ArcheoInf, the CIDOC-CRM and STELLAR: Workflow, Bottlenecks, and Where do we Go from Here?

Geoff Carver
Universität Göttingen, Germany

Abstract:
The following reports on work undertaken to implement the CIDOC-CRM for the ArchaeoInf project. ArchaeoInf (www.archeoinf.de) is intended to act as a data repository, a database which combines and integrates primary archaeological data collected and processed during surveys or excavations and originally stored in individual project databases. Although creating ArchaeoInf presented a number of difficulties (many related to terminology), the following focuses on an aspect of the creation of metadata mapped to the CIDOC-CRM for export in RDF format using the STELLAR "data mapping and extraction utility." The "standard" STELLAR templates enable exports into the English Heritage version of the CIDOC-CRM, but since ArchaeoInf is not concerned with databases or projects using English Heritage documentation standards, modified versions of the "user defined" templates were used instead. This paper describes the process of mapping existing databases into the CIDOC-CRM and then creating STELLAR templates for exporting the results of SQL queries into an XML/RDF format.

Keywords:
CIDOC-CRM, STELLAR, Database Mapping, "How to?"

1. *ArcheoInf*, the CIDOC-CRM and STELLAR

This paper describes the process used to prepare data for export from the *ArcheoInf* Database (www.archeoinf.de) using the STELLAR "data mapping and extraction utility" (Tudhope et al. 2008, 89) (http://hypermedia.research.glam.ac.uk/resources/STELLAR-applications). It is intended to provide a straightforward, "how to" outline to supplement any number of technical papers already available. The author developed the basic process while helping to map the *Arachne* Database for the Universität Köln (http://en.wikipedia.org/wiki/Arachne_%28Archaeological_Database%29). *Arachne* is described as being "a central object-database for a large federal institution which possesses about two millions of images inside their photographic archives and produces even more data each year in the course of its research activities" (Wikipedia 2012).

At the time of the author's involvement during the winter of 2008-2009, *Arachne* consisted of ca. 125 tables originally prepared by the various offices of the German Archaeological Institute (Deutsche Archäologische Institut/DAI) in Rome, Athens, Istanbul, Cairo and Madrid. The various tables document artifacts, photos from photo archives, bibliographic references, etc.

To put these numbers in perspective, *Arachne* might be compared to the 60 tables, ca. 2500 fields, ca. 250 index lists and ca. 110 layouts of iDAI.field (DAI 2012), the DAI's database for recording sites and field surveys. *ArcheoInf* itself has 46 tables and a total of 225 fields.

Following a feasibility study by Robert Kummer (2007), it was decided to map the contents of *Arachne* in CIDOC-CRM (CRM) format. It was later decided to also map the contents into Dublin Core (DC).

Corresponding author: gjcarver@t-online.de

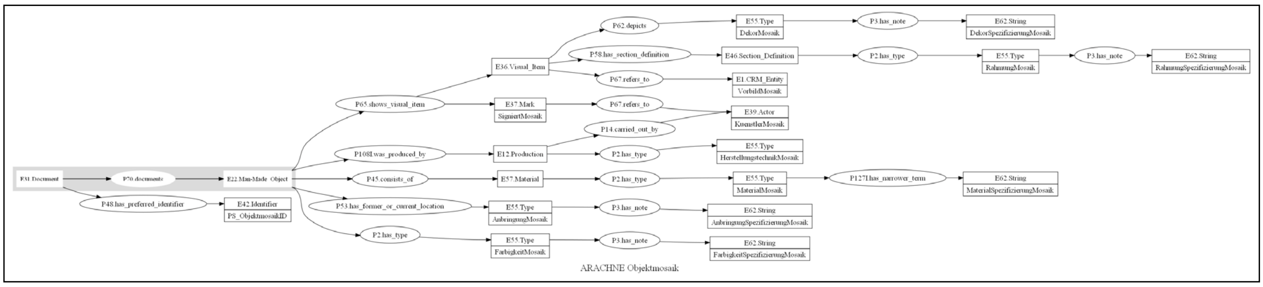

Figure 1. *Example of a full mapping.*

Find [table name]

[short description of what the table is supposed to document, as defined by the database itself, when possible]: Ein Fund enthält alle Attribute eines archäologischen Befunds. Einzelne Funde werden Klassifizierungen über eine oder mehrere Fund-Klassifizierungen angehängt. Messwerte zum Fund werden über Fund-Messungen beschrieben.

FindID [column name]
Scope note: Identification number of Find.
Data format: LongInteger
Sample value: 2204

CIDOC:
E22.Man-Made_Object
P48.has_preferred_identifier
E42.Identifier
Other notes:

Table 1. *Example of an inventory sheet.*

Since the large number of tables and fields did not allow an easy overview and *Arachne* itself had not been documented, the first step was to prepare an inventory of the tables and their fields. An inventory sheet (Table 1) was prepared for each table listing each column heading (exported as a list from the database) plus scope notes, data format (integer, text string, Boolean, etc.), examples, comment, and the CRM and DC mappings. Each inventory sheet also had a link to a PNG version of a flow-chart which illustrates the mapping.

When examining the contents for the purpose of writing the scope notes and comments, no attempt was made to correct errors within the actual content of the database. Examples were chosen which not only showed what kind of information was intended to be recorded within a given column, but also included examples of entries which clearly did not belong. In addition, since many columns were empty, their function had to be assumed, often on the basis of very unclear column headings.

The overall situation was similar to that encountered during the course of the STAR project:

> *In most cases we did not have specific metadata nor descriptive scope notes on what the field content was intended to represent. It was therefore important to review the actual data within the field, as well as the field label, to judge the intended 'meaning' of each field. We recommend assessing the data content, even when a field label seems familiar* (Tudhope et al. 2011, 2.2).

In the case of *Arachne*, the lack of documentation reflects the context (multiple locations) within which data was entered (Carver and Lang 2013).

These inventory sheets were linked to a master document which could in turn serve as the documentation for the *Arachne* database as a whole. These inventory sheets and the master document were prepared using *Open Office Writer*. One clear advantage of *Open Office* over, for example, *Microsoft Word*, was the ease of producing PDFs without the need for additional software or installing a "printer." Thus an updated version of the master document was produced almost daily with the intention that the final version would eventually be posted online in PDF and XML format.

These inventory sheets were intended to provide the basis for some form of template which would in turn be used to export data as XML tagged with the relevant CRM and DC mappings suitable for downloading in response to SQL queries, in addition to displaying an SVG form of the flow-chart online. The process for creating these templates and export documents had not yet been determined when the author quit the project.

For the purposes of the CRM, each individual table was considered a document as part of a larger document (the *Arachne* database itself). Each table thus documented a CRM "Entity": a place, artifact, site, image, etc.

2. CRM Mapping

Mapping was done by consulting PDF versions of the *Definition of the CIDOC Conceptual Reference Model* (Crofts et al. 2009; Tudhope et al. 2011, 1.2) and the online Web Ontology Language (OWL) version maintained by the Friedrich-Alexander-University of Erlangen-Nuremberg (http://erlangen-crm.org/) viewed in Protégé (http://protege.stanford.edu/). The advantage of using the PDF version reflects the ease of searching electronically rather than leafing through a long printed document, while the online version in Protégé enables exploration of the hierarchical relations between Entities graphically.

The CIDOC-CRM is a powerful formal (and occasionally infuriatingly abstract; Tudhope et al. 2008, 88) metalanguage. Its logic is not immediately apparent, but may be illustrated through an example of one of the many conceptual problems the author has encountered using legacy data from a cemetery in Italy (Heitz 2009). Since the human remains in this case had either decayed (after burial) or could no longer be located (after excavation), there was really no "person" involved (i.e. they were only documented indirectly), and the documentation should have centred on graves as "Man-Made Features" (i.e. holes in the ground). As a grave itself does not have a gender (in some cases this could be inferred on the basis of grave goods or the size and/or depth of the grave [assumed to reflect status]), nor an age class (juvenile, adolescent, adult) – despite not actually having been documented in the database itself – the "person" had to be modelled in the CRM.

At the same time, the grave also had to be modelled. In some cases, for example, although no body had been found, there was evidence of a "crouched" burial. And just as a grave cannot be "adolescent," the person who had been buried there was not "crouched."

Partly in order to develop methods and the experience necessary for dealing with such problems, it was decided to begin with simple cases and gradually work towards more complex mappings. A number of tables which simply listed "internal" links to other tables were quickly mapped and the knowledge gained used to provide a work flow and overview of the way forward. In this way, duplication sped the mapping: sections which had already been mapped for one table could be reused by simply copying and pasting relevant strings of text. Since each table, for example, was considered to be a "document" documenting a given "CRM Entity" (i.e. photo, book, architectural fragment, statue, etc.), and each "CRM Entity" had, in turn, a "Preferred identifier" (index or primary key) which identified it within the database, these "roots" could then be copied and pasted into each inventory sheet.

After this, relatively easy (and often repetitive) elements were identified and combined like building blocks. The process of constructing *Arachne* by combining existing tables from a number of sources had led to a good deal of duplication. Information on places was repeated on any number of different tables, for example, as the location of sites, places where stray finds had been discovered, books published, places depicted in photos, etc. In

`digraph g {`	Indicates the beginning of a “directed graph” (opens brackets)

`graph [`
`rankdir = “LR”`
`];`
	Declares that the graph’s layout (“rank direction”) will be Left to Right (i.e. not vertical)

`node [`
`fontsize = “16”`
`shape = “ellipse”`
`];`
	Gives information about the format the “nodes” will take: ellipses with 16 point text

`“E84” [`
`label = “<f0> E84.Information_Carrier |`
`The sculpture ‘Monument to Balzac’”`
`shape = “record”`
`];`
	Defines an entity called “E84” with a label “E84.Information_ Carrier” (CIDOC-CRM entity) in a box separated from “The sculpture ‘Monument to Balzac’” (in the horizontal layout, the vertical line placed the text under the CIDOC-CRM identifier); the shape is defined as “record” (i.e. a box)

`“E21_Balzac” [`
`label = “<f0> E21.Person | Honore de`
`Balzac”`
`shape = “record”`
`];`
	Defines an entity called “E21_Balzac”; the <f0> defines an address in this box (in this case the portion containing “E21. Person”; other lines could be similarly numbered to allow more complex links)

`node [shape=ellipse]; {node [label=“P62.`
`depicts”] P62;}`
`edge [`
`];`
	Defines a relation (node) called “P62” and labelled “P62. depicts”

`“E84”:f0 -> “P62” [`
`id = 0`
`];`
	Links “E84” with “P62”; note the direction of the arrow

`“P62” -> “E21_ Balzac “:f0 [`
`id = 0`
`];`
	Links “P62” with the relation “E21_ Balzac”;

`label = “Monument to Balzac”;`
`fontsize=20;`
	Provides a label to the directed graph

`}`	Indicates the end of the “directed graph” (closes brackets)

***Table 2**. The script used to generate the Graphviz diagram in Figure 2.*

many cases the tables recorded who had last edited a given record and the date when the data was entered, and this data was linked to the table or “document” itself rather than to the entity recorded. After such common links had been identified and modelled once, they were simply copied and recombined.

3. Graphviz

GraphViz (http://www.graphviz.org/) was used to render the CRM mappings graphically. The original aim was to provide a means for controlling the mappings as they developed (correcting errors, etc.). Within the inventory sheets the column headings were listed according to their “sequence” within

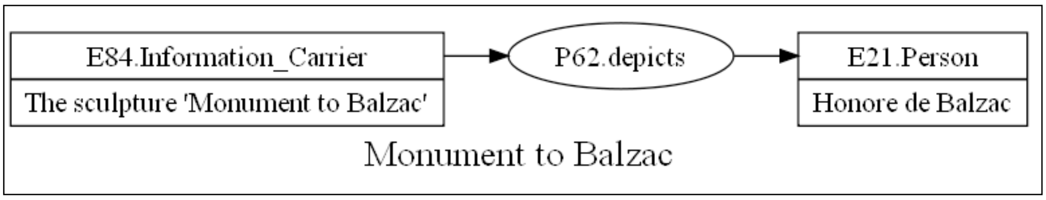

***Figure 2**. Example of a simple Graphviz diagram of a CIDOC-CRM tuple.*

the various tables in order to ease interaction with the database, but with some of the more complex tables this was far from ideal, as the columns were often listed alphabetically, without consideration for logical or semantic integrity (i.e. "collocation" or grouping closely related columns together; Beall 2006, 62). Although the column headings could have been reordered within the documentation to better reflect the semantic structure, it was quickly recognised that the resulting diagrams also helped document the various tables.

When choosing a graphics program, the main concern was that the layout should be generated automatically. Anyone using such flow-chart programs as *Visio* (Visio 1996) generally has to arrange various elements. As the goal was to find a way to quickly check the logic and internal consistency of the various mappings, so long as the final drawing was comprehensible, the layout itself was relatively unimportant.

Since *GraphViz* is also used in the *Protégé* ontology editor, consistency in layout and coding of output files (OWL, RDF) was seen to be an advantage, as was its ability to export diagrams in SVG format for use online. A number of plug-ins are available for *GraphViz* which can, among other things, extract and automatically graph a database structure.

The STELLAR programmers recommend the use of a similar program called *Gruff* (http://www.franz.com/agraph/gruff/) as a means for querying the resulting RDF. Use of one or the other may simply be a matter of personal preference.

GraphViz has the benefit of being freeware, and is relatively simple to use. Various objects and their relations are defined as text, meaning that the process can be automated to some extent by exporting a list of the column names and using a universal search and replace function within a text editor[2].

The column headings were exported and copied into the text-editor. In this way, complete lists could be turned into drawing elements quickly, without error, and those entities which had not been linked up in some relation were immediately visible.

Each "Entity" was mapped in a box (`shape = "record"`). Those Entities which corresponded to a given column were drawn to show the column name in a lower box. The Entities were linked through "Properties" mapped as ellipses (`shape = "ellipse"`).

Table 2 is a sample of the script used to draw a simple *Graphviz* diagram (Fig. 2; Doerr and Kritsotaki 2006, 5 and fig. 3)).

The mappings were drawn horizontally, leading from the name of a given table (at the far left) to the name of a given column at the end of each branch (Fig. 1). Although the diagrams could also have been generated vertically (top to bottom), the results were less consistent (and more difficult to read). Among other things, when a number of fields (the various spatial coordinates of Cartesian space, for example, or length, width, thickness or diameter and weight of a given artifact) all mapped onto the same path (thereby requiring a list of column headings), the lines joining individual elements from any given list become especially complicated in a vertical format.

[2] The author used TextPad (www.textpad.com) largely because it allows the simultaneous editing (search and replace) of numerous documents.

As has been noted, each table was considered to be a "document" which documents a given "CRM Entity" (i.e. photo, book, architectural fragment, statue, etc.). Each such document was given a CRM "root": the table (E31.Document) documents (P70) photos (E38.Image). These "roots" were marked in gray to distinguish them from the contents of the table themselves, thus allowing the activity of entering data to be linked to the table/document and not to the entity being documented.

This "P70" solution actually caused some debate. Some of the computer programmers insisted that an Entity is documented in (P70i) a given table (E31), but this caused problems with those tables which only combined "internal" links to other tables, and thus did not document anything per se. It also caused conceptual problems when trying to link the individual tables back to *Arachne*, since *Arachne* – as a Document – is composed (P148. has_component) of parts (i.e. the individual tables, each of which is in turn a Document), each of which documents (P70) some type of CRM-Entity (E1). Otherwise, if an Entity is documented by (P70i) a given table, then *Arachne* documents a type of Entity which is in turn documented in a table (Document). As a database, *Arachne* does not document objects; *Arachne* merely links a series of tables, each of which documents a type of object in the real world. These tables can exist separately from *Arachne*, and may be added or deleted at will.

Some of the entities which were recorded in two or more tables were not distinguished in the CIDOC-CRM. The database records artifacts. Some of these are sarcophagi, recorded in the "sarcophagus" table, and some were statues, recorded in their own table. It is simply easier to divide the mapping according to existing tables and thus maintain consistency with the database itself, especially considering that it was continually being modified.

It was also intended to eventually make it possible to link each table diagram in a diagram of the entire *Arachne* database. This would have required some minor editing (fixed when it came time to map *ArcheoInf*): instead of using just the column name as an identifier ("ID_ number"), the "entity" nodes were identified

```
Alter

E31.Document
  P70_Documents
          E21.Person
                  P2_has type
                          E55.Type
                          txt_alter
                                  P48.has_preferred_identifier
                                          E42.Identifier
                                                  ID_Alter
                                          E42.Identifier
                                  P48.has_preferred_identifier
                          E55.Type
                  P2_has type
          E21.Person
  P70_Documents
E31.Document
```

***Table 3**. Example of a completed CRM map.*

Alter		
	CIDOC-CRM	DC
txt_alter	E55.Type	Type
ID_Alter	E42.Identifier	Identifier

Table 4. *CIDOC-CRM mappings reduced to Dublin Core.*

using a binomial combination of "table-name" and "column-name" ("person-ID_number") to insure that all the elements from each table were uniquely identified. In this case, the larger E31.Document "*Arachne*" would have all the separate tables as parts.

4. *Arachne* Export

Once each inventory sheet had been prepared, a copy was made where all of the extraneous information (scope notes, examples, etc.) was stripped away, leaving a simple version listing just the CRM mapping and column names, ready for conversion into XML (including the use of tabs to mimic the hierarchical structure (Table 3)). The column order was also rationalized ("collocation"), reflecting the CRM rather than the alphabetic or apparently random sequence found in the database itself. Copies were also made of these secondary files, where the CRM maps were reduced to DC (Table 4).

Scope notes, SKOS, etc., could have also been reintegrated into the list at this point, if desired.

5. Lessons Learned from *Arachne*

One of the most important (and obvious) lessons learned from working with *Arachne* (and reinforced by subsequent experience working on the *ArcheoInf* project) is that databases (and software) need to be documented (Backhouse 2006: 54-55), as this saves a great deal of guesswork later when trying to determine what a given column was meant to represent. Although this need for documentation may be obvious, corners are cut too often. An especially difficult case was that of a table where the column headings correspond to a large number of the 650+ numbered (*not named!*) fields of the *MARC 21 bibliographic standard* (http://www.loc.gov/marc/bibliographic/).

It also helps if the database itself has been well-designed. The spatial information stored in *Arachne* provides a case in point. Ideally, one table would contain the name of a place, its local jurisdiction (county, province, etc.), which country it lies in, plus spatial coordinates and perhaps a cross-reference to a map reference and/or an online gazetteer (such as the Getty Thesaurus of Geographic Names® Online (http://www.getty.edu/research/tools/vocabularies/tgn/)). As has been noted, spatial information was entered repeatedly on numerous different tables. Given the circumstances under which Arachne was put together in different locations, this probably could not have been avoided, but it was not efficient. More important, perhaps, is control over data entry, to insure that only *one* type of information is entered into any given field: *only* Spatial Coordinates (E47) and not town Place Names (E48), for example.

Experience with mapping *Arachne* in CIDOC-CRM shows the value of beginning with something simple and working up to more complexity (Woolley 1961: 54). The CIDOC-CRM is complex enough without starting at the most complex table possible. Starting with something relatively easy allows someone to find out how the CRM works, become familiar with the various aids (the handbook, ontology editors like Protégé, etc.), and to develop a work flow process that suits the given needs and available resources. It is worth noting that, for example, whereas the interactive interface in the Erlangen OWL makes it easy to follow entities (i.e. E59.Primitive Value, E60.Number, E61.Time Primitive and E62.String) through their hierarchical relationships, some of the elements listed in the handbook are not represented online. And, although having a "hard" copy of the handbook is always a good idea, the "search" function in the PDF format is probably more helpful in the long run (among other things, the handbook is too long to search through manually).

Potential users really have to learn how to use or think in terms of the CIDOC-CRM:

> *many of the entities in the ontology are fairly abstract; understanding the conceptual complexity of the CIDOC CRM poses a challenge to some non-specialists. It can also be possible for different people to make alternative valid mappings to the ontology for the same situation, raising difficulties for semantic interoperability* (Tudhope et al. 2011, 4.2).

Eventually, with practice and familiarity, it is possible to model something as initially intimidating as a list of the items carried by a given deity depicted on a certain portion of a coffin in the CIDOC-CRM (`"E22.Man-Made_Object" – "P58.has_section_definition" – "E46.Section_Definition" – "P62.depicts" – "E1.CRM_Entity" "P3.has_note" – "E62.String"`), but to figure out how to do this efficiently takes time and practice, as was underlined when attempts were made to map several very complex tables "out of sequence."

On a practical note: when working simultaneously with *Open Office Writer*, a text editor, *Gruff* or *GraphViz*, the online Web Ontology Language (OWL) version of the CIDOC-CRM viewed in Protégé, and a PDF version of the *Definition of the CIDOC Conceptual Reference Model*, it is useful to work with two monitors.

Since some specialised software will not run on all operating systems, hardware and operating systems also play a role. While working on *Arachne* the author had one machine running Mac and another running a Windows emulator.

6. *ArcheoInf*

The purpose of *ArcheoInf* differs from that of *Arachne*. As described elsewhere (Carver, Lang, and Türk 2012, Carver and Lang 2013), *ArcheoInf* is intended to function as a long-term document repository for combining and storing primary archaeological data recorded in individual project databases constructed for use on surveys and excavations undertaken by German university departments of Classical Archaeology. As such it might be compared to the ADS in the UK (http://archaeologydataservice.ac.uk/), or DANS in the Netherlands (http://www.dans.knaw.nl/en) or tDAR in the USA (http://www.tdar.org/). The major difference being that, whereas the ADS stores documents, *ArcheoInf* aggregates data from any number of databases in a new database, thereby allowing inter-site queries. A closer analogy might be the STAR project (May et al. 2011).

Funded by the Deutsche Forschungsgemeinschaft (DFG), *ArcheoInf* was developed as a collaborative project involving the following institutions:

- Archäologisches Institut und Sammlung der Gipsabgüsse der Georg-August-Universität Göttingen;
- Institut für Archäologische Wissenschaften der Ruhr-Universität Bochum;
- Fachbereich Vermessung und Geoinformatik der Hochschule Bochum;
- Universitätsbibliothek Bochum;
- Lehrstuhl für Softwaretechnologie in der Fakultät für Informatik der Technischen Universität Dortmund;
- Universitätsbibliothek Dortmund.

Building on previous experience, the CRM and DC mapping followed much the same process as that used for *Arachne*. Where relevant, the columns were also mapped in

SKOS (Lang 2013) and Europeana (Isaac and Clayphan 2011) format.

Because the *ArcheoInf* database was planned from the beginning and not assembled from existing tables and databases, it is simpler and more coherent than *Arachne*. On a technical level, however, some of the elements unique to *ArcheoInf* (the thesaurus and the relations between various concepts) present special problems. The CRM allows for hierarchical relationships between "types," for example (a sword is a "type of" weapon), but unlike SKOS, makes no allowance for related terms (i.e. near-synonyms or the foreign translations of general terms in the thesaurus which is one of *ArcheoInf* strengths (cf. Baines and Brophy 2006, 239-240)).

7. Using STELLAR to Export *ArcheoInf*

It was decided to use STELLAR to export the data tagged with CRM mappings (May et al. 2011). One of the advantages of this software is that it, too, is free. STELLAR includes templates built for use with the English Heritage version of the CIDOC-CRM (CIDOC-CRM-EH) (http://hypermedia.research.glam.ac.uk/resources/crm/), with columns relating to particular fields within English Heritage documentation system, and alternate "user-defined" templates. Since ArcheoInf does not use the English Heritage documentation system, a text editor was used to modify the "user-defined" templates to reflect the contents of the existing tables.

The STELLAR example files were downloaded and their contents examined using a text editor in order to determine the function of each one while working through the examples in the tutorials (http://reswin1.isd.glam.ac.uk/stellar/tutorials/tutorial1.html, http://reswin1.isd.glam.ac.uk/stellar/tutorials/tutorial2.html). This method made up for the fact that the tutorials were generally instructive without being explanatory (focusing more on *what* to do without explaining *why* any given step should be performed, a common failing in computer documentation).

One minor criticism: when consulted, some of the weblinks on the STELLAR website were broken. These have since been fixed, but the lack or absence of samples can be frustrating to anyone working to deadline or trying to choose between one program and another, or decide whether to trust software to the degree of investing time and effort getting it to run (Carver and Lang 2013). There are also minor formatting issues with the output (a lot of unnecessary blank lines), and some problems opening the results in Protégé (possibly some formatting incompatible with RDF or OWL).

Another minor problem reflects the fact that, like many other projects, STELLAR syntax is dependent upon imported code. Those seeking "more details of syntax and more advanced usage examples" – such as information on the use and constraints on some of the "if" statements – are directed towards the site of StringTemplate (described as "a java template engine", http://www.stringtemplate.org/), which, like many such websites, is not particularly "user friendly."

Of more concern, though, was the fact that, as a terminal-application, the STELLAR user interface was not the easiest to use; typing mistakes are inevitable. A new, user-friendly GUI has since been added for use in Windows. The process was worked through by experiment first with carried out simple CSV tables (Klammt 2011), then a slightly more complex Access database (Heitz 2009), before starting to work with *ArcheoInf* itself using SQL queries. As has been noted,

> *In practice, getting to use someone else's data implies mapping the fields and codes used to one's own system. This is a time-consuming exercise, that has to be repeated each time a different data set is imported. And even then, data that cannot be adequately translated may be lost in the process* (Verhagen et al. 2011, 153).

The data first had to be formatted properly. The CSVs exported from MS-Excel caused problems (probably due to the inclusion of "illegal" – and invisible – formatting characters), so were re-exported from *Open Office Calc*. The German "sz" (ß) and letters with umlauts (ä, ö, ü) were also converted to the standard character set (ss, ae, oe, and ue).

The next step will be to put the query and export process function online, either using STELLAR or some alternative. Further development has been limited in part by the need to acquire data (Carver and Lang 2013).

8. Conclusions

Overall, the process works. It is possible to map complex archaeological databases and export tagged data with the CRM mappings in XML using STELLAR templates. What is less clear at this point is how this tagged data can then be imported into other databases (i.e. how to "match" mappings) and perhaps whether or not there is even a demand for the ability to do so. At present it may be possible that extensive effort is being invested into export functions nobody wants. The question then becomes one of whether or not that investment represents a waste of time and effort, or whether – once widely available – tagged data will create its own market (Tudhope et al. 2011, 4.6; Eckkrammer et al. 2011, 84).

This paper was intended to help make the mapping process just a little – it is hoped – easier for beginners.

References

Backhouse, P. 2006. "Drowning in data? Digital data in a British contracting unit." In *Digital Archaeology: Bridging Method and Theory*, edited by T. L. Evans and P. T. Daly, 50-58. London: Routledge.

Baines, A., and K. Brophy. 2006. "What's another word for thesaurus? Data standards and classifying the past." In *Digital Archaeology: Bridging Method and Theory*, edited by Thomas L Evans, and Patrick T Daly, 236-50. London: Routledge.

Beall, J. 2006. "The Death of Metadata." *Serials Librarian* 51 (2): 55-74.

Carver, G., and M. Lang. 2013. "Reflections on the rocky road to E-Archaeology." In *Computer Applications and Quantitative Methods in Archaeology (CAA), Southampton, March 26-29, 2012*, edited by G. Earl, T. Sly, A. Chrysanthi, P. Murrieta-Flores, C. Papadopoulos, I. Romanowska, and D. Wheatley, 224-236. Amsterdam: Pallas Publications.

Carver, G., M. Lang, and H. Türk. *in press*. "Recent Developments in the ArcheoInf Project – Towards an Ontology of Archaeology." In *Fusion of Cultures. Proceedings of the 38th International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA), Granada, April 6-9, 2010* , edited by F. J. Melero, and P. Cano.

Crofts, N., M. Doerr, T. Gill, S. Stead, and M. Stiff. 2009. "Definition of the CIDOC Conceptual Reference Model. Version 5.0.1." ICOM/CIDOC CRM Special Interest Group. Accessed 25.9.2013. http://www.cidoc-crm.org/docs/cidoc_crm_version_5.0.4.pdf.

DAI. 2012. "iDAI.field." Accessed 18 March 2012. http://www.dainst.org/en/project/idaifield?ft=all.

Doerr, M., and A. Kritsotaki. 2006. "Documenting Events in Metadata." In *The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2006)*, edited by M. Ioannides, D. Arnold, F. Niccolucci, and K. Mania. The Eurographics Association. Accessed 25.9.2013. http://www.cidoc-crm.org/docs/fin-paper.pdf.

Eckkrammer, F., R. Feldbacher, and T. Eckkrammer. 2011. "Cidoc CRM in Data Management and Data Sharing. Data Sharing between Different Databases." In *On the Road to Reconstructing the Past. CAA 2008: Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 36th Conference, Budapest, April 2-6, 2008*, edited by E. Jerem, F. Redö, and V. Szeverényi, 80-

85 . Budapest: Archaeolingua.

Heitz, C. 2009. *Ripacandida project database. Ver. 17.03.09*. Göttingen: Archäologisches Institut, Georg-August-Universität Göttingen.

Isaac, A., and R. Clayphan (eds). 2011. *Europeana Data Model Primer*. City: Publisher.

Klammt, A. 2011. *Wetdat project database. Ver. 07.06.11*. Göttingen: Archäologisches Institut, Georg-August-Universität Göttingen.

Kummer, R. 2007. *Towards semantic interoperability of cultural information systems - making ontologies work*. Köln: Universität zu Köln.

Lang, M. 2013. "Standardised Vocabulary in Archaeological Databases." In *Computer Applications and Quantitative Methods in Archaeology (CAA), Southampton, March 26-29, 2012*, edited by G. Earl, T. Sly, A. Chrysanthi, P. Murrieta-Flores, C. Papadopoulos, I. Romanowska and D. Wheatley, xxx. Amsterdam: Pallas Publications.

May, K., C. Binding, and D. Tudhope. 2011. "A STAR is Born: Some Emerging Semantic Technologies for Archaeological Resources." In *On the Road to Reconstructing the Past. CAA 2008: Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 36th Conference, Budapest , April 2-6, 2008*, edited by E. Jerem, F. Redö, and V. Szeverényi, 111-16. Budapest: Archaeolingua.

Tudhope, D., C. Binding, and K. May. 2008. "Semantic interoperability issues from a case study in archaeology. Semantic Interoperability in the European Digital Library." In *Proceedings of the First International Workshop SIEDL 2008, associated with 5th European Semantic Web Conference, Tenerife*, edited by S. Kollias, and J. Cousins, 88-99. City: Publisher.

Tudhope, D., K. May, C. Binding, and A. Vlachidis. 2011. "Connecting archaeological data and grey literature via semantic cross search." *Internet Archaeology* 30.

Verhagen, P., C. Sueur, and M. Wansleeben. 2011. "Setting a Standard for the Exchange of Archaeological Data in the Netherlands." In *On the Road to Reconstructing the Past. CAA 2008: Computer Applications and Quantitative Methods in Archaeology. Proceedings of the 36th Conference, Budapest, April 2-6, 2008*, edited by E. Jerem, F. Redö, and V. Szeverényi, 152-55. Budapest: Archaeolingua.

Visio 1996. Visio 4.0c for Microsoft Windows.Visio Ver. 2010. Microsoft.

Microsoft. 2010. Visio, version 2010.

Wikipedia. 2012. "Arachne (Archaeological Database)." Web page, Accessed 18 March 2012]. Available at http://en.wikipedia.org/wiki/Arachne_%28Archaeological_Database%29.

Woolley, L. 1961. *Digging up the Past*. Harmondsworth: Penguin.