

ElKowmGIS: A New Program for the Documentation of Archaeological Sites

Schuhmann, D.

Institute for Prehistory and Archaeological Science, University of Basel, Switzerland
d.schuhmann@unibas.ch

The documentation of the Palaeolithic site of Hummal (El Kowm, Syria) is carried out by digital utilities like EDM and computer. During the work with commercial documentation systems, the Hummal research team identified several shortcomings with the solutions initially chosen. The paper presents a new documentation system, developed as PhD-thesis based on the guidelines defined by the team of Hummal and the author. The paper also discusses the advantages and disadvantages of the already used solution and the new system.

Keywords: Documentation System, Java, GIS, CAD, Archiving.

1. Introduction

The Palaeolithic site of Hummal is situated in the region of El Kowm in the centre of Syria, about 100 kilometres north-east of Palmyra/Tadmor. The boundaries of the plateau of El Kowm are marked by the palmyrids and the hills of Djebel al Bishri (Figure 1). In this area several wells and Palaeolithic sites can be found.

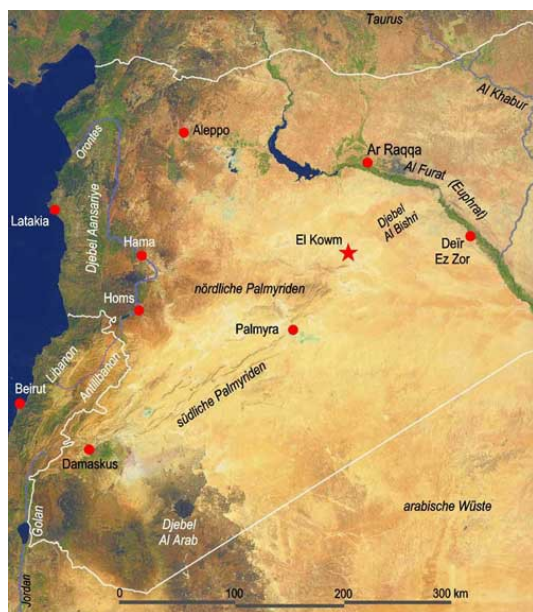


Figure 1: Situation of El Kowm (cartography: R. Jagher).

The documentation of the site has been carried out with the help of computers, since the 2005 excavation

campaign. The measurements on the excavation have been taken by an EDM, which is directly connected to a notebook. Using TachyCAD and Photoplan, the former enables the communication between the EDM and AutoCAD and the registration of points, while the latter is responsible for the rectification of images of surfaces and crosscuts.

Due to the fact that TachyCAD produces AutoCAD-Blocks, the internal AutoCAD Block-Attribute Extractor was used in order to export the three-dimensional information of the measured points. With this information it is possible to create a journal of finds, 3D-Models of the site (SCHUHMANN, 2011), or greater-scaled surfaces and plans of the excavation. All the files containing three-dimensional information, finds and sample descriptions, were then stored on an external hard-disk and later transferred to a simple Samba file-server in our Institute in Basel. During the work with this system and its containing data, we realised that there are some weak points in this method.

First of all, the system of measurement can not be adapted to our specific needs during the excavation and we do not have the possibility to use several tags for one object. A single entry field called “Information” (for up to 5 different tags) is therefore required. This Information is stored within the Block in one single field, and the extracted information of the blocks has to be corrected before using the data. For example a burnt flint, which was moved during the excavation, would get the tags ‘-b’ for burnt and ‘-v’ for not in situ (in German: verlagert). So the value in the journal of finds

would be: 'sx-b-v' or 'sx -b -v'. In an Excel-sheet this is rather unpractical as the abbreviation for the find (sx) and the two tags (b and v) are stored in two separate columns. This system allows sorting the data by different tags, e.g. the category of find or the state of the object. (For example: with this method it is easily achievable to sort out all burnt flints or all calcined bones.)

The second point was the missing option of definition, in which data is stored during the measurement. It is clear, that normally you only need the coordinates of a point (including also the height) and its meaning. But we needed more information, such as a simple field for the date and time, the point that was measured, the field where a remark is stored, and if there was a reflector used or not. The name of the excavator, who measured the point, is also missing in the entry-system of TachyCAD.

The third problem we found was archiving the data in the field. We therefore decided to use an additional excel sheet to store the metadata of the measurement, so that every measurement brings up at least three files:

- A dwg-file, where the measurement itself is stored.
- An excel-sheet, where the metadata (time, place, section, excavator, number of finds, meaning of the measurement, etc.) is recorded
- Another excel-sheet, which contains the journal of finds of this removal.

Additionally, there can be general views of the surface and also photos for rectification. The amount of files for one surface can be rather high, and also the size of the files, especially for the rectified images. Moreover, during the analysis in the lab in Basel more and more files needed to be produced (e.g. drawings of objects and crosscuts, plans of the excavation, journal of the fieldwork, and so on). At the moment we have an overall number of about 23'000 files with an overall amount of 56 GB. These are rather high numbers and therefore unpractical to work with.

As a result we realised that we needed a new solution to acquire and store our data. The answer was 'ElKowmGIS', a combination of CAD, GIS and database.

The source of the name 'ElKowmGIS' is easily to determine:

- El Kowm is the region, where our project is situated.
- GIS: The software, which is developed should have all the abilities GIS-, CAD- and database offer, only combined within one single solution.

2. The aims of ElKowmGIS

For the documentation software several guidelines were given, due to the system described in section 1 (see above).

First of all, it should be platform independent. This means that the user should be able to install and use the software on every computer, regardless of the operating system. Most solutions of measurement systems only offer support for windows or unix operating systems. Especially in Switzerland, where often Mac-Systems are in use, this is rather unsatisfying. ElKowmGIS on the contrary should provide a software solution, which can be used on every operating system.

Secondly, it should be a modular system for several processes that have to be done during the excavation and the evaluation or analysis of the material after the fieldwork. As in many GIS- or CAD-systems, ElKowmGIS should provide the possibilities to extend the system itself. This can be done by writing own modules in java, or by asking the author to implement a special feature. Due to the package solution both ways are easily done.

The third point of the guidelines is the ability to use the software without adapting the documentation system of the excavation to the software. Furthermore the software should be adaptable to the documentation. This means, that the user can decide, which information is stored during the measurement and how the number format is structured. Every excavation has its own needs and the software should be able to handle almost every configuration.

Fourthly, the software should be 'intelligent'. The standard procedures, which have to be done during acquisition and archiving (e.g. tagging and storing of images, storing of documents, numbering of finds, etc.), should be done automatically by the software following the user-defined rules mentioned before. For this the software features a kind of artificial intelligence.

The fifth point concerns the data structure and file size. All the data should be stored in one database, not in a separate file system. With the right system of data management and 'compression' this would save server capacity and additional time during the analysis.

Finally, it should stay a 'simple to use software'. That means that users are able to gather and gain all the information they need in an easy way, without studying handbooks and forums for days. You could call it a kind of 'self-explaining software'.

3. Implementation

The following part concerns some aspects of the implementation of ElKowmGIS, also discussing the advantages and disadvantages of the used solutions.

3.1. Platform independency

As already mentioned, there are several Operating systems in use in archaeology. To guarantee the platform independency of the software, we decided to develop a JAVA application. This allows the user to run the program on most platforms available on the market. This includes also UNIX and Mac systems. Also its porting on a mobile device or cell phone is possible. For the implementation the Java Development Kit 1.6 and several libraries (e.g. PostgreSQL JDBC Driver or the JAVA API for XML Processing), are employed.

Another big advantage of JAVA is the possibility to create a jar-file, containing the executable and the necessary libraries. This leads to a simple installation and a high compatibility as no additional installation of libraries into the root path of the JAVA environment is necessary. Also no moves of the libraries have to be done, as with an update of JAVA the root directory changes.

And last but not least, JAVA has only one restriction: the Java Runtime Environment has to be installed on the machine, which should run ELKowmGIS. But as the Runtime Environment is a kind of standard on new machines nowadays, this is a small restriction, every user can accept.

3.2. Modularity

Also in the field of modularity, JAVA is the first choice. Java offers an easy way to create packages, which can be added to a main-routine. The separation of the program allows differentiating between several parts of a program. For the implementation of ELKowmGIS the following separations were conducted:

First of all, the GUI-concerning classes were separated from each other, in order to change the look and feel of the several windows, frames and entry fields separately. This will allow the user, to change the look and feel of the software. This point is mainly interesting for archaeologists working under different climatic conditions (e.g. in a dessert with high sun irradiation the user needs another contrast between fore- and background as a user working in the European region).

Secondly, the classes regarding the SQL-Statements can be separated from all other packages. This guarantees an easy access to the statements, and with this a fast way to correct wrong or add new statements.

Also, the actions triggered by several GUI-elements are separated. This guarantees the creation of standard routines, which are recyclable. So the routines do not have to be written completely new, every time a new button or another GUI-element is added to the software. For archaeologist these separations are rather irrelevant, as they concern only the developing process.

On the other hand, more important is the separation of different modules concerning the documentation

process. At the moment, the following modules are already finished or under construction:

- Database (finished)
- Data acquisition via EDM, GPS or manually
- Visualization of data, plans, etc. (CAD)
- Surface creation and analysis etc (GIS)
- Project entry (finished)
- Image Viewer with integrated management (finished)

In the following sections, the several modules will be presented and discussed. Focus will be placed on the technical implementation of the modules, not on the archaeological system which was used. The archeological system will only be described, when it is needed in a special case.

3.3. Database

For the database module we decided to use the PostgreSQL-Server Version 8.4. This server allows using SQL-Statements, as the name suggests, and is widely spread in the field of geospatial data management. Especially in open source GIS programming, it emerged as a kind of standard database (DUCKE, 2008).

A significant advantage of this database is the possibility to store not only strings and integers; it is also possible to store streams of data. With the help of this possibility, it is feasible to store all forms of data within the database, without the need of an external directory structure. The storage of streams also allows saving the data with only a small amount of space. These two arguments have therefore led us to the decision to use this database, which allows storing all data together within one single database. That means also that all data is accessible through the software solution.

The database itself is divided into two schemes.

One scheme contains 5 tables which store the data and documents:

- Measurements
- Documents
- Dating
- Context
- Excavations

With the help of the PostgreSQL-Server it is possible to store all of the above mentioned information objects. An Export-Module within ELKowmGIS allows the export of the data in different formats for further work in specialized software, such as Adobe Illustrator or Esri's ArcGIS.

For every object the corresponding excavation, owner and context is stored. Therefore the data is connected to the definition tables and to the other objects by a unique identifier, similar to Shannon McPherron's proposition (MCPHERRON and DIBBLE, 2002). This identifier exists of the unique values excavation-id, unit, subunit and id (Figure 2). With the help of this identifier the data of every single object can be found within the database.

The table measurements contains all the information about the recorded points, lines and areas. Therefore it is stored besides the number of the object, to which excavation the record belongs, (including the coordinates, the type of object that was measured, and how the database entry was modified).

The table documents stores all documents in all conceivable formats, including Microsoft Office, OpenOffice, Photos, Videos and Audio-files. The number of the document is saved, also the type of the object, so that the users can decide in advance if they want to open or export the specific document. Additionally, the software also stores the history of changes and field remarks.

For the storage of dating information, a table was created. Hereby the user can decide whether it is a relative or an absolute date. And, as in the two tables before, within the table dating, one can see the information about the excavation, owner, remarks, change-log, and the number of the excavation and dating.

All these tables are directly connected by the unique identifier to the tables excavations and context. For the storage of all excavation-relevant data, the table excavations contains fields for the excavation-id and the excavation-name, the place, the date, the state of the excavation (active, finished, etc.), history, remarks, and additionally how the orientation of the finds (MCPHERRON and DIBBLE, 2007) is recorded for the specific excavation.

In the table context all the metadata of the entries in the tables measurements, documents and dating is recorded. The record includes:

- a code-id, with which EIKowmGIS can separate between points, areas and lines,
- a type-id, which defines the character of the object (find, sample, topographic feature, ...)
- level: to which archaeological (or geological) layer the object belongs
- state: saves the state (burnt, moved, finished, etc.)
- and the fields remarks, owner and history.

The second scheme contains at least 12 tables for the definitions of the objects, the dating, the features, the symbols etc. These tables are predefined, but can be

changed by the user. Here only the excavation-id is used as relation to the tables (Figure 3). We only use this single key as identifier, because it is sufficient to know to which excavation a definition rule belongs. With this method it is ensured, that also rules can be defined, which should be available in all excavation projects.



Figure 2: Scheme of the tables for the data within the database.

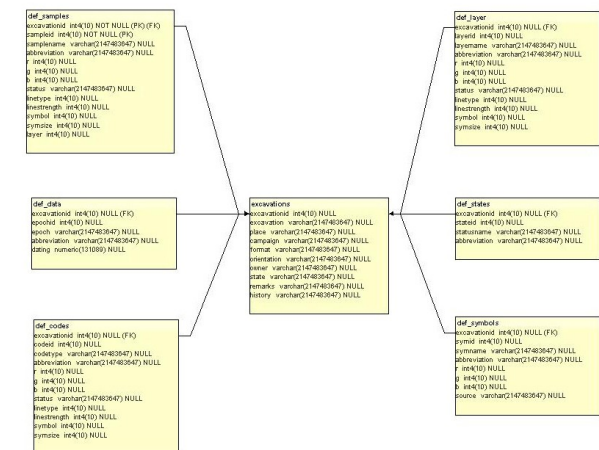


Figure 3: Excerpt from the Scheme of the definition tables.

These definition tables describe in detail, how a single object is represented in the GIS- or CAD-module and do not influence the data itself. This point was the main reason to distinguish between data tables and definition tables. This solution guarantees that no data entry can be overwritten by a definition rule accidentally. It is also possible to change the representation of a whole category at once, by just changing one entry in the database.

Moreover, EIKowmGIS uses the definition tables as templates for new excavations. During the setup of a new project, the user has the possibility to change the several definitions and save the changes as new entries in the definition tables. This guarantees a maximum of self-determination.

3.4. Module Project Entry

The project entry mask is a simple frame in the EIKowmGIS software. Therefore an internal frame (with 6 tabs) was placed on a desktop pane.

The first tab represents the metadata of the project/excavation (id, name, place, campaign, remarks), while the second tab contains a table for the information about the relative or absolute dating (or phases) of the project.

The four remaining tabs concern the several rules of the excavations, which are divided in find, feature, sample, and layer for the CAD- and GIS-modules. These tabs have all the same architecture:

Each tab contains another row of tabs, where the different special rules can be defined (e.g. what abbreviation is used to represent which find, in which color should the object be illustrated, etc.). Therefore the entry mask of all necessary tabs was equalized (Figure 4).

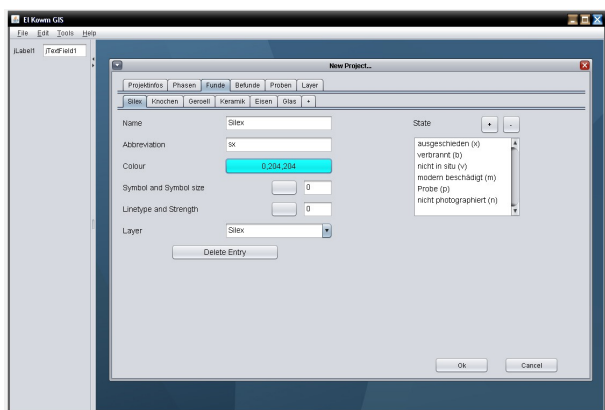


Figure 4: Screenshot of the entry mask of EIKowmGIS.

It contains entry fields for the following rules:

- Name
- Abbreviation
- Color
- Symbol type and size
- Line type and strength
- Layer
- State

All the entry fields are directly linked to the database, so that the entry mask also acts as a template. Already entered information in the database is displayed in the different fields. After the entry of the data and rules, the fields are cross-checked if they are filled-out and then transferred directly to the database. If the user changed a rule, the data in the database is updated. If he/she created a new rule, this rule is added to the database. At

this moment the user will be asked, if the new rule should be available in all projects, or only in the new registered one. If the rule should be available in all projects, the excavation-id is changed into 0, the id for the rules used in all projects. Otherwise the excavation-id is not changed.

3.5. Module Image Viewer

The next module, which should be presented here, is the image viewer with integrated management. Therefore also a single frame is placed on the desktop pane. This frame is divided into two sections (Figure 5).

On the left handside several elements for managing images can be found. This includes fields for tagging the images, changing the preview size of the images and search options.

The fields for the entry of the information are directly connected to the database. This provides an easy-to-use image manager, with the certainty of avoiding spelling errors in the entries. No tag has to be written by hand, only the remarks and the number of the object, surface or crosscut have to be added manually.

On the right handside the several images are shown as small preview images. The images can be imported from every folder (hard disk or external medium). Therefore the images are read with the help of the Java Advanced Image I/O package, provided by Sun Microsystems. With this package it is possible to read images from any device as 'Image'. The reading process is outsourced as a thread. At the moment this method allows to read about 150 images in one single import in a small amount of time (> 1 minute). The software also calculates the right size for the preview image automatically so that it is not necessary to calculate thumbnail images in advance to use in EIKowmGIS.

After the import, the user can select the images, which should have the same tags, choose the tags on the right side of the image manager and press the import-Button. EIKowmGIS then exports the images to the database and stores the images as Streambuffers. It also fills out all necessary fields inside the database tables.

The Image Viewer module forms the search mask for the search inside the database for specific images. Therefore, the user has to activate only a checkbox at the bottom of the data entry area. After the activation of this checkbox, it is possible to use the entry fields as filter for the search. The advantage of such a solution is that the user already knows the User Interface and no other Interface has to be implemented.

As a last feature, the image module offers the possibility to export the images from the database to a folder structure outside the server. If the necessary directories do not already exist, the export function will create them at the place that the user defines.

At the moment, the Image Viewer is not fully implemented in EIKowmGIS, but as a stand-alone

version. However, Image Viewer is fully functional and already connected to the database. The only thing missing is the transformation of the JFrame into a JInternalFrame in order to use it in EIKowmGIS like the other modules.

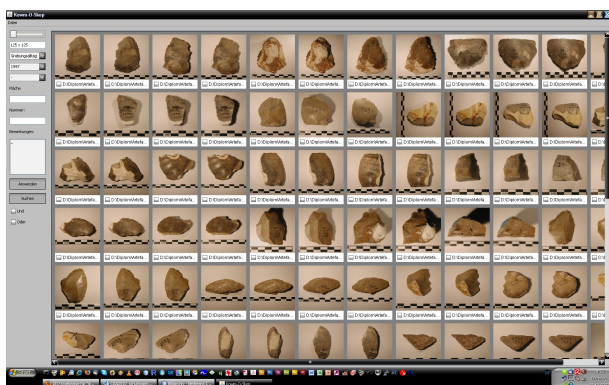


Figure 5: Image viewer and manager (Screenshot of the standalone version).

3.6. Module GIS

The GIS module is the module that is currently under construction. For the implementation, the author decided to utilize the Java GeoTools and the JavaFX package.

Java GeoTools offers routines and tools for the work with geospatial data. The GIS-module GeoTools is particularly suitable, as it is planned to implement different geospatial routines like the presentation of geodata, calculation of surface models, including slope and analyses, calculation of groundwater models (INGEBRITSEN, SANFORD and NEUZIL, 2006), and the calculation of density maps. Also the creation of maps and plans of excavations or single surfaces can be done using the GIS-module. Furthermore, the rectification and/or georeferencing of images, analyses of line of sight, and other analyses a user can think of, will be made possible by EIKowmGIS.

The package JavaFX delivers graphical routines and elements to give the module a nice and uncomplicated look and feel.

For the implementation of the GIS-module also a JInternalFrame is used and placed on the JDesktopPane of the main Window of the presented software. Like any other GIS-software, the navigation in EIKowmGIS is based on either keyboard-commands or on buttons, arranged in menus. The main part of the Internal Frame is the map area, where the data is shown, edited and created. Therefore the Java 2D-Graphics package is utilized. With this package it is possible to use simple predefined graphic routines to illustrate complex geospatial data.

Furthermore mouse navigation is achieved by using MouseListeners from Java, which are adapted to the needs of the GIS-module. For Example, the combination of pressing the CTRL-key and moving the mouse wheel allows the user to zoom in or out the plan. Therefore the

MouseListener was enlarged by a KeyBoardListener, which recognizes which key is actually pressed. By this combination, several actions can be triggered, beside the normal up and down movement, which is implemented by default (e.g. the movement of the mouse wheel causes a ScrollPane to move up or down). With several combinations of Keyboard- and Mouse-actions an easy-to-use interface can be provided. For instance, the movement of the mouse wheel while pressing the Shift-Key causes the map screen to move to the right or the left, depending on the direction of the movement of the wheel.

At the moment, only the first steps of the GIS-module are implemented, including the Internal Frame, the mapping area on a ScrollPane, the display of coordinates and the mouse movements.

Conclusions

B.C. DUCKE and J.M. REEVES (2008) start their conclusion with the argument, that archaeologists normally work under high pressures, due to financial and seasonal restrictions. Therefore an appropriate workflow is needed. It is obvious that the documentation software has to be integrated in this workflow and the software uses open protocols and file formats in order to guarantee the communication of the specific software components. But here a simple question should be allowed: Why do we have to use open protocols and formats, when it is possible to combine the specific components in one application? There is no need to generate a .CSV-file of coordinate-data in order to use it in a GIS-solution, when a program can handle the data from a CAD, a database, and/or a GIS at the same time. For ongoing investigations I agree with Ducke and Reeves, when they promote the use of open file formats, but for the basic workflow, the so-called daily business, it is easier to use only one application for all the work that has to be done.

The next point of interest is the usability and the understanding of a software solution. A program written for archaeologists should follow the workflow on site (e.g. excavation) and in the laboratory. But as there are many different kinds of systems and strategies in archaeology, no software can handle this demand. There has to be a solution, which follows certain guidelines that the user defines, not the programmer who develops the program.

EIKowmGIS could be such a solution. It allows archaeologists to define their own rules, concerning their specific excavation. It also makes it possible to manage several projects with different rules and standards in one single application. Furthermore, the workflow of the excavation is not influenced by the use of EIKowmGIS, but rather the other way round - e.g. the presented documentation system adapts the workflow. The users themselves can define which step has to be taken first, and in which way it should be taken.

Additionally, the software allows to reusing already defined rules and adding new rules to the existing ones.

As a last point, the software opens up different possibilities to standardize the data directly during the measurement. With the presented solution it is possible to implement documentation standards like the CIDOC CRM or any other systems much more easily. This means that it is not necessary to work with the data a second time, only to fit them into a certain standard. Even if the user wants to use another documentation system, the Java-based solution offers a lot of facilities to transform the existing data into another data scheme. The worst case scenario would be to write a small module, which translates one system into another. But with the help of Java, the translation can be automated without the need to control every single entry in the database.

But as the project is still under construction, a lot of work is still to be done. Several modules have to be implemented, such the CAD-module, which allows not only drawing or illustrating the excavation, but the acquisition of data with the help of an EDM or a GPS-device.

Also the GIS-module needs a large amount of time, as there are several functions (e.g. Manipulation of raster data, creation of density maps, slope/aspect calculation, etc.).

Since there are so many possibilities and needs of functions, the possibilities are unlimited. But the advantage of the presented solution is the modular development, which allows appending modules as needed.

References

DUCKE B.C., REEVES J.M., 2008. *The Open Archaeology Research Initiative: Spatial Data Infrastructures and ubiquitous computing*. Oxford Archaeology, Oxford.

INGEBRITSEN S. E., SANFORD W. E., NEUZIL C. E., 2006. *Groundwater in Geologic Processes*. Cambridge University Press, Cambridge.

MCPHERRON S., DIBBLE H. L., 2007. Artifact Orientations from Total Station Proveniences. In: *The World is in your eyes* edited by Alexandra Figueiredo and Gonçalo Velho, pp. 161-166. Tomar: *CAA Portugal*.

MCPHERRON S. P., DIBBLE H. L., 2002. *Using computers in archaeology: a practical guide*. McGraw-Hill/Mayfield, Boston.

SCHUHMANN D., 2011. A three-dimensional model for the palaeolithic site of Hummal (Central Syria). In: *Proceedings of the Symposium 'The Lower and Middle Palaeolithic in the Middle East and Neighbouring Regions'* (in press). Basel, Switzerland.

