A Framework for Supporting the Workflow for Archaeo-related Sciences: Managing, Synchronizing and Analyzing Data

Daniel Kaltenthaler[1] Johannes-Y. Lohrer[1] Peer Kröger[1] Henriette Obermaier[2]

Abstract: In this paper we analyze the challenges of the workflow in archaeo-related disciplines. We extract the main parts of the workflow and consider them in our solution for an eScience service in this domain. First, we describe the dynamic framework *xBook* to be used for different archaeo-related databases. Afterwards, we show the *Synchronization* supporting the collaboration with colleagues and sharing data. Finally, we describe the *Analysis Tool*, that allows analyses to be executed directly inside the framework. Concluding, we show a sample analysis in *OssoBook*, a zooarchaeological database.

Keywords: Analysis, Framework, Synchronization, OssoBook, Workflow, xBook

1 Introduction

The recovery and analysis of material culture is the main focus of archaeo-related work. The corpus of finds like artifacts, faunal remains, or human burial remains are excavated, described, categorized, and analyzed in projects all over the world. The results of analyzing collected data make us learn about the past. Having access to big data volumes may be essential to run significant analyses. Therefore, sharing data is important. A long time data was only recorded analog, the digital recording, if available, was normally done in Excel sheets. This made the exchange of primary data, which are often not published, complicated and difficult, thereby complex analysis of archaeo-related data were infrequent.

The usage of digital databases for the recording of data has increased in the last decades. There are some databases that offer complex and versatile options to enter data, like the zooarchaeological database *The York System* [Ha03] or the archaeobotanical database *ArboDat* [PDK11]. However, these databases lack other convenient features – they provide methods to export data, but sharing data comfortable via the Internet is not supported. For the basic and frequent problem statements the data is retrieved with predefined queries, but there are no complex analyses available that are directly built into the application. Data has normally to be analyzed manually or in extern tools using the exported data. Additionally, there are a wide range of databases that were customly created by freelanced archaeologists and bioarchaeologists, which functionalities are limited to their requirements, and are not published.

[1] Ludwig-Maximilians-Universität München, Institute for Informatics, Database System Group, Oettingenstraße 67, 80538 München, {kaltenthaler,lohrer,kroeger}@dbs.ifi.lmu.de

[2] Bavarian State Collection for Anthropology and Palaeoanatomy Munich, Kaulbachstraße 37, 80539 München, henriette.obermaier@palaeo.vetmed.uni-muenchen.de

While the workflow of all disciplines is similar, there still are some differences in the nature of the data. Therefore our ambition is to provide a flexible framework, that offers the creation of databases which supports the composition of all input options, where data can be entered, shared, and analyzed within the same application. This would support archaeologists and bioarchaeologists in their workflow.

In summary, the main contribution in this paper are as follows: We classify three main parts of the workflow of archaeo-related work in Chapter 2 and consider them in our solution for the digital realization. Our framework is described in Chapter 2.1. We explain the functionality to collaborate and share data with others, and to support working offline without an Internet connection, the *Synchronization*, in Chapter 2.2. The data collection and sharing forms the basis for documentations and for analyses inside the excavation project, that is provided by the built-in *Analysis Tool*, which we describe in Chapter 2.3. Finally, we show the composition of a frequently used analysis in the zooarchaeological database *OssoBook* as an example with the analysis framework in Chapter 3.

2 Creation of the xBook Framework

Concerning the archaeological and bioarchaeological work, we extracted three main challenges that have to be fulfilled to support the work of archaeologists and bioarchaeologists:

1. **Data Gathering:** Saving of archaeo-related data digitally in a database.
2. **Data Sharing:** Collaborating with colleagues and sharing data with other users.
3. **Data Analyses:** Executing analyses on the available data.

The solution of all three challenges must be as dynamic and flexible as possible to be usable in several archaeo-related databases. Below, we describe our solution for these challenges for the archaeo-related disciplines, which can also be transferred to other disciplines.

2.1 Development and Implementation of the xBook Framework

We set the challenge to provide a generic database solution for all disciplines, that is as customizable as possible to allow all required information about the specific data to be gathered. We created *xBook*, a generic open-source framework including the common and basic features for a database for archaeo-related disciplines. Below, we use the term “Book” to describe an incarnation of *xBook*. Each Book provides these common features:

In *xBook* the gathered data is separated to single data groupings called **Projects**. All findings of an excavation or a partial area of the excavation are combined to a single Project. This enables the collaboration and the sharing of data of specific excavations, or parts of it.

Most of the values that are entered to the database have a basic and simple type, like text, drop-down box, numbers, time, date, boolean values, etc. So the framework offers a

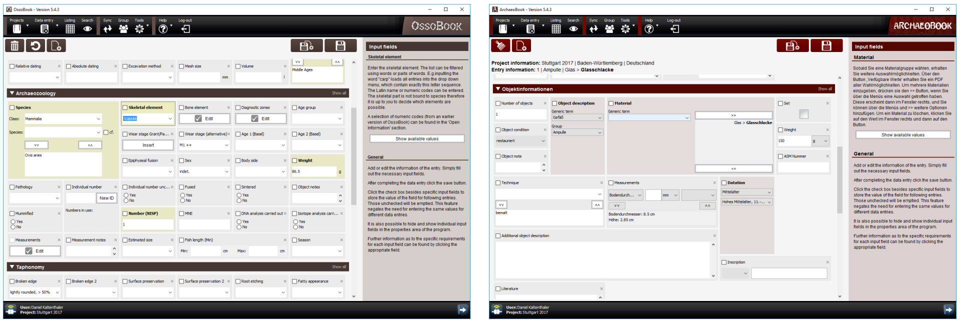


Fig. 1: The input mask of *OssoBook* (left) and *ArchaeoBook* (right), based on the *xBook* framework, that provides a basic GUI and functions, but allows customization like e.g. individual input fields.

dynamic and flexible **Data Entry Mask** that reuses the provided, most common types of input fields for each Book. There are already available several different input fields that can be used without having to implement a new type of input field. However, if a Book needs a specific, maybe much more complex type of input field, it can easily be added and be used for a specific Book. So, the Data Entry Mask of each Book is composed of the necessary, specific input fields and offers the required possibilities to enter the data, dependent on the archaeo-related discipline. As an example, a comparison of the Data Entry Mask of *OssoBook* and *ArchaeoBook*, two incarnations of *xBook*, is shown in Figure 1.

An overview about all data of a specific project can be displayed in the **Listing**. All entries of the Data Entry Mask are displayed as a table form. If there are available more than one Data Entry Mask, a selection of the mask is displayed. The table can be sorted and filtered.

To search the saved data for one or more specific data sets, *xBook* provides two types of **Data Search**. The first one is the Filter. It allows to filter the Listing for entries with specific terms or values. All data sets that do not match the Filter settings are hidden in the Listing. The Filter is made to regain specific entries. The second one is a Project Search where the user can search for local Projects that are matching with specific terms or values.

To be able to collaborate and share data, *xBook* provides an **Project Right Management**. Each Project can be shared to specific users, or to a specific user group. Each user or user group can be granted rights to read or write the data, and permission to edit the Project information. These rights can be extended and set individually.

Further features, like **User Management**, an **Exporter**, and an automatic **Update Function** is also included to the *xBook* framework.

All features are wrapped into a common **Graphical User Interface** which structures can be reused. So every Book provides the same basic frame, like navigation, footer, notifications, etc. The similarity of the graphical representation can be viewed in Figure 1. Still, customizations are possible, like new elements in the navigation, or the adjustment of the application logo. The common features are shared with all Books, that all benefit from new features and special implementations for a single Book. The features match the requests of archaeologists and bioarchaeologists to gather and save data entry for their excavations.

2.2 Synchronization of Data

To collaborate and share the data with others as well as store the data, a synchronization process is required. Since working online is not possible in environments in which a reliable Internet connection cannot be guaranteed, e.g. directly in the field of an excavation, we must provide a way to enter the data locally. These data must then be synchronized to the global server and all updated or added entries in the global database must be synchronized to the local database again. While there are already a wide range of different database systems available that offer possibilities for synchronization (like Microsoft SQL Server, Oracle, and MySQL), none of them fulfills all the requirements needed for our framework. Most of them do not "provide the ability to work offline and none of them allows synchronization of single data sets" [Lo16].

Concept: The *Synchronization* uses timestamps to keep track of the last time the entry was updated in the global database. Additionally the status within the local database is saved in another column. The status can be `synchronized`, if the entry was not updated locally since the last time it was synchronized with the global database; `updated`, if the entry was updated locally since the last synchronization; and `conflicted`, if a conflict was detected. A conflict occurs if the same entry is updated on two different local databases, without synchronizing the entry in between. Because incarnations of *xBook* can have multiple independent input units, the *Synchronization* synchronizes one complete entry – with all information that are saved in connected tables – at a time, one input unit after another.

Implementation: Each time an entry is added or updated in the local database, the status of the entry is set to `updated`. This defines which entry has been changed since the last time the *Synchronization* was executed. If the synchronization process is started, each input unit is checked which data sets are not synchronized. One after the other the unsynchronized data sets are uploaded to the server. If no conflicts are recognized, the status of the uploaded entries is set to `synchronized`. On the server, a trigger sets the timestamp to the current time every time an entry is uploaded or updated. If there is no unsynchronized entry left, the entries are synchronized back to the local database in the order of the timestamp on the server, starting with the entry that has the lowest timestamp on the server, but is higher than the highest timestamp on the local database. If the currently synchronized entry was updated locally or is marked as `conflicted`, this entry is ignored. This procedure is repeated for all available input units.

Conflicts: If a conflict was detected during the *Synchronization* process, the conflict has to be solved before the entry can be synchronized again. This has to be done manually by the users inside an own panel. When solving the conflict, the users gets the differences between the local and the global version displayed in a diff-screen. There, the users can select which value they want to save. After the users have decided the version they want to use for all differences, the entry is saved locally with the updated values as `synchronized`, but without updating the timestamp. Then the entry is additionally saved globally (without the *Synchronization*) with the timestamp of the global entry. Therefore, the timestamp of the local entry is updated during the next synchronization process. This can not be done directly. Otherwise entries would not be synchronized to the local database, if they were updated globally since the last synchronization process, because they would have a lower timestamp than the conflicted entry.

Integration: The integration in the application must consider that most of the users of *xBook* are not familiar to work almost exclusively with a computer. That is the reason why it is absolutely necessary to hide the complexity of the *Synchronization* behind an intuitive input mask that is easy to use. In this Synchronization Panel, all global Projects for which a user has read and/or write permission must be downloadable from the server. Therefore the corresponding Projects can be selected in the Project selection on the right side of the Synchronization Panel. The local Projects that have not been synchronized with the server before must be able to be uploaded to the server. These Projects can be selected in the Project selection on the left side. Existing Projects, that are either saved in the local or global database, must be updatable. For this purpose the corresponding Projects must be selected, as explained above. The synchronization process can be started by pressing the “Synchronize” button.

2.3 Embeddable Analyses Tool

Having provided the framework for creating databases for archaeo-related work (cf. Chapter 2.1) and enabled a feature for sharing the collected data and collaborate with colleagues (cf. Chapter 2.2), we now want to concentrate on possibilities to analyse this data. So far, data is usually exported from a database in the form of spreadsheets — e.g. Microsoft Excel, LibreOffice Calc, etc. — or comma-separated values files to meet the goals of specific researches in the archaeo-related domain. Although this is enabled by using the Export function in *xBook*, this method can be aggravating, time consuming, and error-prone, but it is still common practice. Our ambition is to provide a dynamic, flexible, and powerful tool that allows specific queries from archaeological databases without having any prior knowledge of programming. We want to support archaeologists and bioarchaeologists in their research and to provide the ability to select specific data from the database as is needed, with as few limitations as possible. The target is to provide a flexible tool that scientists can use for visually querying the data from their databases and create almost every composition for their data analysis. Furthermore, we want to offer a possibility to generate graphical results that scientists can use for their analyses and publications, without having to export any data and use external spreadsheet or analysis tools. This tool must work independent of possible changes of the database scheme.

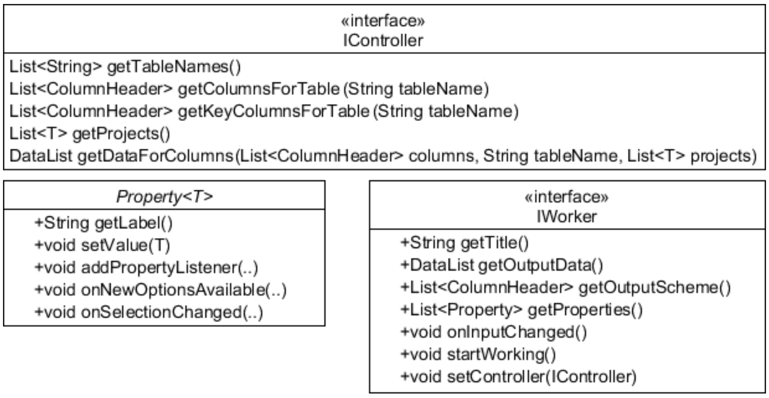
Concept: The *Analysis Tool* was developed for *xBook*, but must be flexible enough to be able to be embedded to other Java applications as well. To achieve this, the tool must provide a well defined set of functions that has to be implemented by the application, the tool is being embedded to. At the same time the tool needs an API that allows new components to be registered that might be specific for an application. This also requires the communication of the components to be generic, because the type of the connected components is not known beforehand. The tool must provide a predefined graphical user interface that can be directly embedded into these applications. However, it must also be possible to implement a custom graphical representation, in case another design of the elements is desired, or if the GUI library is not applicable. So, an abstract connection between the logic and the view layer of the components is required. The *Analysis Tool* is designed to be embeddable to other applications to support the integration of specific solutions in any database application. With these specifications, the tool is flexible and extensible, and supports individual implementations dependent on the requirements of the scientific domain.

Implementation: The different analyses must be composed with several single components ("Workers"), that each represent another task in the composition, like the retrieval of data, the filtering or sorting of tables, or for the joining of different data sets. Each Worker is composed of several Properties. These describe the structure of the Workers, that means they define which inputs and outputs are available and which settings can be set by the user. The graphical user interface checks for each Worker and displays a composition dependent on the available Properties. The communication with Properties is done in different ways. The communication to the Property is realized by the methods that are defined in the class `Property` (cf. Figure 2). The communication from the Property has to be done with an event handling system. For this, the object that wants to be notified about changes of Property has to be registered directly to the Property, in form of a Property Listener.

Workers can have Properties for the input and output of data which can be used to connect a Worker to other Workers. To carry out their logic, Workers can query the previous Workers for the output. Furthermore, they provide their output for other Workers. To reduce calculation time and resources, the Workers only calculate the output data if specifically queried. But they always provide the current output scheme which allows the connected Workers to use the data structure to display the selectable data to the users. To retrieve data from the application, the Workers can use the interface `IWorker` (cf. Figure 2) that has to be implemented by the base application.

The API of the *Analysis Tool* provides a registry class, which allows Workers to be registered for inclusion in the analysis. The indirect registration allows externally created Workers to be included to the analysis by scanning the .jar files in the folder "./analyses/workers". All classes that implement the interface `IWorker` are added to the list of available Workers in the *Analysis Tool*.

Integration: To be able to reuse the *Analysis Tool* in different base applications, it is important that the integration requires as few changes to the base application as possible. But since the *Analysis Tool* cannot know how the data is stored, or how the connection to the data is realized, the base application must implement some wrapper methods. The *Analysis Tool* itself is composed in a `JPanel` or `JFXScene`. For this, the base application can either create a new `AnalysisSwing` or `AnalysisFX` instance, which both require an implementation of the `IController` (cf. Figure 2). The `IController` is the interface that serves as the connection of the base application to the analysis. The base application therefore has not to know any internals of the analyses or other way around. Since the



```
«interface»
IController
List<String> getTableNames()
List<ColumnHeader> getColumnsForTable (String tableName)
List<ColumnHeader> getKeyColumnsForTable (String tableName)
List<T> getProjects()
DataList getDataForColumns(List<ColumnHeader> columns, String tableName, List<T> projects)

Property<T>
+String getLabel()
+void setValue(T)
+void addPropertyListener(..)
+void onNewOptionsAvailable(..)
+void onSelectionChanged(..)

«interface»
IWorker
+String getTitle()
+DataList getOutputData()
+List<ColumnHeader> getOutputScheme()
+List<Property> getProperties()
+void onInputChanged()
+void startWorking()
+void setController(IController)
```

Fig. 2: Schematic representation of the interfaces *IController* and *IWorker*, and the class *Property*.

analysis is done inside one panel, it can easily be included into the base application, which can decide where and when the analysis shall be displayed.

3 Sample Analysis: Age distribution of Animals in OssoBook

The database we focus on in this paper is *OssoBook* [Ka16]. Since its conversion to Java and MySQL in 2001, it has been continuously developed, and became part of the *xBook* framework with release of version 5.0 in 2013 [Ka15]. The application is currently used by approximately 200 users including scientists, Ph.D. students, and students in institutes, universities, museums, and scientific collections with zooarchaeology as field of research, as well as freelance zooarchaeologists in Europe. In the context of the JANUS[3] project of the *German Archaeological Institute*, Berlin, Germany, *OssoBook* will serve as a standard for the zooarchaeology domain in Germany. It contains internationally agreed standards for this branch of science. There are annually workshops in European countries which serve as an introduction to the application, as well as discussion forums on further development.

OssoBook already provided analyses for different scientific research questions, which were not able to be predefined in a framework like *xBook*. Therefore, a plug-in interface allowed the integration of analyses that were dynamically added to the application. Based on this development, it was already possible to implement different tools for data analyses that were integrated to *OssoBook* as plug-ins. [Lo12] [Ka12] The disadvantage of the plug-in interface was that it was only compatible to a specific database scheme of the *OssoBook* application. Each time the database scheme was updated, the plug-ins had to be updated as well. Most of the plug-ins became incompatible and were not further developed, so they could not be used any longer. Still, these analyses are very important.

We demonstrate the functionality of the *Analysis Tool* described in Chapter 2.3 on the basis of an analysis that was already implemented for the out-dated plug-ins interface of *OssoBook*, the age distribution of animals on base of the age determination of teeth [Ka12].

[3] http://www.ianus-fdz.de

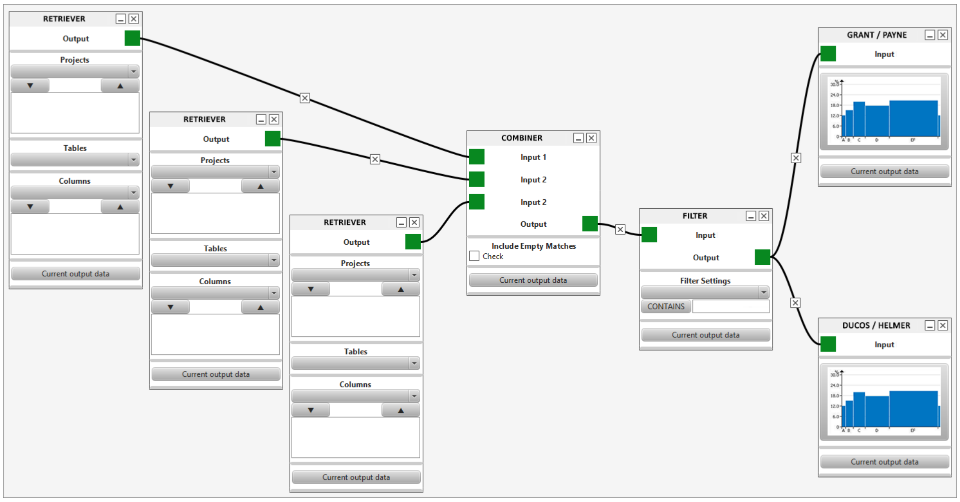


Fig. 3: The simplified composition of the age distribution of animals.

Age distribution of Animals: In zooarchaeology, the age distribution of animals at their slaughtering age is a frequent research question. The age distribution allows conclusions on the main usage purpose of these animals in the past, e.g. for milk, meat, fleece, or lamb production. There are different methods to achieve a result for this analysis. One is described by Grant [Gr82] and Payne [Pa73] and uses the description of the wear stage of the teeth. Another one is defined by Ducos revised by Helmer [He95] and uses the measurements of the crown of the teeth. Both of them distribute the animals dependent on the age of the teeth to groups. Using these improvements in zooarchaeological techniques, actual interpretation of slaughtering age profiles were developed which reflect animal management strategies. [HV07]

Composition in the Analysis Tool: The realization of the two methods consists of several, single Workers. Some of the default Workers can be reused for composing the necessary data. However it is necessary to implement an individual Worker for the calculation of the specific distribution of the animals into the groups. The compositions of the data, that are required to execute the individual calculations, are similar, but differ in one necessary information: While the approach of Grand/Payne uses the wear stage values, the one of Ducos/Helmer needs the measurements to calculate the distribution. So, most of the composition in the *Analysis Tool* can be reused for the two approaches. Due to the nature of the structure of the *Analysis Tool*, there are several possible options to generate the required data for the analyses. We describe a plain way, that allows to easily change the parameters that are used to filter the source data before running one of the two different analysis methods. First all information about the bone of the animal is gathered with three different Retrievers. The first Retriever fetches the species and skeletal element information for all entries, the second one gathers all wear stage information, and the third one gets all measurement information. In the second step all information is combined in a Combiner, that uses the primary key to find matching entries. The combined data can then be filtered in a Filter. Of course it is possible to use multiple filters, if required. Finally, the filtered data is forwarded to both of the two individual Workers for the calculation of the age determination. These Workers were created inside the *OssoBook* source code and are registered to the framework over the API. The Workers require all necessary columns to be available to be able to carry out the analysis. Since the columns are all defined inside *OssoBook* and therefore are known beforehand, no mapping of the columns is needed and the Workers can easily check if all required columns are available. The final composition of the analysis in the *Analysis Tool* can be viewed in Figure 3.

Results: Once the composition is completed and all Workers are connected, the Worker for two approaches (after Ducos/Helmer and Grant/Payne) calculate the necessary data for the analysis. The Workers hold the result of the calculation which can be displayed in table form by clicking the button “Current output data”. Furthermore, the Worker is extended with a functionality to display the data in a histogram, a typical representation of the data of the age distribution.

4 Discussion

In this paper, we explained the workflow of archaeo-related disciplines and described three main challenges of this area: The gathering, sharing and analysis of data. We first explained our solution for the digital data gathering in *xBook*, a dynamic and flexible framework for data gathering. We then explained the realization of the *Synchronization*, a function to work together with colleagues and share data with other scientists. Afterwards we described the *Analysis Tool* that we developed to be able to compose flexible and complex analysis with data from a database. Finally, we showed the functionality of the *Analysis Tool* by composing the age distribution of animals in the zooarchaeological database *OssoBook*.

While the *xBook* framework and the *Synchronization* are already used for scientific and inventory databases, there are also some possible improvements for the performance of the Synchronization process. A possible speed increase could be achieved by using compressing methods for the transmitted data. The usage of a special data type for the transmitted data would decrease the size of the data sent from server to client, or the other way around. Furthermore, the entries are currently only marked as “deleted” on the server, if a data set is deleted by the user. A solution to be able to delete the file sets in the database as well would decrease the necessary amount of data for the database.

The *Analysis Tool* is still at an early stage of development and there are still issues that could be addressed in the future, to make working with it go more smoothly. With the provided basic Workers, the *Analysis Tool* already allows a wide range of possible analyses. But the creation of further Workers would extend the possibilities of the *Analysis Tool*, especially Workers for predefined calculations, including statistical, mathematical or clustering algorithms. Besides, frequent used combinations of single Workers may be integrated as own Workers to increase the usability. Also the number of provided diagrams for the graphical representation of the data could be increased.

For a lot of analysis types, the provided Workers are sufficient to create analyses completely without programming skills. But still there are specific calculations that are not directly possible by using the existing Workers, like the calculation of the age distribution in the example in this paper (cf. Chapter 3). For creating a new Worker like these some programming skills are still required. In addition, access to the source code is also required to add a new Worker for specific calculations.

References

[Gr82] Grant, Annie: The use of tooth wear as a guide to the age of domestic ungulates. In (Wilson, Bob; Grigson, Caroline; Payne, Sebastian, eds): British Archaeological Reports British Series, volume 109, pp. 91–108. Book Publishing Inc., Oxford, 1982.

[Ha03] Harland, Jennifer F.; Barrett, Hames H.; Carrott, John; Dobney, Keith; Jaques, Deborah: The York System: An integrated zooarchaeological database for research and teaching. Internet Archaeology, 13, 2003.

[He95] Helmer, Daniel: Biometria i arqueozoologia a partir d'alguns exemples del Proxim Orient. Cota Zero, 11:51–60, 1995.

[HV07] Helmer, Daniel; Vigne, Jean-Denis: Was milk a "secondary product" in the Old World Neolithisation process? Its role in the domestication of cattle, sheep and goats. Anthropozoologica, 42(2):9–40, 2007.

[Ka12] Kaltenthaler, Daniel: Visual Cluster Analysis of the Archaeological Database OssoBook in Consideration of Aspects of Data Integrity and Consistency. Diplomarbeit, Ludwig-Maximilians-Universität Munich, 2012.

[Ka15] Kaltenthaler, Daniel; Lohrer, Johannes-Y.; Kröger, Peer; van der Meijden, Christiaan; Obermaier, Henriette: Synchronized Data Management and its Integration into a Graphical User Interface for Archaeological Related Disciplines. In: Design, User Experience, and Usability: Users and Interactions - 4th International Conference, DUXU 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part II. pp. 317–329, 2015.

[Ka16] Kaltenthaler, Daniel; Lohrer, Johannes-Y.; Kröger, Peer; van der Meijden, Christiaan; Granado, Eduard; Lamprecht, Jana; Nücke, Florian; Obermaier, Henriette; Stopp, Barbara; Baly, Isabelle; Callou, Cécile; Gourichon, Lionel; Pöllath, Nadja; Peters, Joris; Schiebler, Jörg: OssoBook v.5.4.3. München, Basel, 2016.

[Lo12] Lohrer, Johannes-Y.: Density Based Cluster Analysis of the Archaeological Database OssoBook in Consideration of Aspects of Data Quality. Diplomarbeit, Ludwig-Maximilians-Universität Munich, 2012.

[Lo16] Lohrer, Johannes-Y.; Kaltenthaler, Daniel; Kröger, Peer; van der Meijden, Christiaan; Obermaier, Henriette: A Generic Framework for Synchronized Distributed Data Management in Archaeological Related Disciplines. Future Generation Computer Systems, 56:558–570, 2016.

[Pa73] Payne, Sebastian: Kill-off patterns in sheep and goats: the mandibles from Aşvan Kale. Anatolian studies, 23:281–303, 1973.

[PDK11] Pokorná, Adéla; Dreslerová, Dagmar; Křivánková, Dana: Archaeobotanical Database of the Czech Republic, an Interim Report. Interdisciplinaria Archaeologica, Natural Sciences in Archaeology, 2:49–53, 2011.