a first approximation to the end goal of a fully functional Grid including computing Grid facilities. The pragmatic reason behind this versioning is to as early as possible have a platform for experimentation and interoperation with other projects as well as to give access to the community.

Since two usage scenarios have to be provided for, 1. a user, who is on-line via a standard computer without any additional TextGrid software and 2. a user who wants to work through highly interactive processes, two user interface frameworks will be provided. For the first case a portal, based on GridSphere (http://www.gridsphere.org) will be implemented, which will be accessible via a standard web browser. The second scenario will be provided by implementing Eclipse Framework [7], one of the most advanced platforms for rich client applications, together with TextGrid-specific plugins and already available editor functionality. Its ability to develop user interfaces for eHumanities tasks has already been proven in projects like ARCHWay.

The advantage of separating the service layer from the GUI layer is that GUI components to the services can where applicable be provided in both user interface frameworks, through which the user can modify the service's behaviour by means of configuration. These simple GUI components will produce a configuration file that will be created and stored in the middleware. A pointer to this file will be sent to the service that will then retrieve it from the middleware.

Services and Modules that TextGrid identified as essential for the first phase include an XML editor, a link editor to annotate digital facsimiles, a textual annotator, a tokenizer, a lemmatizer, a collator, a streaming editor, a tool for lexical look-up, a link to a typesetting engine, a web publication component, and others. Powerful streaming editors could be implemented, e.g., by providing an XSLT processor or a Perl interpreter (with reduced functionality to prevent security issues) via a Web Service.

A specific GUI component will enable the user to align the different services in more or less complex workflows. This component will produce an XML-file with the workflow description (containing pointers to input, output and configuration files) which will directly be sent to the workflow enactment service [32]. Such a workflow editor has the additional benefit that it documents the programmatic steps on which the research results are based.

The workflow enactment service will call the specific tool services and monitors their execution. The tool services interact with the TGM for accessing input and output files as well as for accessing the specific configuration and for using a unified logging service. There will be one or more special services in the TGS layer for direct user interaction with the middleware. Via these services the user can for instance search for existing files via metadata, register and publish new files to the middleware, as well as authenticate to the system, etc. The diagram does not contain all functionalities, but only a subset that clarifies the architectural principles.

The TGM is itself a multi-layered construct. The upper level connects with the TGS (and TGG) layer via several Web Services, functioning as a remote API. This "API" encapsulates all Grid functionality. This means that TGM has to provide gatewaying functionality to intermediate between stateless simple Web Services and stateful WSRF based Grid services provided by the underlying Grid in-

frastructure that itself provides all needed data management and information services. The project has decided to deploy Globus Toolkit 4 (GT4, http://www.globus.org) as Grid infrastructure. Depending on the readiness of respective adaptors, an intermediate standard Grid API like GAT (http://www.gridlab.org/gat) or more promisingly SAGA (https://forge.gridforum.org/projects/saga-rg/) might be deployed to gain the possibility to change or additionally connect to other Grid infrastructures like gLite (http://glite.web.cern.ch/glite/). The TGM is also in charge of user management, authentication and authorization services. Here new developments combining GT4 and Shibboleth [13], a software used for federated identity management via standards will be carefully followed, since Shibboleth utilising the user management systems of the home organisations of the researchers seems to be more promising with respect to the target community than establishing a dedicated Public Key Infrastructure.

Often data Grid software is split into three components: The lowest level ensures reliable, efficient and secure data transfer. Data management components address the registration and location, verification, and also the replication of data objects. On the highest level information services cover metadata management, storage allocation, and policy management in the storage Grid. In the Globus middleware, these functions are covered by the Globus Components for Grid Data Management (http://www.globus.org/grid_software/data/) combined with the Metadata Catalog Service as information service (MCS, http://www.globus.org/grid_software/data/mcs.php). These could be combined with other modules implementing specific functions. For example metadata management could be based on an RDF Triple Store such as Sesame (http://www.openrdf.org/) instead of the native Globus components in order to better model semantic relations between objects.

5. Integrating archives

As illustrated above, there is a myriad of exciting initiatives producing scientific texts and other relevant resources. TextGrid aims to establish a network of distributed archives to merge those assets and make them reusable. The architectural model for this archive network aims to be open for any relevant initiative to participate, while retaining its autonomy. However, some organisational and technical agreements are necessary to ensure the reliable integration of the distributed assets on a semantic level. Grid technology weaves the archive network together. While the Grid layer has been discussed above, this section focuses on storage Grid aspects.

The key stakeholders in this are of course the scientific initiatives wishing to include their assets in TextGrid. There are both those projects that produce and process scientific texts for their specific research and archives that accommodate the output of a variety of relevant initiatives, perhaps as part of their institutional mission. The latter may be archives like the Oxford Text Archive in the UK (http://ota.ahds.ac.uk/), a cross-organisational archive with the mission of collecting and making available scientific texts and linguistic corpora. The Monumenta Germaniae Historica, a German institute for researching the middle ages, provides an institutional archive of digitisations and scientific texts from the European middle ages (Monumenta Germaniae Historica digital, http://www.dmgh.de). Besides cross-organisational or institutional text archives, others may be attached to university institutes or be established on a project basis. Such archives may establish the nodes in the TextGrid storage Grid. They establish the basic infrastructure to fulfil the organisational and technical aspects of reliable archival repositories. Ideally, any scientific project with assets to include in TextGrid has access to such a reliable archival repository. Library networks or other organisations could establish hosted archive services for those without such access.

Apart from the TextGrid target community and the data providers, there are secondary stakeholders that influence the TextGrid archive network. The know-how and tools from the Grid community are obviously its technical fundament. Furthermore, the open access [22] and particularly the preservation community [29] may contribute valuable concepts, standards, and tools.

A central standard in digital preservation is the Reference Model for an Open Archival Information System (OAIS) [20]. While the OAIS originates from the scientific community, all organisations active in long-term administration of digital resources – including national archives, libraries, and other cultural institutions – have found the OAIS concepts and terminology valuable. Another current activity in the preservation community are efforts to identify the attributes and responsibilities for an archive to be 'trusted' and define them such that they are auditable [21]. This initiative – even more than the OAIS – emphasises the organisational aspects of an archive. TextGrid partners are active in the preservation community, and they will refer to and promote preservation standards and tools whenever suitable for establishing the TextGrid virtual archive.

In tandem with implementation issues, however, semantic relations, the TextGrid metadata model, and the data model need to be defined. Participation of the text criticism community is particularly important and encouraged in order to create an open platform where any text archive can plug into and any research initiative can contribute to.

6. Acknowledgements

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.

References

[1] S. Anderson. E-science for the arts and humanities: A discussion paper. http://www.ahds.ac.uk/e-science/e-science-discussion-2004.pdf. AHRB E-Research Expert Seminar, 28th April 2004.

[2] J. Bradley. Text tools. In Schreibman et al. [26], pages 505–525.

[3] R. Busa. *Index Thomisticus*. Fromman Holzboog, Stuttgart, 1974.

[4] J. Caesar and T. MacCalla. e-humanities in a digital society. http://k55.nu.edu/resources/CHA/collateral/uploadedFiles/eHumantitiesInDigitalSocietyArticle.pdf. Second International Conference on New Directions in the Humanities: "Future: Human".

[5] D. Colazzo, C. Sartiani, A. Albano, P. Manghi, G. Ghelli, L. Lini, and M. Paoli. A typed text retrieval query language for XML documents. *Journal of the American Society for Information Science and Technology*, 53(6):467–488, 2002.

[6] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, C. Ursu, M. Dimitrov, M. Dowman, N. Aswani, and I. Roberts. *Developing Language Processing Components with GATE*. University of Sheffield. http://gate.ac.uk/sale/tao/index.html.

[7] Eclipse rich client platform FAQ. Wiki article at http://wiki.eclipse.org/index.php/RCP_FAQ.

[8] eHumanities – an NEH lecture series on technology & the humanities. http://www.neh.gov/online/ehumanities.html, 2001.

[9] S. Fish. Better SCM initiative: Comparison. http://better-scm.berlios.de/comparison/.

[10] I. Foster. Globus Toolkit version 4: Software for service-oriented systems. In *IFIP International Conference on Network and Parallel Computing*, volume 3779 of *LNCS*, pages 2–13. Springer, 2005. http://www.globus.org/alliance/publications/papers.php#gt4overview.

[11] I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, Amsterdam, 2004. Second Edition.

[12] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15(3):200–222, 2001.

[13] GridShib. http://gridshib.globus.org/.

[14] S. M. Hockey. *Electronic Texts in the Humanities: Theory and Practice*. Oxford University Press, Oxford, 2001.

[15] K. Kiernan, J. W. Jaromczyk, A. Dekhtyar, and D. C. Porter. The ARCHway project: Architecture for research in computing for humanities through research, teaching, and learning. *LLC*, 20(1):69–88, 2005.

[16] J. Kircz. E-based humanities and e-humanities on a SURF platform. http://www.surf.nl/download/ebased%20humanities.pdf. A Report commissioned by SURF-DARE.

[17] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, and T. Freeman. A multipolicy authorization framework for grid security. In *Proc. Fifth IEEE Symposium on Network Computing and Application*, 2006.

[18] W. McCarty. What is humanities computing? Toward a definition of the field. http://www.cch.kcl.ac.uk/legacy/staff/wlm/essays/what/, 1998.

[19] W. Morgenthaler. Gottfried Kellers Studienbücher – elektronisch ediert. *Jahrbuch für Computerphilologie*, 3, 2003. http://computerphilologie.uni-muenchen.de/jg03/morgenthaler2.html.

[20] Reference model for an open archival information system (OAIS). http://nost.gsfc.nasa.gov/isoas/. ISO/CCSDS 14721:2003.

[21] R. W. G. on Digital Archive Attributes. Trusted digital repositories: Attributes and responsibilities. Technical report, RLG-OCLC, May 2002. http://www.rlg.org/legacy/longterm/repositories.pdf.

[22] Berlin declaration on open access to knowledge in the sciences and humanities. http://www.zim.mpg.de/openaccess-berlin/berlindeclaration.html. In: Conference on the "Open Access to Knowledge in the Sciences and Humanities", 20-22 Oct 2003, Berlin.

[23] L. Qi, H. Jin, I. Foster, and J. Gawor. HAND: Highly available dynamic deployment infrastructure for Globus Toolkit 4. Submitted for publication, available at http://www.globus.org/alliance/publications/papers.php#HAND., 2006.

[24] A. H. Renear. Text encoding. In Schreibman et al. [26], pages 218–239.

[25] K. Schälkle, W. Ott, and H. Fuchs. TUSTEP: Tübinger System von Textverarbeitungs-Programmen. http://www.zdv.uni-tuebingen.de/tustep/index.html.

[26] S. Schreibman, R. Siemens, and J. Unsworth, editors. *A Companion to Digital Humanities*. Blackwell Publishing, 2004.

[27] C.-M. Sperberg-McQueen and L. Burnard, editors. *Guidelines for Text Encoding and Interchange (TEI P4)*. ACH/ALLC/ACL Text Encoding Initiative, Oxford, Providence, Charlottesville, Bergen, 2002.

[28] C.-M. Sperberg-McQueen and L. Burnard, editors. *TEI P5 Guidelines for Text Encoding and Interchange*. TEI Consortium, ACH, ACL, ALLC, Oxford Providence Charlottesville Nancy, 2005.

[29] UNESCO guidelines and charter for the preservation of digital heritage. http://portal.unesco.org/ci/en/ev.php-URL_ID=8967&URL_DO=DO_TOPIC&URL_SECTION=201.html, 2003. Prepared by Colin Webb (National Library of Australia).

[30] VOMS: Virtual organization membership service. http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html.

[31] P. Willet. Electronic texts: Audiences and purposes. In Schreibman et al. [26], pages 240–253.

[32] J. Yu and R. Buyya. A taxonomy of workflow management systems for grid computing. http://www.gridbus.org/reports/GridWorkflowTaxonomy.pdf.