**Introduction**

There is something rotten in the kingdom of Belgium. Right now there is no solid, central and institutionalized infrastructure for the preservation of social science data in the country. Scientists such as demographers and sociologists collect a lot of data for their research, usually in the form of numbers and figures from which they derive statistics and trends (i.e. quantitative data), but also oral or written answers from respondents to questionnaires or during interviews in more open-ended formats (i.e. qualitative data). Such data collections are usually kept on private storing devices (computers, flash drives or servers) that belong to the researchers themselves and, in the best cases, on a server maintained by the researchers’ research unit. Practices vary widely, as some research units impose data archiving while others merely recommend it and yet others have no specific guidelines at all.

In this context it is difficult for researchers to be sure whether or not they will have the opportunity to access and re-use research data produced by their peers. Re-use of research data, or ‘secondary data analysis’, is critical in the social sciences, just like in any other data-driven scientific field, because it enables the replication of studies. It also allows researchers to skip the stage of data collection and make use of previously collected data for their own research, so that they can apply their own methodologies to said data and test new hypotheses. In addition, it provides invaluable help to young researchers.\(^1\)

The data of social sciences are often collected thanks to public funds, so society at large would (and should) benefit from a ‘data archive’ infrastructure. Furthermore, to make research data available, they should be properly archived. Since they often come in digital format, even for early instances of research data, and since they are often support- or software-dependent, research data pose various preservation problems and challenges.

That is why the SODA project (Social Sciences Data Archive), a pilot study for a Belgian data archive for the social sciences, brings together representatives of the social science community as well as professional archivists, who pool their respective expertises in order to create Belgium’s very own public

---

data archive\(^2\). The ensuing entity will integrate the Consortium of European Social Science Data Archives (CESSDA ERIC\(^3\)) and serve as a central platform for archiving and disseminating research data, following in the footsteps of such predecessors as GESIS (Germany)\(^4\), DANS (Netherlands)\(^5\) or UKDS (United Kingdom)\(^6\).

Many legal and organizational questions remain unanswered at this point: What will be the legal form of this new entity? Can it seamlessly integrate representatives of Belgium’s various political bodies, notably the linguistic Communities, the Regions and the Federal State? Where and how exactly will the data be stored? While the project team investigates such issues, this article will present an early technical realization which relies on the assumption that the future entity will somehow integrate both the State Archives and the Belgian universities, the former as managers and preservers of data, the latter as purveyors and consumers. In this scenario, part of the State Archives’ infrastructure for the preservation and dissemination of data will be re-used for the needs of the future Belgian data archive.

Just like social scientists like to be able to re-use data, an increasingly popular method for putting together new, large-scale research institutions is the re-use of already existing infrastructures\(^7\). Needless to say, relying on existing entities means cost reduction. It also implies that people who hail from different companies as well as different fields of study will have to work together one way or another. While this can lead to friction, it also creates an opportunity for interdisciplinary partnerships and the breaking up of organizational silos\(^8\).

Preserving and disseminating archival objects means documenting them thoroughly, so both archivists and social scientists rely on well-established metadata standards to describe their data. In digital contexts, archivists use the Encoded Archival Description (EAD) to produce machine-readable versions of their finding aids, while social scientists work with the Data Documentation Initiative (DDI) standard to

\(^2\) The SODA project members are the State Archives of Belgium, the Université catholique de Louvain and the Vrije Universiteit Brussel. Because there are two other universities in Belgium who go by the same names but in the other language — the Katholieke Universiteit Leuven and the Université libre de Bruxelles — it is best not to translate the name of either university to avoid ambiguity.

\(^3\) ‘ERIC’ stands for ‘European Research Infrastructure Consortium’. It is a legal form under European Union law for large bodies of research institutions. Such a status facilitates a consortium’s participation in European research projects along with boosting its visibility and influence in the race for funds.

\(^4\) GESIS is part of the Leibniz Institute for the Social Sciences: https://www.gesis.org/

\(^5\) DANS are the Data Archiving and Networked Services: https://dans.knaw.nl/en

\(^6\) UKDS is the United Kingdom Data Service: http://ukdataservice.ac.uk/


document their datasets. Much of the State Archives' infrastructure for data management was built to handle EAD files. So then, re-using that same infrastructure for the storage and distribution of social science data would be greatly profitable, as it would deal with the problem of putting together a whole new machinery from start to finish: we would begin with much of it in our hands already! But this brings about several technical requirements. One of them is the transfer of certain information items from DDI over to EAD, so that, for example, information about datasets (i.e. metadata) can be displayed in the State Archives’ open access public catalogue (OPAC).

Both metadata standards, DDI and EAD, are based on the eXtensible Markup Language (XML). This technical proximity enabled the development of a crosswalk to bridge these two standards and channel some of the information contained in DDI files over to new EAD files. This article presents the subsequent mapping\(^9\).

**The Bigger Picture: Mediating Between Rivals**

Grab a soda; let’s talk about the future.

Before I present the mapping itself, I feel the need to expound its rationale. People often find it hard to understand the purpose of such a crosswalk, not because it is so brilliant and cryptic at the same time but, more likely, because, as good or bad an idea as it might be, its practical realization was assigned to a young and inexperienced researcher (and that would be me). And, as Fate would have it and as I got to find out, in this milieu you often find yourself in situations where you have to explain what you are still grappling with yourself, and more often than not it’s stuff that you did not even come up with in the first place! (No wonder impostor syndrome thrives so well in academia.) So then, even though this mapping might be a top-notch idea, I’ve been struggling to convey its *raison d’être* effectively.

With time however, as I went on doing what I was told to do and as my colleague on this research project, Samira Hajji, a brilliant legal expert, worked out the business model and the legal form of the future Belgian data archive, it all became clearer. Like a daring scientific hypothesis (and in many ways like a bold business venture, too) the whole project rests on several assumptions. First, we assume that

---

\(^9\) This work was first presented in B. Peuch, ‘Elaborating a Crosswalk Between Data Documentation Initiative (DDI) and Encoded Archival Description (EAD) for an Emerging Data Archive Service Provider’, *IASSIST Quarterly*, 42, 2, 2018, p. 1-24, [https://doi.org/10.29173/ijq924](https://doi.org/10.29173/ijq924). The present paper presents it in a less technical, more accessible manner.
there is way — that there must be a way — to create an administrative entity that brings together the various Belgian levels of government, in spite of the intricate division of powers and responsibilities that is the signature of Belgium’s quaint political organization. Secondly, there ought to be a legal form that can somehow bridge the administrative division between the Federal State, to which the State Archives belong, and the (linguistic) Communities, on which the higher education institutions depend. Finally, there must be a way to overcome the academic rivalry between the various universities, who compete for funds, prizes, impact factors and peer recognition; a way to bring them all and bind them, not in the darkness, but in a collaborative interface, in a ‘pre-competitive’ context if you will.

Well then, when you think about it, the idea of a crosswalk between the metadata standards of archives on the one hand and of social sciences on the other connects all these issues like a common thread! It rests on the assumption that part of the Archives’ infrastructure will serve the purposes of the data archive, so that the State Archives and the universities will have to work together. They will be able to do so only through strict, lawfully binding agreements. And because the State Archives, a federal institution which does not carry out social science research per se, will act as the pivotal point of contact and as the platform for sharing data, they will constitute a ‘neutral’ force, a ‘counter power’ of sorts, in this alliance of data purveyors, managers and consumers — something a university, biased as it structurally is, could not achieve in this configuration.

That is not to say that the State Archives do not stand to gain anything in this enterprise. As we embark on a project to develop a data archive which will handle mostly if not exclusively digital data, we have the opportunity to expand our own infrastructure for processing digital records as well as our knowledge and expertise in digital archiving. Also, as the prime administrator of the future data archive’s data, the State Archives will thus enrich its collections with a new kind of data, which might bring in new customers, or incite its designated community to explore new research avenues: consider how genealogists might find it interesting to delve into statistical data about, say, the life expectancy or the birth rates at the time when one of their ancestors was born for example. (Wow, it turns out Uncle Larry is part of the 27.84 % of the general population in this neighborhood who did not fall victim to the plague!)

So really, I’m not saying that this mapping binds everything together, but it’s coming pretty close! Although in order to do that, it has to be functional and well integrated to the computer architecture of the State Archives. With this article I mean to demonstrate the first of these two aspects. (The latter will probably be the subject of a paper on its own as well, although I will most likely not write this one alone, God forbid!)
Just slapping the data of social sciences and the data of historical research together doesn’t work: you only end up with troubled water (i.e. messy data). So then, it’s about building a bridge that somehow connects two metadata standards, DDI and EAD. What are these metadata standards of yours all about though? Why do we need them? Can’t we just punch in our paper finding aids into Microsoft Word or LibreOffice (or even LaTeX, for the kinkier among us), upload the damned thing on a server and be done with it? Okay, maybe fiddle with it a bit more to have it nicely and neatly displayed on our website, but surely that’s about it?

Not quite. To explain the added value of standards like these, we must turn to their genitor, XML. XML is not a metadata standard in itself: it provides a framework for creating standards. It bears its name very well: it is an eXtensible Markup Language — that is to say, it is used to mark things up with tags, and in so doing it helps us tell the computer what those things are and how the machine should consequently process them.

Have you ever seen an HTML page, one of the core technical documents behind the Web pages that your Web browser displays for you? See all those strange tags wrapped in angle brackets (‘<’ and ‘>’) with sometimes cryptic, abridged terms?

```
<!DOCTYPE html>
<html>
  <head>
    <title>Archives of Antwerp</title>
  </head>
  <body>
    <h2>The Story of a Harbor</h2>
    <p>Antwerp is the second largest harbor in Europe after Rotterdam.</p>
  </body>
</html>
```

**Figure 1.** A simple example of HTML.

‘HTML’ stands for HyperText Markup Language. Just like XML, it’s a language whose purpose is to inform computers (or in that case, Web browsers) what is what in a document that we want it to display for us. Likewise, it does so through markup. As you can see here, with the <head> tag, we can tell Internet

---

10 Based on the HTML example of the ‘HTML Introduction’ Web page of the World Wide Web Consortium’s website for computer science courses, [w3schools.com](https://www.w3schools.com/html/html_intro.asp) [Accessed 26th July 2018]. Notice that, if the need ever arises, you can hide a body in an HTML page!
Explorer, Mozilla Firefox, Google Chrome or any other browser that, effectively, the head that we want for our Web page is the title (<title>) ‘Archives of Antwerp’. Next, for the body (<body>) of our Web page, we would like to have ‘The Story of a Harbor’ as the first heading (<h>) and, for our first paragraph (<p>), we would like to have the sentence ‘Antwerp is the second largest harbor in Europe after Rotterdam.’ (And bam! we have us a whole website. Easy, right?)

**VERSUS MAKEUP: WHAT IT LOOKS LIKE vs. WHAT IT IS**

Here we need to introduce a key distinction, one that does not instictively come to mind: the difference between markup and makeup. Strictly speaking, languages such as HTML and XML only tell computers and/or browsers what things are, and not what they should look like\(^{11}\). The problem is that, in many cases, computer applications take on both roles, which can cause confusion. For a long time HTML did just that, until the advent of CSS, the Web development language for Cascading Stylesheets. Now HTML is mainly used to define objects on a Web page while CSS takes on the prerogative of the layout\(^{12}\) (more about that in a few paragraphs).

But, you might ask, why do computers need to know ‘what is what’ in this context? Why should it matter that my browser knows that this is the ‘body’ and that that is the ‘head’? That’s shown by the layout, and as users can see a line in a larger font hovering over a paragraph in a smaller font, they instinctively get that this is the title and that that is the main body of the text. Isn’t that good enough? The user needs to know what is what; computers are just big calculators! (Unless you’re a huge fan of Terminator-like theories about machines developing a consciousness.)

Well first of, modern Web pages look so neat and stylized and coherent (for the best among them at least) precisely because the layout — the ‘makeup’ — can be so efficiently controlled through the use of the markup. Once you have tagged your text with all those <head> and <body> labels (to the point where you might feel something like a forensic pathologist by the end of this), you can then turn to the CSS language and, by the input of formal commands, basically instruct it: ‘All right now CSS, I want you to display all the titles (the <head>-tagged sections) in Segoe UI, size 18, boldface, and all the paragraphs

---


(the <body>-tagged sections) in Gadugi, size 14.’ In this way, the code that you cleverly put together (the combination of HTML and CSS) will apply those instructions to the whole document and display it with unbroken consistency! (Assuming there are no typos in the code, but fear not: HTML is very accommodating.)

That is the smart way of doing things. The not-so-smart way goes something like this:

![HTML code snippet]

**Figure 2.** An bad example of HTML where markup is actually makeup.

Just think: upon display, you suddenly find that the visual rendering of your Web page does not appeal that much to you. Titles are too big, or the paragraphs are too small; it’s all hard to read; it’s not pretty. If you went with HTML as shown in Figure 2... then good luck with systematically modifying all the display parameters for each and all instances of titles and paragraphs *one by one*!

However, if you went with HTML as shown in Figure 1 and had a CSS file in the wings, who rendered everything in a cohesive manner based on such tags as ‘head’ and ‘body’ wherever they appear, then you need only tweak *this one file* in which all your graphical, typographical, chromatic (and what not) specifications are coded. So then, ‘[t]he solution is to separate content from display of information’\(^{13}\). As Seth van Hooland and Ruben Verborgh put it: ‘This is the idea behind the “write once, publish many” principle. [...] The content stays the same, you just use another style sheet which indicates how the different elements of the web page should be rendered’\(^{14}\).

---


But then you might think this is just an efficient way of managing the looks of things and not so much things themselves. But think of other such examples as: If you have properly labeled all the titles in a document with the <head> tag, then you can automatically generate a list of those titles and thus obtain the table of contents for your work! Likewise, you can generate lists of tables and figures. You can also ask a program to systematically check all the hyperlinks (which you tagged as such, with the <url> tag for example) that appear in your text and see whether they still redirect to active Web pages or if any of these went defunct in the meantime. The point is, with careful markup, you can automatize all sorts of commands and operations that would otherwise eat up so much of your time if you had to perform them ‘by hand’. The machine loves nothing more than to execute a precise set of instructions again and again; markup is particularly useful to this end (and it means less grunt work for us).

So then, what do archives have to do with all of this? Well, XML, the eXtensible Markup Language, because it is so very ‘extensible’, allows you to create your own computer language for tagging information! And that is just what archivists did with the Encoded Archival Description, also known as ‘EAD’. EAD is used to encode finding aids and make them available and searchable on an archive’s website. More specifically, it can make the descriptions of archival fonds machine-operable. For example, the <name> tag allows one to mark all the instances of names of people of relevance in a finding aid like so:

This collection contains personal the papers of painter and poet <name>Kathleen van Meulder</name>. It comprises sketches, diaries and various notes, as well as much correspondence between the artist and members of her family, such as her spouse, social critic <name>Lucille Saint-Jérôme</name>, her father, architect <name>Yves van Meulder</name>, and her son and daughter, <name>Skylar van Meulder</name> and <name>Rolande Porter</name>.

Figure 3. A fictional example of the use of the <name> tag in an EAD document.

Maybe you’ve already guessed it: by tagging all the names of individuals who played an important role in the history that pertains to a specific fonds, we can then generate a list of all those names and link them,

---

for example, to authority files, bibliographic records, archive creator profiles and so on. That is only one of the many things EAD can do for you!

EAD: MORE THAN A LIST-MAKER

The other key functions and potentialities of EAD we can best demonstrate by looking at the structure of this standard. Just like HTML, EAD consists of those angle bracket-wrapped tags. The first tag of all, logically enough, is <ead>; it contains all the other tags. Like those Russian nesting dolls, you have to ‘open’ it to see all of the other tags which are nested inside and which neatly encode the information within specific sections.

The three main sections are <eadheader>, <archdesc> and <dsc>. The first one, ‘EAD Header’, does not contain information about the archives themselves but about the finding aid that describes them — so it’s about the metadata, not the data. You can use this section to encode such things as:

● the title of the finding aid (<titleproprer>);
● the name of the archivist(s) who compiled the historical materials (<author>);
● the date when this was done (within <publicationstmt>, <date>);
● the publisher of the finding aid (<publisher>);
● the language of the finding aid (not of the archival materials!) (<langusage>);
● the rules followed during the production of the finding aid (<descrules>);
● information about the creation of the EAD file itself (<creation>).

Then comes the big section about the archives proper, ‘Archival Description’. This is where you’ll find such essential elements as:

- the title given to the fonds (<unittitle>);
- the time frame of the archival materials (<unitdate>);
- the physical and/or digital extent of the materials (<physdesc> and <extent>);
- the name of the entity responsible for providing intellectual access to the materials (<repository>);
- the physical location of the materials (<physloc>);
- the creator(s) of the archives (<origination>).

All these are grouped within the first instance of <did>, which stands for ‘Descriptive Identification’. This one <did> applies to the whole fonds, contrary to those that will come later to describe the various parts of the materials.

Then come various several semi-large sections which were made for typically larger chunks of text and which can bear upon:

- the biography of the archive creator(s) and the history of the materials (<bioghist>);
- the custodial history of said materials (<custodhist>);
- the circumstances in which the acquisition of the materials took place (<acqinfo>);
- the scope and content of the archives (<scopecontent>);
- the work of appraisal which was performed upon them (<appraisal>);
- the way in which they were arranged (<arrangement>);
- whether the fonds will likely grow in the future (<accruals>);
- under what conditions the materials may be accessed (<accessinfo>);
- under what conditions they might be used, for instance whether photocopies can be made of paper documents (<userestrict>);
- the language(s) of the archival materials (<langmaterial>);
- the technical peculiarities of the materials if there are any (<phystech>);
- other archives or finding aids which the reader is advised to consult because they are related to the present materials one way or another (<relatedmaterial> and <otherfindaid>);
- where the originals are kept if applicable (<originalsloc>);
- whether there exist alternative forms (<altformavail>).

Final, there’s the no-less-vital section <dsc>, which stands for ‘Description of Subordinate Components’. This is where the Russian doll structure is most clearly illustrated, for this is where you’ll be able to encode all the hierarchical groupings into which you’ve arranged the materials of the fonds as sections, sub-sections, sub-subsections… You can do this by the use of the <c> element, which stands for ‘Component’. Here is an example of a fictional <dsc> section with a fairly deep-reaching structure:

```
<archdesc>
  <head>
    Hierarchical groupings of the materials
  </head>
  <did>
    <unittitle>1. House Committee on Interior and Insular Affairs</unittitle>
  </did>
  <c>
    <did>
      <unittitle>A. General Information Department</unittitle>
    </did>
    <c>
      <did>
        <unittitle>1. Statistics</unittitle>
      </did>
      <c>
        <did>
          <unittitle>2. Agencies and Branches</unittitle>
        </did>
      </c>
    </c>
  </c>
</archdesc>
```

**Figure 6.** A fictional example of a <did> section within <archdesc>.

Notice how all the <c> sections and subsections are embedded within each other? Of course there can be ‘childless’ sections with no supersections above them and/or ‘orphan’ sections with no subsections...
below them. So really, the <dsc> section mirrors the structure of the fonds, which is usually placed at the end of a finding aid after all the introductory information and materials.

All in all, EAD provides a whole template for encoding already existing finding aids and for writing up new ones. This can lead to problems if your own model greatly differs from this one, but EAD can be adapted in many ways.\footnote{See also, for a case where the structure of the original finding aids was completely revamped, D. MEISSNER, ‘First Things First: Reengineering Finding Aids for Implementation of EAD’, The American Archivist, 60, 1997, p. 372-387, \url{https://doi.org/10.17723/aarc.60.4.6405275227647220}.}

Being the version that was around for the longest period of time, EAD 2002 received a good deal of criticism. Authors pointed out that, because only too few tags were actually mandatory, the form of EAD documents varied enormously from one repository to another, thus leading to sloppy documentation in some cases and heterogeneity at large.\footnote{E. J. SHAW, ‘Rethinking EAD: Balancing Flexibility and Interoperability’, New Review of Information Networking, 7, 2001, p. 117-131, \url{https://doi.org/10.1080/13614570109516972}; L. FRANCISCO-REVILA, C. B. TRACE, H. Li and S. A. BUCHANAN, ‘Encoded Archival Description: Data Quality Analysis’, Proceedings of the American Society for Information Science and Technology, 51, 1, 2014, p. 6.}

Also, the sheer scope of the standard (with its 146 tags) and the counter-intuitive look out of XML and of some tag names threw many archivists into disarray and hindered the early adoption of EAD.\footnote{E. YAKEL and J. KIM, ‘Adoption and Diffusion of Encoded Archival Description’, Journal of the American Society for Information Science and Technology, 56, 2005, p. 1436, \url{https://doi.org/10.1002/asi.20236}; E. DOW, ‘Encoded Archival Description As a Halfway Technology’, Journal of Archival Organization, 7, 2009, p. 109-110, \url{https://doi.org/10.1080/15332740903117701}.}

As Sonia Yaco aptly said to summarize this situation: ‘it’s complicated’.\footnote{S. YACO, ‘It’s Complicated: Barriers to EAD Implementation’, The American Archivist, 71, 2008, p. 456-475, \url{https://doi.org/10.1080/13614570109516972}.}

That being said, back in the late 1990s and early 2000s, many of the obstacles to widespread EAD adoption pertained not so much to EAD itself than to the situation of archival repositories in general, many of whom lacked full-fledged computer infrastructures and tested-and-tried workflows, especially the smaller ones.\footnote{E. J. SHAW, ‘Rethinking EAD’, p. 122; S. YACO, ‘It’s Complicated’, p. 457-459 and 461.}

Adoption of EAD 1.0 had also been very limited,\footnote{J. ALISON-BUNNELL, ‘Review of Encoded Archival Description Tag Library — Version EAD3’, Journal of Western Archives, 7, 1, 2016, Article 6, p. 2.} so this was a first start with XML for the majority of archivists. Nowadays however, there is no denying that EAD truly has become the most
widespread norm for archives’ metadata in Europe and the United States, and because EAD3 solves many issues that weakened EAD 2002, it is likely to remain the dominant standard.

MORE DATA THAN YOU COULD SHAKE A STICK AT! THE EXTRA-LARGE DATASETS OF SOCIAL SCIENCES

Now that we know more about XML and EAD, we need to look at DDI, the Data Document Initiative standard. DDI was developed for the needs of the social sciences, especially for studies that yield quantitative data, from which charts and statistics can be easily derived. Such data constitute the bulk of the social sciences. Just like EAD, DDI was devised in the late 1990s; and just like EAD is meant to encode the typical element of documentation of its field of study, i.e. the finding aid, DDI provides a template for encoding one of the most frequent metadata document in the social sciences: the codebook.

Studies and research projects whose methodologies involve collecting much numerical data are most of the time documented with ‘books of codes’, or codebooks, which serve as ‘dictionaries of codes’. But what kind of ‘codes’ are we talking about?

Let’s say you want to learn more about the people who smoke in a certain area, the city where you live for example. You would need to procure data from a representative sample of the local population and gather information about these people such as their age, their gender, their income, their family circumstances, their medical history, their place of residence and so on. These features would then constitute the ‘categories’ of your investigation, which you defined before launching the collection

process, while the actual data that you gathered afterwards constitute your ‘values’ or, more frequently in this field of study, your ‘variables’\(^\text{29}\).

Should you enter those categories and variables inside a spreadsheet program, such as Microsoft Excel or SPSS, it could look something like this for the number of individuals who smoke out of all the people you interviewed in the course of your study:

<table>
<thead>
<tr>
<th>Age of Smokers</th>
<th>16-21</th>
<th>22-30</th>
<th>30-40</th>
<th>40-50</th>
<th>50-60</th>
<th>60-70</th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>357</td>
<td>835</td>
<td>1,037</td>
<td>569</td>
<td>324</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** An example of data organization in a chart from a fictional dataset.

So you see that, in fact, ‘categories’ designate the headings and subheadings of columns (here ‘Age’ and the various age ranges) and that ‘variables’ designate the data *per se* (how many people between the ages of 16 and 21 smoke, between the ages of 22 and 30, etc). However, most codebooks do not use such a neat, layperson-friendly format — for two reasons.

First off, thorough research methods require that social scientists keep track of all of the information about all of the *instances* of elements that they are investigating in their surveys. In other words, if you are going to study what people do (if they smoke for example), you’re going to need to record the information for each of those categories (age, gender, etc) *for each and every individual!*

In our fictional example shown in Table 1, we have surveyed at least 3,289 individuals (and those are only just the smokers, so we are not taking the non-smokers into account here). This means that, if you did your research well, you should have record cards like this...

<table>
<thead>
<tr>
<th>Record number</th>
<th>00357</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>18</td>
</tr>
<tr>
<td>Gender</td>
<td>F</td>
</tr>
<tr>
<td>Monthly income (gross)</td>
<td>n/a</td>
</tr>
<tr>
<td>District</td>
<td>Farnham</td>
</tr>
<tr>
<td>Family situation</td>
<td>Student housing, 2 flatmates</td>
</tr>
<tr>
<td>Medical history</td>
<td>Asthmatic</td>
</tr>
</tbody>
</table>

**Table 2.** An example of an individual record from a fictional dataset\(^\text{30}\).


\(^{30}\) Notice how the name of the respondent is not recorded. It likely was at one point during the study, mainly for practical reasons (especially for follow-up procedures, in case it becomes necessary to contact the person again) but it is always imperative at the end of a research in social studies to anonymize respondents. Even so, it often
... at least 3,289 times! — even if some of the data are missing (in case a respondent was not able or willing to provide a certain piece of information for example).

Table 1 contains sums, i.e. aggregated data. When social science research data is presented in the media (on TV, on the radio, in non-specialized literature...), data usually takes this form: statistics. But such aggregates necessarily stem from large quantities of individual records like this one, and all must be neatly recorded and formatted so that the information can be verified, preserved and re-used.

The second reason is that ‘categories’ are seldom put in such common, human-readable terms as ‘Age’, ‘Gender’ etc. That is because most researchers in quantitative social sciences use powerful software programs like SPSS, SAS or R, which were specially designed for handling, compiling and manipulating large volumes of complex data. And because there are so many subdivisions within the data and because researchers need to be able to aggregate the data in various ways while keeping track of everything — say for instance that you want to know exactly how many individuals between the ages of 30 and 50 who have lung-related medical conditions and who live in two particular districts smoke — these programs need to record all this information and categorize it in a systematic and unambiguous manner. That is why they generally assign generic identifiers to these categories, such as ‘V12’ or ‘Q0A’.

And these are the ‘codes’ which the codebook helps us elucidate. Datasets in the social sciences usually feature one or several spreadsheet files with long, cryptic lists of numeric values and no-less-cryptic names of columns such as these. Without the ‘dictionary of codes’ that is the codebook, it is generally impossible for users to work out their meaning. Without proper context it just looks like a nonsensical jumble of numbers and letters — see for yourself:
Table 3. An excerpt from the United States’ 2010 Decennial Census.

Right here, the first column seems intelligible enough: as indicated by the title, ‘LANGUAGE’, it’s a list of varieties of American Indian languages spoken on American soil. But what about ‘POP’? Might refer to respective populations, but can we just rely on an assumption? And what about ‘VAPOP’, ‘MVAPOP’, ‘VACIT’ and all the others?

Obviously, a lot of documentation and study hours are required to grasp the general context in which those data were collected and the particular meanings of those rows and columns. Hopefully there is a formal, standard way to document such things! Right?

WHAT’S NICE ABOUT STANDARDS IS THAT THERE ARE SO MANY TO CHOOSE FROM!

As noted by Karsten Boye Rasmussen and Grant Blank: ‘Analysing undocumented data is impossible’:

‘But even with documentation the process of analysis is often difficult (e.g., the user must be able to understand the jargon of the documentation), error prone (e.g., the documentation might be imperfect, and/or the user might misunderstand the documentation), and time-consuming (e.g., users have to familiarize themselves with the documentation and the software). Providing a standard format for machine-readable metadata can reduce errors and simplify analysis. [For] these reasons, the DDI is intended to become the cornerstone of many scientific infrastructure projects.’

For this is where DDI comes into play. With its three big sections, its shortened tag names and its three already existing versions (EAD 1.0, EAD 2002 and EAD3), EAD might seem rather complex to you. In this respect however, DDI takes the cake!

First of all, it is worth noting that there are no less than seven version of DDI. It’s not as bad as it looks, for there are two main ‘branches’ with incremental versions within: on the one hand, DDI-Codebook, which is very ‘document-centric’, and on the other hand, DDI-Lifecycle, which encompasses the whole lifecycle of research data from the moment when they are conceived to the moment when they are finally deposited into an archive and ready for re-use. DDI-Codebook comprises DDI 1, 2.0, 2.1 and 2.5. DDI-Lifecycle extends from DDI 3.0 to DDI 3.1 and 3.2. Also, DDI 4 is coming!

Why such a proliferation of standards, you might ask? How can things be ‘standard’ if there are so many different versions in circulation?

The answer is complex because it hinges on the evolution of research in social sciences and the whole development of computer technology throughout recent history. The picture greatly changed, what with the many transformations that affected academia in general in the last few decades and, more particularly, the emergence of a growing network of research infrastructures. New needs and problems arose — think about big data, interoperability of formats, technical and technological obsolescence, increased focus on data and privacy protection (we’ve all heard about the GDPR now) and so on... From here, the DDI Alliance, which coordinates the development of and provides technical for the DDI had to take initiatives to address these challenges. This entails improving DDI, updating it, shaping it so that it can tackle those issues. Like all technologies, DDI is perfectible.

So then, what are we working with here? Well, back when I was instructed to put together a crosswalk between DDI and EAD, several choices had to be made. The first was, which version of DDI to choose? By

34 See the website of the DDI Alliance for more information about the different versions, their history and their field level documentation: http://www.ddialliance.org/.
gathering information about the way social sciences function nowadays, the following correspondence seemed to emerge between common types of data and of metadata in their respective domains:

<table>
<thead>
<tr>
<th>Field of study</th>
<th>Social sciences</th>
<th>Archives (Archival Science)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Print Datasets</td>
<td>Archive fonds</td>
</tr>
<tr>
<td></td>
<td>Digital Born-digital documents + scans</td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td>Print Codebooks</td>
<td>Finding aids</td>
</tr>
<tr>
<td></td>
<td>Digital DDI</td>
<td>EAD</td>
</tr>
</tbody>
</table>

Table 4. Parallels between the kinds of data and metadata used in the social sciences on the one hand and in archives on the other hand.

If the crosswalk was to make some sense, it would have to bridge two objects that are somewhat similar in shape and purpose. This seemed to be the case when laying things out in this manner: just like finding aids, codebooks are discrete, originally book-like documents that help readers appropriate source materials. A ‘codebook’ certainly felt closer to a finding aid, intuitively, than the general ‘life cycle of research data’, since the latter concept felt so abstract and, in itself, devoid of a stereotypical form. All in all, both DDI-Codebook and EAD are ‘document-centric’.

That being said, when you think about it, finding aids themselves do document the ‘lifecycle of archives’: they trace their origins — who produced them, where, when, why, and what do they consist of, what do they look like, what are they all about — up to the point where they were delivered to a repository — and what happened there and then too, how were they arranged, where are they stored now, whether we can consult them, etc. So then, it would not have been entirely inappropriate to draw parallels between EAD and DDI-Lifecycle in this respect.

But because a choice had to be made and because, in the way both variants of DDI were structured, Codebook felt much simpler and easier to handle to a layperson than Lifecycle did, Codebook was the winner.

Further, the latest version of DDI-Codebook, No. 2.5, was selected because it builds atop all of

---


37 Incidentally, it is commonly admitted in DDI-savvy circles that the vast majority of currently existing DDI files is still DDI-Codebook-compliant — quite simply because DDI-Codebook has been around for longer than DDI-Lifecycle. That being said, the Consortium of European Social Science Data Archives (CESSDA) pushes a lot for DDI-Lifecycle and various elements seem to indicate that the day of DDI-Codebook are numbered, at least on the
the other DDI-Codebook versions, so that a software program that can process DDI 2.5-compliant files can also process files that follow the rules of DDI version 1.0. As for EAD, I selected EAD 2002 simply because this was the version of EAD in use at the State Archives at this particular moment.

THE BURDEN OF THE PIONEER, OR ‘HOW I WISH SOMEONE HAD MADE THAT MISTAKE BEFORE I DID’

At this point, it might seem like I forgot to answer an important question: wasn’t there already a mapping from DDI (be it Codebook or Lifecycle) to EAD, or the other way around? There is actually!

Part of the multi-faceted crosswalk which was put together by the Emory University Libraries & Information Technology department links between EAD and the Dublin Core metadata standard — which is yet another standard, although an extremely simple one, perhaps the simplest one around as it consists of just 15 tags and was conceived to describe about any kind of document in a summary way. But really, that’s that: the Emory U mapping bridges between only about 15 tags from EAD and DDI among other standards. This seemed insufficient to me, as I envisaged a much more comprehensive transfer of information from the DDI documentation of social science dataset over to new, EAD-compliant records that would be channeled through the State Archives’ pipeline in order to enrich the institution’s collections. As I uncovered, while there are very light DDI files in circulation, some can be extremely long, especially when the documentation goes as deep as the variable level — which is pretty much the equivalent of our ‘item level’ in archive fonds. For that reason I could not be content with this one minimalistic mapping, commendable as it is otherwise.

This however is where I must confess that, in spite of those wise reflections, my methodological approach turned out to be somewhat lacking. Because I was a young, feather-brained researcher with little hands-on experience, and because I was given almost all latitude to determine how best to

long term. The author gathered this anecdotal evidence by attending such events as the 9th EDDI Conference in Lausanne, 2017, and the work sessions of the working group CESSDA Metadata Management (CMM) Phase 2.


41 For example, the DDI file for the United Kingdom’s 2002-2015 National Travel Survey is no less than 66,951 lines long — Department for Transport, National Travel Survey, 2002-2015, 11th ed., distributed by the UK Data Service, serial number 5340, https://doi.org/10.5255/UKDA-SN-5340-7.
proceed, I found myself free to determine what was the best angle to tackle the problem at hand — and it became clear at one point that said angle mattered less than turning up with actual results because time and funding were short.

I like to say that I went at it in a very heuristic manner, but this is mostly scientific gloss to express the fact that I had all too few ideas about what I was doing exactly, and I eventually decided to map all the tags of DDI-Codebook 2.5 over to EAD 2002.

That’s roughly 2,000 tags (elements and attributes) in total. Good times.

Later on, I discussed the matter with the Head of Computer Developments at the State Archives, a pragmatic, down-to-earth engineer. He explained to me that completely and entirely mapping a DDI file over to EAD made little sense: if we were in a situation where we would want to collect all or most of the information contained within DDI files, then we would build a system that can read DDI as-is. For if our goal was to make use of all the information inside there, then we could adapt our tools to the container, and not the other way around, since the container — DDI — had been especially devised for that kind of information, giving it precedence over whatever external system or infrastructure. A parallel would be: we have this huge source of fuel, but it won’t work in our engine! So we must build a large refinery that will process the fuel in costly, clunky and intricate ways in order to alter it so that it can power our engine! … Or… you know… we could simply… build a new engine?

A COHORT OF BRIDGES (BUT EACH WITH A NEAT LITTLE SIGN): EXAMPLES OF MAPPINGS FROM DDI TOWARDS EAD

So there you are: a 2,000-line long Excel file that contains all of the DDI-Codebook 2.5 tags in one column with EAD 2002 equivalents in the next column. It also has a column with identifying numbers for all DDI elements — formatted as 1, 1.1, 1.2, 1.2.1… — plus another column that describes the DDI elements and another one for problems of all sorts.

The column with EAD elements gives the full ‘path’ for the machine to ‘reach’ the final tag. If you look at Figure 5 for example, supposing you want to retrieve the information concerning the repository that provides access to a fonds, then the path will be ‘ead / archdesc / did / repository’, as the computer will need to jump through all of these elements — jump down these elements, you might say — to reach its final destination and read what’s stored in there (here, ‘State Archives of Belgium’).
Let’s illustrate. Let us take DDI element 2.3.1.16.9 (deep one!). The path for that element reads as follows: \textit{stdyDscr / method / dataColl / sampleFrame / useStmt}

- \textit{<stdyDscr>}

  stands for ‘Study Description’. This is the second big section out of five in DDI-Codebook and, simply enough, its purpose is to record information about the study in the course of which the dataset which that DDI file documents was put together.

- \textit{<method>}

  stands for ‘Methodology and Processing’ and, quite self-explanatory, it’s about how the various operations which constitute the study were performed.

- \textit{<dataColl>}

  stands for ‘Data Collection Methodology’. This subsection addresses the ways in which the data were collected (by phone, through online questionnaires, by approaching people in the streets, etc).

- \textit{<sampleFrame>}

  stands for ‘Sample Frame’. While the word ‘sample’ refers to the part of a larger ensemble which you have surveyed for your study (e.g. a sample of a certain population), the ‘sample frame’ designates the general frame which was used to constitute your sample. For example, if you conducted some phone interviews, then you likely relied on a phone book of sorts: that is your sample frame.

- \textit{Finally, <usestmt>}

  stands for ‘Use Statement’. This section was made to record all the information that the (potential) user of the dataset should be aware of before they proceed to request permission to download a copy of it.

Now that we finally got there to this pretty deep level, let’s see what we can find.

First we have \textit{<citeReq>}, which stands for ‘Citation Requirements’. How do you cite a dataset in your list of references? This element should tell you. For example, note no. 33 of this text contains the (slightly adapted) preferred citation for the dataset which Table 3 comes from. Notice how it has a DOI (a Digital Object Identifier)? DOIs are unique identifiers which the indexing robots of companies such as Thomson Reuters or Elsevier harvest in order to calculate the so-called ‘impact factors’ of researchers, journals and publications. Impact factor are supposed to determine how influential those three are compared to the others, so the idea is that you’ll rack up more points if you manage to get published by Oxford or Cambridge University Press than by a young, budding and open access editor.

So this means that even datasets can be aggregated into this whole calculation thing to determine (most rigorously and objectively) what a good researcher you are (forgive the sarcasm). These types of documents now factor in in those various metrics and are automatically taken into account by bots who

---

\textsuperscript{42} See the DDI 2.5 online field documentation:  
produce these scores. This can give you an idea of how interesting (how *vital*, in fact) it now is for scientists to put their data out there.

Back to `<usestmt>`, we have other such elements as the fairly self-explanatory `<disclaimer>`; `<confDec>`, which stands for ‘Confidentiality Declaration’; `<contact>`, which is meant to record the ‘Contact Persons’ for the dataset; `<specPerm>`, in case you need ‘Special Permissions’; among others.

Now how do we connect all these to EAD elements? Well many correspondences are fairly obvious (as people who like to conflate idioms like to say, ‘it’s not rocket surgery’):

- For instance, DDI’s `<citeReq>` naturally corresponds to EAD’s `<prefercite>` (‘Preferred Citation’).
- DDI’s `<disclaimer>` neatly fits within EAD’s `<accessrestrict>` (‘Conditions Governing Access’) section — just don’t forget to add a `<p>` (for ‘Paragraph’) and with it a `<head>` (‘Heading’) in order to get a nice-looking section with its own title and paragraph. (Also, this `<head>` and `<p>` dynamic is just the same as what we said earlier about HTML and making formatting issues easier to handle!)
- DDI’s `<confDec>` also logically has its place in EAD’s `<accessrestrict>`, although you’ll want to distinguish it from the contents of `<disclaimer>` with a meaningful title.
- The information recorded in DDI’s `<contact>` can also go in EAD’s `<accessrestrict>`, although we can make use of other EAD elements which will make the nature of this data even clearer: add a subsection `<address>` in `<accessrestrict>` and tag the information within with `<addressline>` elements, with which you’ll be able to retrieve and sort out physical or post addresses, e-mail addresses, phone numbers and what not in a more systematic manner.
- Finally, DDI’s `<specPerm>` can go into EAD’s `<userestrict>` (‘Conditions Governing Use’), which is close yet not identical to `<accessrestrict>`: the latter should inform users about the potential need to make an appointment or receive permission from donors to access archival materials, while the former should tell you for example whether you can make photocopies of the materials, take pictures of them or cite them.

To summarize, this is what the two main columns of the mapping will show you for element 2.3.1.16.9 `<useStmt>`:

<table>
<thead>
<tr>
<th>DDI-Codebook 2.5</th>
<th>EAD 2002</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;citeReq&gt;</code></td>
<td>ead / archdesc / prefercite / head + p</td>
</tr>
<tr>
<td><code>&lt;disclaimer&gt;</code></td>
<td>ead / archdesc / accessrestrict / head + p</td>
</tr>
<tr>
<td><code>&lt;confDec&gt;</code></td>
<td>ead / archdesc / accessrestrict / head + p</td>
</tr>
<tr>
<td><code>&lt;contact&gt;</code></td>
<td>ead / archdesc / accessrestrict /address / addressline</td>
</tr>
<tr>
<td><code>&lt;specPerm&gt;</code></td>
<td>ead / archdesc / userestrict / head + p</td>
</tr>
</tbody>
</table>
They say the intelligent person learns from their mistakes, but the wise person learns from the mistakes of others. Hopefully this was an occasion for me to become more intelligent! I will have to leave the wisdom to you.

The engineer I mentioned earlier further explained to me that ‘overriding’ the DDI files with an army of EAD clones made little sense because you had to envisage things like this: the future SODA entity, which will be part of the State Archives one way or another, will manage the DDI files on the one hand for the specific needs of social science researchers, and it will also, on the other hand, create an opportunity to enrich the collections of the State Archives by deriving some of the information contained in the DDI files and by processing it through the Archives’ pipeline thanks to a DDI-EAD mapping.

Now what specific pieces of information would that be? Which particular elements of DDI should transfer to EAD in order to align with the rules and practices that govern the management of the Archives’ catalogue? Well that will be for next time! Refining this big ol’ clunky 2000-line long mapping is the next step. Now we’ve got to make it fit for our uses, which is one of the main criteria of quality in modern industry and project management: a product must be consistent with the initially set requirements and it ought to satisfy the user’s needs to the maximum for it to be a quality product.\(^{43}\)

For now, it seems like this huge file has little theoretical or practical potential. At worst it can serve as a case study for, let’s say, asymmetric time allocation\(^{44}\). At best it will be the foundation for a shorter, cleverer, all-round better mapping that actually meets the requirements of the future Belgian data archive. Further, the mapping will likely be published in open access format, making it available to whoever might want to re-use it, improve it or adapt it for other purposes. From here, the next steps include:

- Refining the mapping into a more ad hoc one;
- Formatting the larger mapping (and potentially the second one, too) to make them publication-friendly, hopefully in open access;


\(^{44}\) You could also say ‘giving a fastidious young fool too much time to do a simple task’.
Integrating the new mapping in the technical and technological architecture of the upcoming Belgian data archive for the social sciences.

This paper introduced an early technical realization in the course of an ongoing project. Because the project is still running, and because it involves different actors and disciplines, and because the mapping’s technicality reflects this quaint but fruitful union — for all these reasons, much contextualization was required before the mapping proper could be discussed.

That being said, the idea of such mapping accounts for the complexity of the SODA endeavor, as well as for its rich potential. Two worlds — archives and social sciences — and two types of institutions — a large federal entity on the one hand and several universities who are related to linguistic communities on the other — are coming together to build a new kind of administrative and scientific entity from which both types of actors will benefit. However this entails devising practical means in order to rely on the strengths of both worlds so that both stand to gain something from this venture.

The Belgian political and administrative landscape is reputedly convoluted and hard to navigate, so the legal and organizational challenges of the SODA project are reflected in the ensuing technical and technological constructs and adaptations, such as this mapping.

---