Importing legacy data into Arches – Best Practices

This document describes how project specific legacy data can be imported into Arches. The examples are based on a use case that is currently being implemented at the i3mainz – Institute for Spatial Information and Surveying Technology: In course of a joint research project the i3mainz and the Römisch-Germanisches Zentralmuseum Mainz (RGZM) work on an archaeological information system serving for complex landscape-based settlement structure analyses. In the past decade a series of heterogeneous datasets of the West-Central European Neolithic have been generated through numerous MA-theses, dissertations and research projects. Given this background this document addresses the tasks arising when integrating legacy data and the models they are based upon into Arches, rather than the task to build up an inventory from scratch.

Extending Arches to import project specific legacy data consists of the following steps:

1. Identifying which Resource Types (graphs) are needed
2. Identifying the entities (nodes) within these graphs, the legacy data shall be mapped to and missing nodes that are necessary to map the legacy data
3. Extending the graph(s) by the necessary nodes
4. Defining Controlled Vocabularies for these entities (nodes) and the ones that make sense in the future (and appear in the GUI)
5. Adapting the GUI
6. Transforming the legacy data into Arches import format (Mapping)
7. Importing Data

To extend Arches, a new package can be defined (that can be based on the CDS package – simply copy packages/cds).

The following Arches build scripts need to be run (source) upon modifications above:

1. build/install_arches_db.sh: clears DB and resets Arches
2. build/build.sh: (re-)builds JavaScript code
3. build/install_packages.sh: imports graphs, authority files, legacy data and JS code from package and rebuilds JS code

While executing these scripts the development server should be running (in a different shell window):

- source /arches-web/runserver.sh

1. Identifying Resource Types (Arches Graphs)

A specific use case might not need all resource types that are defined in the Arches CDS package by default. Therefore Resource Types that Arches should load can be set in:

- /arches-web/archesproject/settings.py
- /arches-web/archesproject/packages/[package-name]/settings.py (overwrites the above)

```
LIMIT_ENTITY_TYPES_TO_LOAD = None #(
    # override this setting in your packages settings.py file
```

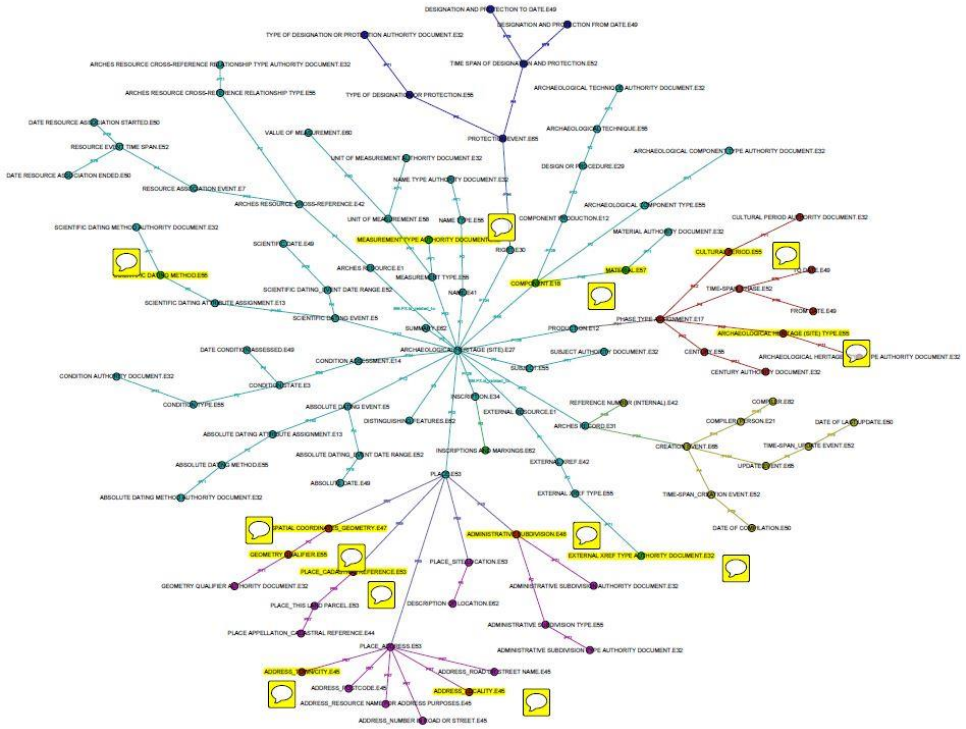
```
#     'ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18',
#)
```

Note: We experienced a bug in Arches 2 when limiting resource types! https://groups.google.com/forum/?hl=en&fromgroups=#!searchin/archesproject/modified/archesproject/f4Fl32NiGyE/G9tEdw8g2FkJ

In our exemplary use case we only need the Resource Type: ARCHAEOLOGICAL HERITAGE (SITE).E27

2. Identifying entities (graph nodes)

Based on the attributes in the legacy data the required entities (nodes) within the identified Resource Type(s) can be identified. This information is later needed to map the data onto the graph.



ARCHAEOLOGICAL HERITAGE (SITE).E27 graph: Nodes to map legacy data (exemplary Fetsch dataset) are marked in yellow.

3. Extending Resource Types

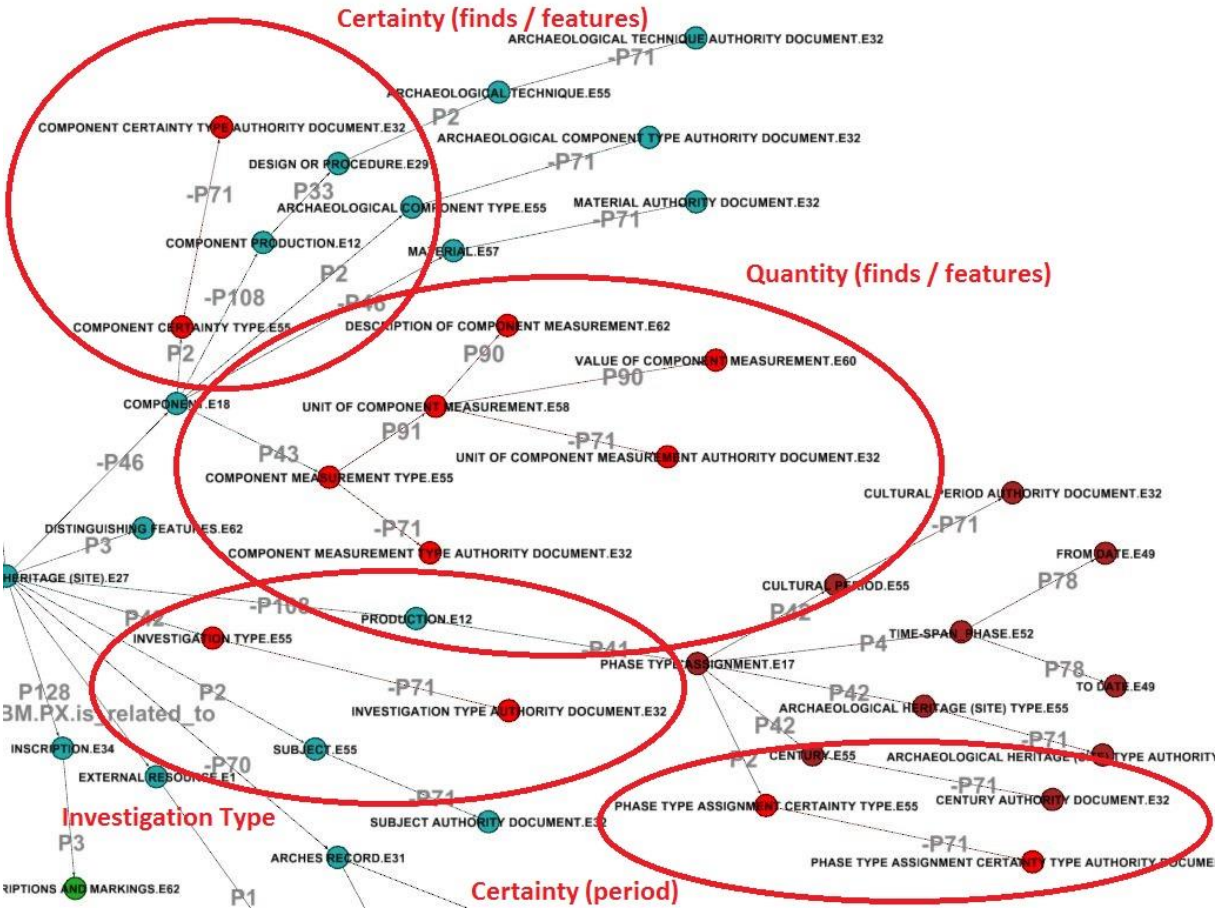
Incase not all attributes of the legacy data can be mapped to the graph(s), the graph(s) can also be extended by further nodes.

In the example use case the ARCHAEOLOGICAL HERITAGE (SITE).E27 graph is extended to represent the following information:

- The certainty that a site belongs to a particular period.
- The certainty that a find or feature was found or identified at a particular site.
- The quantity of finds or features that were identified at a particular site.

Without digging too deep into CIDOC modelling, this document pursues a relatively simple model to keep a balance between detail and feasibility: The proposed model defines (un-)certainty as a type (E55) that is attached to a component (find or feature) via “has type” (P2). The quantity of identified components is modelled in form of a measurement (E16) that has a type (E55) and a unit (E58). Values can be stored, both numerical and descriptive (many, multiple, etc.).
In addition, the integrated data comprises the information, how a site was investigated. By default Arches provides a proper resource type (INVESTIGATION.E7) to manage such information. However, if the legacy data does not possess more detailed information on the investigation the use of a proper resource type can mean a large overhead and impractical utilization for the user (creation of investigation entity and its linkage to a site). Instead this document proposes to further extend the graph ARCHAEOLOGICAL HERITAGE (SITE).E27 by an investigation type (E55). This way a user can directly indicate the investigation type to a site.

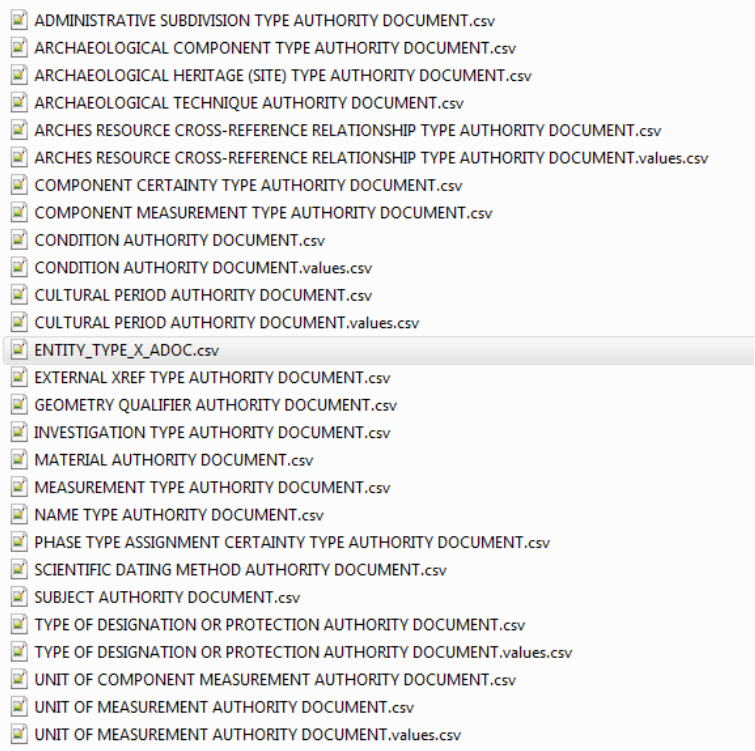
Note: The described CRM modelling should be improved regarding the following comments: https://groups.google.com/d/msg/archesproject/J61-gPSAyVQ/EqejLFGth24J



Technically the graph can be extended using Gephi. In the “Data Laboratory”, nodes and edges can be exported into .csv. The Gephi project file and .csv outputs can be found in resource_graphs.

4. Defining Controlled Vocabularies (Authority Documents)

For all entities within the used graphs, controlled vocabularies need to be defined in Authority Documents. For the exemplary use case these were the following (see also the included authority_files):



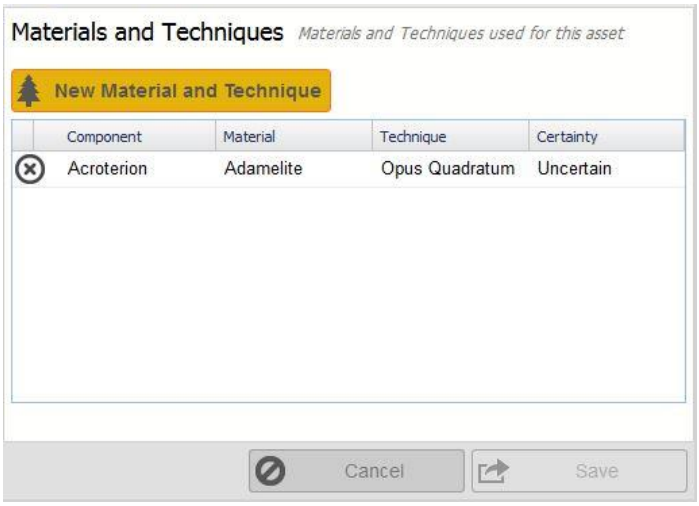
5. Adapting the GUI

In order to make any extended entities and their values available to the user in the graphical interface, the JavaScript forms and their underlying models need to be modified. This step is illustrated by extending the GUI to present the values from the COMPONENT CERTAINTY TYPE AUTHORITY DOCUMENT only.

Three components in the GUI need to be adapted:

- Editing forms (below)
- Report forms
- Advanced Search

Editing forms



To add the COMPONENT CERTAINTY TYPE.E55 in an extra column to the Material and Techniques dialog (above), the following JS classes need to be adapted in archesproject/packages/packagename/media/js/forms and models (see JS code in example package for details):

- Arches.forms.Materials: Add certainty column
- Arches.models.Materials: Add certainty field
- Arches.forms.ArchaeologicalMaterials: Add certainty field
- Arches.models.ArchaeologicalMaterials: Add certainty field

Report forms

- Arches.reportsection.Description: Add certainty column

Advanced Search

- Arches.widgets.AdvancedSearchPanel: Add Certainty
- In addition to the modification in the JS-code, new entities need to be indexed by elastic search. This is achieved by adding an entry in /arches-web/archesproject/packages/[package-name]/settings.py under 'ADV_SEARCHABLE_ENTITY_TYPES'

Note: Don't modify /archesproject/arches/Media/js/i18n/en-us.js as it is (re-)built each time Arches is built!

6. Transforming Legacy Data into Arches Import Format (Mapping / Script)

Before importing legacy data, they need to be transformed into the Arches import format that must comply with a certain form. Data needs to be arranged by the following columns (if an Authority File defines the vocabulary, the attribute value must be given by the value's ID):

- RESOURCEID|RESOURCETYPE|ATTRIBUTENAME|ATTRIBUTEVALUE|GROUPID

Examples of this format can be seen in:

- packages/[package-name]/source_data/resource_info_small.csv

At i3mainz we have implemented a data specific script that does this task for our dataset(s) that can be found in the attached data. For the future we propose a more generic method, however (discussed in: http://caa2014.sciencesconf.org/browse/session?sessionid=5230 and the attached presentation)

7. Importing Data

The data import is executed by the script listed above: build/install_packages.sh
However, running import_packages.sh does not only import the legacy data, but all content of the package (including graphs, authority files and JS code). A faked example dataset is included in the attached data.

This document was created by:

Tobias Kohr
Institute for Spatial Information and Surveying Technology (i3mainz)
University of Applied Sciences Mainz
Lucy-Hillebrand-Str. 2, 55128 Mainz, Germany