Chapter 1
Collaborative Web-based Tools for Multi-layer Text Annotation

Chris Biemann, Kalina Bontcheva, Richard Eckart de Castilho, Iryna Gurevych, Seid Muhie Yimam

Abstract Effectively managing the collaboration of many annotators is a crucial ingredient for the success of larger annotation projects. For collaboration, web-based tools offer a low-entry way gathering annotations from distributed contributors. While the management structure of annotation tools is more or less stable across projects, the kind of annotations vary widely between projects. The challenge for web-based tools for multi-layer text annotation is to combine ease of use and availability through the web with maximal flexibility regarding the types and layers of annotations. In this chapter, we outline requirements for web-based annotation tools in detail and review a variety of tools in respect to these requirements. Further, we discuss two web-based multi-layer annotation tools in detail: GATE Teamware and WebAnno. While differing in some aspects, both tools largely fulfill the requirements for today's web-based annotation tools. Finally, we point out further directions, such as increased schema flexibility and tighter integration of automation for annotation suggestions.

1.1 Introduction

In this chapter, we discuss the topic of scaling annotation with multi-user web-based tools. Making annotation tools available via the web, on any computer running a web browser, and without installation efforts, facilitates the work of annotators significantly, and unlocks a distributed, not necessarily tech-savy workforce. At the same time, a web-based architecture has ramifications regarding tool engineering, workflow management, and data flow modeling. After motivating the use of web-based tools for annotation more elaborately in Section 1.1.1 and providing a survey of tools that only partially support web-based collaborative and/or distributed multi-user annotation projects in Section 1.1.2, we list requirements and desiderata for such tools in Section 1.2 and discuss the various ways in which these can be implemented, as well as lay out user roles. In Section 1.3, two open-source, collaborative annotation tools are discussed in detail: the GATE Teamware tool, a project that leverages the well-known GATE NLP platform over the web, and WebAnno, a more lightweight tool for linguistic annotations with an interface to crowdsourcing. After their presentation, both tools are compared and evaluated against the requirements in Section 1.3.3. Finally, Section 1.4 concludes and gives a further outlook on future developments.

1.1.1 Motivation

Collaborative annotation with general-purpose/multi-layer web-based tools has several advantages over domain specific annotation tools. Below are important characteristics and benefits of such tools:

- *Enhanced flexibility*: A general-purpose annotation tool provides better flexibility, in such a way that any type of annotation layers can be created, depending on the data collection need of the target application.
- *Lower training effort*: A tool that can easily be employed by users with basic web browser experiences does not require specific training. It also runs flawlessly without extra installation efforts, and can be updated centrally.
- *Unlocking a larger workforce*: The main goal of an annotation tool is to generate large annotated corpora. Similar to crowdsourcing platforms, it is possible to generate larger amount of annotated corpora more quickly by making them accessible to larger workforce.
- *Distributed annotation*: The collaborative annotation tool will be used in a distribution fashion with the only requirement being internet connectivity. Annotators can work at any time and from anywhere, without concerns for data losses and continuous intervention to save the data.
- *All-in-one solution*: The re-use of generic infrastructure for e.g. annotator management, agreement computation, and project workflows.
- *Open source*: An open source annotation tool can be extended with new functionalities, and is thus subject to a collaborative (programming) process. This flexibility makes a tool more attractive for people that conduct and oversee annotation projects.

1.1.2 Related Work

While web-based tools clearly have advantages for multi-user scenarios, for a long time, web technology was not suited for doing any complex annotation tasks. The visualization of annotation structures like constituency trees, dependency relations, or co-reference relations requires graphical capabilities that were difficult to realize in a web browser and, in particular, across different browser implementations. The annotation process also relies heavily on interactions such as marking spans of texts, dragging relations, connecting elements, or aligning data, which were difficult to implement in web browsers.

Due to the rapidly developing browser technology, maintenance efforts for sophisticated browser-based applications could hardly be handled by the scientific community. Consequently, researchers had to decide between a simplistic browser-based annotation tool, or a more sophisticated implementation as a specialized application. These latter applications were usually single-user applications.

Specialized Single-User Tools

The fact that many annotation tools focus on specific types of annotations, e.g. treebank structures, co-reference relations, or span-based annotations, may also be the consequence of the difficulties of adequately modelling the annotation data and implementing a sophisticated user interface on top of the data model. Examples for such tools are MMAX2 [34], WordFreak [33], Knowtator [37] and the NITE XML toolkit (NXT) [10]. MMAX2 focusses on annotating relations, e.g. co-reference chains. WordFreak is supporting several types, with different interfaces for e.g. span and constituency annotation inside the same tool. Knowtator provides support for very complex schemata, and is deployed as a single-user Protégé plugin. NXT targets speech and video annotation and transcriptions, implementing sophisticated search capabilities over the annotated data. TrEd [38] is a tool that supports all kinds of annotations that involve tree structures, and Annotate [5] is a treebanking tool that can interact with external programs for automatic pre-annotation. A more flexible framework in the single-user space is Callisto [18], which is a configurable linguistic annotation workbench that allows plugging in specific interfaces for different types of annotations.

Multi-User Standalone Tools

An attempt of adding multi-user capabilities to a stand-alone annotation application was undertaken with ELAN [6]. This tool targets the annotation of video and audio. It was attempted to integrate peer-to-peer networking technology to enable users to share data with each other. However, this idea appears to have largely been abandoned.[1]

A more simplistic but effective approach was undertaken with SALTO [8], an application for relation annotation (mainly semantic roles). Documents can be distributed to specific annotators by placing them in folders, e.g. on a shared network drive. Annotators receive a document via the *in* folder, place them in the *work* folder while annotating, and finally in the *out* folder when the annotation is complete. Eventually, annotations from different annotators are merged and reconciled via an extension of the specialized interface. Thus, even though not web-based, without

[1] The corresponding code still is present in ELAN 4.6.1, but is disabled and appears not to have been touched for several years.

real user and workflow management, and with a comparatively primitive approach, SALTO fully implements a distributed multi-user annotation scenario – albeit for a specific annotation type, and with installation efforts by annotators.

Shared Database Tools

A shared database for accessing corpora and storing annotations is used in the annotation tools developed by the Linguistic Data Consortium [31]. Development of this suite of tools was largely driven by project requirements and covers aspects of project management, adjudication and quality control. The tool collection, however, is still not web-based, as they are typically used by professional annotators producing a high volume of annotations, which offsets the time investment of local installation and training.

Web-based Tools

A web-based annotation solution was provided by Serengeti [41], a tool for annotating anaphoric relations and lexical chains. Serengeti also supports a multi-user distributed scenario in which multiple annotators work in parallel on a set of texts, then annotations are compared to each other, and quality is measured before the annotations are merged in a specialized comparison UI. To realize its sophisticated user interface, however, Serengeti had to make a compromise: it ties in heavily with a single specific browser, Firefox, which makes it prone to becoming outdated as the browser landscape changes.

Arborator [24] is a web-based tool for the purpose of annotating dependency structures. It employes a distributed annotation mode. Adjudicators can are allows to view all annotations from all users, to compare, and merge them. A single installation of Arborator can accomodate multiple annotation projects in parallel.

A more browser-independent web-based annotation tool is brat [40]. Its annotation interface is based on the SVG[2] standard supported by most modern browsers. Still, it works best on browsers based on the WebKit[3] engine, a software component designed to allow web browsers to render web pages. Brat supports the annotation of spans and relations between spans, but it does not support the higher-level annotations required for treebanks, such as constituency structures. Brat supports a collaborative annotation scenario, in which multiple users work on the same annotations in parallel: Changes made by one annotator are immediately visible to other annotators working on the same document at the time. In this survey, brat is the only annotation tool that advocates this collaborative mode, as opposed to supporting distributed per-user annotation. In collaborative mode, there is no need to compare and

[2] http://www.w3.org/TR/SVG/

[3] https://www.webkit.org/

merge annotations from different users. However, there is also no way to compute inter-annotator agreement, contributions per user and other user-related metrics.

Recently, Anafora [12] was released, which is a general-purpose web-based annotation tool. It is targeted to annotations regarding information extraction tasks and span annotations, and supports annotation as well as curation. User roles and access restrictions are modeled directly on the Linux file system of the server on which Anafora is installed. Anafora stores its data in a proprietary XML format, and is available under a permissive open-source license. Of the tools discussed in this section, it comes closest to the desiderata that we will discuss next.

The annotation workbench Argo [39] also contains a web-based annotation editor. It serves to inspect and optionally correct output that has been produced by an automatic processing pipeline that was built and run using the workbench. The visualization capabilities of the editor appear to be limited to a colored highlighting of spans. Neither interlinear labels nor relations appear to be supported. Further, each user appears to be able to only view and edit results produced by their own pipelines. Collaborative annotation and adjudication seems to be beyond the current scope of Argo.

Table 1.1 compares some of these tools with selected properties.

Table 1.1 Comparison of annotation tools with their selected properties

Properties	Anafora	Annotate	Arborator	Argo	brat	MMAX2	NXT	WordFreak
Reference	[12]	[5]	[24]	[39]	[40]	[34]	[10]	[33]
License	ASL 2.0	proprietary closed source	AGPL 3.0	proprietary closed source	MIT	ASL 2.0	GPL v2	MPL
Web-based	yes	no	yes	yes	yes	no	no	no
Annotate in browser	yes	NA	yes	yes	yes	NA	NA	NA
Adjudication support	yes	no	yes	no	NA	NA	NA	NA
Multi-user	yes	yes	yes	no	yes	no	no	no
Mode	distributed	collaborative	distributed	NA	collaborative	NA	NA	NA
User management	yes	unknown	yes	yes	yes	NA	NA	NA
Manage users in browser	no	no	yes	unknown	no	NA	NA	NA
Global roles	yes	unknown	yes	unknown	no	NA	NA	NA
Project roles	annotator, adjudicator, administrator	unknown	annotator, adjudicator, administrator	no	NA	NA	NA	NA
Project support	yes	no	yes	yes (Collection is comparable to projects)	yes (Pipelines comparable to projects)	yes (Collection is comparable to projects)	NA	NA
Manage projects in browser	no	NA	no	no	no	NA	NA	NA
Corpora shared between projects	no	NA	no	NA	no	NA	unknown	NA
Configurable types	yes	no	no	no	yes	yes	unknown	yes
Configurable tag sets	yes	no	yes	no	yes	yes	unknown	yes
Workflow support	no/automatic	unknown	yes	yes	no	NA	NA	NA
Automatic pre-annotation	no	yes	no	yes	no	no	no	via plugins

From Table 1.1, it is evident that discussed tools do not support all of the desired annotation tool properties such as web-based annotation and configuration, configurable annotation types, workflow management, and automatic pre-annotation.

Comparisons of Annotation Tools

In [19], the authors develop criteria and requirements for XML-based (not web-based) linguistic annotation tools. As requirements, they define diversity of data, multi-level annotation, simplicity, customizability, quality assurance, and convertibility and compare five tools with respect to their usability as well as these requirements. The management of annotation tools in focus of [29], who address especially the notion of extensibility and adaptability of annotation tools in an environment that supports user, project and configuration management.

1.2 Distributed Annotation

In this section, we discuss aspects and requirements for annotation tools that collect annotations from multiple users that work in a distributed fashion and give recommendations for tool design.

1.2.1 Requirements for Web-based annotation tools

As annotation efforts have been continuously ongoing ever since the release of the Brown corpus [22], and it is commonly regarded as advantageous to annotate the same text in multiple ways, and it is common to have corpora with multiple and overlapping annotations that can be consumed by different NLP applications [35]. A web-based annotation tool should support the visualization and annotation of such annotation layers in the most convenient way for annotators. Architecture of such a system should also facilitate the integration of arbitrary annotation layers with minimal efforts.

Web-based collaborative text annotation is a complex process, which involves different kinds of actors and requires a wide range of automatic pre-processing, user interfaces, and monitoring tools. From a high-level methodological perspective, web-based text annotation frameworks need to support annotation efficiency, consistency, scale, good interfaces, and clear procedures [26]. Corpus management and quality control are very important components of a distributed web-based collaborative annotation tool. Adjudicators should have the possibility to analyse different annotations so as to maintain a quality corpus output. It is also a requirement to display inter-annotator agreement (IAA), which provides information about the reliability and consistency of annotations. These translate into a set of functional requirements, which need to be met:

1. *Multi-role support*, including user groups, access privileges, annotator training, quality control, and corresponding user interfaces.
2. *Shared, efficient data storage* to store and access text corpora and annotations.
3. *Support for automatic pre-annotation services* and their configuration, to help achieve time and cost savings.
4. *Flexible workflow engine* to model complex annotation methodologies (e.g. [26]) and interactions.
5. *Web-based user interfaces*, that are easy to learn and use, without a need for local software installation. They also need to include customisable templates for common annotation tasks, and support annotator comments.
6. *Support for open linguistic annotation standards* (e.g. ISO/TC 37/SC 4 [27].), and compatibility with a wide range of exchange formats

Next, we will discuss the first four functional requirements in further detail. The fifth one, user interfaces, will be discussed on an exemplary basis. The last one regarding standardization of formats is not in the focus of this chapter.

1.2.1.1 Multi-Role Support and Division of Labour

For a distributed, web-based collaborative annotation tool, role-based access control is a crucial component of the system. Project managers should create and define projects including their tagsets and annotation layers, create users with differing roles, and handle corpus management. Depending on the roles, users need to execute different stages of an annotation project workflow: annotators can add/remove annotations to a document, while curators are responsible for reconciling conflicting annotations. As annotation projects differ in complexity and size, there is no reason why the same user should not be assigned multiple roles, e.g. being project manager and annotator in the same project. In more detail, we argue that it is necessary to distinguish the following four user roles:

Annotators are given a set of annotation guidelines and often work on the same document independently and concurrently. In order to be able to employ less-specialised annotators, annotation interfaces need to be easy to learn. In addition, it is desirable to provide an automatic training mode for annotators where their performance is compared against a known gold standard and all mistakes are identified and explained to the annotators, until they have mastered the guidelines.

Since annotators and project managers are often working at different locations, there needs to be a communication channel between them, e.g. instant messaging. If a manager is not available, an annotator should also be able to mark an annotation as requiring discussion and then all such annotations should be shown automatically in the manager console. The platform should automatically save annotations without user interventions so that if they close the annotation tool, the same document must be presented to them for completion next time they log in. Optionally, some projects might need to restrict the annotators to a maximum of *n* documents (given as a number or percentage), in order to prevent an over-zealous annotator from introducing an individual bias.

From a user interface perspective, there needs to be support for annotating document level metadata (e.g. language identification), word-level annotations (e.g. named entities, POS tags), and relations and trees (e.g. co-reference, syntax trees). Ideally, the interface should offer some generic components for all these, which can be customised with project-specific tags and values via an XML schema or web based configurations. The framework also needs to be extensible, so specialised UIs can easily be plugged in, if required.

Project managers are typically in charge of defining new corpus annotation projects and their workflows, monitoring annotation progress, dealing with annotator performance issues, and carrying out annotator training. They also define the annotation guidelines, the associated schemas (or tagsets), and prepare and upload the corpus to be annotated. Managers also make methodological choices: whether to have multiple annotators per document; how many; which automatic NLP services need to be used to pre-process the data; and what is the overall workflow of annotation, quality assurance, adjudication, and corpus delivery.

Managers need a project monitoring tool where they can see:

- Whether a corpus is currently assigned to a project or, what annotation projects have been run on the corpus with links to these projects or their archive reports (if no longer active). Also provides links to the annotation schemas for all annotation types currently in the corpus.
- Project completion status (e.g., 80% manually annotated, 30% adjudicated).
- Annotator statistics within and across projects: which annotator worked on which document, how long it took, and what was the IAA.
- The ability to lock a corpus from further editing, either during a project, or after it has been finished.

Curators are responsible for annotation adjudication and creating the gold-standard. Therefore, in addition to the standard annotation interfaces, they have access to IAA statistics and a curation user interface (appropriate for comparing the differences between multiple annotators). The curator, therefore, generates a single annotation document out of the annotation documents the annotators have provided. Even though manual curation adds to the cost of corpus annotation, it is typically very beneficial to include that as part of the workflow, since it improves the annotation quality in hard-to-solve cases, and acts as a quality check [26].

Administrators define roles for other users, create user accounts, create and configure services, and monitor workflow processes.

1.2.1.2 Remote, Scalable Data Storage

Given the multiple user roles and the fact that several annotation projects may be running at the same time with different remotely located teams, the data storage layer needs to scale to accommodate large, distributed corpora and have the necessary security in place through authentication and fine-grained user/group access control [7].

For commercially conducted projects, data security is paramount and needs to be enforced as data is being sent over the web to the remote annotators. This is often less of a concern in publicly funded scenarios. Support for diverse document input and output formats is also necessary, especially the stand-off ones (e.g. XCES [28]), which can minimise network traffic by transmitting only a relevant subset of all annotations.

Since multiple users must be able to work concurrently on the same document, there needs to be an appropriate locking mechanism to support that: either every user works on her own copy, or assigning documents to single users at a time is handled by the server. The data storage layer also needs to provide facilities for storing annotation guidelines, annotation schemas, and, if applicable, ontologies or other lexical resources. Last, but not least, a corpus search functionality is often required, at least one based on keywords, but ideally also including document metadata (e.g. author, year, domain, etc.) and linguistic annotations.

1.2.1.3 Automatic Pre-Annotation Services

Automatic pre-annotation services can reduce significantly annotation costs (e.g. annotation of named entities), but unfortunately they also tend to be domain or application specific. Also, several services might be needed in order to bootstrap all annotation types, e.g. named entities, co-reference, and relation annotation modules. Therefore, the architecture needs to be open so that new services can be added easily. Such services can encapsulate different NLP modules and take as input one or more documents (or an entire corpus). The automatic services also need to be scalable in terms of processing time, in order to minimise their impact on the overall project completion time. The project manager should also be able to choose services based on their accuracy on a given corpus.

Machine Learning (ML) modules can be regarded as a specific kind of automatic service. A mixed initiative system [17] can be set up by the project manager and used to facilitate manual annotation behind the scenes. This means that once a document has been annotated manually, it will be sent to train the ML service which internally generates an ML model. This model will then be applied by the service to any new document, so that this document will be partially pre-annotated. The human annotator then only needs to validate or correct the annotations provided by the ML system, which makes the annotation task significantly faster [17].

There are principally two ways to integrate automatic pre-annotations: One way is to include this mechanism in the annotation tool, which makes its use a more seamless experience but adds to the size and complexity of the tool. Another way is to keep automatic processing outside of the tool and provide a way to import automatically pre-annotated documents for correction, and export the annotated data in order to train an ML module. This keeps the use of the specific automatic method more flexible and thus supports a wider range of different annotation layers. However, this comes with increased effort for the project manager, who has to manually handle import and export, as well as to train and to apply the ML module.

Since most of its future annotation use cases are unknown during tool development, users should be able to leverage pre-automatic annotation both ways in a maximally flexible tool.

1.2.1.4 Flexible Workflow Engine

In order to have an open, flexible model of corpus annotation processes, we need a powerful workflow engine which supports asynchronous execution and an arbitrary mix of automatic and manual steps. For example, manual annotation and adjudication tasks are asynchronous. Resilience to failure is essential and workflows need to save intermediary results from time to time, especially after operations that are very expensive to re-run (e.g. manual annotation, adjudication). The workflow engine also needs to have status persistence, action logging, and activity monitoring, which form the basis of the project management tools.

In a workflow, it should be possible for more than one annotator to work on the same document at the same time; however, during adjudication, all affected annotations need to be locked to prevent concurrent modifications. For separation of concerns, it might be useful for the same corpus to be part of more than one active projects. Similarly, the same annotator needs to be able to work on several annotation projects.

1.2.2 Tool Design Principles

A web-based collaborative corpus annotation tool should support well-designed client-server architecture that facilitates efficient annotation. The server should support concurrent access to resources where annotators can work on single/multiple copy of their own annotation document. There should be a clear separation between the UI, the server structure and the data. Ideally, the architecture should enable a replacement of user interfaces or the server implementation with minimum effort. On the other hand, the generated data should be consumed easily by different implementations of a server or UIs. The annotators or curators can concentrate on the main annotation task where persistence of annotations is managed transparently by the system. This further saves the annotator's time, as well as preventing data loss. The amount of data transmitted over the network strongly affects the availability of the system.

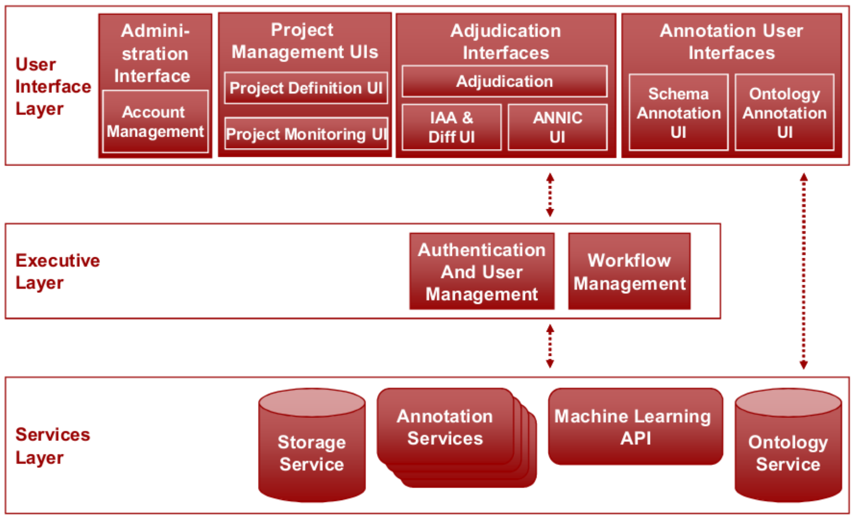


Fig. 1.1 GATE Teamware architecture diagram showing three layers: the user interface layer, the executive layer and the services layer

1.3 Two Web-based Collaborative Annotation Tools

In this section, we discuss two collaborative web-based multi-layer annotation tools in detail: GATE and WebAnno. Both tools adhere largely to the design principles and desiderata given in the previous sections. While some parts are very similar between both tools, they also differ in particular aspects.

1.3.1 GATE Teamware

This section presents GATE Teamware[4] [4], an open-source, general-purpose text annotation framework and a methodology for the implementation and support of complex annotation projects. It has a web-based architecture, where a number of web services (e.g. document storage, automatic annotation) are made available via HTTPS and the users interact with the text annotation interfaces through a standard web browser.

It is based on GATE [15, 14], a widely used, scalable and robust open-source NLP platform. GATE comes with numerous reusable text processing components for many natural languages, coupled with a graphical NLP development environment and user interfaces for visualisation and editing of linguistic annotations, parse trees, co-reference chains, and ontologies. GATE Teamware however was created

[4] Source code and documentation are available from http://gate.ac.uk/teamware/

specifically to be used by non-expert annotators, as well as to enable methodologically sound, efficient, and cost-effective corpus annotation projects over the web.

In addition to its research uses, GATE Teamware has also been tested as a framework for cost-effective commercial annotation services, supplied either as in-house units or as outsourced specialist activities. Several test annotation projects have been conducted in the domains of bio-informatics and business intelligence, with minimal training and producing high quality corpora. For example, Meurs et al [32] apply GATE Teamware to the task of building a database of fungal enzymes for biofuel research. Their results show that using GATE Teamware for automatic pre-annotation and manual correction increases the speed with which papers can be processed for inclusion in the database by a factor of around 50%.

GATE Teamware's novelty is in being a generic, reusable, web-based framework for collaborative text annotation. Unlike other tools (see Section 1.1.2), GATE Teamware provides the required multi-role methodological support, as well as the necessary tools to enable the successful management of distributed annotation projects. It has a service-based architecture which is parallel, distributed, and also scalable (via service replication) (see Figure 1.1). Each section of the architecture diagram will be explained in more detail below, from the bottom up.

Similar to other server-side software, GATE Teamware installation is a specialised, non-trivial task with associated costs, in terms of significant time and staff expertise required. In order to lower this barrier and provide zero startup costs, we have made available cloud-based GATE Teamware virtual machines[5], that can be turned on and off as required. In addition, the GATECloud.net [42] integration makes it easy to choose a set of automatically annotated documents and send these into a GATE Teamware instance. There is also a virtual machine distribution that can be downloaded and run locally instead.

1.3.1.1 Services Layer

The services layer includes the GATE document service, serving the data structures used in GATE Teamware and the GATE annotation services, coordinating the computational tasks.

The document storage service provides a distributed data store for corpora, documents, and annotation schemas. Input documents can be in all major formats (e.g., XML, HTML, PDF, ZIP), based on GATE's comprehensive support. When a document is uploaded in GATE Teamware, the format is analysed and converted into a single unified, graph-based model of *annotation*: the one of the GATE NLP framework. Then this internal annotation format is used for data exchange between the service layer, the executive layer and the UI layer. The main export format for annotations is currently stand-off XML, including XCES [28].

GATE Annotation Services (GAS) provide automatic pre-annotation services, e.g. running the ANNIE named entity recogniser from GATE [14]. Annotation

[5] Available to use and trial at http://gatecloud.net.

pipelines, installed in GATE Teamware as a GAS, are used in projects to prepare data. GATE Teamware includes a number of pre-packaged GASes to perform common functions, such as moving and copying annotations between different sets. Managers and administrators can view and edit GASes.

1.3.1.2 The Executive Layer

The executive layer includes authentication and user management, as well as configuration of which UI components are accessible to which user roles (the defaults are shown in Figure 1.1).

The second major part is the workflow manager, which is based on JBoss jBPM[6] and has been developed to meet the requirements discussed in Section 1.2.1.4 above. It not only assigns dynamically annotators to available jobs, but also measures how long annotators take, how good they are at annotating, as well as reporting overall progress and costs.

1.3.1.3 The User Interfaces

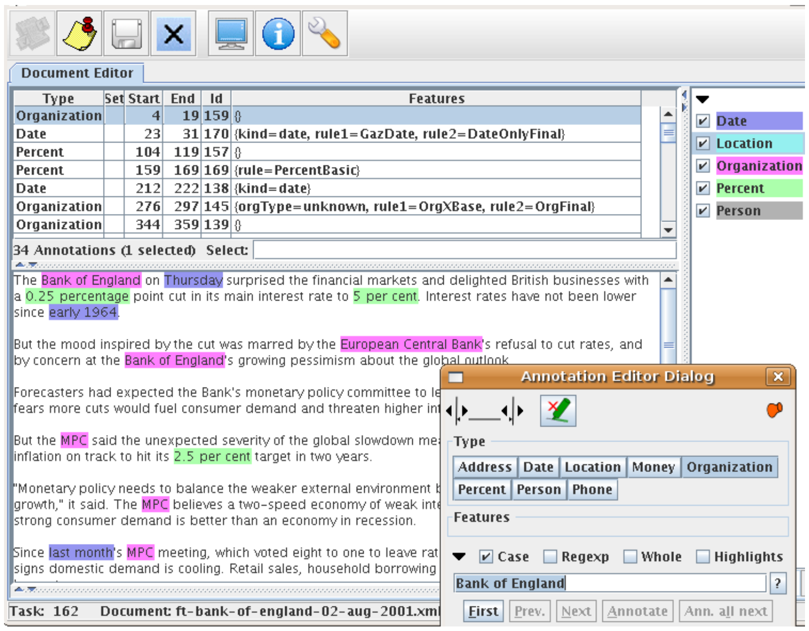
The GATE Teamware user interfaces run in a web browser and do not require prior installation. After the user logs in, the system checks their role(s) and access privileges, to determine which interface they are shown (annotator, manager, or administrative). Annotators only see the annotation interfaces, whereas managers see the project management and adjudication interfaces. GATE Teamware administrators have access to all user interfaces, including a dedicated administration interface.

Annotators carry out manual annotation, from scratch, or by correcting automatic annotation generated by the GATE processing resources. The most frequently used annotation UI is the generic schema-based annotator UI (see Figure 1.2). The annotation editor dialog shows the annotation types (or tags/categories) valid for the given project and optionally their features (or attributes). These are generated automatically from the annotation schemas assigned to the project by its manager. Annotation schemas define the acceptable types of annotations and attributes and thus allow the user interface to be customised, in a manner similar to other tools, such as Callisto [18] and MMAX2 [34].

The annotation editor also supports the modification of annotation boundaries, either through mouse clicks or keyboard shortcuts. In addition, advanced users can define regular expressions to annotate multiple matching strings simultaneously.

To add a new annotation, one selects the text with the mouse (e.g., "Bank of England") and then clicks on the desired annotation type in the dialog (e.g., Organization). Existing annotations are edited by hovering over them, which shows their current type and features in the editor dialog.

[6] http://www.jboss.com/products/jbpm/



Type	Set	Start	End	Id	Features
Organization		4	19	159	{}
Date		23	31	170	{kind=date, rule1=GazDate, rule2=DateOnlyFinal}
Percent		104	119	157	{}
Percent		159	169	169	{rule=PercentBasic}
Date		212	222	138	{kind=date}
Organization		276	297	145	{orgType=unknown, rule1=OrgXBase, rule2=OrgFinal}
Organization		344	359	139	{}

34 Annotations (1 selected) Select:

The Bank of England on Thursday surprised the financial markets and delighted British businesses with a 0.25 percentage point cut in its main interest rate to 5 per cent. Interest rates have not been lower since early 1964.

But the mood inspired by the cut was marred by the European Central Bank's refusal to cut rates, and by concern at the Bank of England's growing pessimism about the global outlook.

Forecasters had expected the Bank's monetary policy committee to le
fears more cuts would fuel consumer demand and threaten higher inf

But the MPC said the unexpected severity of the global slowdown me
inflation on track to hit its 2.5 per cent target in two years.

"Monetary policy needs to balance the weaker external environment b
growth," it said. The MPC believes a two-speed economy of weak inte
strong consumer demand is better than an economy in recession.

Since last month's MPC meeting, which voted eight to one to leave rat
signs domestic demand is cooling. Retail sales, household borrowing

Task: 162 Document: ft-bank-of-england-02-aug-2001.xml

[✓] Case [ ] Regexp [ ] Whole [ ] Highlights

Bank of England

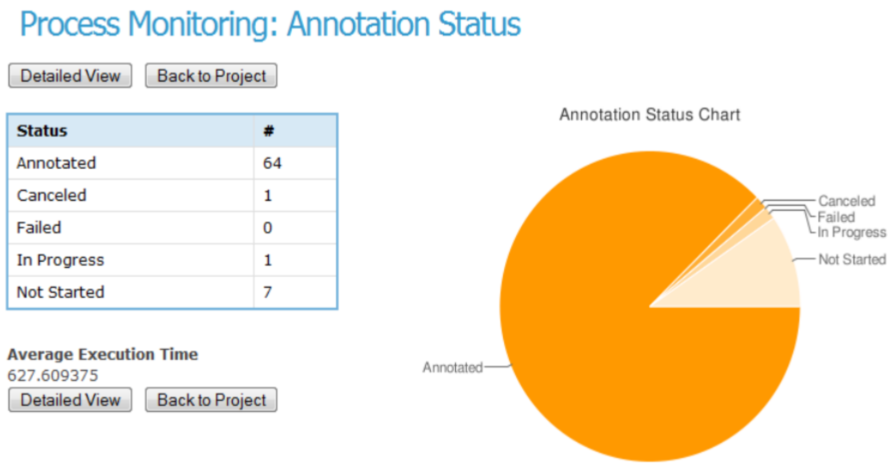
Fig. 1.2 The GATE Teamware schema-based annotator user interface, showing the document displayed with annotations indicated in coloured highlighting

Annotators can also control which annotation types are highlighted in the text, by selecting the corresponding check-boxes, shown at the top right side of Figure 1.2. By default, all types are visible, but this functionality allows users to focus on one category at a time, if required.

As discussed in Section 1.2.1.1, quality assurance is a key element of annotation projects. In GATE Teamware it is carried out by project managers. Tools available include IAA metrics (including f-measure and Kappa) to identify if there are differences between annotators; a visual annotation comparison tool to see quickly where the differences are per annotation type; and an editor to edit and reconcile annotations manually (i.e. adjudication) or by using external automatic services. See [4] for details.

Apart from adjudication, project managers are responsible for defining annotation guidelines and schemas. They choose relevant automatic services with which to pre- or post-process the data (optional), benchmark annotator performance and monitor the project progress. Project managers define annotation workflows, manage annotators, and liaise with the system administrators.

The project management web UI provides the front-end to the executive layer (see Section 1.3.1.2). In a nutshell, managers upload documents and corpora, define the annotation schemas, specifying the allowed annotation types and attributes, choose and configure the workflows and execute them on a chosen corpus. Workflows may be as simple as passing the documents to n human annotators, or more


Process Monitoring: Annotation Status

Status	#
Annotated	64
Canceled	1
Failed	0
In Progress	1
Not Started	7

Average Execution Time
627.609375

Fig. 1.3 The GATE Teamware progress monitoring interface

complex, for example, preprocess the documents to produce automatic annotations, pass each document to three annotators and then adjudicate the differences. There is a workflow wizard to facilitate this step [4]. The management console also provides project monitoring facilities, e.g. number of annotated documents, number in progress, and yet to be completed, as shown in Figure 1.3. Per annotator statistics are also available – time spent per document, overall time worked, average IAA, as well as per document statistics.

1.3.2 WebAnno

In this section, we provide an in-depth view of WebAnno [45, 46], a general purpose web-based annotation tool for a wide range of linguistic annotations. WebAnno offers annotation project management, freely configurable tagsets and the management of users in different roles. WebAnno uses technology from *brat* [40] for visualizing and editing annotations in a web browser. The architecture design allows adding additional modes of visualization and editing, when new kinds of annotations are to be supported. WebAnno can perform automatic pre-annotation of spans learned from provided or currently annotated data.

The overall architecture of WebAnno is depicted in Figure 1.4. The modularity of the architecture, which is mirrored in its open-source implementation[7], makes it possible to easily extend the tool or add alternative user interfaces for annotation

[7] Available for download at: http://webanno.googlecode.com/

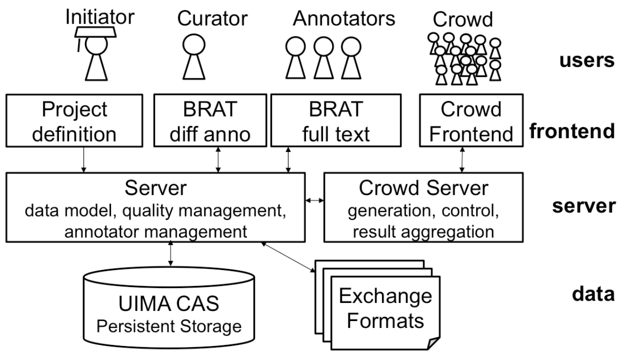


Fig. 1.4 System architecture of WebAnno, organized in users, front-end, back-end and persistent data storage.

layers are rather displayed with different annnotation front-ends, e.g. constituent structure or frame-based annotation.

In Section 1.3.2.1, we illustrate how different user roles are provided with different graphical user interfaces, and show the expressiveness of the annotation model. Section 1.3.2.2 elaborates on the functionality of the back-end, and describes how data is imported and exported, as well as our implementation of the persistent data storage.

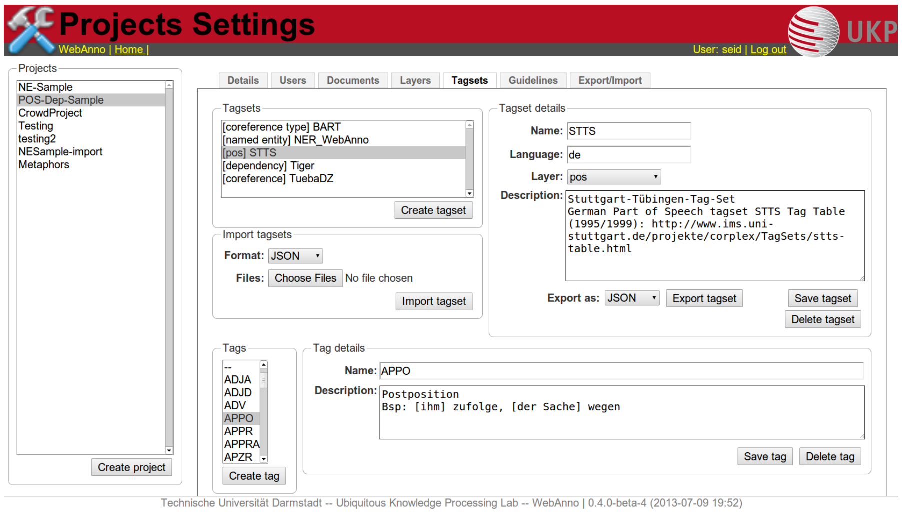
1.3.2.1 Front-end

The definition and the monitoring of an annotation project is conducted by the initiator (a project manager) (cf. Figure 1.4) in a project definition form. It supports creating a project, loading un-annotated or pre-annotated documents in different formats[8], adding annotator and curator users, defining tagsets, and adding/configuring the annotation layers. Only a project manager can administer a project. Figure 1.5 illustrates the project definition page with the tagset editor highlighted.

Annotation is carried out with an adaptation of the brat editor, which communicates with the server via Ajax [23] using the JSON [30] format. Annotators only see projects they are assigned to. The annotation page presents the annotator different options to set up the annotation environment, for customization:

- *Display window size:* For heavily annotated documents or very large documents, the brat visualization is very slow both for displaying and annotating the document. We use a paging mechanism that limits the number of sentences displayed at a time to make the performance independent of the document size.
- *Annotation layers:* Annotators usually work on one or two annotations layers, such as part-of-speech and dependency or named entity annotation. Overloading the annotation page by displaying all annotation layers makes the annotation

[8] Formats: plain text, CoNLL [36], TCF [25], UIMA XMI [21]



Projects Settings

WebAnno | Home | User: seid | Log out

Projects: NE-Sample, POS-Dep-Sample, CrowdProject, Testing, testing2, NESample-import, Metaphors

Tabs: Details | Users | Documents | Layers | Tagsets | Guidelines | Export/Import

Tagsets: [coreference type] BART, [named entity] NER_WebAnno, [pos] STTS, [dependency] Tiger, [coreference] TuebaDZ

Tagset details

Name: STTS

Language: de

Layer: pos

Description:
```
Stuttgart-Tübingen-Tag-Set
German Part of Speech tagset STTS Tag Table
(1995/1999): http://www.ims.uni-
stuttgart.de/projekte/corplex/TagSets/stts-
table.html
```

Import tagsets — Format: JSON — Files: Choose Files No file chosen

Export as: JSON

Tags: --, ADJA, ADJD, ADV, APPO, APPR, APPRA, APZR

Tag details

Name: APPO

Description:
```
Postposition
Bsp: [ihm] zufolge, [der Sache] wegen
```

Technische Universität Darmstadt -- Ubiquitous Knowledge Processing Lab -- WebAnno | 0.4.0-beta-4 (2013-07-09 19:52)

Fig. 1.5 Project definition: tagset editor. Note the hidden tabs "Details", "Users", "Documents", "Layers", "Guidelines", and "Export/Import"

and visualization process slower. WebAnno provides an option to configure visible/editable annotation layers.

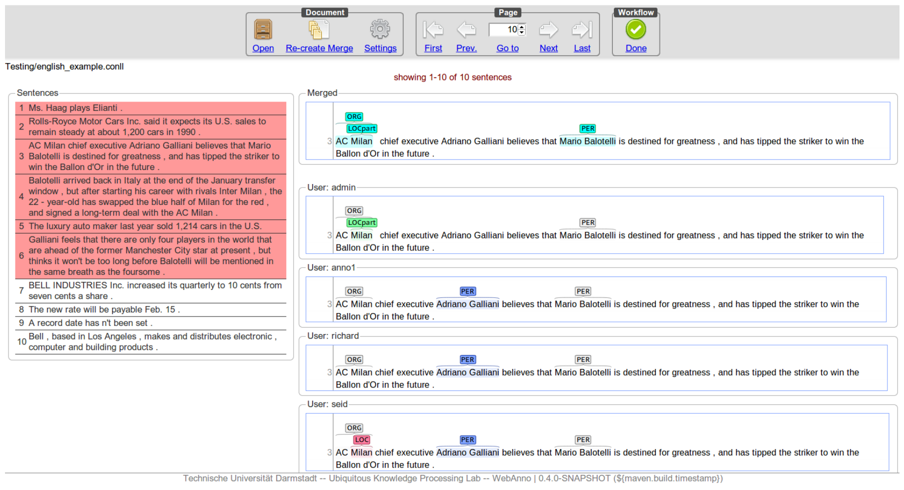
- *Immediate persistence:* Every annotation is sent to the back-end immediately and persisted there. An explicit interaction by the user to save changes is not required.

WebAnno implements a simple workflow to track the state of a project. Every annotator works on a separate version of the document, which is set to the state *in progress* the first time a document is opened by the annotator. The annotator can then mark it as *complete* at the end of annotation at which point it is locked for further annotation and can be used for curation. Such a document cannot be changed anymore by an annotator, but can be used by a curator. A curator can mark a document as *adjudicated*.

The curation interface allows the curator to open a document and compare annotations made by the annotators who already marked the document as *complete*. The curator reconciles the annotation with disagreements. The curator can either decide on one of the presented alternatives, or freely re-annotate. Figure 1.6 illustrates how the curation interface detects sentences with annotation disagreement (left side of Figure 1.6) which can be used to navigate to the sentences for curation.

Similar to the curation interface, the correction interface is implemented for projects with automatically annotated or pre-annotated documents where the user's task is correcting those annotations, as well as adding missing annotations.

WebAnno offers a tight loop to automatic pre-annotation: as soon as annotations are performed, they are used by the system to improve the pre-annotation machinery. This is realized by two different modes of automatic prediction: In *repetition mode*, further occurrences of a word annotated by the user are highlighted in the



Testing/english_example.conll

showing 1-10 of 10 sentences

Sentences

1 Ms. Haag plays Elianti .
2 Rolls-Royce Motor Cars Inc. said it expects its U.S. sales to remain steady at about 1,200 cars in 1990 .
3 AC Milan chief executive Adriano Galliani believes that Mario Balotelli is destined for greatness , and has tipped the striker to win the Ballon d'Or in the future .
4 Balotelli arrived back in Italy at the end of the January transfer window , but after starting his career with rivals Inter Milan , the 22 - year-old has swapped the blue half of Milan for the red , and signed a long-term deal with the AC Milan .
5 The luxury auto maker last year sold 1,214 cars in the U.S.
6 Galliani feels that there are only four players in the world that are ahead of the former Manchester City star at present , but thinks it won't be too long before Balotelli will be mentioned in the same breath as the foursome .
7 BELL INDUSTRIES Inc. increased its quarterly to 10 cents from seven cents a share .
8 The new rate will be payable Feb. 15 .
9 A record date has n't been set .
10 Bell , based in Los Angeles , makes and distributes electronic , computer and building products .

Merged

3 AC Milan chief executive Adriano Galliani believes that Mario Balotelli is destined for greatness , and has tipped the striker to win the Ballon d'Or in the future .

User: admin

3 AC Milan chief executive Adriano Galliani believes that Mario Balotelli is destined for greatness , and has tipped the striker to win the Ballon d'Or in the future .

User: anno1

3 AC Milan chief executive Adriano Galliani believes that Mario Balotelli is destined for greatness , and has tipped the striker to win the Ballon d'Or in the future .

User: richard

3 AC Milan chief executive Adriano Galliani believes that Mario Balotelli is destined for greatness , and has tipped the striker to win the Ballon d'Or in the future .

User: seid

3 AC Milan chief executive Adriano Galliani believes that Mario Balotelli is destined for greatness , and has tipped the striker to win the Ballon d'Or in the future .

Technische Universität Darmstadt -- Ubiquitous Knowledge Processing Lab -- WebAnno | 0.4.0-SNAPSHOT (${maven.build.timestamp})

Fig. 1.6 Curation user interface (left: sentences with disagreement; right: editor)

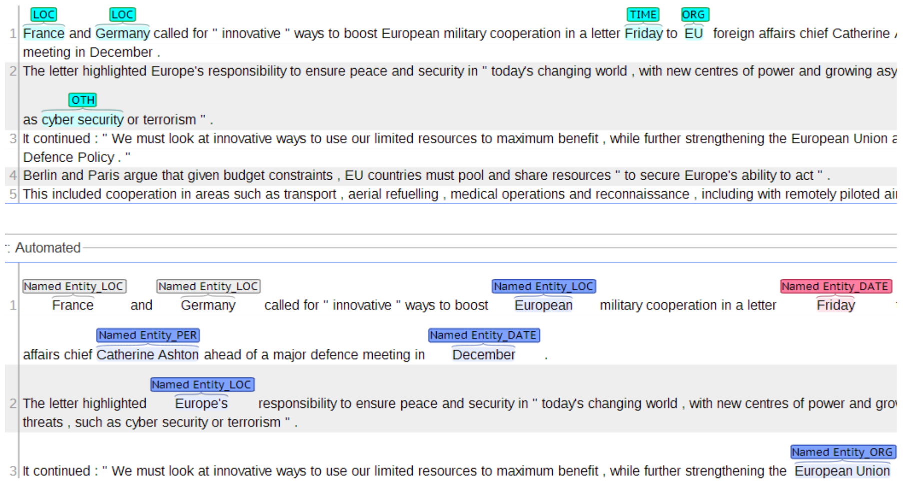
suggestion pane. To accept suggestions, the user can simply click on them in the suggestion pane. This basic – yet effective – suggestion is realized using simple string matching. The *learning mode* is based on MIRA [13], an extension of the perceptron algorithm for online machine learning which allows for the automatic suggestions of span annotations. MIRA was selected because of its relatively lenient licensing, its good performance even on small amounts of data, and its capability of allowing incremental classifier updates. The setup allows for maximum flexibility as it does not assume language-specific preprocessing – at cost of pre-annotation classifier performance, which for this reason cannot match highly specialized NLP components, whose output, however, can be imported for correction.

The lower panel in Figure 1.7 displays pre-annotated documents, while the upper panel presents the annotation panel where annotations are copied from the lower panel or new annotations are added by the user.

WebAnno has a monitoring component, which tracks the progress of a project. The project manager can check the progress and compute agreement with Kappa and Tau [9] measures. The progress is visualized using a matrix of annotators and documents displaying which documents the annotators have marked as *complete* and which documents the curator marked as *adjudicated*. Figure 1.8 shows the project progress, progress of individual annotators and the overall completion statistics.

Crowdsourcing is a way to quickly scale annotation projects. Distributing a task that otherwise will be performed by a controlled user group has become much easier. Hence, if quality can be ensured, it is an alternative to high quality annotation using large number of arbitrary redundant annotations [44]. For WebAnno, we have designed an approach where a source document is split into small parts that get presented to micro-workers in the CrowdFlower platform[9]. The crowdsourcing com-

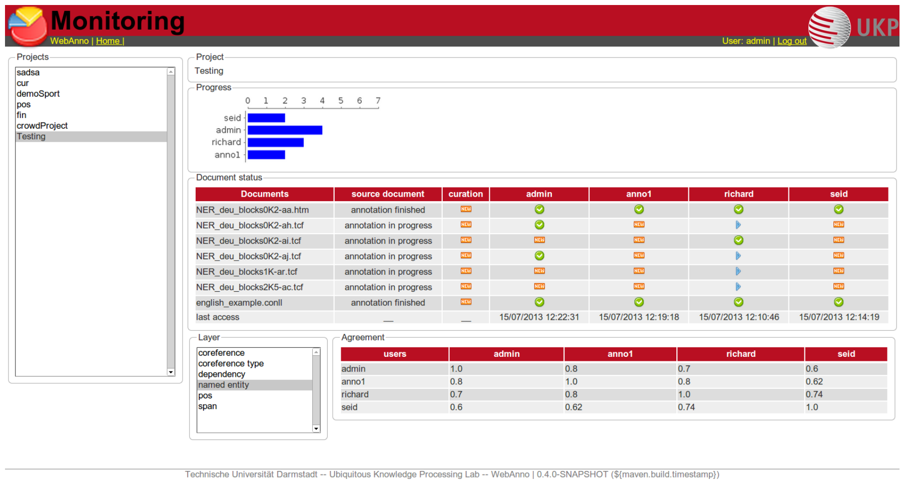
[9] www.crowdflower.com



1 France and Germany called for " innovative " ways to boost European military cooperation in a letter Friday to EU foreign affairs chief Catherine
meeting in December .
2 The letter highlighted Europe's responsibility to ensure peace and security in " today's changing world , with new centres of power and growing asy
as cyber security or terrorism " .
3 It continued : " We must look at innovative ways to use our limited resources to maximum benefit , while further strengthening the European Union
Defence Policy . "
4 Berlin and Paris argue that given budget constraints , EU countries must pool and share resources " to secure Europe's ability to act " .
5 This included cooperation in areas such as transport , aerial refuelling , medical operations and reconnaissance , including with remotely piloted ai

1 France and Germany called for " innovative " ways to boost European military cooperation in a letter Friday
affairs chief Catherine Ashton ahead of a major defence meeting in December .
2 The letter highlighted Europe's responsibility to ensure peace and security in " today's changing world , with new centres of power and gro
threats , such as cyber security or terrorism " .
3 It continued : " We must look at innovative ways to use our limited resources to maximum benefit , while further strengthening the European Union

Fig. 1.7 Correction user interface (lower: sentences with pre-annotations; upper: correction view)



Projects: sadsa, cur, demoSport, pos, fin, crowdProject, Testing

Project: Testing

Documents	source document	curation	admin	anno1	richard	seid
NER_deu_blocks0K2-aa.htm	annotation finished	NEW	✓	✓	✓	✓
NER_deu_blocks0K2-ah.tcf	annotation in progress	NEW	✓	NEW	▶	NEW
NER_deu_blocks0K2-ai.tcf	annotation in progress	NEW	NEW	NEW	✓	NEW
NER_deu_blocks0K2-aj.tcf	annotation in progress	NEW	✓	NEW	▶	NEW
NER_deu_blocks1K-ar.tcf	annotation in progress	NEW	NEW	NEW	▶	NEW
NER_deu_blocks2K5-ac.tcf	annotation in progress	NEW	NEW	NEW	▶	NEW
english_example.conll	annotation finished	NEW	✓	✓	✓	✓
last access	__	__	15/07/2013 12:22:31	15/07/2013 12:19:18	15/07/2013 12:10:46	15/07/2013 12:14:19

Layer: coreference, coreference type, dependency, named entity, pos, span

users	admin	anno1	richard	seid
admin	1.0	0.8	0.7	0.6
anno1	0.8	1.0	0.8	0.62
richard	0.7	0.8	1.0	0.74
seid	0.6	0.62	0.74	1.0

Technische Universität Darmstadt -- Ubiquitous Knowledge Processing Lab -- WebAnno | 0.4.0-SNAPSHOT (${maven.build.timestamp})

Fig. 1.8 The monitoring component showing project progress, annotators progress and document completion status (red and blue).

ponent is a separate module that handles the communication via CrowdFlower's API, the definition of test items and job parameters, and the aggregation of results. The crowdsourced annotation appears as a virtual annotator in the tool. As different layers need different crowdsourcing templates to address the limitations of crowd workers and crowdsourcing platforms, we currently only support named entity annotation.

1.3.2.2 Back-end

The back-end of WebAnno was implemented using Java (Wicket [16], Spring Framework [43], DKPro Core [11]). Hibernate and JPA [1] are used for persisting objects in a MySQL database. We store serialised UIMA CAS objects [21] in the file system for every annotation document.

Project definitions including project name and descriptions, user-defined annotation layers, tagsets and tags, and user details are kept in a server-side database, whereas the documents and annotations are stored in the server file system. WebAnno supports limited versioning of annotations, to protect against the unforeseen loss of data. To enable versioning of WebAnno annotations, the administrator sets the interval between backups, and how long backups should be stored. Figure 1.9 shows the database entity relation (ER) diagram.

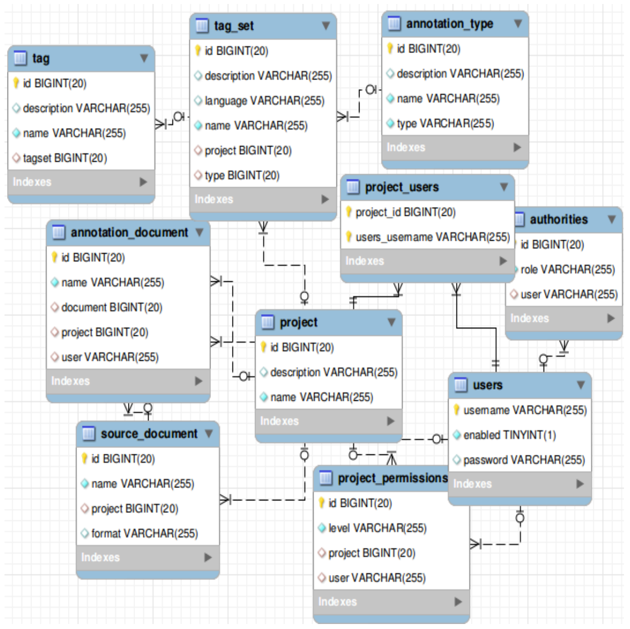


Fig. 1.9 WebAnno: Diagram, showing the persistence storage structures.

Although WebAnno has only recently been released to the public, it is already being used by a number of industry projects as well as research projects. Below are some of the projects WebAnno is being used for.

Current schemata and guidelines for linguistic annotation have been developed predominantly for the description of newspaper language. Also, automatic annotation tools continue to be evaluated mainly on newspaper language. A project at Humboldt-Unversity Berlin and Ruhr-Universität Bochum [20] has been compiling a small corpus of texts from different domains of so called "non-standard varieties" like spoken, diachronic, second language learner, prosaic and chat data. Such data comprise a variety of linguistic structures and phenomena, which are not covered by current guidelines. Within this project, three types of annotations (dependency relations, named entities and coreference) have been annotated using WebAnno. This is possible as the GUI allows for the simultaneous annotations of nested spans (NER) and typed pointing relations inside sentences (dependencies) and between markables in distant sentences (coreference). Being a pilot annotation study, the tagsets and edge label sets have been iteratively adjusted, which is supported by the tagset editor. In a second project on historical German [3] (15th/16th century), the corpus was semi-automatically annotated with POS information, and the standard tagset was adopted for this purpose. The focus of this project is on verbal syntax (i.e. verbal complex phenomena, infinitival complement constructions, sentence frame). Finally, during WebAnno development, we conducted a Named Entity Recognition annotation project for German [2], to be able to get early feedback from annotators and curators.

1.3.3 Comparison, Discussion towards requirements

Having described two instances of open-source, multi-layer collaborative web-based annotation tools, we now contrast and discuss them, based on the requirements stipulated above. One or the other tool might better suit the needs of a project at hand, and better fit technical and/or organizational constraints.

A main difference to note is the comprehensiveness and maturity of GATE Teamware, including its connection to pre-annotation services in the GATE platform. WebAnno can import pre-annotated formats and offers a close-loop online machine learning for learning span annotations during annotation. While both tools allow configurable annotations, GATE Teamware is more targeted towards information extraction tasks, while WebAnno is especially suited for linguistic annotations, and applications in the Digital Humanities: when interested in non-standard language phenomena and when performing explorative annotation for singling out linguistically interesting examples, an annotation tool has to support the incremental adjustment of tagsets, and it has to provide high flexibility with respect to the length and the structure of documents, as well as annotation layers. Pre-annotation machine learning must cope with heterogeneity of languages.

Regarding extensibility and licensing, both tools are available as open-source projects, with permissive licenses for commercial, as well as academic use.

On the user interface side, WebAnno uses SVG technology to visualize the span and arc annotations while GATE Teamware uses background colors for highlighting different annotation types, and annotation templates for properties. During annotation, assigning tags for annotation is faster in GATE Teamware using the keyboard short-cuts, while the visualization of WebAnno is more intuitive for span-and-arc annotations.

Both WebAnno and GATE Teamware have very similar user roles and project workflows. Besides the four roles mentioned above, WebAnno supports an additional *REMOTE_USER* role where users can import and export data to WebAnno from external systems, as well as a special *CROWD_USER* to model annotations from crowdsourcing.

As an annotator or curator, there is zero installation effort in WebAnno. GATE Teamware requires that a Java web start bundle is downloaded in the browser, but its installation and running is seamless to the user. Installation is only required on the server side, unless a GATE Teamware server is launched via the GATECloud platform, where it comes ready to use.

While GATE Teamware handles a larger number of import formats than WebAnno, it supports only a single output format, stand-off XML, while WebAnno allows exporting to a range of formats. Both WebAnno and GATE Teamware support multi-layer annotation. In GATE Teamware the configuration of the annotation layers is specified by each project manager, as part of the workflow defining the specific annotation project. Similarly, WebAnno has a web-based annotation layer configuration support, which is configurable by project managers.

An interface to crowdsourcing as a means to scale out small annotation tasks to a large anonymous workforce is not currently available in GATE Teamware, although it is being developed as part of the uComp project[10]. WebAnno provides this functionality, however only for Named Entity annotations on an exemplary basis.

1.4 Conclusion and further directions

In this chapter, we have discussed the use of web-based tools for scaling and distributing collaborative annotation efforts amongst many users at different locations. After motivating the need of web-based tools for this purpose, and highlighting important characteristics and requirements towards such tools, we presented a comprehensive survey of the state-of-the-art existing annotation tools.

When comparing the tools along these requirements, we demonstrated that very few tools natively support all required and desired functionality. In particular, it is important to support multiple user roles, which perform different tasks during the workflow of an annotation project. This workflow should be modeled in the tool, and should be flexible enough to handle a large variety of project setups. Further, storage of the results should be scalable, and certain project settings demand data security. The possibility to be able to supply automatically pre-annotated data was identified as a very important means for increasing annotation speed.

For web-based tools, a multi-layer architecture consisting of at least one server and multiple web-based clients, seems the only reasonable architecture. We further highlighted aspects of modularity, and data persistence.

[10] http://www.ucomp.eu

The concepts and design principles have been exemplifie through an in-depth description of two web-based annotation platforms. While both tools adhere to best-practice design principles and fulfill the requirements to a large extent, they still differ in some aspects. GATE Teamware is built on top of the well-known and very mature GATE framework, which enables a tightly integrated automatic pre-annotation, and is targeted mostly towards Information Extraction tasks. WebAnno, on the other hand, supports more linguistically oriented annotation projects, is more lightweight, and offers an interface to crowdsourcing. In conclusion, based also on the comparison to other tools, there is no single best web-based annotation tool. Instead, the choice of tool depends on the nature of the annotation project at hand.

There are several directions for future work in this area. As web-based technology has already moved the location of the annotation tool away from the annotator's computer, virtualization and cloud-based solutions will alleviate the requirement for the project manager or administrator to take care of an installation on a web server, but rather use a service for that. This development has already started, as briefly described in Section 1.3.1. Along the same lines, infrastructures like CLARIN[11] provide automatic annotation services, which can be integrated seamlessly in annotation workflows.

Another direction is the further modularization of the architecture to enable more differentiation of user interfaces to support more diverse types of annotation layers. Regarding tool engineering, producing open-source components under permissive software licenses is imperative for ensuring interoperability and reusability.

Finally, to facilitate projects that are less rigidly defined, such as exploratory annotation for the Digital Humanities, the on-the-fly extension of tagsets and schemata, coupled with automatic annotation, is a promising direction with a high potential impact for automatic and semi-automatic processing of text and other modalities. GATE Teamware already supports managers with changing annotations and their properties from one project to the next, coupled with automatic pre-processing (either GATE Teamware-internal or external via the GATE platform). The next step would be to give exploratory projects further flexibility, to change schemas during the annotation process. In this case, there would need to be infrastructural support for identifying all annotations which no longer conform to the new schema definition, and thus need to be modified by the human annotators or curators.

References

[1] Bauer C, King G (2007) Java Persistence with Hibernate. Manning Publications Co, Bruce Park Avenue Typesetters, Greenwich, CT, USA

[2] Benikova D, Biemann C, Reznicek M (2014) NoSta-D Named Entity Annotation for German: Guidelines and Dataset. In: Calzolari N, Choukri K, De-

[11] http://www.clarin.eu/

clerck T, Loftsson H, Maegaard B, Mariani J, Moreno A, Odijk J, Piperidis S (eds) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Reykjavik, Iceland, pp 2524–2531

[3] Bollmann M, Dipper S, Krasselt J, Petran F (2012) Manual and Semi-automatic Normalization of Historical Spelling – Case Studies from Early New High German. In: Proceedings of the First International Workshop on Language Technology for Historical Text(s) (LThist2012), KONVENS, Vienna, Austria

[4] Bontcheva K, Cunningham H, Roberts I, Roberts A, Tablan V, Aswani N, Gorrell G (2013) GATE Teamware: a web-based, collaborative text annotation framework. Language Resources and Evaluation 47(4):1007–1029, DOI 10.1007/s10579-013-9215-6

[5] Brants T, Plaehn O (2000) Interactive corpus annotation. In: Calzolari N, Carayannis G, Choukri K, Höge H, Maegaard B, Mariani J, Zampolli A (eds) Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'00), European Language Resources Association (ELRA), Athens, Greece, pp 453–459

[6] Brugman H, Russel A (2004) Annotating Multi-media / Multi-modal resources with ELAN. In: Lino MT, Xavier MF, Ferreira F, Costa R, Silva R, Pereira C, Carvalho F, Lopes M, Catarino M, Barros S (eds) Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), European Language Resources Association (ELRA), Lisbon, Portugal, pp 2065–2068

[7] Brugman H, Crasborn O, Russel A (2004) Collaborative annotation of sign language data with peer-to-peer technology. In: Lino MT, Xavier MF, Ferreira F, Costa R, Silva R, Pereira C, Carvalho F, Lopes M, Catarino M, Barros S (eds) Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), European Language Resources Association (ELRA), Lisbon, Portugal

[8] Burchardt A, Erk K, Frank A, Kowalski A, Pado S (2006) SALTO: A versatile multi-level annotation tool. In: Calzolari N, Choukri K, Gangemi A, Maegaard B, Mariani J, Odijk J, Tapias D (eds) Proceedings of the 5th international conference on language resources and evaluation (LREC'06), European Language Resources Association (ELRA), Genoa, Italy, pp 517–520

[9] Carletta J (1996) Assessing agreement on classification tasks: the kappa statistic. Computational linguistics 22(2):249–254

[10] Carletta J, Evert S, Heid U, Kilgour J (2005) The NITE XML Toolkit: data model and query language. Language Resources and Evaluation 39(4):313–334, DOI 10.1007/s10579-006-9001-9

[11] Eckart de Castilho R, Gurevych I (2009) DKPro-UGD: A Flexible Data-Cleansing Approach to Processing User-Generated Discourse. In: Online-proceedings of the First French-speaking meeting around the framework Apache UIMA, LINA CNRS UMR 6241 - University of Nantes, France

[12] Chen WT, Styler W (2013) Anafora: A web-based general purpose annotation tool. In: Proceedings of the 2013 NAACL HLT Demonstration Session, Association for Computational Linguistics, Atlanta, Georgia, pp 14–19, URL http://www.aclweb.org/anthology/N13-3004

[13] Crammer K, Singer Y (2003) Ultraconservative online algorithms for multiclass problems. Journal of Machine Learning Research 3:951–991, DOI 10.1162/jmlr.2003.3.4-5.951

[14] Cunningham H, Maynard D, Bontcheva K, Tablan V (2002) GATE: an Architecture for Development of Robust HLT Applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02), Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp 168–175, DOI 10.3115/1073083.1073112

[15] Cunningham H, Tablan V, Roberts A, Bontcheva K (2013) Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. PLoS Computational Biology 9(2):e1002,854, DOI 10.1371/journal.pcbi.1002854

[16] Dashorst M, Hillenius E (2009) Wicket in Action. Manning Publications Co, Sound View Court 3B, Greenwich, CT, USA

[17] Day D, Aberdeen J, Hirschman L, Kozierok R, Robinson P, Vilain M (1997) Mixed-initiative development of language processing systems. In: Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLC '97), Association for Computational Linguistics, Washington, DC, pp 348–355, DOI 10.3115/974557.974608

[18] Day D, McHenry C, Kozierok R, Riek L (2004) Callisto: A configurable annotation workbench. In: Lino MT, Xavier MF, Ferreira F, Costa R, Silva R, Pereira C, Carvalho F, Lopes M, Catarino M, Barros S (eds) Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), European Language Resources Association (ELRA), Lisbon, Portugal, pp 2073–2076

[19] Dipper S, Götze M, Stede M (2004) Simple annotation tools for complex annotation tasks: an evaluation. In: Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora, Lisbon, Portugal, pp 54–62

[20] Dipper S, Lüdeling A, Reznicek M (2013) NoSta-D: A Corpus of German Non-Standard Varieties. In: Zampieri M, Diwersy S (eds) Non-standard Data Sources in Corpus-based Research, Shaker, pp 69–76

[21] Ferrucci D, Lally A (2004) UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering 10(3-4):327–348, DOI 10.1017/S1351324904003523

[22] Francis WN, Kucera H (1979) Brown corpus manual. Tech. rep., Department of Linguistics, Brown University, Providence, Rhode Island, USA, URL http://icame.uib.no/brown/bcm.html (Last accessed: 2015-02-11)

[23] Garrett JJ (2005) Ajax: A New Approach to Web Applications. http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/ (Last accessed: 2015-02-11)

[24] Gerdes K (2013) Arborator - A tool for collaborative dependency annotation. URL https://launchpad.net/arborator (Last accessed: 2015-02-08)

[25] Heid U, Schmid H, Eckart K, Hinrichs E (2010) A Corpus Representation Format for Linguistic Web Services:the D-SPIN Text Corpus Format and its Relationship with ISO Standards. In: Calzolari N, Choukri K, Maegaard B, Mariani J, Odijk J, Piperidis S, Rosner M, Tapias D (eds) Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10), European Language Resources Association (ELRA), Valletta, Malta, pp 494–499

[26] Hovy E (2010) Annotation. In: Tutorial Abstracts of ACL 2010, Association for Computational Linguistics, Uppsala, Sweden, p 4, URL http://www.aclweb.org/anthology/P10-5004

[27] Ide N, Romary L (2007) Towards International Standards for Language Resources. In: Dybkjær L, Hemsen H, Minker W (eds) Evaluation of Text and Speech Systems, vol 37, Springer Netherlands, chap 9, pp 263–284

[28] Ide N, Bonhomme P, Romary L (2000) XCES: An XML-based Encoding Standard for Linguistic Corpora Encoding Standard for Linguistic Corpora. In: Calzolari N, Carayannis G, Choukri K, Höge H, Maegaard B, Mariani J, Zampolli A (eds) Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'00), European Language Resources Association (ELRA), Athens, Greece, pp 825–830

[29] Kaplan D, Iida R, Nishina K, Tokunaga T (2011) Slate – a tool for creating and maintaining annotated corpora. Journal for Language Technology and Computational Linguistics 26(2):89–101

[30] Lin B, Chen Y, Chen X, Yu Y (2012) Comparison between JSON and XML in Applications Based on AJAX. In: Guerrero JE (ed) Proceedings of the International Conference on Computer Science & Service System (CSSS'12), IEEE Computer Society, Nanjing, China, pp 1174–1177, DOI 10.1109/CSSS.2012.297

[31] Maeda K, Lee H, Medero S, Medero J, Parker R, Strassel S (2008) Annotation Tool Development for Large-Scale Corpus Creation Projects at the Linguistic Data Consortium. In: Calzolari N, Choukri K, Maegaard B, Mariani J, Odijk J, Piperidis S, Tapias D (eds) Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08), European Language Resources Association (ELRA), Marrakech, Morocco, pp 3052–3056

[32] Meurs MJ, Murphy C, Naderi N, Morgenstern I, Cantu C, Semarjit S, Butler G, Powlowski J, Tsang A, Witte R (2011) Towards evaluating the impact of semantic support for curating the fungus scientific literature. In: Baker CJO, Chen H, Bagheri E, Du W (eds) Proceedings of the 3rd Canadian Semantic Web Symposium (CSWS'11), Vancouver, British Columbia, Canada, pp 34–39

[33] Morton T, LaCivita J (2003) WordFreak: An Open Tool for Linguistic Annotation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations - Volume 4 (NAACL-Demonstrations '03), Associa-

tion for Computational Linguistics, Stroudsburg, PA, USA, pp 17–18, DOI 10.3115/1073427.1073436

[34] Müller C, Strube M (2006) Multi-Level Annotation of Linguistic Data with MMAX2. In: Braun S, Kohn K, Mukherjee J (eds) Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods, Peter Lang, Frankfurt a.M., Germany, pp 197–214

[35] Nakov P, Schwartz A, Wolf B, Hearst M (2005) Supporting Annotation Layers for Natural Language Processing. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, Ann Arbor, Michigan, pp 65–68, DOI 10.3115/1225753.1225770

[36] Nivre J, Hall J, Kübler S, McDonald R, Nilsson J, Riedel S, Yuret D (2007) The CoNLL 2007 Shared Task on Dependency Parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, Association for Computational Linguistics, Prague, Czech Republic, pp 915–932

[37] Ogren PV (2006) Knowtator: A protégé plug-in for annotated corpus construction. In: Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume: Demonstrations, Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL-Demonstrations ’06, pp 273–275, DOI 10.3115/1225785.1225791, URL http://dx.doi.org/10.3115/1225785.1225791

[38] Pajas P, Štěpánek J (2008) Recent advances in a feature-rich framework for treebank annotation. In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING’08), Manchester, UK, pp 673–680, URL http://www.aclweb.org/anthology/C08-1085

[39] Rak R, Rowley A, Black W, Ananiadou S (2012) Argo: an integrative, interactive, text mining-based workbench supporting curation. Database 2012, DOI 10.1093/database/bas010

[40] Stenetorp P, Pyysalo S, Topić G, Ohta T, Ananiadou S, Tsujii J (2012) brat: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Avignon, France, pp 102–107, URL http://www.aclweb.org/anthology/E12-2021

[41] Stührenberg M, Goecke D, Diewald N, Mehler A, Cramer I (2007) Web-based annotation of anaphoric relations and lexical chains. In: Proceedings of the Linguistic Annotation Workshop (LAW’07), Association for Computational Linguistics, Prague, Czech Republic, pp 140–147

[42] Tablan V, Roberts I, Cunningham H, Bontcheva K (2012) GATECloud.net: a Platform for Large-Scale, Open-Source Text Processing on the Cloud. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 371(1983), DOI 10.1098/rsta.2012.0071

[43] Walls C (2011) Spring in Action, 3rd edn. Manning Publications Co, Sound View Court 3B, Greenwich, CT, USA

[44] Wang A, Hoang CDV, Kan MY (2013) Perspectives on Crowdsourcing Annotations for Natural Language Processing. Language Resources And Evaluation 47(1):9–31, DOI 10.1007/s10579-012-9176-1

[45] Yimam SM, Gurevych I, Eckart de Castilho R, Biemann C (2013) WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Sofia, Bulgaria, pp 1–6, URL http://www.aclweb.org/anthology/P13-4001

[46] Yimam SM, Biemann C, Eckart de Castilho R, Gurevych I (2014) Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Baltimore, Maryland, pp 91–96, URL http://aclweb.org/anthology/P14-5016