eHumanities
Royal Netherlands Academy of Arts and Sciences

CEDAR project

Technical report: Miniproject advances, Iteration 2

Ashkan Ashkpour

Albert Meroño-Peñuela

Revision history

Version	Date	Description/changes
0.1	22/01/2013	First draft
0.2	27/01/2013	Second draft
0.3	20/02/2013	Third draft
1.0	21/02/2013	Final version

Contents

Introduction

Goal

The goal of this document is to provide an overview of the second iteration of the CEDAR miniprojects. It describes all implemented steps from a previously agreed roadmap, with an emphasis on results (with external files gathering them) and issues encountered (with their current status and possible solutions).

Scope

Within the CEDAR project, an iterative development cycle has been established to study, correct, harmonize and publish the Dutch historical censuses (1795-1971) in a constrained scale environment. We refer to it as the *miniprojects*. Miniproject 1 (MP1) is more focused on data quality and retrieval from one single census occupational table (Noordbrabant in 1899), while miniproject 2 (MP2) wants to harmonise demographical tables from different census years to enable longitudinal queries (namely, 1859,1869,1879 and 1889).

References

Agreed roadmap

Tables for MP1

Tables for MP2

Whole dataset

Background

CEDAR closed the first miniproject iteration by July 2012. The results of the first iteration can be found here. The first iteration essentially consisted of the selection, conversion to RDF, and querying of the mentioned census samples. The second iteration was planned (see document here) to scale up the system, solve issues and follow research paths devised during the first one.

Overview

The document is divided as follows. First, we describe the development of the second iteration of the miniprojects following the structure of our agreed roadmap. We put a special emphasis on issues found, highlighting whether we found a workaround. If we did, a pointer to an external file or algorithm is provided. Otherwise, we explain why the solution is not trivial to be obtained, and thus why the problem is not yet solved, gathering a set of further paths to follow. In this respect, in this document we will use the following layout when spotting such problems:

Summary of the problem encountered	● One possible solution ● Another possible solution

The outline of this development is heavily inspired by this agreed roadmap document.

Second, we collect all our relevant findings in a set of appendixes. In the first we present an inventory of produced files, like summary tables or query lists, and provide online links to reach them. In the second we gather a list of source code repositories of the several scripts developed. Finally, we provide an overview of all problems and solutions presented across the iteration description, that can be seen as a further work section.

Finally, a glossary with terms we consider important to clearly define is included.

Miniproject developments

STEP A. Inventory, quality of the tables, and conversion to RDF

Inventarisation

In order to have a clear summary of the contents of the dataset, we analyzed it systematically with a script we wrote, TabExtractor. TabExtractor is meant to offer a summary of a collection of Excel spreadsheets at the data and metadata levels. The results of this analysis count 2288 tables with 33283 annotations in 507 Excel files. This analysis can be easily repeated if new versions of the dataset come, by simply executing the script again on that new version. A complete result file is located in the project Dropbox.

Dropbox/CEDAR/Inventory/Census summary.xls

TabExtractor also generates inventory metadata about annotations contained in these Excel sheets. For an annotation summary and distribution of these annotations over the census tables, see

Dropbox/CEDAR/Inventory/Annotations/Table annotations distribution.xls

For a complete dump of the textual content of these annotations, plus additional metadata and English automated translations, see

Dropbox/CEDAR/Inventory/Annotations/annotations-dump-translation.csv

The naming of the digitized Census tables is currently done by using a coding scheme which does not give information about the contents of the table itself (see example below). This may not be a problem when users browse the volkstellingen.nl website where descriptions are given, butit is a problem for CEDAR where we need to download and work with a couple of thousand tables offline. Identifying the tables, by changing the names in something which gives more information would save us a significant amount of time when *working* with the files. However it will also make it easier to document and keep it understandable for all. For example the table containing information with regards to the 'Population of the Netherlands by religion" for 1971 has the code "VT_1971_B3_T1". Or the 'Population census of the province Noord Holland with age and civil status" information is coded as "VT_1899_04_H4". This ways of coding is very systematic but not self-descriptive. As this information is meaningless to the users, changes have been made to *all* the Excel files in the context of CEDAR (1795-1971). Although this makes it easier to work with , it is less structured compared to the original coding. However by keeping track of the original names we can easily reconstruct the file names. See examples of the above files with more readable information:

VT_1899_04_H4.xls	VT_1899_Nh_age_civils.xls

VT_1971_B3_T1.xls	VT_1971_Pop_by_rel.xls

The new file names include; the type of census (VT, BT or WT), the year and up to 3 variables contained in that specific table. Moreover, the new names are also very useful to use in the context of the HISTEL project. The Excel file containing all the translations can be found here:

Dropbox/CEDAR/MiniProjects/Mini Case 2/Census Table Name List.xls

The study of the content of the tables and their inventorisation using TabExtractor produced several inventory files with spread content. These files are not easy to follow, making it difficult to the user to get a clear overall picture of the dataset.

Inventarisation produced a lot of messy spreadsheets	● Reimplement TabExtractor to produce one single coherent database with the whole census inventory

Checking of tables

One of the first steps in the second round of the mini projects concerns the quality of the tables. During the digitization process to images and manual processing to Excel, different types or errors / mistakes were made (source and data entry errors). In general, two main checks exists. The first option which takes significantly more time and effort, and which has been the focus of Tom Vreugdenhill concentrates on the actual content of the sheets, i.e. on the cell level, comparing the numbers of the Excel files with the original sources and if needed adding annotations to it. The other perspective is to look whether the tables are structurally correct, i.e. do we have shifted or merged columns and variables ?

In order to continue with the mini projects this option is not feasible at this stage. This project cannot wait for the double check of the accuracy of the digitization and infinitely incorporate new versions. This is an ongoing process and will take an unknown amount of time according to Tom. Furthermore we do not have other resources to do this. We aim to provide a workflow where future versions, corrections, or other alterations to the data which result in a new version of the file can be incorporated.

Prior to starting this process we checked for an inventory of all the tables which have currently been processed and checked by Tom. As far as we know there is no available list but Tom shared which year's still have to be checked. Unfortunately, in the case of the mini project, the years 1889, 1869, 1879 are not checked yet and could contain many errors and lacking figures.

We do the checking by comparing the Excel files (1 table) with its corresponding images of that table (from the books) which are on volkstellingen.nl. This is currently a *manual* process. Accordingly i.e. 1889 alone has over 200 images, which represent *one* single table. For mini projet two we have five tables with over 500 images to be compared. As we have over 2200 *tables* it is not feasible to continue this way. Even using a sample would result in more than ten thousand images to be compared manually.

However, the years with relatively less data have been compared quite extensively; for the other years we choose a random sample and specifically look for 'bad translations' of the tables into the Excel files, i.e. shifted columns/rows/cells, missing variables , annotations in columns and rows etc.. (Sometimes variables are not included in the original, i.e. if there are only 3 women in a certain category they were *joined* with another variable instead of creating its own variable). Moreover, although a source oriented representation is applied, the images are not always represented as a one to one copy. For example in some cases, the Excel files have joined separate variables into one string which makes it very difficult to extract the right information. Therefore, the focus of the checking is not on individual cell level where we compare the actual numbers but the detection of structural problems.

The two main methods are either manual or automatic, however none of these methods works as needed. Doing this manually is too absorbing and time consuming in the framework of the PhD's; automating this process still requires a (substantial) amount of manual corrections as we expect however it is necessary stage. Some options to do the checking which come to mind are crowdsourcing, outsourcing (for a relatively small fee) or even with students which look at the possibilities of digital conversion methods and checking of tables. In the context of the CEDAR project this still remains an open problem.

Conversion of tables to RDF

We used TabLinker, a supervised Excel/CSV to RDF converter by Data2Semantics, to convert the census Excel files to RDF. To do so, all Excel spreadsheets must be formatted beforehand one by one, using predefined cell styles to draw bounding boxes around sheet areas with common contents (like column and row headers, data, properties, etc.). Since the datasets for MP1 and MP2 are small, this task was performed manually without a high cost.

As a successful result, we produced a set of TTL files containing RDF data of MP1 and MP2 tables.

Dropbox/CEDAR/MiniProjects/TabLinkerOutputFiles

However, during the conversion process two issues were encountered that remain unsolved. First, manual markup of the entire dataset is impracticable, since it would require a very high labour cost. How to scale from small samples to the whole dataset is one important research question of this project.

Conversion to RDF does not scale. We cannot mark manually the whole dataset.	● Crowdsourcing ● Students labour ● Outsourcing

Second, due to the verbosity (i.e. number of produced triples to describe a cell) of the RDF graph data model TabLinker produces, it needs very high amounts of memory to convert the Excel files. A very high number of files of the dataset are currently above a reasonable threshold: the requirement is about 1GB of memory per 1MB of Excel spreadsheet, which makes dealing with files higher than 6MB unaffordable for most laptops or desktops. This means that these files cannot be converted without expensive hardware.

TabLinker takes about 1GB of memory per 1MB of Excel file size. Conversion of big Excel files is unmanageable with our current hardware.	● Get additional hardware resources ● Rewrite TabLinker to be less memory consuming (without giving up verbosity)

Additionally, we identify the problem on how to integrate all the conversion processes as they are right now. Ideally, the whole framework of CEDAR should be deployable in one single straight-forward step, in order to be able to generate, convert, harmonize and publish census Linked Data executing automatically the entire workflow once (with user intervention as needed). We have begun the study on how to integrate all scripts and tools in a single branch. The placeholder for this artifact, which should evolve into an integration platform, is this repository.

STEP B. Census annotations

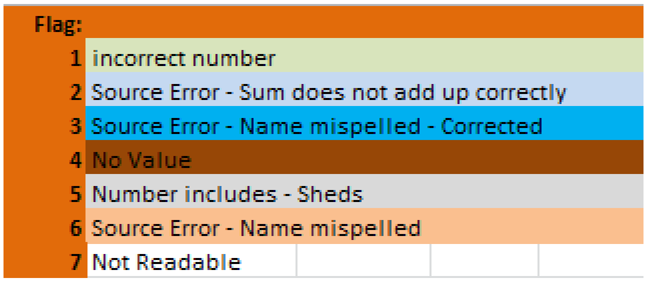
Annotations

In order to deal with the various annotations throughout the census tables we have applied the "Flag Classification System" proposed by Kees Mandemakers. This system uses standard numbers (original values), interpretations and flags to indicate the type of correction when dealing with annotations. The analysis of the census revealed 33,283 annotations in total for the census years 1795-1971. The annotations range from incorrect numbers which have been corrected, comments (e.g. this number contains 1 shed and 3 barns), or interpretations which give additional information mostly based on some expert knowledge. Currently (this is an ongoing process) 89% of all the annotations are 'textual' (i.e. comments or interpretations on the text) and 11% are numerical annotations which refer to wrong numbers in the cells. However, at this time we cannot distinguish between annotations made originally in the source (which were also digitized later) and 'new' annotations. We have to consult with experts for this issue.

In order to integrate the various annotations we need a common classification system and put the different kinds of annotations into meaningful groups. In the case of the mini project 2 we have experimented with this for the given 4 years (1859,1859,1879,1889) and extracted all the annotations, see

Dropbox/CEDAR/MiniProjects/Mini Case 2/Flag Classification System/Annotations from the mini case files.xlsx

The "plaatstelijke indeling" of the Populations Census (1859,1869,1879,1889) currently has around 50 annotations for the mini case years. This is a very small number compared to the total, i.e. 1889 alone has over 14000 annotations. Thus far, this process has been done manually however to deal with the large amounts yet to be classified, we need better and (semi) automatic solutions. Especially in the future, where annotations will be added and more censuses are digitized. The annotations were consolidated and given correct names and a corresponding Flag Number. For example, in case of ' incorrect numbers' we have assigned the Flag "Code" 1, or Flag 4 for "No Value". In total we have classified 7 Flags from the mini case 2 years and classified by using Flags and corresponding Color Codes. See Example:



Flag:	
1	incorrect number
2	Source Error - Sum does not add up correctly
3	Source Error - Name mispelled - Corrected
4	No Value
5	Number includes - Sheds
6	Source Error - Name mispelled
7	Not Readable

Annotations are not made consistently throughout the census which makes it difficult to automatically group certain annotations into a Flag number. For example :

Bronfout. De gedrukte bron geeft 501. Optelling van de kolomdata geeft 441.
de gedrukte bron geeft 1690. Optelling van de kolomdata geeft 1590.

The above annotations are both from the Populations Census (plaatselijke indeling) of 1859 and have the same meaning (i.e. Source error – sum does not add up) but differ in the way they are noted. The goal is to gradually expand this Classification system until all types of annotations have an assigned Flag Number. In the above example we give this Flag # 2.

One of the challenges is how to interpret these annotations as they are sometimes very detailed. Ideally we would like to use the expert knowledge of key users which have been annotating the Census over the past years (especially experts like Tom Vreugdenhill). The next step should be to look whether we can do this process mainly automatically and which percentage (of a larger annotations file with annotations e.g. 1899) can be classified. The remaining annotations are those which cannot be matched based on labels and would require a human input.

Data model

We improved the data model of the first iteration by adding annotations representation using the Open Annotation Core Model (OACM). We extended TabLinker to generate additional annotation graphs according to this data model. The following picture shows how an annotated cell in the tables is exported in RDF. The OACM defines an annotation as a target (the cell being annotated), a body (triples expressing the contents of the annotation), and some additional triples describing the author, date and other annotation metadata. The full size picture of the data model can be found in the Dropbox.

Dropbox/Diagrams/Annotations.png

One addition with respect to the OACM is the *flag system*. One of our requirements was to describe explicitly the content of these annotations using a set of pre-established flag values. In the figure example, the annotation contains a numeric value that does not match the original value of the cell, and hence we link this annotation with the flag identified with the number 0 (*"incorrect value"*). At the moment, this is the only flag type that is being automatically detected and generated while converting the Excel tables to RDF.

We identify the problem on how to cover the rest of the cases; namely, the task of classifying an annotation into a certain flag label, according to its textual contents. Although some of them may be classified easily (like the shown type 0), the classification of the rest may not be trivial at all.

Annotations need to be classified into one of the pre-established flag labels	● NLP technique ● Expert knowledge (with some support tool) ● Crowdsourcing

Furthermore, this annotation data model shall allow not only descriptive annotation contents and flags characterizing them, but also harmonization actions that we call interpretations. In general, these annotations describe issues with cell values: for instance, a cell made of the sum of other cells may not contain a correct value. In such a case, the interpretation associated with this issue must be an action to be carried out to correct it: for example, retrieve the sum from another table, or impute its value. The actions that have been taken to correct or improve data with problems must be shown to the user when she retrieves them.

In further releases, the data model needs to be extended to allow provenance. We aim at specific nodes pointing to the original census images, so that users can always retrieve these images while exploring structured data. Work with DANS to allow using resolvable URIs pointing to the original census scanned images stored in EASY is in progress.

Consistency checking

The annotations processing is not mature enough to answer the question on to what extent do annotations improve the quality of the data. However, as an easy first implementation we identified Benford's Law as an important data quality measurement. This law refers to the frequency distribution of the first digit of numbers in many datasets (including censuses).

This law can be checked with MP2Demo. All our trials showed that MP1 tables and MP2 tables meet Benford's Law with great accuracy.

STEP C. Classification systems, data consistency

Harmonization

Harmonization of the four census years was at the core of MiniProject 2 in the second round. After the first round of MiniProject 2, changes have been made to the harmonization scheme, applying a more consistent and structural approach. This approach builds on the type of information contained in the Census tables themselves. With regards to harmonization of the Population Census we can distinguish two types of information which are contained in the census, namely numerical variables *and* 'context' variables which mainly refer to geographical aspect. Variables which describe numbers make up the largest part of the Population Census. These variables for example describe 'the number of' for example of Houses under construction, Inhabited Houses, Ships, Total Males etc.. The 'context' variables however contain 'textual' information with regards to the geographical context of the Netherlands. For example variables such as 'Province', 'Gemeente', 'Kom' etc describe places and their geographical disposition. It is important to note that, we we don't have an overview of all the different types of 'numerical' and context 'variables' (as we have only looked at a limited number of tables in the context of the mini projects), however the latter has proven to be more difficult to harmonize and requires a different approach. Ideally we would like to know the total number of variables e.g. per tabel or per type of census. As the detail per census differs from year to year we cannot make estimations on the total number of variables. In order to extract this information we need to 'Style or Mark' all the tables with TabLinker which is currently still an ongoing process. The variables which describe numbers require a straightforward approach of linking similar variables across time or in some cases certain variables need to be joined, e.g. the 'total number of inhabitants' is not given in all censuses, however this can be easily created by adding the 'total number of males and females' up. In the case of "context" variables, this is very different as we deal with historical geographical information and variables which cannot be simply linked without e.g using / creating classification systems.

Currently we have described two different types of variables in the Census (context variables and variables describing number). For these two variables we have distinguished three different types of information, namely; *geographical*, *demographic* and *Housing information*.

CONTEXT – GEO	Textual

Housing info	Numbers
Demographic	Numbers

Several key changes have been made to the harmonization overview since the first mini project. The first scheme matched variables (on a high level) without making any distinction between the different types of variables (Geographical variables, housing variables and demographic variables). The first step therefore was, to redesign the harmonization overview by making a more structured table were, variables of the same type are grouped (see table above). After we found a classification which fitted all the variables in the mini project years, we harmonized the variables on a more content level. For example, not only do we record that we have the variable "municipality" or "housing types" across the different years, but also harmonized all the variables on a more content level. So we now know *which* municipalities or types of houses we have per census year. This method of harmonizing both on a high and lower level was inspired by the IPUMS International harmonization system.

The findings from the inventarisation and checking of the tables were taken into consideration but, as the main goal was to identify the problems we did not spend much time on cleaning of all the tables. We therefore started with variables which were more easily comparable across time (e.g. number of inhabitants per Province or at a more local level such as 'municipality' or 'plaats'). During the checking of the tables we noted that, although the tables represent the images closely, this is not always the case and therefore not easily comparable. To give an example, certain variables were recorded separately in the images (the source) but joined into one string in the Excel files. The restructuring and extracting of this is currently still a bottleneck. The variable "Plaatselijke indeling" in the Population Census of 1859 or 1909 contains: Kom buiten de kom, Wijk and Plaats in one string, whereas this information is recorded *separately* in the other years. In another example we see 'housing types' popping up under the 'municipality' header which is obviously a data entry error (specific flags are needed for these type of errors). As no consistent logic is applied it is very difficult to extract the right data without extensive manual input. With the use of semantics and NLP techniques we aim to identify these 'labels' automatically. Besides problems such as merging columns and strings we also found some (seemingly) gaps in the data. In the Population Census of 1920 "Kom" only appears 3 times (out of 15000 rows). These type of errors were initially not identified previously as we mainly focus at the years 1859,1869,1879 and 1899. In order to get a better grasp on the data its quality we also harmonized the years 1899, 1909 and 1920.

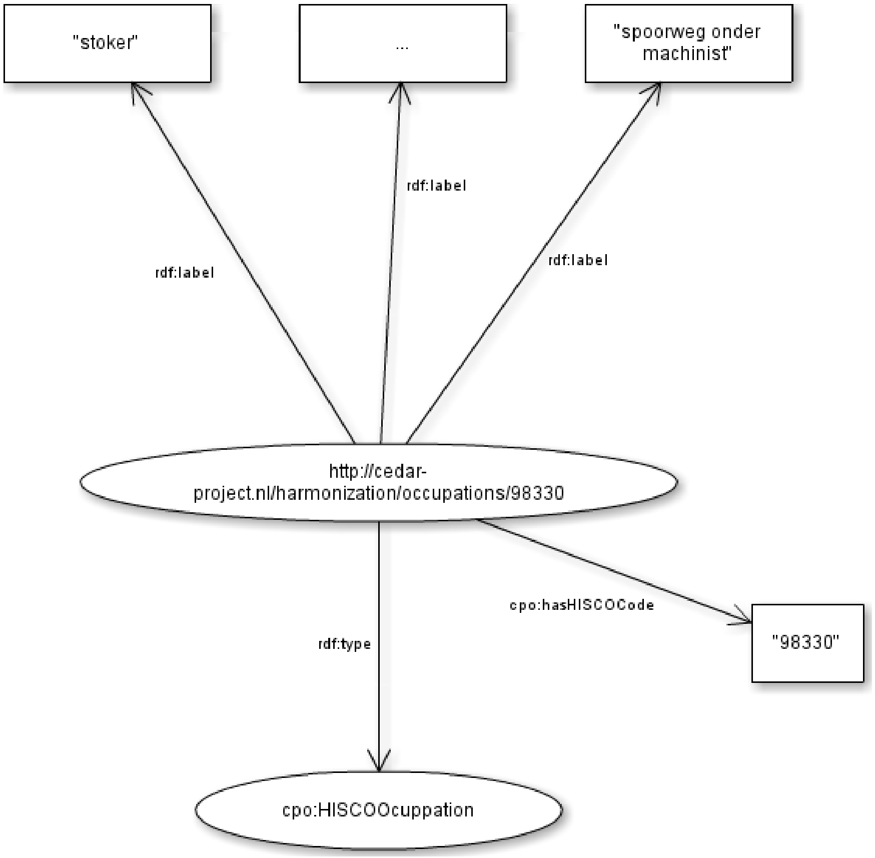
One of the major issues which we encountered during the harmonization process is very much related to this issue (i.e. the quality of the tables) and how to extract and restructure this data before we can even continue. The main focus during the next phase should therefore be, identifying the different types of errors and to create scripts and general methods for the restructuring of the data so it can be used for further harmonization.

STEP D. Linking to external datasets

Historical International Standard Classification of Occupations (HISCO)

In order to harmonize the occupations of both miniprojects conveniently, we decided to link the occupations with the Historical International Standard Classification of Occupations, HISCO. This way, HISCO can work as a interface between the user and the tables, gathering occupations together in a consistent way to facilitate longitudinal queries (e.g. by means of using HISCO codes in SPARQL queries).

The first problem we encountered is that there is no RDF conversion of the HISCO database. We produced a script, hisco2RDF, that generates a simple RDF model from an Excel table containing the original database. The data model is depicted below. It maps HISCO occupation identifiers with all possible literal names that these occupations may get in historical sources like the census. We leave the question of generating a proper SKOS hierarchy representing the HISCO vocabulary opened as a possible spin-off of this project.



The full size picture can be found in the Dropbox.

Dropbox/CEDAR/Diagrams/HISCO datamodel.png

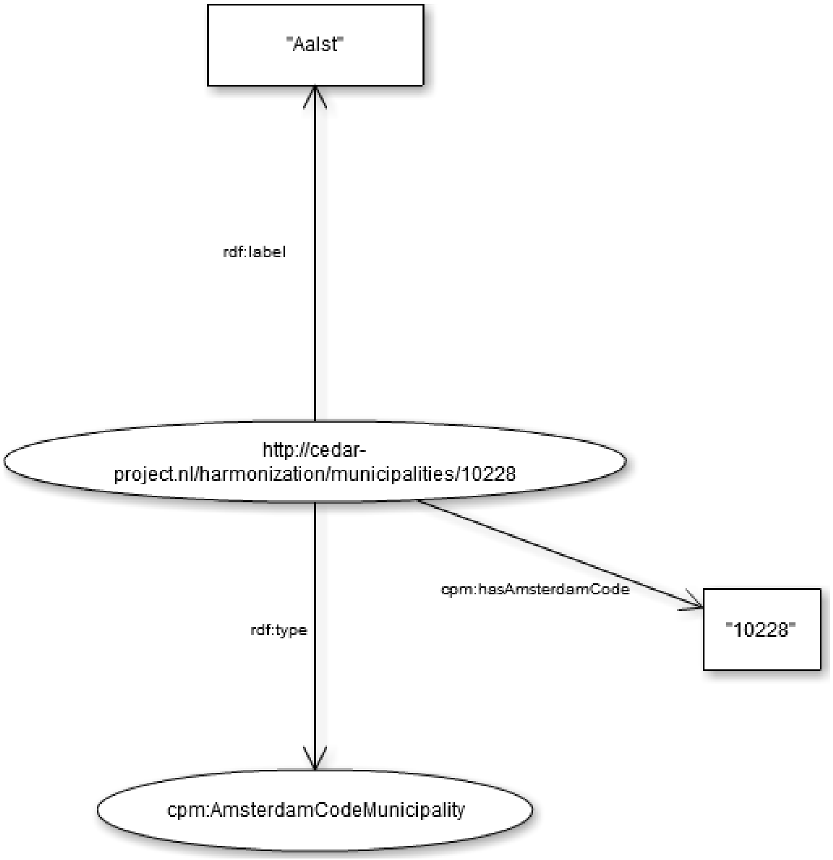
One problem we encountered while generating this graph is that we are not sure about the data source we are currently using. We found this HISCO Excel table in the volkstellingen.nl website that we take as faithful, but its quality and convenience has to be checked with experts.

Questions about HISCO source data faithfulness	● Consult with experts

Amsterdamse Code (AC)

In order to harmonize the municipalities that appear in both miniproject datasets, we decided to link these municipalities with the Amsterdam Code, AC. This way, the AC can work as an interface between the user and the tables, gathering municipalities together in a consistent way to facilitate longitudinal queries.

Again, the problem we encountered is that there is no RDF conversion of the AC. We produced a script, ac2RDF, that generates a simple RDF model from an Excel table containing the original database. The data model is depicted below. It maps AC municipality identifiers with all possible literal names that these municipalities may get in historical sources like the census. Again, we leave the question of generating a proper SKOS hierarchy representing the AC vocabulary opened as a possible spin-off of this project.


"Aalst"
rdf:label
http://cedar-project.nl/harmonization/municipalities/10228
cpm:hasAmsterdamCode
"10228"
rdf:type
cpm:AmsterdamCodeMunicipality

The full size picture can be found in the Dropbox.

Dropbox/CEDAR/Diagrams/AC datamodel.png

One problem we encountered while generating this graph is that we are not sure about the data source we are currently using. We found an ACExcel table in the volkstellingen.nl website that we take as faithful, but its quality and convenience has to be checked with experts.

Questions about AC source data faithfulness	● Consult with experts

Census, HISCO and AC RDF mapping

We produced a script, Harmonize, that links municipality and occupation names that appear in the census tables of both miniprojects with their corresponding entities in the HISCO and AC RDF graphs. If these mappings are done correctly, they can help in harmonizing the census and allowing longitudinal queries, since the ambiguity introduced by different spellings, concept drift, etc. is eventually solved. We call this set of mappings the *harmonization layer.*

As a first trial, we focused on literal string comparisons to decide which municipality and occupation names in the census match with their most suitable equivalents in HISCO and AC. We mapped occupations and municipalities in the census with the highest Levenshtein ratio (string similarity) found in the set of all HISCO occupations and AC municipalities. This way, the occupation and municipality strings that appear in the census are linked with the HISCO code or AC code that contains the most similar occupation or municipality description.

We encountered two problems when we finished this. The first is that there is not a clear method on how to evaluate the quality of the resulting mappings. The second is that, given the case that these links are not good enough, there is not a clear answer on how to improve the mapping method.

No trivial evaluation method on occupation and municipality mappings	● Manual expert evaluation ● Query results using these mappings ● Use an heuristic to evaluate how good a mapping is
Semantically enabled mapping method	● Enrich HISCO/AC with more structured descriptions ● Use linkage to other datasets to add criteria

STEP E. Data publishing

Command-line interface demo

MP2Demo is a command-line, python written script that shows the advances of this iteration. It allows designing queries, exploring the tables, dealing with annotations, and checking table data quality through Benford's Law verification.

The implementation of this demo shows how all necessary SPARQL queries have lots of patterns in common, and that the main differences between them come when user-defined parameters take different values. This makes our case especially suitable for existing query parametrization work and workflow systems, like LarKC. This possibility can be further investigated.

Two minor issues arose while developing this demo. First, INSERT SPARQL statements were tricky to implement using a local laptop development station. We aim at testing it with the currently working Virtuoso server of OPS (http://ops.few.vu.nl:8890/sparql). Second, longitudinal queries making use of the harmonization layer are still not available, pending on a final evaluation of our automated harmonization mappings.

Issues with SPARQL INSERT in annotations layer	● Test with OPS Virtuoso backend
Harmonization layer not used in longitudinal queries	● Quality check of the harmonization layer

Visualizations

Maps of iteration 1 suggest that some kind of pattern is present in SPARQL queries that retrieve almost the same data with light user-defined variations. This reinforces the possibility of studying how these queries can be parametrized systematically. The visualization is available here:

Men population in Noord Brabant, 1879-1889

The SPARQL queries that generate them are available in the Dropbox:

Dropbox/CEDAR/MiniProjects/SPARQL/

Academic publications

Albert Meroño-Peñuela, Ashkan Ashkpour, Marieke van Erp, Kees Mandemakers, Leen Breure, Andrea Scharnhorst, Stefan Schlobach, and Frank van Harmelen. *Semantic Technologies for Historical Research: A Survey* – submitted to Semantic Web Journal – under review

Meroño-Peñuela, Albert; Ashkan Ashkpour; Laurens Rietveld; Rinke Hoekstra; and Stefan Schlobach. *Linked Humanities Data: The Next Frontier? A Case-study in Historical Census Data*, In: Proceedings of the 2nd International Workshop on Linked Science 2012 (LISC2012) – Tackling Big Data (in conjunction to ISWC 2012), Boston, USA. Ed. by Tomi Kauppinen, Line C. Pouchard, Carsten Keßler. Electronic resource. http://ceur-ws.org/Vol-951/

Literature reviews

Great part of the survey paper *Semantic Technologies for Historical Research: A Survey* was produced from this literature review written in March/April 2012.

Appendixes

Appendix A. File inventory

This appendix provides a list of all relevant files produced during this iteration, with pointers to their location.

File	Description	Location
Census inventory summary	Summary of the CEDAR dataset, with an inventory of files, tables, annotations, authors, etc.	*Dropbox/CEDAR/Inventory/Census summary.xls*
Annotations summary	Distribution of annotations across all files and tables	*Dropbox/CEDAR/Inventory/Annotations/Table annotations distribution.xls*
Annotations dump and translation	Dump of all annotations of the dataset, with additional annotation metadata and content translation from Dutch to English (automated using Google Translate)	*Dropbox/CEDAR/Inventory/Annotations/annotations-dump-translation.csv*
Census table name list	Renaming conventions for all census tables, giving a basic description of the contents using the table title	*Dropbox/CEDAR/MiniProjects/Mini Case 2/Census Table Name List.xls*
Inventory of tables checked	Checking of all the mini project years for structural errors	*Dropbox/CEDAR/MiniProjects/Mini Case 2/Checking of tables*
TabLinker output files directory	All produced RDF from the miniprojects, in Turtle format	*Dropbox/CEDAR/MiniProjects/TabLinkerOutputFiles*
Classification of types of annotations	Study of all the different kinds of annotations present in the miniprojects tables	*Dropbox/CEDAR/MiniProjects/Mini Case 2/Flag Classification System/Annotations from the mini case files.xlsx*

Annotations diagram	Schema showing the data model used to represent census annotations in RDf	*Dropbox/Diagrams/Annotations.png*
Harmonization	Harmonization of the Cenus	*Dropbox/CEDAR/MiniProjects/Mini Case 2/Variables/Variable list.xls*
HISCO datamodel	Diagram showing the HISCO data model used to translate HISCO into RDF	*Dropbox/CEDAR/Diagrams/HISCO datamodel.png*
AC datamodel	Diagram showing the AC data model used to translate AC into RDF	*Dropbox/CEDAR/Diagrams/AC datamodel.png*
Iteration 1 visualizations	Visualizations produced in iteration 1 using sgvizler, a Javascript package to retrieve SPARQL query results and draw them using the Google Visualization API	Men population in Noord Brabant, 1879-1889
SPARQL queries directory	Placement for all SPARQL queries produced	*Dropbox/CEDAR/MiniProjects/SPARQL/*

Appendix B. Source code repositories

This appendix provides a list of all tools and scripts developed to the date, with pointers to source repositories.

Tool	Description	Location
Project repository index	Summary of all source code repositories in CEDAR	https://github.com/CEDAR-project
TabExtractor	XLS structured variable extractor, cleaner and summary generator	https://github.com/CEDAR-project/TabExtractor
TabLinker	Supervised Excel/CSV to RDF Converter	https://github.com/Data2Semantics/TabLinker
CEDAR Workflow	Platform integrator	https://github.com/CEDAR-project/Workflow
hisco2rdf	Script that generates an RDF graph for the History of Work Information System	https://github.com/CEDAR-project/hisco2rdf
ac2rdf	Script that generates an RDF graph for the Amsterdam Code	https://github.com/CEDAR-project/ac2rdf
CEDAR Harmonize	Prototype for harmonizing RDF census data using variable, occupation and municipality mappings	https://github.com/CEDAR-project/Harmonize
MP2Demo	Demo testing the achievements of the second iteration of the MiniProjects	https://github.com/CEDAR-project/MP2Demo

Appendix C. Problems & solutions

This appendix provides a summary of all unsolved issues encountered while developing this iteration, and re-displays the problem textboxes. There is a table for each substep of the process. Each row of these tables represents an issue. For each problem, a description and a set of possible solutions are given in two separate columns.

Inventarisation / Checking

Unreadable Names	Changed the coding system /human readable
Impractical when working 'offline'	Can be Used for inventarization but also the HISTEL project

Inventarisation produced a lot of messy spreadsheets	● Reimplement TabExtractor to produce one single coherent database with the whole census inventory

Inspection / Checking

Too many images	e.g. Crowd sourcing
Poor quality of some tables	Needs more advance clean up
Merged tables	Cannot continue harmonization without restructuring the data
ongoing process *many tables yet to be checked	Use expert knowledge

Conversion of tables to RDF

Conversion to RDF does not scale. We cannot mark manually the whole dataset.	● Crowdsourcing ● Students labour ● Outsourcing
TabLinker takes about 1GB of memory per 1MB of Excel file size. Conversion of big Excel files is unmanageable with our current hardware.	● Get additional hardware resources ● Rewrite TabLinker to be less memory consuming (without giving up verbosity)
Tagging of files depends on interpretation	● need uniform workflow or rules to avoid inconsistencies

Annotations

Can’t distinguish between original and new	● Consult with experts
Too many annotations to deal with	● Automatic grouping of similar labels (annotations)
Annotations need to be classified into one of the pre-established flag labels	● NLP technique ● Expert knowledge (with some support tool) ● Crowdsourcing

Harmonization

we have systems such as A’dam & CBS Code	(automatic) alignment of variables
Currently Top Down	Restructuring of data
	NLP
	Going for more difficult harmonization

	problems

Linking to external datasets

Questions about HISCO source data faithfulness	● Consult with experts
Questions about AC source data faithfulness	● Consult with experts
No trivial evaluation method on occupation and municipality mappings	● Manual expert evaluation ● Query results using these mappings ● Use an heuristic to evaluate how good a mapping is
Semantically enabled mapping method	● Enrich HISCO/AC with more structured descriptions ● Use linkage to other datasets to add criteria

Data publishing

Issues with SPARQL INSERT in annotations layer	● Test with OPS Virtuoso backend
Harmonization layer not used in longitudinal queries	● Quality check of the harmonization layer

Glossary

This section contains definitions of all terms which meaning we consider very important to describe, spot or constrain.

Variable	In CEDAR we understand variables in the applied statistical sense. Variables refer to measurable attributes, as these typically vary over time or between individuals. For instance, “age of workers in the leather industry in 1889 in Groningen” would be a valid variable.
Mapping	Data mapping is a task in data integration that establishes relationships between elements of two different data models. The two elements thus related are said to be mapped, or that there exists a mapping (of some kind) between them.