



Usability Criteria for External Requests of Collaboration (R 1.2.2)

Version 20.03.2015

Arbeitspaket AP 1.2 Usability and (G)UIs

Verantwortlicher Partner DAI

DARIAH-DE Aufbau von Forschungsinfrastrukturen für die e-Humanities

This research and development project is / was funded by the German Federal Ministry of Education and Research (BMBF), fund number 01UG1110A to M, and managed by the Project Management Agency of the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, PT-DLR).

Project: DARIAH-DE: Aufbau von Forschungsinfrastrukturen für die e-Humanities

BMBF Fund number: 01UG1110A to M

Duration: March 2014 - February 2016

Document status: Final

Dissemination level: DARIAH-DE-internal

Contact: dariah-de-cluster-1@gwdg.de

Authors:

Matteo Romanello, DAI

Juliane Stiller, MPIWG

Klaus Thoden, MPIWG

Revisions:

Date	Author	Comment
17.02.	Matteo Romanello	First complete draft
18.02.	Juliane Stiller	Feedback, editing
20.02.	Klaus Thoden	Feedback, editing
23.02.	Matteo Romanello	Minor revisions
27.02.	Matteo Romanello	Draft sent to the consortium
17.03.	Matteo Romanello	Revision based on the feedback received

Table of Contents

[1. Introduction](#)

[1.1 Goals](#)

[1.2 Use of these Usability Criteria for Evaluation](#)

[1.3 Overview of Usability Criteria](#)

[2. The User Groups targeted by DARIAH-DE](#)

[3. Usability Criteria](#)

[3.1 Documentation](#)

[3.2 Localisation and Internationalisation](#)

[3.3 Licencing](#)

[3.4 Platform](#)

[3.5 Stability](#)

[3.6 Interoperability and Data Exchange](#)

[3.7 Data Examples](#)

[4 Integration with the DARIAH-DE Infrastructure](#)

[4.1 User Authorisation and Authentication](#)

[4.2 Persistent Identifiers](#)

[4.3 Data Storage](#)

[4.4 Content Harvesting](#)

[Appendix: The DARIAH-DE Style Guide](#)

[HTML/CSS/JS Framework](#)

[Global Navigation](#)

[Language](#)

[User Profile](#)

[Terminology](#)

[Capitalisation](#)

[Documentation](#)

[Bug Reporting](#)

[User Feedback](#)

[Icons](#)

[Search](#)

[UI layout](#)

1. Introduction

1.1 Goals

The goal of this report is to provide a list of criteria for the evaluation of external tools that may be integrated into the DARIAH-DE infrastructure. These criteria aim to provide a means of establishing how desirable the integration of a certain tool into DARIAH-DE is and what the foreseeable obstacles or challenges that are likely to raise during the integration process are.

The criteria defined in this report are targeted on usability as they are conceived with the needs of the DARIAH-DE users in mind. At the same time, they address some high-level aspects of the DARIAH-DE technical infrastructure that are critical and/or very likely to require some adaptation of the existing tool (e.g. user authentication).

The *DARIAH-DE Style Guide* provided in the appendix addresses the challenge of integrating into a single ecosystem a variety of tools that are often developed independently from each other and serve a wide array of purposes. The challenge is to maintain a reasonably homogenous look and feel and recognisable basic interactions across tools in order to provide a consistent user experience and to reduce user disorientation. The main goals of the style guide are: a) to guide the adaptation of external tools to a design and style that characterises the DARIAH-DE ecosystem of tools; b) to ensure that the tools developed by DARIAH-DE are informed by the same set of guiding principles and design rules.

The report is structured as follows: in the next section (1.2) we outline how we envisage the use of the usability criteria specified in this report and how this fits into the life cycle of a DARIAH-DE tool. In section 2 we examine the usability requirements determined by the user groups that DARIAH-DE targets. Section 3 describes in detail each criterion, while section 4 addresses some high-level aspects of the DARIAH-DE infrastructure that may require some technical adaptations of the tool being evaluated. Finally, the appendix contains the above mentioned Style Guide, which provides a set of guidelines related to interface design and user interaction.

1.2 Use of these Usability Criteria for Evaluation

The main aim of these criteria is to evaluate, from a usability point of view, requests for collaboration that concern the integration of an external tool into the DARIAH-DE ecosystem.

The workflow developed by the Workgroup Service Life Cycle (fig. 1) can be used to describe the workflow for external requests of collaboration that concern the integration of a tool into DARIAH-DE. Two scenarios for the application of these

criteria can be foreseen as requests may come from external projects (passive scenario) but may also be proposed by DARIAH-DE members while being directed to external projects (active scenario).

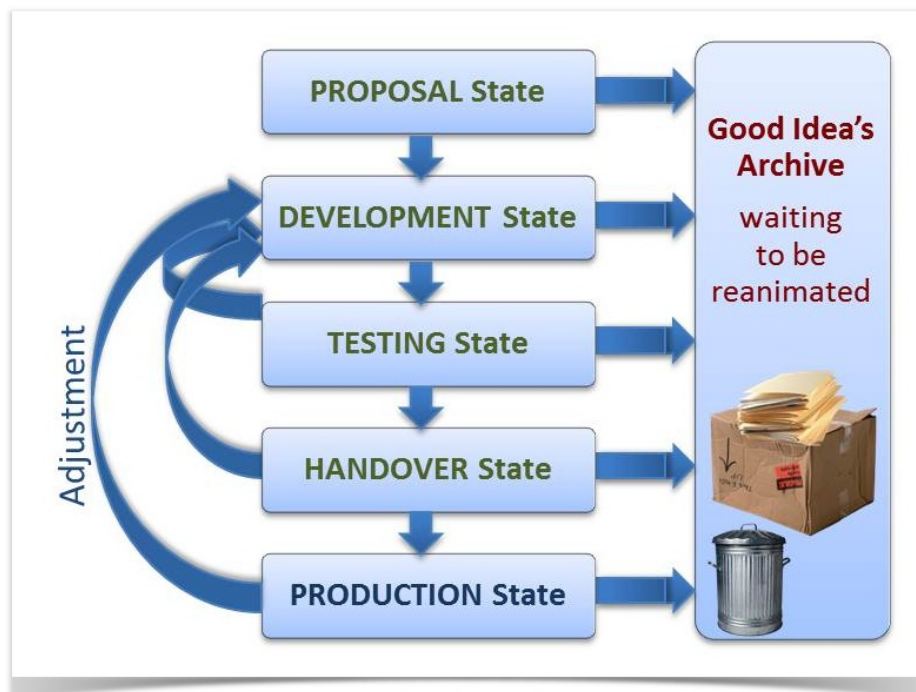


Fig. 1: Service Life Cycle.

The **proposal state** corresponds to when the request of collaboration is made or received by DARIAH-DE: if the request is received, these criteria guide the evaluation of the collaboration request; if the request is made, these criteria should help identifying external tools whose integration is desirable given the users that DARIAH-DE targets and technically feasible given the characteristics of the tool and the requirements of the infrastructure.

This report addresses key questions that should inform any decision made at this stage:

- Do we want this tool? Does it match the target user groups? (section 2)
- Does the tool satisfy entirely or partly the usability criteria? (section 3)
- How difficult is it to plug this tool into the overall technical infrastructure? (section 4)

Once accepted, the proposal enters the **developmental state**: in this phase the necessary adaptations to the tool are carried out both in terms of integration with the overall technical infrastructure (see section 4) and in terms of layout and user interface (see Appendix).

1.3 Overview of Usability Criteria

The table below gives an overview of the usability criteria grouped by category. For each criterion we indicate the section where is described in detail and the priority level.

We distinguish three levels of increasing priority, i.e. *optional*, *recommended* and *required*. These levels are aimed to guide the development of new tools as well as to be used for ranking purposes when comparing the evaluation of several tools.

Section	Category	Criterion	Priority
3.1	Documentation	User documentation	required
3.1	Documentation	Developer documentation	required
3.2	Language	Localisation: support for multiple locales (e.g. German, English, French, etc.)	optional
3.2	Language	An internationalisation framework is used	recommended
3.3	Licence	Source code (and any data required to run the software) released under an open source licence	required
3.4	Platform	Web-based tool with adaptive UI	recommended
3.4	Platform	Desktop tool with mobile app	recommended
3.4	Platform	Desktop tool with cross-platform support	recommended
3.5	Stability	The underlying code is stable: a stable version of the code was released (e.g. v1.0.0).	recommended
3.6	Data Exchange	Import/Export functionality	required
3.6	Data Exchange	Support for standard file formats	recommended
3.6	Data Exchange	Availability of an API	optional
3.7	Data Examples	The tool comes with data examples	recommended

2. The User Groups targeted by DARIAH-DE

Four target user groups were identified during the development of a marketing strategy for DARIAH-DE.¹ The identification of these groups was based on the following requirements for the success of research infrastructures such as DARIAH-DE and TextGrid:

1. institutional support;
2. an offer that matches requirements with regard to contents;
3. an offer that caters for users with a varying degree of prior experience with digital tools.

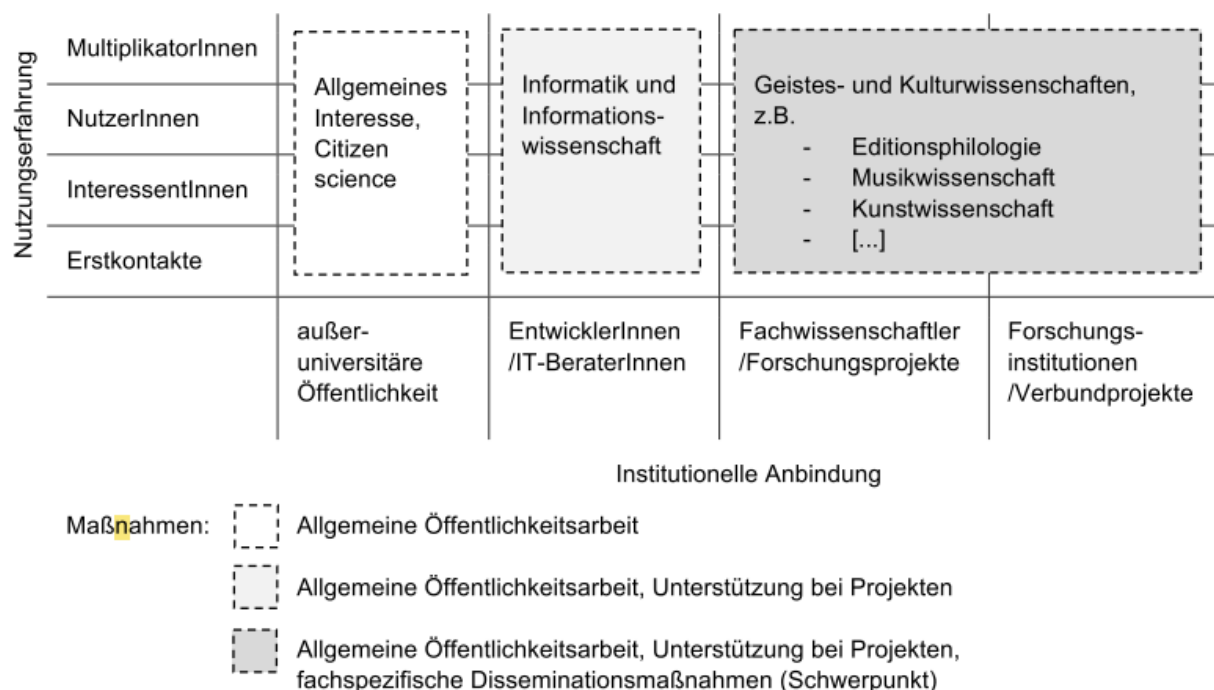


Fig. 2: User groups targeted by DARIAH-DE.

The users that DARIAH-DE targets are scholars from Arts and Humanities disciplines, those working in research projects (including joint collaborative research projects) and/or in research institutions (see fig. 2). They can be divided into the following four groups:

1. first contacts
2. interested persons/parties
3. users
4. evangelists

¹ See Report 7.2, DARIAH-DE / TextGrid Disseminationskonzept.

Each of these user groups have different needs and levels of expertise that need to be taken into account when developing a marketing strategy but also when identifying a set of usability criteria for tools. Making the tools offered by DARIAH-DE usable by and appealing to the different groups is a challenging task as requires to lower the entry point for using the tools while keeping them useful and interesting for more advanced users. Moreover, since learning how to use a new tool is a time-consuming task -- and one that some scholars cannot afford to undertake -- usability aspects such as the availability of documentation and data examples (or even search query examples) become important means to reaching out for interested yet not advanced users.



Fig. 3: The five phases of a prototypical research process.

Another aspect that informed the usability criteria outlined in this report is a division of the research process into five phases, i.e. (fig. 3). Such a model of a research process was used as the underlying structure for a recent survey aimed at gaining insights into the behaviours of users in the Digital Humanities (DH) (R 1.2.1 – *Report Nutzungsverhalten in den Digital Humanities*). Interestingly, a number of survey respondents did support such a model, although such concepts are well established in the DH literature and DH tools are often designed to support users in one or more specific phases of the process. On the contrary, from the survey emerged the tendency towards using one single piece of software throughout the entire research

process even when this means using a tool for purposes very different from those for which it was designed (i.e. “one software to rule them all”). In order to reconcile these contrasting tendencies, this report focuses on **documentation** and **import/export functionalities** as two essential usability criteria among others. The former should it make easier and less time-expensive to learn how to use a new tool, while the latter aims to encourage the use of task-oriented programmes by fostering a greater integration between the different programmes, in which import/export functionalities as well as Application Programming Interfaces (APIs) play a key role.

3. Usability Criteria

3.1 Documentation

The availability of documentation is a required criterion for a tool to be integrated into the DARIAH-DE infrastructure. Documentation is necessary in order to get new users started with the tool but also to carry out routine operations that are needed to guarantee the functioning of the tool in the long run. Ideally, documentation should be thorough, abundant and constantly updated.

Documentation includes:

1. *user documentation* targeting the users and aiming to elucidate the functionalities of a tool; this kind of documentation should address total beginners as well as users that had previous experiences with the tool;
2. *technical documentation* targeting especially developers or advanced users who need to make changes to the codebase or to reuse any of the tool components (e.g. API, libraries, etc.).

User documentation can be offered in a variety of forms such as written tutorials, interactive tutorials and screencasts.

Technical documentation can be offered in any of the above mentioned forms and should include also code documentation, i.e. documentation present within the code for example in the form of comments from which additional standalone documentation can be generated.²

3.2 Localisation and Internationalisation

Localisation is the process of adapting a tool to meet the language and other cultural requirements of a specific locale. Localisation is usually carried out by translators and entails, in addition to language adaptation, adjustments concerning the formatting of time, dates and currency, as well as the way text is displayed (e.g. directionality).

Internationalisation, on the other hand, is what enables localisation and concerns the mechanisms that a tool uses (or offer) to adapt its content and interface to another language.³ Decisions on internationalisation strategies should occur in the early phase of tool development. The use of an internationalisation framework is

² See the section 'documentation' of the *Style Guide* provided in the Appendix for some practical advices on how to provide access to the documentation accessible from within a User Interface (UI).

³ For the difference between the two see the W3C page on "Localization vs. Internationalization", <<http://www.w3.org/International/questions/qa-i18n>>.

recommendable also in cases where localisation is not yet required but is a foreseeable requirement.

Internationalisation is **required**, while **localisation** remains **optional** (i.e. nice to have). The importance of making such a distinction can be appreciated in cases where a given tool provides a robust internationalisation mechanism but may currently support just one language. The localisation of such a tool requires a smaller effort than it would require to change the code in order to support internationalisation.

Internationalisation and localisation are especially important given the transnational nature of DARIAH-EU, the bigger framework of which DARIAH-DE is part. Tools developed by other national DARIAH-DE initiatives may require to be translated into German and, likewise, tools developed in the framework of DARIAH-DE may be integrated into other national infrastructures. Making internationalisation a required criterion should help towards both scenarios.⁴

3.3 Licensing

A **required** criterion is that the **code** of the tool -- as well as any data required to run it -- is **released under an open source licence**, which allows software to be freely used modified and shared.⁵

The open source licence of choice for all code developed within DARIAH-DE is the Apache License v. 2.0⁶. Other licences are accepted as well insofar as they comply with the definition of open source (e.g. GNU General Public Licence, MIT Licence, etc.).⁷

Having direct access to the source code of a tool is required in order to:

- *modify* the code, which may be required for the medium- and long-term maintenance of the tool (e.g. bug fixing, update of single components, etc.);
- *adapt* the tool so as it becomes possible to integrate it into the DARIAH-DE ecosystem (e.g. localisation);
- *extend* the functionalities of the tool in order to meet the needs and requirements of the user groups that DARIAH-DE targets

⁴ See the section 'language' of the *Style Guide* provided in the Appendix for some guidelines on the display of internationalised interfaces.

⁵ Open Source Initiative (OSI), *Open Source Definition* <<http://opensource.org/osd>>.

⁶ <http://www.apache.org/licenses/LICENSE-2.0.html>

⁷ For some recommendations on licencing research data see Nikolaos Beer et al. (2014), "Datenlizenzen für geisteswissenschaftliche Forschungsdaten - Rechtliche Bedingungen und Handlungsbedarf", *DARIAH-DE Working Papers* Nr. 6, Göttingen, <<http://nbn-resolving.de/urn:nbn:de:gbv:7-dariah-2014-4-8>>.

Moreover, the use of an open source licence will ensure that further improvements or developments of the tool, which will result from its integration into DARIAH-DE, can be fed back to the broader community.

3.4 Platform

Although the majority of tools that are currently part of DARIAH-DE are web-based, this is not made a strict requirement for the integration of external tools into the DARIAH-DE ecosystem. As a result, **suitable tools may be either web or desktop applications.**

Similarly, support for handheld devices is desirable (and strongly encouraged) but remains optional. In fact, some tools may be based on patterns of user interaction that are hard to transfer onto handheld devices.

However, we recommend:

- **web tool + adaptive user interface:** the interface of web-based tools should be adaptive -- i.e. adapts well to the screen of handheld devices -- or, alternatively, follow a principle of graceful degradation when viewed on smaller devices (i.e. some features are disabled without compromising the overall functioning of the application);
- **desktop tool + mobile app:** desktop-based tools should provide also support for mobile applications so as to allow for synchronisation between desktop and mobile version;
- **[web | desktop] tool + front/back-end separation:** if both support for mobile applications and an adaptive interface are not provided by the tool, a good separation between front and back-end is recommended, as this facilitates the development of alternative interfaces for a given tool or even a mobile application.

3.5 Stability

It is possible to distinguish between two meanings of the word *stability* in relation to a piece of software.

1. **Stability as reliability:** from a user's perspective a software is stable when it does not crash; in this sense, the use of the word **reliability** is preferable. The results of the survey conducted in the context of R1.2.1 show that users tend to use tools that they find stable and tend to complain whenever the software appears to be unstable. User testing is essential to assess the reliability of a tool.

2. **Code or Software Stability:** from a developer's perspective a software is stable when its public API or functionalities remain stable (i.e. compatible) until the release of a new major version (e.g. from 1.8.1 to 2.0.0). When the versions of a software are named following the so-called *semantic versioning specification*⁸ pre-release versions (e.g. 1.x.y-alpha) or major version zero (e.g. 0.x.y) are not to be considered stable.

However, the reliability and stability of a tool should be considered also in light of the **activity of the developer community** responsible for its development. The fact that a given tool is not yet fully reliable (i.e. still crashes sometimes) has a different weight depending on whether this tool is being actively developed and maintained or not.

3.6 Interoperability and Data Exchange

One essential criterion for the integration of a tool into the DARIAH-DE ecosystem is its ability to facilitate data exchange. Mechanisms for data **import** and **export** are of essential importance if we are to build an ecosystem of distributed and specialised tools that, nevertheless, can be integrated seamlessly to support the research process. Therefore, the availability of mechanisms to import and export data is a required criterion.

The tendency emerged from the R1.2.1 survey consisting of using one single tool to manage all the different phases of the research process -- i.e. -- may be seen as the attempt of users to overcome the limited ability of current tools of "talking to each other". Phases where data exchange is especially crucial were deemed to be *Research* as well as *Consolidation of Research Question*. Researchers, for example, want to be able to import easily search results into their reference management software of choice. This software should also provide the ability to annotate documents and ideally provide support for doing so while using handheld devices. When it comes to publishing their research, users need to cite these documents from within their word processing tool of choice. In order to support such an articulated and distributed workflow two main aspects of data exchange come into play:

1. the support for various standards and file formats (optional);
2. the availability of machine interfaces (i.e. APIs) to exchange data (optional).

However, we recommend as practice to provide support at least for one standard format as well for one relatively flat file format that makes it easier to transform data into other formats.⁹ For example, the DARIAH-DE GeoBrowser supports KML as

⁸ Semantic Versioning 2.0.0, <<http://semver.org/>>.

⁹ For recommendations concerning the choice of standard formats to support see Nikolaos Beer *et al.*: "Interdisciplinary Interoperability", *DARIAH-DE Working Papers* Nr. 3. Göttingen: DARIAH-DE, 2014. <<http://nbn-resolving.de/urn:nbn:de:gbv:7-dariah-2014-1-0>>.

standard for import/export and, additionally, allows users to import and export data also as CSV.

3.7 Data Examples

Providing some data examples that can be used (i.e. visualised, analysed, etc.) with a given tool is recommended as it makes easier for first time users (i.e. first contacts and interested persons) to test the tool and have an idea of what they can achieve with it. This criterion applies to those tools that require some data in order to function such as editors, visualisation tools, etc. The DARIAH-DE GeoBrowser¹⁰, for example, allows users to upload their own data or to choose one dataset among a number of pre-loaded examples (see fig. 4).

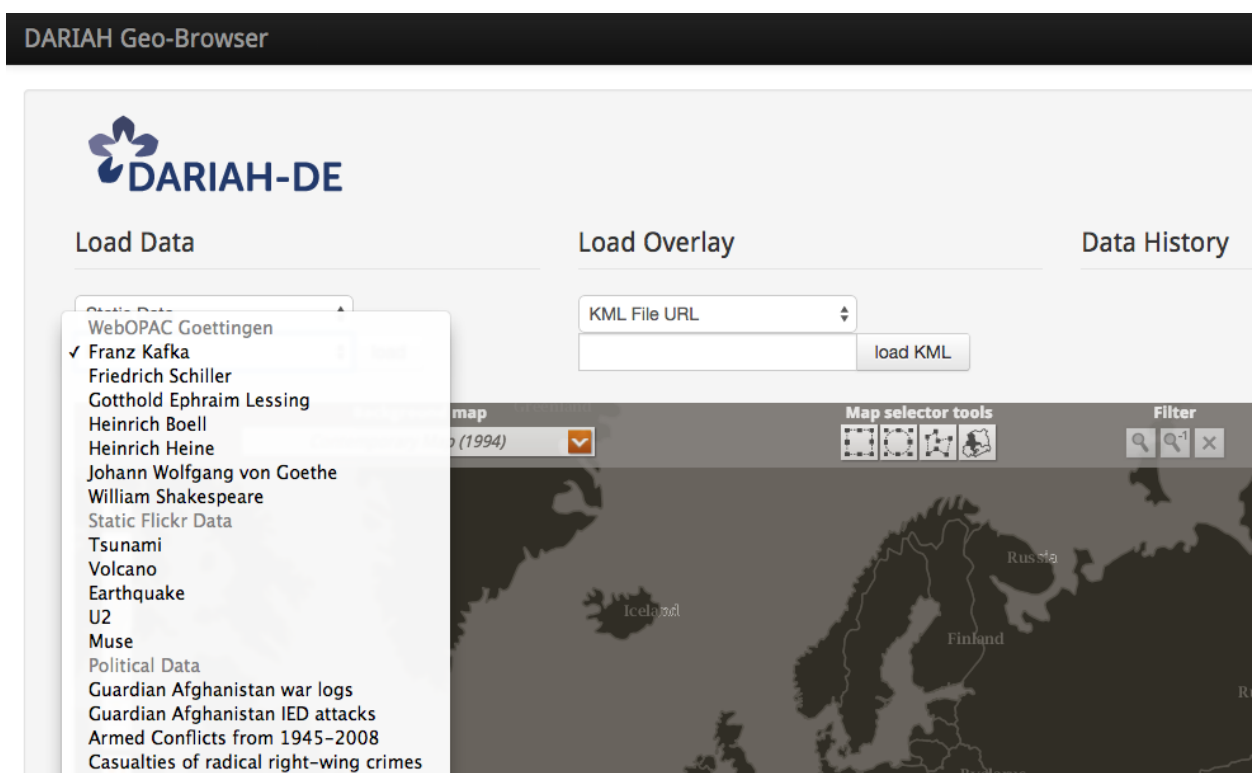


Fig. 4: Data examples offered by the DARIAH-DE GeoBrowser.

¹⁰ <http://geobrowser.de.dariah.eu/>

4 Integration with the DARIAH-DE Infrastructure

In addition to the criteria described in the previous section, the assessment of the effort needed to integrate a given tool into the DARIAH-DE infrastructure should take into account some adaptations to the tool that are required in order to interact with the technical infrastructure. Whether these modifications will be needed or not depends on the implementation (e.g. programming language and libraries used) as well as on the functionalities of the tool.

This integration process was tested and demonstrated by the development of Archive in a Box (AiB)¹¹, a repository for digital objects based on the open source software DSpace¹². The motivation behind developing AiB was to provide a data repository software that integrates seamlessly with the technical infrastructure of DARIAH-DE, specifically with regards to the following aspects:

- user authorisation and authentication
- creation and maintenance of persistent identifiers (PIDs)
- long term data storage (also known as bit-stream preservation)
- content harvesting

4.1 User Authorisation and Authentication

If a tool needs to authenticate a user, for example in order to decide if the user has enough rights to access a given resource, this tool needs to support DARIAH-DE Authorisation and Authentication Infrastructure (AAI)¹³.

AAI is based on two key standards -- i.e. LDAP and SAML -- and uses the implementation of the SAML standard provided by the open source software Shibboleth¹⁴. In order to reduce the effort needed to interact with the AAI layer, a Java Shibboleth client was developed as extension of the Apache HttpClient¹⁵. This client can be integrated into the back-end of an existing tool and makes it possible to let DARIAH-DE users log in with their institutional accounts.

4.2 Persistent Identifiers

Integration with the PID system is necessary if the tool is to rely on DARIAH-DE's PID service¹⁶ for the creation and maintenance of persistent identifiers. Tools that may want to do so are in general those that allow users to create new content (e.g.

¹¹ <https://de.dariah.eu/archive-in-a-box>

¹² <http://www.dspace.org/>

¹³ <https://de.dariah.eu/aai>

¹⁴ <https://shibboleth.net/>

¹⁵ <https://github.com/reckart/shib-http-client>

¹⁶ <https://de.dariah.eu/pid-service>

editors, writing platforms, etc.) but also tools that allow users to create persistent pointers to online resources (e.g. bookmarking applications).

The PID service uses an identifier system called Handle -- the same system on which Digital Object Identifiers (DOIs) are based on -- and provides a HTTP API to create, update, delete persistent identifiers¹⁷. The availability of such an API, which is independent from any specific programming language, makes it possible to integrate this service into any software, no matter the programming language of choice.

4.3 Data Storage

It is possible for a tool to rely on the technical infrastructure of DARIAH-DE for the storage and long term preservation of data. To this end DARIAH-DE provides the HTTP Bit Preservation API, which allows client applications to create, read, update and delete files without having to worry about how or where such files are stored by the back-end. It is possible to interact with this API by using the DARIAH-DE Storage Client¹⁸, a client library written in Java.

An example of the use of this layer of the infrastructure is offered by the Datasheet Editor, a tool that allows users to prepare geographical data for their display in the GeoBrowser. Users can import data from a CSV file or input them from scratch and use this tool to enrich the data with geographical coordinates. In the background, the tool uses the Storage Client to store the data persistently.

4.4 Content Harvesting

It may be desirable for a tool to expose its content for harvesting, i.e. to publish it in a way that makes it possible for other applications to access it programmatically. The standard supported by DARIAH-DE to achieve this is the Open Access Initiative Protocol for Metadata Harvesting (OAI-PMH). This protocol may be used, for example, by a publication platform to disseminate its content for access and indexing purposes. OAI-PMH can also be used to import content into a tool from a repository, provided that an OAI-PMH interface exists. In a DARIAH-DE context, this protocol is used to harvest contents that can then be searched by using the DARIAH-DE Generic Search¹⁹.

¹⁷ EPIC API version 2, <http://doc.pidconsortium.eu/guides/api-create/>

¹⁸ DARIAH-DE Storage Client,

<http://dev.dariah.eu/nexus/content/groups/public/de/tudarmstadt/ukp/dariah/de.tudarmstadt.ukp.dariah.storage.client/>

¹⁹ <https://search.de.dariah.eu/search/>

Appendix: The DARIAH-DE Style Guide

The *DARIAH-DE Style Guide* described in this section provides some guidance for the design or adaptation of the UI of tools that are part of the DARIAH-DE ecosystem.

We envisage the following scenarios for the use of this document:

1. to review tools that have been already developed in order to achieve further integration and a more consistent layout;
2. to adapt the design of already implemented tools that are to be integrated into the DARIAH-DE ecosystem;
3. to guide the development of new tools in DARIAH-DE.

The User Interfaces of the following tools were analysed during the development of this style guide:

- DARIAH-DE Portal
- GeoBrowser
- Datasheet Editor
- Generic Search
- Schema Registry
- Collection Registry

HTML/CSS/JS Framework

We recommend the use of the **Bootstrap** framework²⁰ for the implementation of the User Interface (UI).

Bootstrap, which was tested during the implementation of a number of DARIAH-DE tools, proved to be suitable to ensure a certain degree of consistency and homogeneous appearance across tools. In fact, Bootstrap already implements a number of UI elements that are normally found in web applications such as headers, menus, toolbars, dropdowns, buttons, etc. Moreover, it provides a responsive layout that scales well also on handheld devices (i.e. phones, tables, etc).

Global Navigation

The global navigation provides the user with information about the tool currently being used while putting it into context by means of links to the DARIAH-DE homepage and to other tools that are part of the ecosystem. A consistent global navigation is important to prevent user disorientation, i.e. to let the user move through the tools of the ecosystem without losing sight of the big picture.

²⁰ <http://getbootstrap.com/>

The global navigation is to be placed in the top left corner.

A comparison of how the global navigation has been implemented in the different tools developed by DARIAH-DE is provided in fig. 5. The recommended layout is shown in fig. 6.

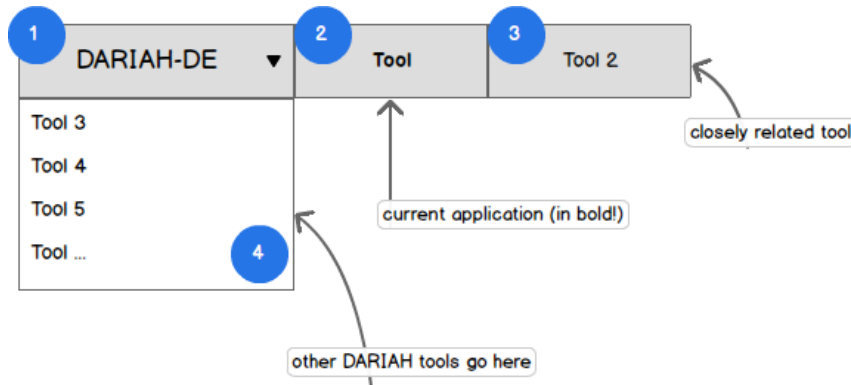


Fig. 5 Global navigation: mockup of recommended layout.

The global navigation consists of the following elements:

- **Home button:** links to <<https://de.dariah.eu/>> and opens in a new page (1)
- **Name of the tool:** use a bold typeface to indicate that this is the tool currently being used (2)
- **Name of closely related tools:** two or more tools are *closely related* when they are often used in combination with one another or when the user is likely to switch often between them (e.g. GeoBrowser and Datasheet Editor or Generic Search, Schema Registry and Collection Registry). The link to the tool opens in a new page. (3)
- **Drop-down menu:** lists and links to other tools that are part of the DARIAH-DE ecosystem but are not closely related to the tool in focus. (4)



Fig. 6 Global navigation: comparison across DARIAH-DE tools.

Language

If a tool is localised into more than one language, a dropdown menu is provided whereby the user can change the language (fig. 7).

The drop down menu keeps this control compact and the use of flag icons²¹ make it also visually intuitive. The language selection menu is to be found in the top right corner of the UI.

The behaviour expected by the user upon selection of a given language is that the entire application is translated into that language, i.e. the content as well as the text in menu, buttons, etc. The user should be notified if no translation in the target language is available for a certain content.

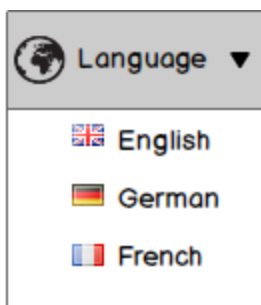


Fig. 7 Language selection: mockup of recommended layout.

User Profile

The login button is to be found in the top right corner of the UI. If the user is already logged in, display the username. A drop down menu groups together actions that are centered around the user, such as logging out, viewing the user profile, changing the tool preferences, etc.

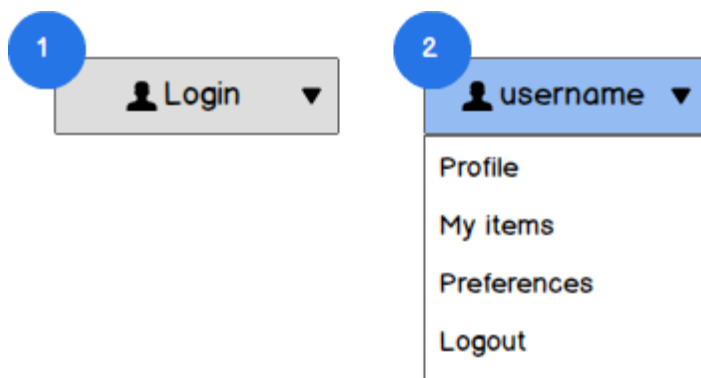


Fig. 8 User area: mockup of recommended layout. Two modes are represented: 1) user not logged in and 2) user is already logged in.

²¹ <http://www.famfamfam.com/lab/icons/flags/>

Terminology

For a tool to be usable it is highly important that the terminology used in all UI elements that contain some text (i.e. headings, tooltips, buttons, etc.) is descriptive, concise and can be easily understood by all users. Therefore, avoid using jargon or technical terms as far as possible; in case where this is inevitable, make good use of tooltips to clarify the meaning to non-expert users.

Capitalisation

Consistent capitalisation of text leads to a more polished feel, and greater perception of quality. Headline capitalization should be applied to all titles, menus, tooltips, tabs, and buttons (e.g. “Register Schema”). Sentence style capitalisation should be applied to all check-boxes, radio buttons, and group labels (e.g. “Please select a schema in the list on the left!”).

Documentation

We recommend to keep the documentation as far as possible within the tool itself rather than in separate documents.

If additional documentation needs to be provided in a separate document, make sure that the link is found in each page of the tool, always in the same position (e.g. in the footer).

The use of tooltips is recommended to provide the user with some concise documentation about the action performed upon click of a certain button. Bootstrap implements tooltips and gives control over their behaviour: they can be triggered by a certain event, they can be destroyed -- this can be useful when they are used to provide an initial interactive walkthrough of the tool -- and it is possible to specify where they should be displayed (i.e. their position in relation to the element to which they are attached).²²

Bug Reporting

Enable users to report bugs and unexpected errors. This can be done in a variety of ways:

- by providing ways to get directly in touch:
 - a dedicated email address (or email form);
 - handles for social media profiles, where applicable (e.g. Twitter);
- by linking to the issue tracker for the tool where the bug or error report can be logged (e.g. the issue tracker connected to code repositories on GitHub);

²² See <http://getbootstrap.com/javascript/#tooltips> for more documentation.

- by requesting user input when a certain error is raised;

User Feedback

Give feedback to the user about the estimated duration of long operations (e.g. export generation) as well as about the current position of the user within a process made of several steps (e.g. survey completion). To provide such kind of feedback make use of progress bars²³ or “loading” icons depending whether the duration can be estimated or not.

Icons

When using icons for command buttons, choose symbols that reflect well the underlying action and avoid those that are commonly associated with other specific actions. For example, the trashbin icon is strongly associated with the *delete* action, thus should not be used with a different meaning.

Bootstrap comes with a library of 260 icons that can be easily attached to UI elements by means of CSS classes.²⁴ Moreover, it provides ways to use such icons while keeping the interface accessible to users of assistive technologies such as screen readers.

Search

(This section applies to tools offering a full-text search functionality.)

The search function is suitable for tools that contain or make accessible contents (e.g. a text repository) but may be less suitable for example for tools like content editors.

If a search function is provided, offer the user also an alternative way of exploring contents such as (faceted) browsing or visualising the content items displayed on a map.

Be clear with the user about what is being searched and what is the search result. Provide information about the number and type of objects that can be searched for (see fig. 9).

²³ <http://getbootstrap.com/components/#progress>

²⁴ See <http://getbootstrap.com/components/#glyphicons> for implementation notes and examples.



Fig. 9 Detail of the search box in DARIAH-DE's Generic Search.

New users -- and especially non expert users -- may not know what to search for. To make the search function of a tool usable by a wider audience we encourage developers to include and make visible some query examples. This applies to basic as well as advanced search functionalities.

In addition to search examples, autocomplete functionalities allow users to get insights about what can be searched for.²⁵

²⁵ Bootstrap's plugin Typeahead provides an easy to use implementation of autocomplete, see <http://getbootstrap.com/2.3.2/javascript.html#typeahead>.

UI layout

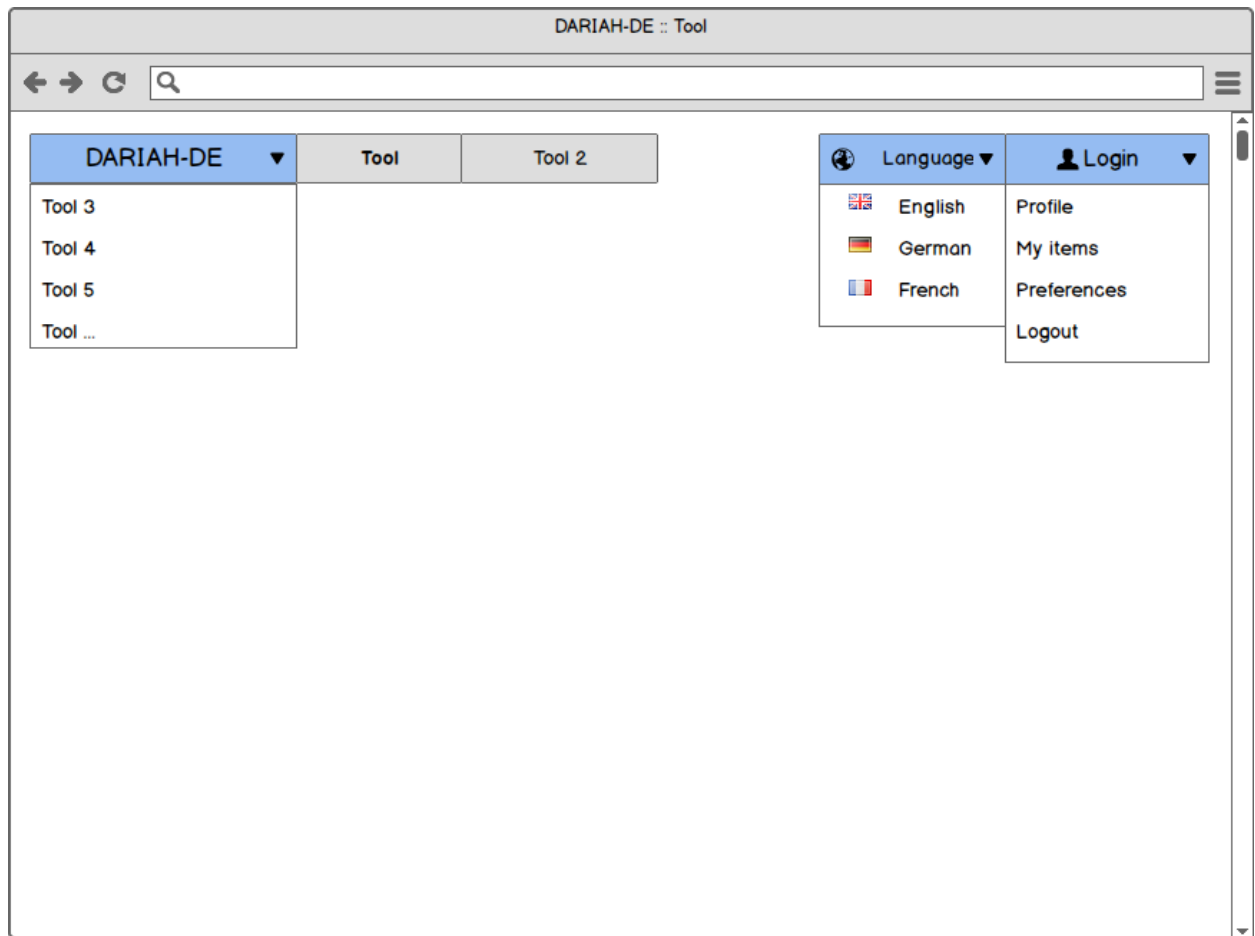


Fig. 10 Overall UI layout with global navigation, language selection and user area.