SIKB0102: Synchronizing Excavation Data for Preservation and Re-Use

WOUTER BOASSON, RAAP, The Netherlands
RONALD M. VISSER, Saxion University of Applied Sciences, The Netherlands

A key issue in re-using data from excavations is the need to understand the meaning of the contents. Using old datasets can be difficult, for obvious reasons like finding the right data in the first place, understanding unknown codes, and the inherent difficulty of combining data from different excavations. These problems are commonly addressed by archiving and publishing harmonized data, which enables searching through combined datasets, but at the cost of losing important detail. An interchange format for digital archaeological data was clearly needed. The authors played a major role in the drafting group for what would later become the "SIKB0102" interchange standard. The standard focuses on: 1) keeping the original level of detail while providing a harmonized view; 2) serving archiving as well as data interchange in active projects; 3) control of versions; and 4) making sure that relationships, the key to solving archaeological mysteries, are central. An unusually flexible interchange format was created that can hold detailed data together with, and linked to, harmonized data. Having the harmonized data makes it easy to search and combine datasets, while having the related detailed data makes it possible to drill down to the original level of detail. Archaeological data is all about structure and location; therefore, the authors have taken care to include vector geo-location data in the specification as well. Combining all these aspects in one interchange format makes the SIKB0102 specification stand out. In The Netherlands, archaeological research data must be provided to the National Archival Institute (DANS), and the KNA (quality standard for the Dutch archaeology) requires submission of the data to the national archive. Today the KNA requires archaeological excavation data to be provided according to the SIKB0102 specification, which is a big step forward in re-using archaeological excavation data.

Key words:
Exchange, Archival, Excavation, Harmonization, XML.

SDH Reference:
Wouter Boasson & Ronald M. Visser. 2017. SIKB0102: Synchronizing excavation data for preservation and re-use. SDH, 1, 2, 206-224.

DOI: 10.14434/sdh.v1i2.23262

1. INTRODUCTION

During the last decades in the Netherlands, there has been a substantial increase in the number of excavations and excavators, and also an increased use of digital tools in archaeological research

Author's addresses: Wouter Boasson, RAAP Archaeological Consultancy, Leeuwenveldseweg 5b, 1382LV, Weesp, The Netherlands; email: w.boasson@raap.nl. Ronald M. Visser, Saxion University of Applied Sciences, Handelskade 75, 7417DH Deventer, The Netherlands; email: r.m.visser@saxion.nl.
Permission to make digital or hardcopies of part or all of this work is granted without fee according to the open access policy of SDH.
© 2017 SDH Open Access Journal

[Visser et al. 2016]. Archaeological excavations are destructive by nature. This implies the need to properly document the discovered objects, for two reasons: 1) the in situ cultural heritage is destroyed, and 2) it should always be possible to re-interpret the primary data, an important part of scientific research in general. This makes it necessary to document the data in a way that will be understood by current and future generations of archaeologists. However, since the emergence of databases in archaeological research, each archaeological company or institute has developed its own digital systems. When data have to be exchanged, this leads to various, often undocumented, conversions of data. Problems that can arise during these conversions have ranged from fairly easy to solve issues, such as non-matching object and property (table and field) names, through different but easy to understand data-structures, to a complete lack of understanding of the codes or structure. Dutch archaeology is blessed in having DANS (Data Archiving and Networked Services; https://dans.knaw.nl/nl) whose mission is to make all data technically available by converting multiple formats to a few open and future-proof, ASCII text-based formats (e.g. XML and CSV) with inclusion of the requisite metadata.

In the Netherlands, archaeological contractors are required to send the excavation data to at least three different repositories or databases, each with its own rules with regard to formats:

1) All the digital documentation of excavations is required to be sent to the DANS-EASY repository.
2) All the excavated finds, along with both digital and analog documentation, must be delivered to an archaeological depot for permanent storage, including descriptive documentation.
3) The ARCHIS database [Roorda and Wiemer 1992] from the Cultural Heritage Agency of the Netherlands is designed to keep track of indicative data for archaeology, which could be seen as a harmonized summary of the excavation data.

Until a few years ago, each of these institutes had its own regulations and formats for delivering data. DANS was the least demanding: every dataset that seemed to be accompanied by proper metadata was accepted. Archaeological depots had very diverse demands, ranging from almost none to the very specific. Updating ARCHIS required manual data entry. This used to be a very time-consuming and thus expensive process for archaeological contractors.

Clearly, this was not a desirable situation. Providing excavation data in various formats is costly, and there is no defining standard that makes sure that the data will be easily reusable for future research. A common data format to exchange data would be of benefit to the archaeological community, but designing it was hampered owing to the many different requirements. The most interesting part of such an undertaking is to align all the data providers and institutes involved; the more people, the more opinions that have to be harmonized. Nevertheless, the benefits were acknowledged by all institutes and organizations involved. The SIKB (foundation for infrastructure quality control of soil data, including archaeology; www.sikb.org) was invited to manage the development and availability of a data exchange format for the core data that must accompany the physical objects sent to the archaeological repositories, which would solve at least part of the problem. The SIKB had successfully designed a data exchange format targeted at soil characteristics data (SIKB0101 protocol: http://www.sikb.nl/datastandaarden/sikb0101-bodembeheer), and it already safeguarded the Dutch Archaeology Quality Standard [Willems and Brandt 2004;

http://www.sikb.nl/archeologie/richtlijnen/brl-4000]. The SIKB also has the full infrastructure in place to manage standards, both for data exchange as well as other standards, such as formal guidelines (e.g. the Dutch Archaeology Quality Standard, also known as the "KNA"). Both existing Dutch standards formed the starting point for the development of a new exchange standard.

Since 2011, the authors have played an important role in designing the SIKB0102 exchange standard (http://www.sikb.nl/datastandaarden/richtlijnen/sikb0102). The format will be presented in this paper. It started out as a digital "packing slip" when delivering finds for long-term storage in the depots, but during the discussions we realized that this provided a unique opportunity to develop a standard that enables a harmonized exchange of data. The new SIKB0102-format enables the exchange of an archaeological dataset while keeping full detail. The qualities of the format are also reflected in its acception as a national "open standard," which one should comply with unless it is not possible (Forum Standaardisatie: https://www.forumstandaardisatie.nl/standaard/sikb0102).

For the past couple of years it has been obligatory to use the SIKB0102 format when exchanging archaeological data in the Netherlands. However, the format was never presented to an international public, and it is not well known outside the Netherlands. For example, the development of the emerging CRMArchaeo-model (http://www.cidoc-crm.org/crmarchaeo/) would have benefitted from our experience in developing the SIKB0102-format. This paper aims to share the format, its strengths and shortcomings. However, before describing this format, we will first briefly explain the Dutch excavation and documentation method.

2. EXCAVATION DATASETS IN "EXPLODED VIEW"

The typical Dutch excavation dataset reflects the fieldwork methods, which will be briefly described here. Two major different excavating types can be distinguished:

1) Planes and cross-sections for feature-based excavations, such as settlements, or
2) Sieving grids when features are virtually absent, generally for the lithic periods.

A distinction is made between the observations, the analysis and the metadata. Typically, in the analysis phase of a research project, relationships are discovered, and these should be stored. This is a different type of information: interpretations instead of observations, and storing interpretative information should ideally not touch the observations.

This section also includes examples of common implementation strategies.

2.1 Data objects for feature based excavations

The feature-based excavation is the more common, having the following workflow and administrative objects:

- The larger excavations in The Netherlands are usually segmented into *trenches* on the order of meters to several tens of meters wide, and sometimes over 100 meters long.
- Within this trench excavation, planes are prepared for documentation and further inspection. These planes are created based on stratigraphy.

- While digging to the desired level, care is taken to collect *finds*.
- The plane will subsequently be investigated for *features*. The features will be mapped and related to the plane where they were seen.
- The features are cut in half and dug out, to check for *fillings* and finds. A cross-section of a feature is generally drawn and is called a "coupe" in Dutch (Fig. 1).

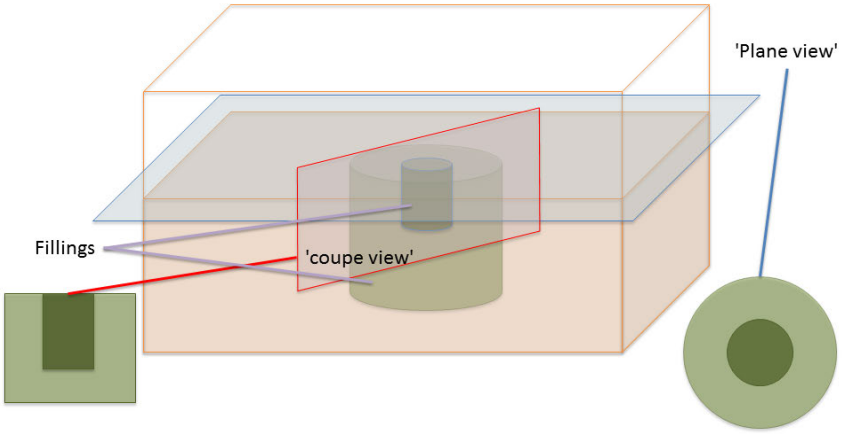

Figure 1. Conceptual view of a Feature (posthole), with two Fillings, with the contours as observed in the Plane as well as in the "coupe."

- Collected finds are linked either to the plane and an administrative subdivision of the plane, or to a feature and possibly a filling. For large features sometimes administrative subdivisions of the feature are made (*segments*).
- The walls of the trench give insight into the stratigraphy; in fact they are *cross sections*, and they are treated much like the excavation plane, since it is the vertical version of a plane. Features and finds observed are documented. If possible, the features are related to features in the excavation plane.
- The entire process could be repeated multiple times when several stratigraphic levels of interest are expected.

The basic (field) *observations* translate to a very basic and common logical set of data objects, mapping 1:1 to the observed objects, as laid out in Fig. 2. Note that not all data objects are present; in general, the interpretative information layer is omitted for clarity, as well as descriptive data about drawings, documents, images, find packaging information and project metadata.

Note that the relationship between Feature and Plane has a many-to-many cardinality, which is logically true: one Feature may be observed in more than one Plane, and multiple Features can be observed in one Plane.

When implementing a data structure, various excavators deal differently with this phenomenon, which is one of the challenging areas when creating a standardized data exchange model. There are two common approaches:

1) Every time the same feature is recognized, it is re-entered as a new Feature in the database, referencing the actual plane where it was observed, but with the same Feature number.
2) Each Feature is only entered once, referencing the first plane where it was observed.

Disadvantages: the first approach creates phantom objects in the database, while the second method lacks the links with other observations of the same feature. This could easily be solved by inserting an extra relation layer that links a feature to a Plane, instead of directly referencing the Plane table, but despite being the solution that best resembles the real-world situation, it is regarded as complex and/or time consuming.

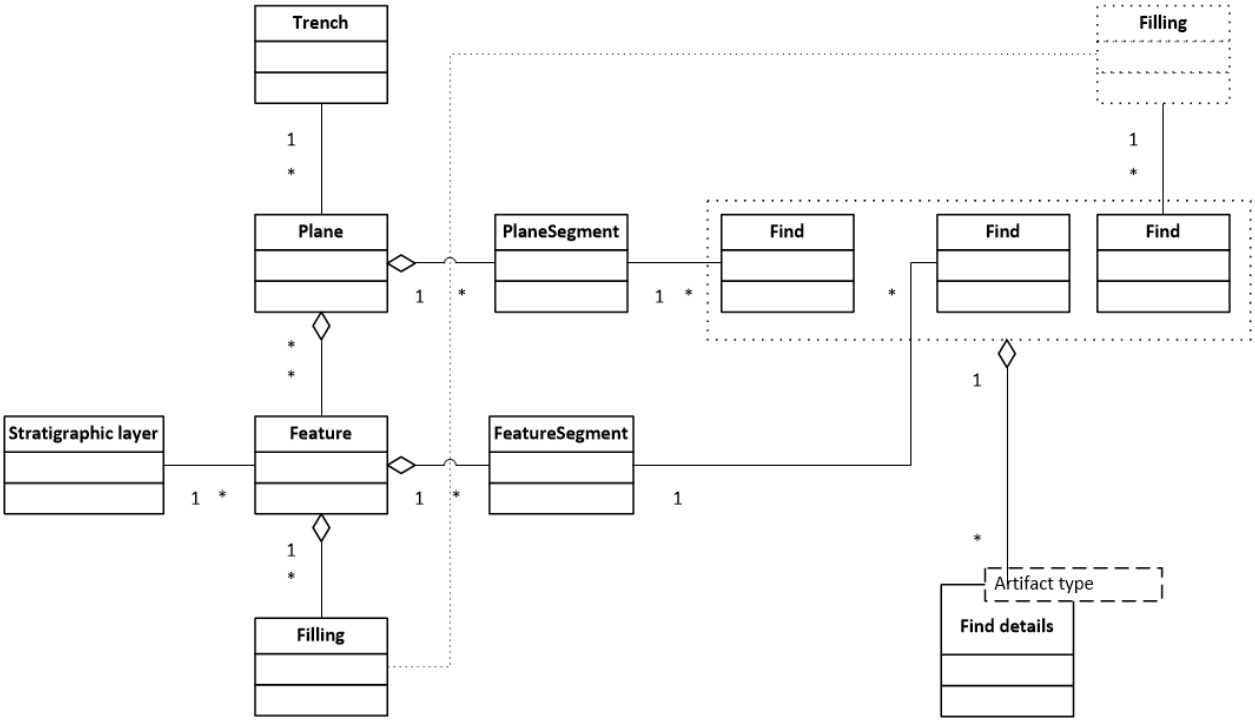

Figure 2. Basic logical data structure of the administrative objects, field observations and artifact details in a Dutch plane/feature based excavation. Note that the three Find objects represent Finds collected in a different way, but they are basically all Find objects, with the only exception that the relationship differs.

Databases are often not correctly normalized to the 3NF [Codd 1972] or BCNF form [Codd 1974], possibly a left-over from paper-based documentation, where providing redundant information is of great help, not to say necessary, during the analysis phase of a project. Examples:

- With each Find not only the feature number is registered, but also the Trench and the Plane, which is redundant as long as a feature object can be uniquely referenced. The feature itself will reference the Plane.
- Composed keys (e.g. a Find number) are often created by separating the different parts of the key with a . (dot), where they are actually different properties of the object. E.g., Find one may reference feature 2.5.7 (which is Feature 7, observed in Plane 5 in Trench 2).

Both examples are considered bad practice as they are error prone (contradictory information) and difficult to manage in a database.

2.2 Sieving grids

In this type of excavation, the digging is restricted to the removal of the topsoil. Within our layer of interest, primarily the soil is lifted in small quadrants and sieved to detect (small) finds. The major difference with the plane/feature is that when sieving a grid, the plane is subdivided into small segments (e.g. 0.5m x 0.5m), and the finds are linked to these small segments as well as to a layer (Z-level), not to be confused with the stratigraphic layer. Fig. 3 shows the data objects and their relations; here the interpretative layer as well as various metadata objects are omitted for clarity.

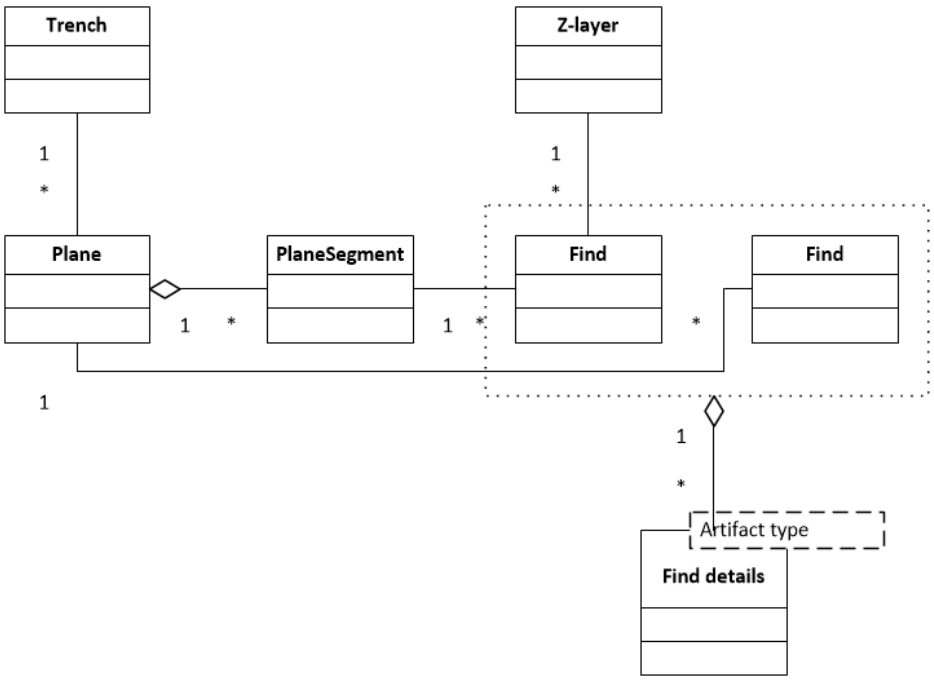

Figure 3. Logical data objects matching the field observations in a grid-wise filtering excavation.

2.3 The analysis layer

New insights acquired by analyzing the excavation data must be stored in the database as well. It is important to separate interpretation from observation, as for re-use the observed data must be available for re-interpretation.

The results of analyzing the observations could be stored in the form of setting additional properties of already existing objects, e.g., a concluding age for a specific Feature, based on find information. It could also lead to new data objects, the most notable example being the structures (e.g., a set of postholes making up a building). This is all additional information on top of the observed objects.

- Structures: each structure should be represented by a data object, which can be linked to multiple features.
- Stratigraphic layers: Features (and finds) are linked to a stratigraphic layer; the link must be observed in the field. But often there are uncertainties, especially when linking stratigraphic layers between geographically separated locations (separated Trenches). Errors are usually detected and corrected later, and in an ideal dataset we should have the initial observation separated from the final interpretation. We should have an object representing a stratigraphic layer, which can be referenced from the features.
- Detailed find descriptions: additional information, usually stored in objects with a data structure matching the properties of interest. They link to the Finds, in order to have them coupled to the correct Features and Structures.
- Dating: Finds, Features and Structures are usually given an age or date. Most often this is bi-directional, derived from the supposed stratigraphic layer in combination with the artifact characteristics. This information is added to the respective objects.

2.4 Geodata

The geodata provides information on where the features and finds were located. The level (height information) of the Planes is measured, to create a digital elevation model (DEM) of the historic site. The contours of Features are registered as well. The locations of the finds are not always registered, in which case they can be located by linking them to a Feature when found in a sufficiently spatially defining Feature (small Features, e.g., postholes).

Data collection is most often done using a GPS or RTS (Robotic Total Station) set, but the measurements are taken in a way that does not deliver a full 3D dataset. Geometry data collection is aimed at creating a 2D-documentation of planes and cross-sections instead of 3D volumes.[1] Therefore it is often not possible to derive relationships between Finds and Features from the geodata. This is solved by always registering these relationships in the database.

Excavations datasets usually have their geodata stored in geodata files, separated from the object descriptions (Features, Finds, etc.), and therefore a formal link is impossible. Common practice is the

[1] Digitizing the contour of a Feature in a plane does give the Z-component of each measuring point, but a Feature is not registered as a volume.

use of naming conventions, but they are by no means standardized. For example, the geodata file "project_t1_p1_ft" probably contains the features in plane one, trench one, but it is also possible that all features were combined, where the geodata table has additional attribute columns for trench and plane number.

2.5 Metadata

To be able to understand the data, additional information on the data and documents is required, including code books (thesauri), relations between different data files, and descriptions of documents (e.g. reports and images) that are part of the dataset. General information may also be of value for the discovery of datasets and judging the quality.

From a technical point of view within the context of this paper, the metadata is not very interesting, as there are already many standards (such as the well-known Dublin Core; http://dublincore.org). However, the relations between various files containing tabular data is of high importance for linking the data. The code books used in a dataset are also of crucial importance for understanding the data, as the stored information is in many cases somehow coded for the sake of size and search options, and not having proper code books renders a dataset useless.

3. SIKB0102

The SIKB0102 specification is an attempt to make archaeological research data exchangeable by using an exchange format that accommodates all relevant information from any Dutch excavation. In order to accomplish this, it should be possible to map common data structures in use at excavations to the exchange specification. This leads to a basic functional specification:

- The various object types that should be included are determined by the requirements for carrying out an excavation. These are described in the KNA standard (NL: "bouwstenen referentie," EN: "building blocks reference"). Also included must be the objects needed to contain the information required by the archaeological repositories.
- The dataset should include observations and interpretations.
- The required properties were determined by the absolute minimum requirements to get a basic idea of observed objects at the excavation site, together with enough information to handle the objects handed over to the archaeological repository. In plain language: the minimum set of attributes that is always present, even in a worst case excavation scenario.
- All possible relationships between the various objects (regardless of type) must be included.
- Common code books must be used.
- Metadata documenting the project, images, documents and additional data files (e.g., geodata files) must be included.

To add the option of exchanging an excavation dataset in full detail, the following requirements have to be met too:

- It should be possible to add additional attributes to any predefined (standard) data object.
- Original terms should be included for enhanced detail where the default code books fall short. The original terms must be provided with a description.
- It should be possible to integrate spatial data.

It would be useful to have the option to exchange data multiple times, and update, remove or insert records. This would be especially helpful when exchanging data with other researchers during the lifetime of a research project.

3.1 Mapping original excavation datasets onto a generic structure

In the case of the Dutch excavation datasets, the (data) objects that should be present in an exchange format are known. Difficulties arise when trying to map the various methods and data structures in use for the Trench, Plane, Feature, Filling, Segment and Find objects.

There are multiple issues involved that together form the challenge of designing a proper data structure (objects and relations).

Structure challenges:

- Not all object types are always present (often Segments are left out).
- Various solutions are in use for linking Finds with their containing objects (Feature, Plane, sieve grid), and constructions typically become complex because different Find collection strategies are used . For example, some implementations use different Find tables for each collection strategy (Fig. 2), while others introduce dummy objects to keep the relational integrity or just plain ignore errors in the presumed database constraints.

Key/numbering issues:

- Numbers could even have a meaning (such as that Filling [0] represents the intersection between the excavation Plane and the Feature, at its first occurrence), or, because Finds must be linked for relational integrity in the database, a dummy Feature [999999] exists, which represents finds collected in the topsoil dumping area beside the Trench.
- Primary keys could be business keys or pseudo keys (an id assigned by the database).
- Composed primary keys are not always present as such in the database; they could be just strung together, separated by a dot.
- Some data, notably the spatial data, is typically stored in separate files. Here the direct link is missing, makig it necessary to describe the relationships in the metadata.

Relationships and stratigraphy:

- Not every relationship can be foreseen, unless for each and every possible relationship a link object exists. This is unwanted, as the number of object link tables explodes. Given the goal of accommodating every single excavation dataset, it should be possible to create any link.

An example of a commonly overlooked candidate for a relationship: two Features may be linked when they appeared to be the same (separating observation and interpretation).

- Stratigraphic relationships are extremely important. These are usually relationships between Features, but also Structures may be stratigraphically related, if desired. A provision must be made to accommodate these relationships.

In an exchange format, these issues must be addressed in such a way that a source dataset can be converted to the exchange format, with no loss of detail.

Solutions

The structural issues require first of all a conceptual and coherent approach, as the multiple issues are somehow interrelated. The conceptual approach is necessary here, as in the excavation database many object-identifying numbers, often used as (primary and/or foreign) keys in a database, also have a meaning. As relationships differ among the various database implementations and the meaning of the numbers as well, the data exchange format must be agnostic to these differences, but nevertheless be uni-interpretable.

1) Relationships in the exchange format must use internal object identifiers to create the links between the various data objects. The original keys of the dataset may be left out, as long as they are not necessary in order to link to external files or documents.

2) The flexibility required by the different solutions of the researchers with regard to Feature relationships was solved by introducing the concept of an "Observation." This is a required object, in fact an extra link layer, which makes explicit which intersection with a Feature is described (which Plane, and optionally which Filling; the method has to be added too, in order to distinguish between making a "coupe" (Fig. 1), and a horizontal or vertical Plane view). This allows for relating the single instance Feature model to one or more Planes, as well as multiple instances of Feature models (of actually the same Feature) observed on multiple planes. See Fig. 4.

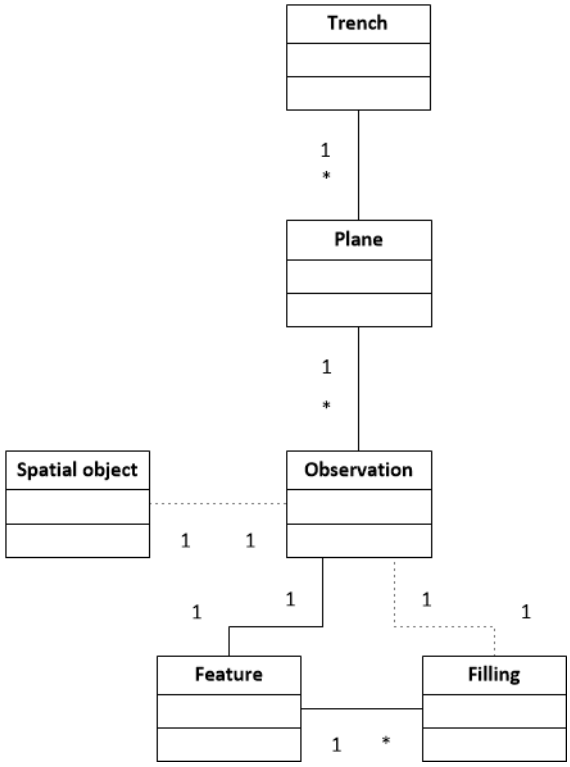

Figure 4. The observation layer, which describes the observation method for a Feature contour, and allows for linking a Feature with multiple Planes (horizontal and vertical).

3) The complexity of linking Finds to the context where they were encountered, which can be of different types (see Fig. 2 and Fig. 3), is solved by introducing a second link layer. The Find itself connects to the Find context, which is a link object designed to link with objects of different types; a spatial object can always be included. This context link object can link directly to, among others, a Feature or a Plane, to the Observation layer in order to more precisely describe the context, or to a Segment. This cleans up the Find objects, and optionally allows for re-use of the relationships in case many Finds were encountered in the same context, which compacts the dataset.

4) Geo-information should be fully integrated and linked to the appropriate (descriptive) data objects, while validating links to external files is hardly possible. All data objects that could have a related geo-data object may reference a chunk of GML (Geography Markup Language, http://www.opengeospatial.org/standards/gml), a broadly supported XML standard for exchanging geo-data.

5) The most common relationships are part of the structure of the SIKB0102 formatted dataset, which facilitates validating, and hints to the user that the relationhip should be made. Besides this, the SIKB0102 format includes a “relation store," where any object can be related to any other object, in a structured way. Each relation must have a code designating the type

of relationship. The standard code list is currently targeted at stratigraphic relations, and one generic type, but could easily be expanded by allowing more values.

3.2 Custom attributes

Archaeology is science, with the invariable diversity in registered attributes, depending on a specific research subject. It is possible to include many common attributes, but it would be impossible to be complete. In addition, it was considered not acceptable to force excavators to rewrite and translate their entire coding system, especially given that many excavations that still must be exchanged using the SIKB0102 format are already in a finished state.

Solution

Part of the SIKB0102 specification is an additional "Attribute store," that allows for adding custom attributes in a structured way, including metadata. The basic principle is to create a data object representing the attribute, which specifies:

- to which standard data type the attribute belongs (e.g. a Find).
- a description of the content.
- data type.

Each instance of a data object that has extra attributes specified may add an attribute value object, containing the link to the data object as well as to the attribute definition.

3.3 Code book usage and original terms

The typical code book (thesaurus) is incomplete. From a scientific point of view, it cannot be complete by definition, as new insights are gained during research. However, a common code book is needed in an interchange format, as this is the only way to create a common picture and make datasets uniformly searchable. On the other hand, a dataset would be of much more value when the original terms used by the excavator, including the descriptions, were available too.

In Dutch archaeology, a standard code book exists (https://cultureelerfgoed.nl/dossiers/archis-30/archeologisch-basisregister-plus), but it is widely acknowledged that it lacks detail.

Solution

The SIKB0102 exchange format stands out, as it allows for adding an extra information layer with codes (the *code reference list*) for each property that has a standard code book. This enables a harmonized conversion and exchange of code books or thesauri. The additional information layer links original codes with the required code book, and provides room for a description, and it also must be used when providing codes for custom attributes.

3.4 Handling updates: versions

We have to look at different aspects of versions:

- Updates to the code books: it is necessary to be able to handle changes, as the national standard code books change every year.
- Updates to the datasets: when working together in a project (e.g. a consortium of excavators, subcontractors) it would be nice to be able to send updates of a dataset, so that instead of fully replacing it, only updated, deleted or inserted records would be processed.
- Updates to the user provided code reference lists: with changing standard codebooks, the links with the original terms must also be updated; this has to be done by the data provider, the dataset constructor.

Solutions

For each of the above-mentioned types of updates, a mechanism is in place in the exchange format.

Code book updates are being handled by a classic mechanism where every code is annotated with:

- a version number (x.y.z)
- date of version
- state

In this way it is possible to check which codes are not valid anymore (withdrawn), and which ones are valid. Only valid codes may be used during exchange, but values cannot be updated. This is sufficient for many purposes, especially when the code represents the state at a specific moment in time.

Dataset updates are handled by assigning Universally Unique Identifiers to each data object exchanged in the SIKB0102 format (https://en.wikipedia.org/wiki/Universally_unique_identifier). They should be considered stable and unique, so that, e.g., Feature 1 of Project X by contractor Y always has the same UUID, but no Feature 1 of any other project should have the same UUID. These identifiers can in that case double-act as identifiers for linking data (within a SIKB0102 dataset), as well for handling updates. Updates can be handled by checking for changes, deletions and new identifiers in a dataset. Note that this eliminates the problem of renumbering keys in the original dataset; they are replaced by the UUID in the SIKB0102 dataset.

The *code reference list* includes a property to reference a previous version of the code reference [Boasson and Boasson 2010]. This is an enhanced form of updating that makes it possible to actually replace records with new versions, and also to look back at previous versions. In the SIKB0102 format, this allows for updates to original terms, in combination with changes to the related standard code books.

4. SIKB0102 IMPLEMENTATION

The SIKB0102 exchange format is implemented in XML. XML is a well-known standard for data exchange, and provides several advantages over any other option:

- ASCII text: durable; readability is not depending on a particular piece of software.

- XML offers good options for validation using the XML Schema Document (XSD).
- Incorporation of other standards, most notable GML for spatial data, is easy.
- A well-known standard helps in getting the (SIKB0102) standard accepted.
- Other similar standards (e.g. SIKB0101) were already implemented in XML.

However, one should be aware of the pitfalls of using XML, which was in the beginning of the process a major discussion point. These have been overcome. The most notable issues and the chosen solutions are outlined below.

4.1 General object structure

Traditionally, relations in XML are created by creating an object hierarchy. In archaeology, where there are many relationships, and many relations to and from the same objects, this would create multiplication of complete data records. To clearly demonstrate the issue, imagine the following. An image was taken of a Find, so the image data has to be included in the Find object. The Find record itself must be included in the Feature object, but then includes the image as well. The Feature record should be included in the Plane record, etc. Then there is another set of objects that includes Feature objects: the Structures. The entire Feature object must be included in that case, too. Displayed in a simplified structured way, it looks like this (note the repetition):

```
Trench 1
 Plane 1
 Feature 1
  Find 1
  Image 1
Structure 1
 Feature 1
  Find 1
  Image 1
```

This is classic XML. Normally, this is technically easy to navigate, but size matters: the file size would easily exceed 100MB, and then it quickly becomes too big to handle with a classic XML document parser which fully loads the data.

The solution was to use a relational database-like approach. Every object occurs only once, and is uniquely identified. XML offers a good construct to accomplish this, by combining the “unique” constraint with the “key” option and the “keyref” construct. Effectively an XML document becomes a relational database in this way. Each and every object has an id (UUID) attribute (in the SIKB format called: “sikb:id”), with a unique constraint on all objects. A key is defined on every object type, and the keyref creates the references. See the code example in Fig. 5.

```
<!-- unique constraint -->
<unique name="uniqueId">
 <selector xpath="sikb:*"/>
 <field xpath="@sikb:id"/>
</unique>

<!-- key on the Feature objects ('spoor' in dutch) -->
<key name="spoorKey">
 <selector xpath="sikb:spoor"/>
 <field xpath="@sikb:id"/>
</key>

<!-- keyref linking a Filling object ('vulling' in dutch) to a Feature object
-->
<keyref name="vullingSpoorRef" refer="sikb:spoorKey">
 <selector xpath="sikb:vulling"/>
 <field xpath="sikb:spoorId"/>
</keyref>
```

Figure 5. Overview of the constructs to effectively turn XML into a relational database.

The keys act as primary keys, and are implemented as attributes in the data objects; the foreign keys are stored within an element. For an example, see Fig. 6.

```
<sikb:spoor bronId="spoor:spoor:1" sikb:id="3041d5e6-1ee2-4f31-aaf6-
12fa3dd26892">
 <sikb:naam>spoor 1</sikb:naam>
 <sikb:grondspoortype>PAALKUIL</sikb:grondspoortype>
 <sikb:diepte uom="cm">20</sikb:diepte>
</sikb:spoor>

<sikb:vulling bronId="vulling:spoor/vulling:10/0" sikb:id="fd7e8d0e-7597-
4597-b248-087f2746db9c">
 <sikb:naam>vulling 10/0</sikb:naam>
 <sikb:spoorId>3041d5e6-1ee2-4f31-aaf6-12fa3dd26892</sikb:spoorId>
 <sikb:kleur>DY</sikb:kleur>
 <sikb:textuur>Zs2</sikb:textuur>
</sikb:vulling>
```

Figure 6. Example of a relation: Filling (vulling) points to Feature (spoor). Blue =key, green =foreign key or reference.

Note that the UUID data type does not exist in XML, so it is implemented as a string, and constrained with a regular expression (Fig. 7):

```
<restriction base="string">
 <pattern value="[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
[a-fA-F0-9]{12}" />
 </restriction>
```

Figure 7. String constraint to make sure UUs are used (as far as possible).

4.2 Code books

The code books for archaeology contain thousands of codes. This is hard to handle, as classic XSD constraints in the XML, the XSD explodes, and there is not much controlled room for adding additional information, such as a validity date and state. For this reason, the code books are stored in a regular XML data file, accompanying the XSD structure document.

4.3 Context links of Finds

As explained above, Finds are linked to intermediate link layer objects, the "Find contexts." These contexts must link to one of several object types (e.g. a Plane (Segment), or a Feature), and additionally a spatial object must be linkable too. XML offers an elegant way of handling this, using a combination of the aforementioned "keyref" constructs, in combination with the "choice" element. The choice element effectively says: one of the following elements must be present ("element" is the official XML term for what often is called a property, attribute, field or column name; in XML an attribute has a special meaning).

In SIKB0102 this is implemented as follows (Fig. 8):

```
<complexType name="VondstcontextType">
<annotation>
 <documentation>Relatieobject om de context van vondsten en monsters te
duiden.</documentation>
</annotation>
<complexContent>
 <extension base="sikb:BasisLocatieType">
      <sequence>
       <element name="contexttype" type="sikb:CodeType" minOccurs="1"
maxOccurs="1"></element>
       <element name="stortId" type="sikb:uuid" minOccurs="0"
maxOccurs="1"></element>
       <choice minOccurs="0" maxOccurs="1">
            <element name="planumId" type="sikb:uuid" minOccurs="1"
maxOccurs="1"></element>
```

```
            <element name="spoorId" type="sikb:uuid" minOccurs="1"
maxOccurs="1"></element>
            <element name="vullingId" type="sikb:uuid" minOccurs="1"
maxOccurs="1"></element>
    ...
       </choice>
      </sequence>
 </extension>
</complexContent>
</complexType>
```

Figure 8. Partial specification of the Find context object: the "choice" element makes it possible to enforce the rule that one of the enclosed elements is present, in this (stripped) example a reference to a Plane, Feature or Filling. A keyref relation constraint may be set on all of these elements, whereby only the existing elements will be validated; missing elements do not cause errors. Note that the type has an extension base: in a "master" object definition, the general attributes (key) and elements for referencing spatial objects are present.

4.4 Original terms - code reference

The original terms used in an excavation can optionally be sent alongside the harmonized code. Harmonized exchange is necessary to enable searching, but the details are relevant for interpretation. Technically, the original terms are in separate data objects, which can be re-used whenever relevant, to avoid duplicates. As already mentioned, they are versioned in order to cope with updates. An example of a record with a standard code together with the original term is shown in Fig. 9.

```
<sikb:vondst bronId="artefact:artefact:13" sikb:id="f400e32b-e4f8-49d2-9276-
857d6feb17e7">
 <sikb:naam>vondst 6.13</sikb:naam>
 <sikb:veldvondstId>f64402f8-08fd-4ce6-8512-6617399de305</sikb:veldvondstId>
 <sikb:aantal>1</sikb:aantal>
 <sikb:gewicht uom="g">33</sikb:gewicht>
 <sikb:materiaalcategorieKER</sikb:materiaalcategorie>
 <sikb:artefacttype codereferentieId="5f6b3a2c-6ee8-56b0-bb19-
a9f0655f58f0">STGL</sikb:artefacttype>
 <sikb:beginperiode>MELB</sikb:beginperiode>
 <sikb:eindperiode>NTV</sikb:eindperiode>
 <sikb:geconserveerd>false</sikb:geconserveerd>
 <sikb:exposabel>false</sikb:exposabel>
 <sikb:gedeselecteerd>false</sikb:gedeselecteerd>
 <sikb:verpakkingseenheidId>36c644f0-e0de-486c-b2d7-
734759e3c77a</sikb:verpakkingseenheidId>
</sikb:vondst>
```

```
<sikb:codereferentie sikb:id="5f6b3a2c-6ee8-56b0-bb19-a9f0655f58f0">
 <sikb:bronCode>s2.kan</sikb:bronCode>
 <sikb:bronOmschrijving>steengoed (met
glazuur/engobe)</sikb:bronOmschrijving>
 <sikb:bronCodelijst>artefacttype_codes</sikb:bronCodelijst>
 <sikb:standaardCode>STGL</sikb:standaardCode>
 <sikb:naamCodelijst>ArtefacttypeValueType</sikb:naamCodelijst>
</sikb:codereferentie>
```

Figure 9. Example of a Find where another code book was used by the excavator. The standard code is present in the Find (‘‘vondst”) record (‘‘artefacttype”), and the original term is referenced using the ‘‘codereferentieId’’ (in blue).

5. DISCUSSION OF KNOWN ISSUES AND QUALITIES

The SIKB0102 format is not perfect. Although great care has been taken to make the XSD as specific as possible and to check as many relationships as possible, there are still many holes in the specification. From the point of data integrity issues, there are still multiple problems.

1. The most obvious problem is that links with external data files cannot be enforced, but knowing how to link them is a necessity for using the full dataset. External data files typically consist of geodata files, and possibly of specialized information that is not part of the required elements. By agreement, it is still allowed to deliver this in external files to the archaeological repositories.
2. The many-to-many nature of the Observation object makes it impossible by design to verify if the required relationship between a Feature and the Plane is in place (the Observation refers to these objects).
3. There is a logical bug in the Codereferences section: with custom attributes, it is impossible to link to a standard code, but a standard code is required (a dummy could be inserted).

Given the additional constraints on integrity and the option to add external (data) files, a full integrity check can only be performed using dedicated software.

Unfortunately, it is impossible to check whether all the data is included in the XML data exchange file. Conversion is certainly possible, but not necessarily easy.

Although the SIKB0102 format should be sufficient to deliver only one dataset for all required institutes, the national Archis database requires a slightly different organization of Find location information, and requires the notion of an “Archaeological Complex” type, which does not exist in the current format.

Reading and querying the XML data file. Even though the relational database approach reduces file size dramatically, the use of XML has one major drawback—its size. Using standard XML querying techniques such as XPath is virtually impossible on larger datasets, and the limits will be reached even sooner when the spatial data is integrated as GML. On the other hand, it is still easy to read

using a streaming parser, which requires programming. This will slow down querying, but is not an issue when the data has to be imported into another system, which is mostly the case.

6. CONCLUSION

The various institutes involved are in the middle of developing software to actually start using the format; it is acknowledged as a usable format by those who have started using it. Still, many excavators have to implement a solution to deliver data in SIKB0102 format, but they will be forced to do so by the archaeological repositories, which already, or in the near future, will not accept data in other formats anymore. There is a certain risk of sending less information to the archaeological repositories, as it takes more effort to convert all the data into the SIKB0102 format then simply send the excavation data "as is," the regular practice in the past.

Recommendations:

- To improve the re-usability of the datasets, the SIKB0102 exchange format needs better specifications on how to make sure a future user can establish links with external data files.
- Make the necessary adjustments to improve the link with the Archis database.
- Put in place smart checks on the contents, in order to provide warnings for the receiving institute when the data seems incomplete.

Altogether, the standard can be seen as a success, not in the least because it has been adopted as an open standard, because it is well documented, in the progress of adoption, and well maintained.

7. REFERENCES

Erik Boasson & Wouter Boasson. 2010. Data integration technology in cultural research. In *Proceedings of the 15th International Conference on Cultural Heritage and New Technologies*. Vienna: Museen der Stadt Wien – Stadtarchäologie, 459-469.

E. F. Codd. 1972. Further normalization of the data base relational model. In R. Rustin, ed. *Data Base Systems, Courant Institute Computer Science Symposia Series*, vol. 6. Englewood Cliffs, N. J.: Prentice-Hall, 66-98.

E. F. Codd. 1974. Recent investigations in relational database systems. In *Proc. IFIP* 74. Amsterdam: NorthHolland, 33-36.

Iepie Roorda and Ronald Wiemer. 1992. The ARCHIS project: towards a new national archaeological record in the Netherlands. In C. U. Larsen, ed. *Sites and monuments: national archaelogical records*. Copenhagen: National Museum of Denmark, 117–122.

R. M. Visser et al. 2016. Teaching digital archaeology digitally. In Stefano Campana et al. , eds. CAA 2015. *Keep the Revolution Going. Proceedings of the 43rd Annual Conference on Computer Applications and Quantitative Methods in Archaeology*. Oxford: Archaeopress, 11–15.

W.J.H Willems and R.W. Brandt. 2004. Dutch Archaeology Quality Standard, Den Haag: Rijksinspectie voor de Archeologie.

Received March 2017; revised July 2017; accepted August 2017.