https://doi.org/10.2352/issn.2168-3204.2019.1.0.34

Digitizing, archiving... and then? Ideas about the usability of a digital archive

André Kilchenmann, Flavie Laurens, Lukas Rosenthaler; Data and Service Center for the Humanities DaSCH; Basel, Switzerland

Abstract

Digitizing is everywhere. Archives and libraries bring their holdings into the digital world. New sources are created exclusively digitally. The archive is and will become digital. The process of digitization has been going on for more than twenty years and has increased considerably since the appearance of the first smartphone. It is now possible to be online, read and generate digital content around the clock. And the amount of data is enormous.

At the Data and Service Center for Humanities, we are confronted with this amount of data in the research field, and we try to bring the different data models and different media types into a uniform but flexible system. The difficulty is not the data storage anymore but the presentation and usage of the data. The aim is not to archive data, but to keep them alive. Availability and usability are playing an important role. Right now, we are two front-end developers building our infrastructure web applications and think about new possibilities to bring the data to the users.

Digitizing, digitizing

In the last twenty years, one topic has increased strongly: The Digitizing. Today, almost all data are created only digitally. In addition, many archives and libraries invest a lot into digitizing their analogue data. There are many reasons for this process. A simple reason is data recovery. Dynamic media like moving images or audio documents are always bound to a technical device which are gradually becoming obsolete. So, the work is enormous. But what happens to all these data?

Too often analogue sources are digitized so that the sources are digital, but the filing remains archival in nature. There are efforts to open the data and to make them public on an application programming interface (API). There are a few user interfaces to present the data in a nice and simple way but working with the sources remains difficult. With the digitizing of data there should always be answered two questions: How should we store the data with a long-term perspective? What could be the presentation and the usage of the data?

Much has already been written and discussed about the storage of data. In this paper, we will discuss our data storage system only briefly. Now, the next step is about the presentation and the use of the data on the client side. At our Data and Service Center for Humanities (hereinafter called DaSCH), we have to deal with the most different data and sources. "The main task is to operate a platform for humanities research data that ensures access to this data. In addition, the networking of data with other databases is to be promoted (linked open data), thus creating added value for research and the interested public." [1]

The "digital turn" has changed research in the humanities to a large extent: many new digital tools and methods exist with which to access and analyze texts, videos, sound, and music. Scholars are eager to explore these new approaches to understand and create new knowledge. However, cutting edge IT-technology can be complex and time-consuming to master, especially in time-limited, small, or under-funded projects such as early career scholarly work and PhD theses. The heterogenous field of data was already a challenge on the data storage side but even a bigger challenge on the front-end side.

Data storage: Knora

Our mission at the DaSCH is the preservation of many project data from the different disciplines in the humanities. This includes the work with still images, moving images, sound, music notation, books, facsimiles, arts and so on. To provide qualitative data handling services to Swiss and international researchers, the DaSCH develops and maintains a software platform called "Knora" [2] consisting of a database based on a Resource Description Framework (RDF) triple store and Application Programming Interfaces (APIs):

- Knora is a software framework that can store texts, images, audio and video recordings, metadata, annotations, text markup, and any other data created by humanities research. Knora also provides powerful tools for searching, annotating, extending, linking, sharing and reusing data and is designed for long-term preservation. The architecture of Knora goes well beyond the Open Archival Information System (OAIS) reference model for digital archives. OAIS basically emulates the processes of an analogue archive containing physical artifacts in the digital domain. However, for qualitative research data, this model of emulating an archive is not sufficient. The data itself, not only its descriptive metadata, has to be searchable at any time; data has to be annotatable and linkable on a very fine-grained level as a full data set. In addition, the data objects must be changeable, e.g., if new findings emerge, these findings can be added, while previous data versions are preserved.
- RDF allows great flexibility of data modeling, which enables the DaSCH to use one single infrastructure for data, metadata, models, and structures for any project regardless of the data concept used. Thus, the DaSCH has to maintain only one single infrastructure to provide sustainability. Data from any one project can be analyzed and compared with data from other projects.
- APIs based on open standards (RESTful and JSON-LD) are designed to be used by virtual research environments for querying and updating data.

Media server: Sipi

An important aspect of qualitative data in the digital humanities is that, in most cases, the preservation of data sets alone makes little sense. The way the data sets are accessed and re-used, researchers' queries and views, etc., often form an integral part of the knowledge represented by the data sets. Thus, the infrastructure of DaSCH provides components to emulate queries (Knora) and to store different kind of

media files. For images we use exclusively the International Image Interoperability Framework (IIIF), which is a set of shared application programming interface (API) specifications for interoperable functionality in digital image repositories. In the near future, IIIF will also include standards for videos and audio files. Texts can be imported/exported as TEI/XML.

All of the mentioned file types, still images, moving images, sound and text files, can and will be handled by our own high-performance media server Sipi, written in C++. Sipi implements IIIF, an authentication standard for access control and offers many extensions. New extensions can be written in Lua, which expand the Sipi service in a fast and easy way.

Sipi already converts efficiently between image formats, preserves metadata (such as EXIF, IPTC and XMP) contained in original image files or transforms ICC color profiles. In particular, if images are stored in JPEG 2000 format, Sipi can convert them on the fly to formats that are commonly used on the Internet. It offers a flexible framework for specifying authentication and authorization logic in Lua scripts, and supports restricted access to images, either by reducing image dimensions or by adding watermarks. This allows to store only one working image copy on the server. So, no additional thumbnails or images with watermarks are needed.

By following the discussion about the implementation of video and audio to the IIIF specification [3], Sipi will support the A/V media as soon as the technical specification for them is released.

Data usage: simple user interface

As mentioned above, Knora provides an API that allows to query and display data with various applications. Of course this fact is appreciated by the individual project managers, as they would like to have their own application for their project. However, most of them do not have the technical know-how to write their own application or they don't have the resources. We also lack human resources to write a separate application for each project, because we are only two front-end developers.

Therefore, we have developed small knora-specific, front-end modules [4] to build a project-specific website quite easily. The code of these modules is open-source – like all our software code – accessible via GitHub.com and published on NPM. But developing modules have not been our main task. We have started to develop one generic web application, based on the already existing modules, where the users can do their research, administrate their projects and their data.

There's already an existing prototype of such a generic web application, a virtual research environment called Salsah. The acronym stands for System for Annotation and Linkage of Sources in Arts and Humanities. This prototype was developed about 10 years ago and is still running on salsah.org [5, 6]. Two dozen projects are using and working with it. Salsah uses outdated technology and we know that it's not really user-friendly. A gap we want to close. So, we have started from scratch to re-design and develop a new simple user interface for the DaSCH.

The technology we are using for the generic web application is the Angular framework, developed by Google. We can reuse code and build apps for any deployment target. For web, mobile web, native mobile and native desktop [7]. We decided to develop a purely web-based application and use the material design guidelines for the styling.

The challenge is to make the user interface as simple as possible. The users of the application have no technical background. Therefore, our task is to find a way to simplify the complexity of the different data and requirements. The application should help to build a data model, called "ontology" in RDF, as simple as in FileMaker, which is part of the administration service. After defining a project and the ontology, the user can add the data, browse the data, and be able to work on this data. There are different tools to annotate, connect, but also transcribe and generate additional data, which means more knowledge at the end. The app should be as generic as possible but still flexible.

The web application will have three main parts. One is the project administration, where the user can set up the general project information, organize the team, and define the data model for his resources. The second part is the search panel with different search options with filters and similar tools. The search panel includes the display of the search results in an ergonomic view. The third part is probably the most complex. The workspace contains the tools to work on the data and the sources. Comment and annotate sources and their metadata fields is one thing, but we also want to support real research tools like A/V transcription tools, as well comparison tools, where we bring the different sources together and display them side by side.

Project administration

In our case, a project can be a PhD research, a pilot or proof of concept, and we keep the general project information as short as possible. There's more flexibility in the creation of the team and the data model. The user who creates a new project will also form the team. Permission roles can be assigned to each individual member. Who is able to edit or to see the data in the project? Data includes the research sources and their metadata. Permissions can be set for an entire source or only for single metadata fields. Sources and metadata fields have to be defined in the data model / ontology editor, which is part of the project administration.

We offer a tool to create data models easily. The user has to know first about the data and sources he wants to work with. The data model can be flexible and customizable, but it's still standardized because of the concept of RDF offered by Knora. The data model definition can follow the FAIR data standard, but compliance is not required to analyze the data. The questions to answer in the creation of the data model: "What kind of data do I have in my project?", "What are the sources and what are their metadata?"

For example, in a humanities field like the cultural anthropology, a researcher interviewed a dozen people. During these interviews he taped, they talked about photographs. Among all the data collected during the start of the project, the most important are: audio-files of the interview, photographs, data about the interviewed person, and probably the location where the photograph was taken if this information is necessary for the project and the research question. Figure 1 illustrates the required resources and the connection of the individual sources with each other. The "interview transcript" is an own resource and can be created from the resource "Interview" which contains the audio file of the Interview. For the transcription, we have developed an appropriate tool.

Our idea is to offer a list of predefined source types, where the user can select from and combine to one's own needs. With a predefined list, the user doesn't have to learn the schema, guidelines of RDF, and the complex ontology setup. In our example with the interview and the photographs, the user drags and drops the following main source types from the list:

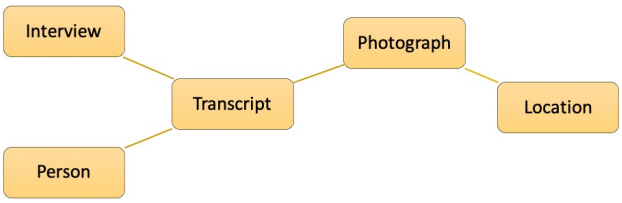

Figure 1. *Example of project data — Five source types: Interview, Transcript, Person, Photograph, Location.*

- Audio / Sound / Interview
- Transcript
- Image / Photograph / Postcard
- Person
- Location / Place

The predefined source types already offer a suggested list of metadata fields specified for this source type. This list helps to create a data model simply and quickly. The selected source type and the suggested metadata fields are customizable as long as the defined ontology is not published or used yet. So, it's possible to deselect the suggested metadata fields, to adapt others and to customize them. Here we offer a list of predefined metadata fields, to add them quite easy by drag and drop as shown in figure 2.

Figure 2. *Select all main source types by drag and drop; e.g. for a taped interview, select the source type "Audio / Sound / Interview".*

Search panel

Once the data model is ready, the user is able to add data and upload his files. The user interface offers several possibilities to add data. When a research project starts from scratch, the user can enter and generate the data directly in the application itself. Generating new data can be done one by one with a form or with a table-based (Excel-like) tool. Augment the metadata is, as usual, provided by a form based on the previously defined data model. If needed and defined, an upload area helps to store the files. Handled by Sipi, we support the storage for still images, moving images, audio files and text documents such as PDF/A. In case of interview transcriptions from audio or video files, the user interface will offer a simple transcription tool (see figure 4). If the project has already started and the user has already his data in FileMaker or similar database software, the application supports an upload (transfer) of standardized CSV or XML files.

The user interface supports the connection of different sources even if they're not in the same project. The link between two sources is an additional data set and bi-directional, which extends the metadata of a source on both sides. Also a comment on a source and on their metadata fields, and the transcription of an audio-visual material expand the data, the sources, and finally, the knowledge. These actions will generate more data and will help to find specific sources and their relations easily. To find the data, and explore new data and connections, the interface has three different search tools implemented.

One search is the simple full-text search well known from Google. Most of the time, users will use this search mode. The second one is the advanced search with many options to filter by source type or by the metadata of source types. Each filter can be standalone or combined. The metadata field can be precisely filtered with criteria such as "contains", "like", "equals to", "exists", or in case of a date value with "before" or "after". In addition, for a metadata field that is connected to another source type, it's possible to filter by this second source type. If you are looking for the source type "Photograph" with the metadata field "Photographer", which is connected to source type "Person", you can search for photograph(s) taken by person(s) who is born before February 1970. The result of this request will be an intersection of the two source types, illustrated in figure 3.

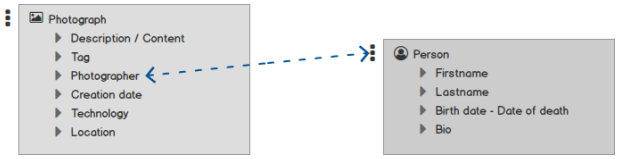

Figure 3. *Photograph and Person are two sources, connected by metadata field "Photographer" in photograph. In advanced search (and expert search) you can find an intersection of both by filtering both sources at the same time.*

At least, there will be an expert search, which can be more powerful than the advanced search, but requires knowing the query language Gravsearch, a syntax based on SparQL and developed by the DaSCH team. With Gravsearch, expert users can build searches by combining text-related criteria with any other criteria. For example, you could search for a photograph in a transcript that contains a certain element and also mentions a person, who lived in the same country as another person, who is the creator of another photograph.

In all search modes, it's possible to filter by own project or to keep it open to get more results from all different public project data. Once a search is submitted, the relevant results are displayed in the user interface. There are three layout choices: a simple list, a light-table-like preview grid list, or a table-based Excel-like view. The last layout is enabled when the search has been performed with only one source type, because the columns correspondence to the metadata fields. The table-view is good to edit more than one entry just by once, well known from duplicate functions in excel. Finally, there will be another view, which will show the connection of the individual sources and their metadata to each other. This visualized graph view can be very helpful because it allows to discover more. It should be possible to follow the connections and to get more information.

Workspace

After adding data and finding the desired sources, the user can (re)view them and annotate the source itself, the media file, or single metadata values. If he selects more than one source, he can compare them in a side-by-side view, link them, edit them all at once, or save them in a collection. A collection is similar to a playlist in a music app or a shopping basket in an online store.

As it exists specific views for the different media types, there are also specific tools to work on the source. With still

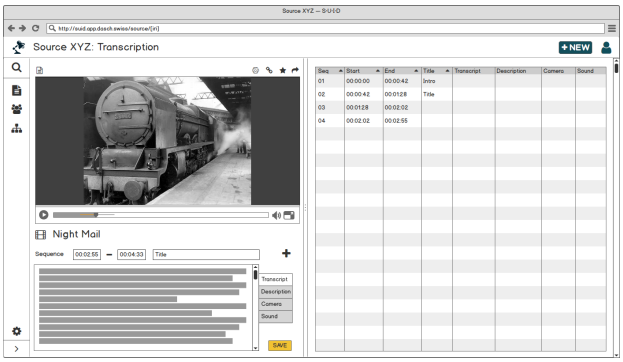

Figure 4. *Single resource full-frame view with the transcription tool at the bottom. The source type in this example is "Video" with a table-based sequence protocol on the right hand-side.*

image source, the user is able to draw regions on the image and annotate or transcribe this region. A still image source is usually used for photographs, post cards, but also book pages, letters, etc. In time-based sources like moving image and audio documents, the user can highlight sequences on the timeline. With the integrated transcription tool, it's possible to transcribe the marked sequence. We have already done some tests and we know that the web technology offers functions to do so. There will be keyboard shortcuts to navigate through the A/V media while writing. This helps to keep the hands on the keyboard and to be faster than switching between keyboard and mouse device.

Figure 4 shows how the transcription tool could look like. In the top left corner is the video player for the moving image source, followed by the transcription tool. In film and media studies, it's important to transcribe various aspects in the moving image. It's not only a transcription of the dialog and texts, but also a definition about the camera view, the sound or a scene description. This will all be implemented in the transcription tool and the transcribed sequences are displayed on the right hand side organized in a table. Here as everywhere, this is customizable and the user can decide if he needs a table-based sequence protocol or only a simple line-based transcription.

At the end, there will be a share and export function to get the data out of the web application if it's necessary. In order to keep the user's data inside of our infrastructure, we are trying to implement as many research tools as possible, so the user doesn't have to switch to an additional third-party software. We shouldn't forget that our infrastructure, especially the data storage in Knora, is prepared for long-term availability. With the export of data and to work on them somewhere else, the data is not anymore under history control in the long run.

Conclusion

Researchers in the Humanities need an accessible and easy to use digital platform to manage, store, work with and share their research data. Interesting IT-tools already exist. However, for small projects, e.g., PhD projects, pilot projects, and proof of concepts, these technologies can be difficult to use due to researchers' limited IT-skills, small amounts of funding, limited project time, or need for specific assistance. Most small humanities projects rely on "homemade solutions" using desktop data management tools, such as FileMaker, MSAccess, etc., but the data modeling often does not follow standards. The data itself may be inconsistent. Often researchers only have access to poor tools for export, analyze, and re-use of the data.

Powerful data tools already exist for humanities research. The Data and Service Center for the Humanities (DaSCH) is a national research infrastructure at the University of Basel that includes all disciplines of the humanities. This infrastructure is clearly focused on qualitative data such as interlinked databases, complex data involving different media with annotations (text, facsimile, photographic images, video, and film), rich linkages, and connections [8]. The DaSCH team has developed a unique and powerful software platform, Knora, to provide services, like data maintenance, long-term access, and research and analysis tools for qualitative data. However, with no simple user interface, this platform is not easily used by researchers with small projects or limited resources. Over 30 Swiss projects are currently queuing to get access to Knora, waiting for developer staff support.

We now develop a simple user interface for the DaSCH. The design is an intuitive, easy to use web-based application placed on top of Knora to directly use its powerful data management functionalities. With this application, the researchers will be able to add data models, search, browse, and work with their qualitative data as easily as they could with a desktop data management tool. In addition, data models and data will automatically follow accepted standards, be interoperable, findable, and re-usable. Researchers and scholars with small data sets will have access to long-term accessibility at minimal cost and time to keep their research data alive, guaranteeing longevity of the data.

References

[1] Data and Service Center for the Humanities DaSCH. URL: http://dasch.swiss/about (accessed February 21, 2019).

[2] Knowledge Organization, Representation, and Annotation. A software framework for storing, sharing, and working with primary sources and data in the humanities. URL: http://www.knora.org. The source code is available on GitHub: https://github.com/dhlab-basel/knora (accessed both on March 10, 2019).

[3] International Image Interoperability Framework™. URL: https://iiif.io/community/groups/av/charter (accessed on March 12, 2019).

[4] Knora ui modules. User Interface Modules for Knora. URL: https://dhlab-basel.github.io/Knora-ui (accessed on March 11, 2019).

[5] System for Annotation and Linkage of Sources in Arts and Humanities. URL: http://salsah.org (accessed October 21, 2018).

[6] cf. Schweizer, Tobias, Rosenthaler, Lukas and Subotic, Ivan (2015). Visualisierung von Annotationen und Verknüpfungen in SALSAH. http://blog.ahc-ch.ch/wp-content/uploads/2015/09/13-Schweizer-et-al.pdf (accessed October 21, 2018).

[7] Angular is a development platform for building mobile and desktop web applications using Typescript/JavaScript and other languages. URL: https://angular.io (accessed on February 21, 2019).

[8] cf. Swiss Academy of Humanities and Social Science (2015). Final report for the pilot project "Data and Service Center for the Humanities". (DaSCH). Swiss Academies Reports, Bern.

Author Biography

André Kilchenmann studied cultural anthropology, media studies and computer science at the University of Basel. During this time, he worked at the museum of cultures in Basel and at the data center of the University. His interests are photography, design and digital work in general. In 2016, he completed his PhD studies at the Digital Humanities Lab in Basel and now works for the Data and Service Center for the Humanities DaSCH.

Flavie Laurens is a front-end designer and web developer. She has a master's degree in "Systematics, Evolution, Paleobiodiversity" and in "Biodiversity Informatics" from Pierre and Marie Curie University, Paris. Since June 2018, she works on different user interfaces for the Data and Service Center for the Humanities DaSCH.

Lukas Rosenthaler studied physics and astronomy and received his PhD at the University of Basel. He worked as a Postdoc at ETH Zürich. He wrote his habilitation in the humanities department of the University of Basel about long-term archiving of digital data. Since 2012, he's the head of the Digital Humanities Lab, and since 2017, he's the director of the Data and Service Center for the Humanities DaSCH.