Digital Mappa – Simple and Web-based Annotations

Digital Mappa – Simple and Web-based Annotations, (unclear, not properly defined) University of Wisconsin - Madison's Center for the History of Print and Digital Culture (ed.), 14.07.2022. https://www.digitalmappa.org/ (Last Accessed: 14.11.2022). Reviewed by [iD] Tobias Hodel (University of Bern), tobias.hodel@unibe.ch, [iD] Anna Janka (University of Bern), anna.janka@unibe.ch, [iD] Jonas Widmer (University of Bern), jonas.widmer@unibe.ch.

Abstract

How does Digital Mappa, a browser-based annotation tool, fit into the current landscape of Virtual Research Environments? How is an intuitive WYSIWYG graphical user interface suited for simple (scholarly) editions and image-text representations? What about involving different users via crowdsourcing? For this review, we took three distinct perspectives (user, teacher, and backend admin) and tested the tool for text annotation but furthermore for linking images with text. Digital Mappa proves to be a stable, not-so-easy-to-set-up tool that covers a lot of ground concerning annotation and image linking. Subsuming, we describe a mature system that allows for easy editing, suitable in the classroom. Critically, it has been assessed that the tool refrains from standards (such as TEI XML) and lacks capabilities to involve larger groups (e.g. for crowdsourcing).

Opening the review

The reviewer

1   This review aims to look from a multitude of perspectives at Digital Mappa 2.0 (henceforth DM). For this reason, we assembled a group of people that wanted to work with the tool as annotator (Anna Janka), scholar and teacher (Tobias Hodel), as well as system administrator and developer (Jonas Widmer).

Identification of the reviewed tool

2   Digital Mappa 2.0 is an open-source DH platform for open-access workspaces, projects, collaborations, and scholarly publications/digital edition platforms, focusing on linking of images and text annotations. DM is well documented, and the descriptions on its [website](#) are exhaustive, it is hosted on [GitHub](#).

Stage of development

3   The first version of Digital Mappa was released in 2018 after more than a decade of development and early releases. In 2020 version 2.0 with an entirely refactored platform was released. Currently, bugs are being fixed.

4   Due to the very recent release: This review did not take the latest features of version 2.1 (especially layering of annotations) into account!

Identification of the reviewing platform

5   We tried to install DM on a virtual server of our own (based on Ubuntu 20.04 LTS) but failed to do so due to miscommunication between the front- and the backend. Furthermore, the proposed installation involving the use of AWS as a storage hub did not lead to a working system. In the end and for this evaluation, we relied on an already running installation of the University of Zurich, hosted by "[Digitale Lehre und Forschung](#)". The front end of DM was accessed through state-of-the-art browsers such as Chrome (version 94.0.4606.71) and Firefox (version 92.0.1).

General introduction

6   Digital Mappa has been developed in order to allow for the annotation of (historical) maps. Due to the simple graphical user interface, developers as well as

involved scholars quickly realized that DM could not only be used for this task but, in general, to create annotated data on any image files and even texts. This led to the development of DM with the idea of providing an exhaustive digital scholarly edition environment (in the sense of an IDE: integrated development environment). The goal of the tool remains to offer easy-to-use annotation capabilities within web browsers.

7 The tool is thus not only suited for very specific projects (focusing on maps) but addresses, in general, the needs of the edition community, especially if the annotation of facsimile is involved. As an end-user, any scholar and student independent of pre-existing knowledge and technical expertise are envisioned.

Credit and funding

8 The tool has been created and is maintained by Martin Foys (University of Wisconsin-Madison) as project director in collaboration with the Center for the History of Print and Digital Culture (also at the University of Wisconsin-Madison) as well as the Schoenberg Institute for Manuscript Studies (University of Pennsylvania).

9 Most of the refactoring for version 2.0, which is the focus of this review, seems to be carried out by Performant Software Solutions, a digital humanities-close software development company. The tool is available open-source and hosted on GitHub. Digital Mappa and its history and development are comprehensible and well documented.

Transparency

10 Information about the installation is available as readme on GitHub. Further information about the project is available on its website. Digital Mappa is licensed under a GPL 3.0-License.

Methods and Implementation

State of the art

11 Currently and very broadly speaking, there exist two ways to create scholarly editions: On the one hand, digital editions based on TEI XML are often carried out in interdisciplinary teams creating back- and front-end parallel to the mark-up of text. On the other hand, there are also text editor-based (e. g., MS Word or LaTeX) editions in

use that mimic the typesetting of print and are – after preparation of the text – exported to professional layout and printing companies.

12   DM has to be located in-between these two approaches, as it allows to annotate images and textual entities. For annotation and linking, no knowledge about mark-up languages is necessary. At the same time, advantages of mark-up languages (ontologies in the broadest sense) are not provided but need to be defined on a project basis. From another perspective, DM offers a WYSIWYG (what you see is what you get) approach, and all information can be accessed online right after entry.

13   DM is a self-sufficient platform that was not built to export annotated documents. Its data model does not rely on TEI XML and DM is therefore not part of the TEI community. Very simple mark-up for text (mostly focusing on formatting, like bold/ italicized) can be added, but the focus lies on marking images and connecting them to textual structures. Of course, from a technical point of view, it would be possible to export annotations, but this would require several transformation processes that needed to be prepared by future users.

14   Very loosely, we can identify relations to manual text annotation tools (like CATMA or Annotation Studio) as well as image annotation tools (e. g. within Mirador). The map annotation capability is comparable to Recogito but doesn't offer the linking to norm data. There is no automatization (e. g. recognition of named entities, identification of visual structures) currently included in Digital Mappa. The combination of text and image annotation is still a rather unique mix and, thus, a unique selling point of DM.

Discipline and methodology

15   The tool facilitates the annotation and interlinking of images and textual entries. This again allows marking certain aspects, for example, on a historical, digitized map. Already during the development of DM 1.0, it has been realized that with the toolset also editions can be obtained that are tightly intertwined and create a close connection between a documentary edition as well as annotated images. Although such a simple, visual edition can be created through the offered toolset, certain design decisions (shortcomings for certain approaches) need to be taken into account: In order to add an apparatus, it would be necessary to either use manually added footnotes (not semantically annotated) or link text entries (which is possible but doesn't really add to the usability of an edition).
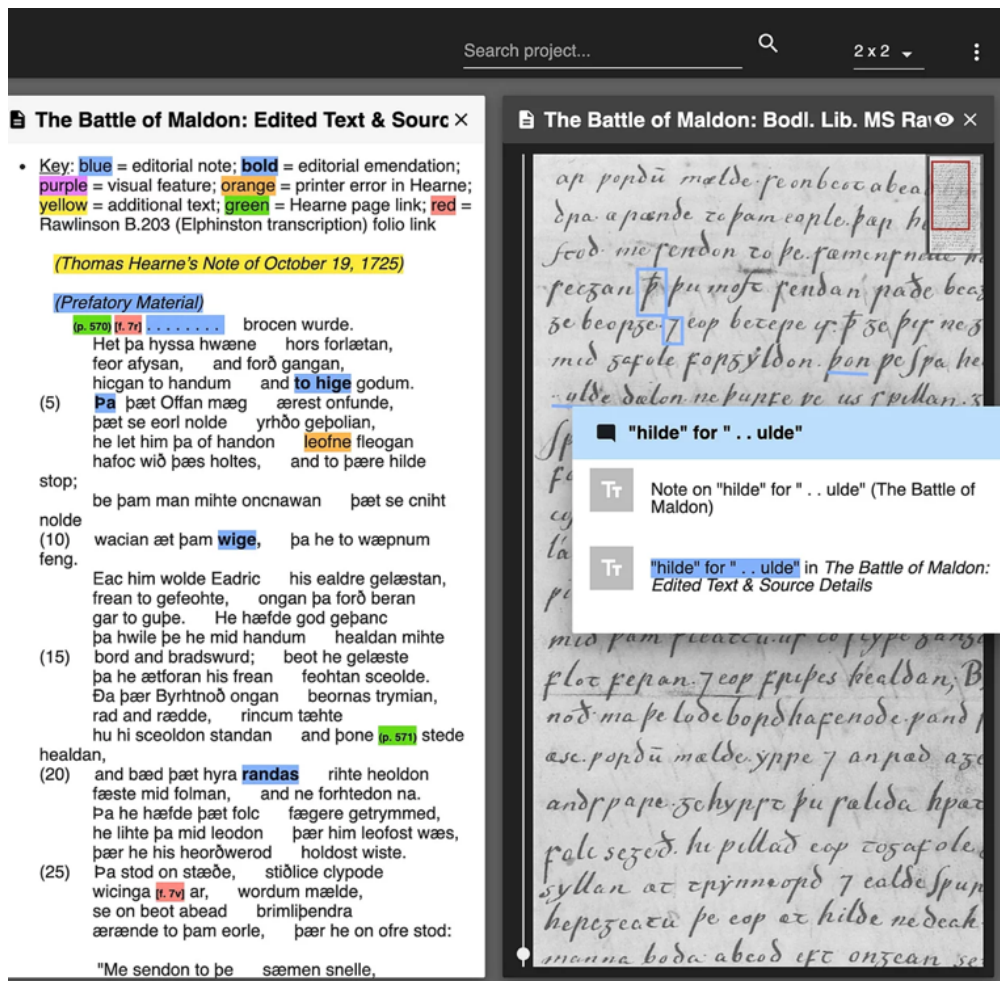
How it works

Fig. 1: Screenshot of image annotation (right frame), text entity (left frame), and connecting annotation (pop-up right frame), provided on the DM web-page.

16   As already stated, DM focuses on annotation and interlinking images with textual entities, which can be done by marking certain image areas and linking them to text boxes (fig. 1). A check-out functionality stops the parallel processing of images or textual entities on the level of one document or one image. The functionality ensures that no entries get lost and no parallel editing occurs. The check-out modus also limits the possibility to use DM as a crowdsourcing application, since certain entries could be locked from adding text/annotations, which could be demotivating.

17   From a technical point-of-view, DM uses Ruby-on-Rails and embeds it in a Heroku-App. Since a browser-based web solution was envisioned from the start of the project, the choice to release DM as a Heroku-App seems sensible and supports (thanks to the environment) a relatively quick setup. Right after finishing this review, further

updates of DM were made which are not covered in this review; amongst other things, a docker-compose/docker-images was released to the GitHub repository.

Input, output, and data modeling

18   Input is very limited, and images can be imported as JPEG. IIIF-compliant images can be imported directly as well. Text is typically imported through copy and paste. Output in other formats, other than in the form of the web-page, is not foreseen. Therefore, the data model of DM cannot be compared to scholarly digital editions based on TEI XML or Linked Data knowledge graphs. Unicode (UTF-8) is supported, which makes the DM language independently usable and interesting for projects involving non-western character sets.

Performance

19   Once installed, the tool performs even on 'slim' virtual machines (a 2 CPU, 4 GB setup was used) decently and without any problems regarding loading or processing time. Access to rather large JPEG files (~15 MB) does not pose any problems. Technically, it would be possible to embed images stored on other servers and made available via IIIF manifests. Access times to such (larger) files have not been tested.

20   With regards to accuracy, no negative experiences were observed or have been reported by instructors or students. The website reacts as intended, clicks get recorded, and polygons (or other shapes) get drawn according to mouse movements.

Logs and trackability

21   Logs are being generated for the installation process. Besides that, we couldn't identify any logs.

Application and results

22   DM features a few showcase projects on their web-page. Besides that, we did not identify further publicly available use cases. The showcases already demonstrate a wide variety of possible scenarios to use DM. From "Virtual Mappa" (the classical use-case of DM), to a facsimile annotation of old English poetry and furthermore a library of stains that have been analysed using spectral analysis. In respect of the last example, DM was used as closely interlinked documentation with visual as well as textual components.

23   Overall, it becomes evident that with some creativity, DM should not only be used in the intended context (annotation of images, esp. maps), as more pathways need to be explored. For art history classes (as an invented example), DM could be used to produce annotated image corpora and thus replace traditional seminar papers.

Example

24   Using DM, our team of four people partially marked and annotated a historical map of the Bernese territorial dominion in the 16th century. In order to work with the map, a user could check out a single document file on DM (e. g. the part of the map or a textual annotation), which was then blocked for the other team members. Our map consisted of nine pages, and therefore nine single documents on DM, and because of that, our team could still work collaboratively.

25   Our goal was to tag different geographical elements, like mountains, lakes, rivers, or woodland, as well as cities and villages under Bernese reign at the time. The tools on DM function very intuitively. Users can choose between eight colors, draw lines with the drawing tool, or select pre-made shapes (rectangle or circle) to drag and drop on the map. A difficulty our team encountered was highlighting longer entities (like rivers) because they were separated into different parts (e. g. interrupted by bridges). DM then generated different tags for every part of the river, and we had to add the annotation to every part of the river since one annotation could not be linked to multiple markings.

26   After marking our geographical elements and localities, our team annotated them by referencing relevant text documents about the elements from the same century and websites from the present day. The annotations were relatively easy to create since the elements of the annotation tool were very basic and limited but sufficient. One could only choose between 5 different font sizes, bold/italicize/underline/highlight text, and choose between different fonts. While writing our text, we could embed website links directly in the annotation tool. The icons were the same as in well-known word processors (like MS Word or GoogleDocs), making the use very easy, straightforward, and intuitive.

Usability, sustainability and maintainability

Access

27   As already mentioned before, a [GitHub Repository](#) gives access to all necessary files, deployment as a Heroku App is foreseen.

Dependencies

28   The GitHub-readme describes DM as a Ruby-on-Rails 5.x application. Since DM is available for deployment on Heroku, there is a good description of how to install and configure it. For Heroku deployment, the Heroku-18 stack is required.

29   The local installation as a development environment indicates the required installation of PostgreSQL v11.1, Ruby 2.5.7, and Bundler 2.2.23. Since there are plenty of online descriptions on how to set up this stack, it is not described in the GitHub-readme. All backend dependencies are in the Gemfile (installation with Bundler). Front-end JavaScript dependencies can be installed through Yarn. Unfortunately, several dependencies are not up-to-date or deprecated, and there is the need to find fitting and working versions.

Documentation

30   The online documentation for viewers and creators on the DM web-page is useful and accurate. The main features are described in video sequences. Some known issues and workarounds are listed, fixes in future versions are noted.

31   The documentation and step-by-step instructions for installation and configuration focus on the Heroku deployment and are available in the GitHub-readme. As mentioned in "Dependencies", the Ruby-on-Rails stack installation for local deployments is not described. The local development environment deployment instructions are not well documented, and there are some skills (like setting up a PostgreSQL database) required to successfully assemble the platform.

Support

32   For our small experiment, and since we didn't plan to deploy the platform, we did not rely on support from the developers. The community is being maintained through a Twitter account and a Facebook group. Due to the publication on GitHub, it is always possible to report bugs right at the source.

Learnability

33   DM works intuitively and does not require a lot of explanation (cf. also the chapter "Example"). Within a classroom setting, DM can be easily introduced and tested out in a

60-minute slot. Although no proper error messages are being shown (if clicks or draws are not allowed), users become quickly aware of shortcomings due to no reaction by the platform. The design follows state-of-the-art expectancies and symbols that guide users to possible pathways.

Build and install

34   As soon as the indicated stack is fitting the current DM-version, the deployment on Heroku is simple and needs a few configurations. For local deployment, there is specific experience required, as the installation of DM is challenging. Dependencies have to be modified since they are not up-to-date: In March 2021, the dependency files were from November 2019. As already mentioned, some bug fixes were released between August and October 2021; it is, therefore, possible that the mentioned shortcomings are already fixed. Uninstallation has not been tested.

Test

35   A basic template for testing the main features is implemented, but it is necessary to write case-specific tests. For our case, we didn't use the test function.

Portability and interoperability

36   As stated already, Digital Mappa is run as a webserver on a Heroku App. Heroku allows the integration of different programming languages, and in the case of Digital Mappa, the well-established Ruby-on-Rails has been chosen. Due to this, the setup ought to be highly portable and interoperable. End-user access to the platform is possible through any recent web browser (Mozilla Firefox, Google Chrome, Safari, etc.). During our evaluation, no problems from this site have been encountered.

37   At the moment, we are not aware of any other tools that have been developed to add to the functionality of DM. We are also unsure if this is necessary since the tool is constructed as a stand-alone edition platform.

Source, license, and credits

38   Digital Mappa 2.0 is provided open source under a GPL-3.0 Licence. A free installation on any infrastructure allowing to set up a Heroku app is possible. As far as we can see, development is entirely taking place on GitHub. At this moment (October

2021), eleven branches have been created, hinting at a somewhat active but not very big community of developers. Due to the fact that most contributors are mentioned on the web page, there is no indication on who contributed in what manner (except, of course, that it would be possible to track past pushes in order to get an idea).

Analysability, extensibility, reusability of the code

39   For this review, we didn't analyse the provided code.

Security and privacy

40   There is no information about data handling provided to the users, and we get no clear indication of where DM is deployed and which services are used (e. g., Heroku, AWS S3 bucket). Accordingly, we recommend deploying the system of its own servers or using DM setup by known providers (like university libraries) in order to not risk misuse of provided data. From the point of the structure of the provided framework, there is not a high risk of losing data due to breaches.

Supportability and maintenance

41   Due to recent updates in August and October 2021 (the first part of this review was prepared in spring, the second part in autumn 2021), we reckon that version 2.1.0 (which would be called 1.0 of DM2) will fix certain problems and allow for easier deployment.

Citability

42   The tool does not offer citations or citation guidelines.

User interaction, GUI, and visualization

User profile

43   The proto-typical user/contributor is probably a humanities scholar or student who does not want to deal with technical requirements and standards in-depth. The classroom setting has already been mentioned and allows for the quick introduction of fundamental approaches to the (digital) humanities. Annotations and especially the need to annotate images or texts is the main capability realized within the platform.

Technical interfaces and GUI

44   The core idea of DM is to offer an easy and intuitive way to annotate images and provide textual information that can be formatted and enhanced with embedded images. The tool is web-based and offers a graphical user interface (GUI) that can be accessed through common browsers. The GUI is oriented towards current web practices and is easily understandable. In general, it needs to be stressed that the visual components of DM are its main advantages, which makes the tool ready for use in teaching and outreach settings.

45   On the left side of the interface is a list of all the different documents in one collection. By clicking on a document, it gets displayed on the right side. The menu on the left cannot be hidden while working on or looking at a document. If one wants to work on a single page, there is a check-out box next to every file that the user can click. There are two main tools the user can work with on DM, a tagging and an annotation tool.
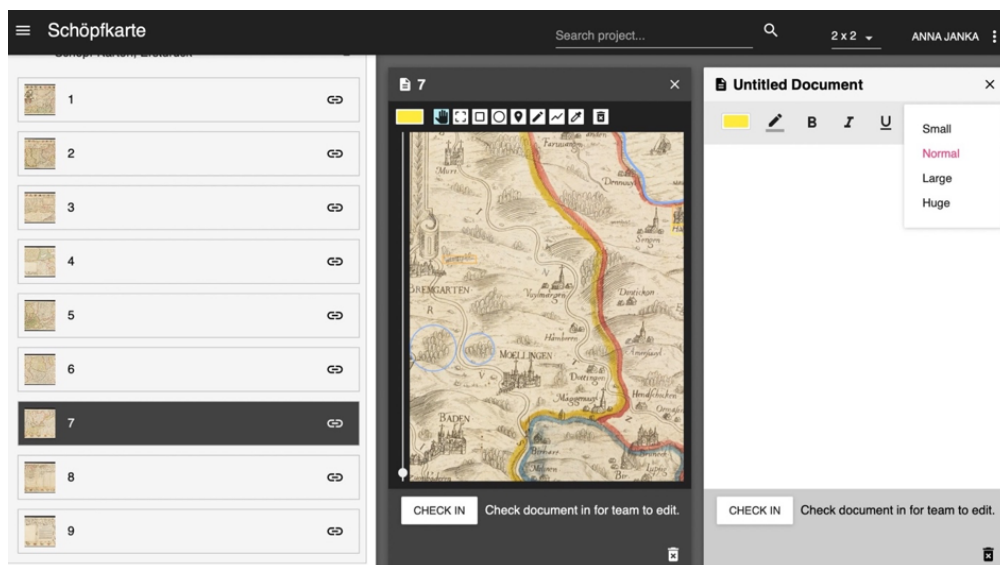

Fig. 2: On the left side of the interface are nine documents, of which the user chose to edit document 7. In the middle is the marking tool with the hand symbol selected (highlighted in light blue), and on the right is the annotation tool.

46   While tagging an element on the document, the toolbar is always visible and cannot be hidden, except for the color choices. Once one of the eight available colors is selected, the color selection can be hidden, and only the selected color appears in the toolbar. When a marking tool is selected, it is highlighted in blue, making it easy to see which tool is in use (fig. 2). Before deleting a mark on the document, a warning pops up,

the same for deleting a whole document. This enables the user to overthink his/her choice(s). On the contrary, there is no error prevention in deleting annotations. Once the user clicks on the delete button, the annotation disappears without further warning and any chance of reparation. One of the main disadvantages of the tool is that there is no Undo-Button in general, one cannot go back to previous document versions, and all changes are irreversible.

47   The toolbars of the annotation and marking tools consist of standard symbols used in other programs, and the variety of tools is wide enough to individualize an annotation or differentiate different markings, but not too extensive. For example, a user can select between four different font sizes to write an annotation. When the user goes over a symbol with his/her mouse, the symbol's function gets explained (B = "bold selected type"). After the documents have been edited and annotations were added, the user can search for different keywords in the top right of the interface. All documents are included in the search. Once a keyword is selected, the corresponding document pops up on the screen.

Visualization

48   DM does not provide specific visualizations. However, we would not expect such a thing in a platform like DM in the first place. Nevertheless, it is an easily individualizable tool. As a user, one can display documents and annotations that are important and hide more irrelevant parts. Additionally, the different documents can easily be moved around on the interface, and therefore it is possible to display, for example, documents 1 and 4 next to each other to compare them with each other and visualize the data in different ways.

User empowerment

49   There is not one fixed workflow defined for the use of DM. Also, no ontologies or gazetteers are implemented in DM that need to be used. On the one hand, we encounter this as formidable liberty towards use in quite different settings. On the other hand, this means that there is no support of existing standards and solutions, e. g., towards norm data, given.

Accessibility

50   All browser-based tools (e. g., zooming, read out loud) are supported. Besides this, we were not able to identify particular features.

Conclusion

General characteristics and realization of aims

51   DM is a mature and performant system that allows for easy and barrier-free use in different settings: from simple digital editions to in-depth text annotations, DM offers a general approach that leads to pragmatic as well as presentable results. Since it is an annotation and publication system in one, contributors will be able to inspect and share their results right away and directly through the platform.

Achievements and scholarly contributions

52   Opposed to a multitude of recently developed tools, workbenches, and standards (e. g., ontologies), DM aims at a very general but active user group in need of easy-to-use annotation and publication tools. Specialization and implementation of the latest technological developments (read, e. g., Linked Open Data) have not been intended but somewhat sidelined, focusing on very few approaches that work flawlessly.

53   We expect that due to DM, a variety of rather short but potentially complex digital artifacts will be created and made accessible. Therefore, DM fits in a small but essential niche for (visually appealing) documents that are not treated within large textual editions or don't merit a stand-alone web-page.

Particularities and limitations

54   DM demonstrates that it is possible to provide an easy-to-use platform with a solid design. Congrats to the developers for this.

Suggestions for improvement

55   In version 0.3.2. (March 2021) the project seemed orphaned since dependencies were from November 2019. In July 2021, the development went on, and version 0.5.0 was released in August. In October, we got aware of the release of version 1.0.0.

56   When we started using DM, we had an application in a classroom setting in mind for which the tool worked very well and flawlessly (from a viewer and contributor perspective). The implementation of norm data (e. g., support of Geonames) is, of course, possible in the form of linking, but currently, there is no clear path towards using such data. In this regard Recogito is currently ahead. At the same time, we started discussions about using DM as a crowdsourcing application in order to deeply annotate one specific historical map. At the current stage of development, DM does not seem to be an adequate tool for this approach. Although the check-out functionality hinders overwriting contributions, the user management is not really convenient (every user needs a dedicated account with an individual password). While collaborating with other users is possible, only one person can work on a document at a time. The ability to check out an annotation box instead of the whole page would make a crowdsourcing approach more feasible. Furthermore, certain text entities were envisioned in specific (fixed) structures. This is, for the moment, not supported but could be an interesting add-on. Moreover, we would like to have more possibilities with regards to extraction/export of textual files (at least in plain text, ideally as TEI XML).

References

Factsheet

Resource reviewed	
Title	Digital Mappa – Simple and Web-based Annotations
Editors	(unclear, not properly defined) University of Wisconsin - Madison's Center for the History of Print and Digital Culture
URI	https://www.digitalmappa.org/
Publication Date	14.07.2022
Date of last access	14.11.2022

Reviewer	
Name	iD Hodel, Tobias
Affiliation	University of Bern
Place	Bern
Email	tobias.hodel (at) unibe.ch

Reviewer	
Name	iD Janka, Anna
Affiliation	University of Bern
Place	Bern
Email	anna.janka (at) unibe.ch

Reviewer	
Name	iD Widmer, Jonas
Affiliation	University of Bern
Place	Bern
Email	jonas.widmer (at) unibe.ch

General information		
Software type	What type of software is it? (cf. Catalogue 0.1.1)	Virtual Reseaerch Environment (VRE)

Identification of the environment	On which platform runs the tool? ([cf. Catalogue 1.4](#))	Web browser
Purpose	For what purpose was the tool developed? ([cf. Catalogue 1.5](#))	developed to accomplish a general task
Funding	Which is the financial model of the tool? ([cf. Catalogue 1.6](#))	Free/open
Maturity	What is the development stage of the tool? ([cf. Catalogue 1.5](#))	Release
Methods and implementation		
Programming Language	Which programming languages and technologies are used? ([cf. Catalogue 2.3](#))	Ruby, Other: JavaScript, CSS, HTML
Reuse	Does the tool reuse portions of other existing software? ([cf. Catalogue 2.3](#))	yes
Input format	Which input formats are supported? ([cf. Catalogue 2.4](#))	.pdf
Output format	Which output formats are supported? ([cf. Catalogue 2.4](#))	.txt
Encoding	Which character encoding formats are supported? ([cf. Catalogue 2.4](#))	utf-8
Encoding preprocessing	Is a pre-processing conversion included?	yes
Dependencies	Does the documentation list dependencies on other software, libraries or hardware? ([cf. Catalogue 3.2](#))	yes
Dependencies installation	If yes, is the software handling the installation of dependencies during the general installation process (you don't have to install them manually before the installation)?	yes
Documentation and support		
Documentation	Is documentation and/or a manual available? (tool website, wiki, blog, documentation, or tutorial) ([cf. Catalogue 3.4](#))	yes
Documentation format	Which format has the documentation? ([cf. Catalogue 3.3](#))	readme, .html

Documentation parts	Which of the following sections does the documentation contain? ([cf. Catalogue 3.3](#))	'Getting Started' section (installation and configuration), Step-by-step instructions, Examples, FAQ
Documentation language	In what languages is the documentation available? ([cf. Catalogue 3.3](#))	English
Support	Is there a method to get active support from the developer(s) or from the community? ([cf. Catalogue 3.4](#))	yes
From of support	Which form of support is offered? ([cf. Catalogue 3.4](#))	Other: Discord Channel
Issue tracker	Is it possible to post bugs or issue using issue tracker mechanisms? ([cf. Catalogue 3.4](#))	yes
Usability and sustainability		
Build and install	Grade how straightforward it is to build or install the tool on a supported platform: ([cf. Catalogue 3.6](#))	tricky
Tests	Is there a test suite, covering the core functionality in order to check that the tool has been correctly built or installed? ([cf. Catalogue 3.7](#))	yes
Portability and interoperability	On which platforms can the tool/software be deployed? ([cf. Catalogue 3.8](#))	Not applicable (if web-based for example)
Devices	On which devices can the tool/software be deployed? ([cf. Catalogue 3.8](#))	Not applicable (if web-based for example)
Browsers	If the tool is web-based: On which browsers can the tool/software be deployed? ([cf. Catalogue 3.8](#))	Mozilla Firefox, Google Chrome, Safari
Plugins	If the tool is web-based: Does the tool rely on browser plugins? ([cf. Catalogue 3.8](#))	yes
API	Is there an API for the tool? ([cf. Catalogue 3.8](#))	yes
Code	Is the source code open? ([cf. Catalogue 3.9](#))	yes
License	Under what license is the tool released? ([cf. Catalogue 3.9](#))	GNU/GPL

Credits	Does the software make adequate acknowledgement and credit to the project contributors? ([cf. Catalogue 3.9](#))	yes
Registered	Is the tool/software registered in a software repository? ([cf. Catalogue 3.9](#))	yes
Possible contribution	If yes, can you contribute to the software development via the repository/ development platform?	yes
Analysability, extensibility, reusability of the code		
Analysability	Can the code be analyzed easily (is it structured, commented, following standards)? ([cf. Catalogue 3.10](#))	yes
Extensibility	Can the code be extended easily (because there are contribution mechanisms, attribution for changes and backward compatibility)? ([cf. Catalogue 3.10](#))	yes
Reusability	Can the code be reused easily in other contexts (because there are appropriate interfaces and/or a modular architecture)? ([cf. Catalogue 3.10](#))	yes
Security and privacy	Does the software provide sufficient information about the treatment of the data entered by the users? ([cf. Catalogue 3.11](#))	yes
Supportability and maintenance	Is there information available whether the tool will be supported currently and in the future? ([cf. Catalogue 3.12](#))	yes
Citability	Does the tool supply citation guidelines (e.g. using the Citation File Format)? ([cf. Catalogue 3.13](#))	yes
User interaction, GUI and visualization		
User profile	What kind of users are expected? ([cf. Catalogue 4.1](#))	Humanities researcher, Digital humanist, General public
User interaction	What kind of user interactions are expected? ([cf. Catalogue 4.1](#))	Reading, Text editing, Image editing, Comparing
User Interface	What kind of interface does the tool provide?	Graphical User Interface (GUI)

	(cf. Catalogue 4.2 and 0.1.1)	
Visualization	Does the tool provide a particular visualizations (in terms of analysis) of the input and/or the output data? (cf. Catalogue 4.3)	yes
User empowerment	Is the user allowed to customize the functioning of the tool and the output configuration? (cf. Catalogue 4.4)	yes
Accessibility	Does the tool provide particular features for improving accessibility, allowing „people with the widest range of characteristics and capabilities" to use it? (cf. Catalogue 4.5)	yes