Source Oriented Harmonization of Aggregate Historical Census Data: a Flexible and Accountable Approach in RDF

Ashkpour, Ashkan; Mandemakers, Kees; Boonstra, Onno

Source Oriented Harmonization of Aggregate Historical Census Data: A Flexible and Accountable Approach in RDF

Ashkan Ashkpour, Kees Mandemakers & Onno Boonstra *

Abstract: *»Quellenorientierte Harmonisierung von Aggregaten historischer Zensusdaten: ein flexibler und nachvollziehbarer Ansatz in RDF«.* Historical censuses are one of the most challenging datasets to compare over time. While many (successful) efforts have been made by researchers to harmonize these types of data, a lack of a generic workflow thwarts other researchers in their endeavors to do the same. In order to use historical census data for longitudinal analysis, a common process currently often loosely referred to as harmonization is inevitable. This process becomes even more challenging when dealing with aggregate data. Current approaches, whether focusing on micro or aggregate data, mainly provide specific, goal-oriented solutions to solve this problem. The nature of our data calls for an approach which allows different interpretations and preserves the link to the underlying sources at all times. To realize this we need a flexible, bottom-up harmonization process which allows us to iteratively discover the peculiarities of these types of data and provide different interpretations on the same data in an accountable way. In this article, we propose an approach which we refer to as source-oriented harmonization. We use the Resource Description Framework from (RDF) as the technological backbone of our efforts and aim to make the process of harmonization more graspable for others to stimulate similar efforts.

Keywords: Historical census data, harmonization, source-oriented, Semantic Web, RDF, social historical research, historical demography.

1. Introduction

Throughout history, the main goal of censuses has been to collect information about a nation's population characteristics. As censuses are meant to accommodate the *information needs* of governments and societies, changing circum-

* Ashkan Ashkpour, International Institute of Social History Amsterdam, P.O. Box 2169, 1000 CD, Amsterdam, Netherlands; ashkan.ashkpour@iisg.nl.
Kees Mandemakers, International Institute of Social History Amsterdam, P.O. Box 2169, 1000 CD Amsterdam, Netherlands; kma@iisg.nl.
Onno Boonstra, Radbout University of Nijmegen, Postbus 9103, 6500 HD Nijmegen, Netherlands; o.boonstra@let.ru.nl.

Historical Social Research 41 (2016) 4, 291-321 | © GESIS
DOI: 10.12759/hsr.41.2016.4.291-321

stances will require different questions and data. Questions and purposes therefore change for each census. This principle is very well reflected and inherent in understanding the changing nature of historical censuses. These changes are valuable snapshots of our history (Higgs 1996) and are embedded in the very structure of the census itself, resulting in changing questions, variables, classifications, structures and processing methods over time.

The first integral enumeration of the Dutch population was held in 1795. Over 30 years later, a Royal Decree initiated the first official census of the Netherlands in 1829. From this year onwards, every ten years (with exceptions), the characteristics of the population of the Netherlands were captured in seventeen historical censuses up until the year 1971 (den Dulk and van Maarseveen 1999). After this period, the traditional 'door to door' enumerations came to an end due to political and budgetary reasons. Because of the obligation of the European Union to provide census data, from 2000 onwards, data was again collected via the municipal registers in combination with surveys (van Maarseveen 2002, 2003).

Unfortunately, the original (micro) data sheets collected by the enumerators were not archived from 1850 until 1960. From the 1830 and 1840 censuses, about half of the micro data have been preserved in municipal archives (Muurlings and Mandemakers 2012). Census results were published in aggregated form in several volumes for each census. In 1997, the first efforts were taken to provide better digitized access to the historical censuses, following a very strict source-oriented approach where even the presentation/layout of the tables were copied. All censuses were first digitized to images[1] and later transcribed into 2249 separate and disconnected Excel tables (Doorn, Jonker and Vreugdenhil 2001). The problematic aspects of using historical census data for comparisons over time are well known (Ashkpour, Meroño-Peñuela and Mandemakers 2015; van Maarseveen 2008; Ruggles and Mennard 1995; van de Putte and Miles 2005; Esteve and Sobek 2003; van Leeuwen, Maas and Miles 2004; St-Hilaire, et al. 2007), especially when dealing with aggregated data. From census to census we find changing questions, enumeration methods, variables, values, classifications etc. all hampering longitudinal analysis of the data. The diversity in data formats, structures and content of historical censuses calls for a unified system. *Harmonization* is therefore a prerequisite in order to do *any* type of longitudinal research. However the harmonization process differs for micro and aggregated data. The main difference is that aggregated data introduces more ambiguity. Whereas with micro data one is able to build classifications systems, variables etc. according to one's need, aggregated census data needs to introduce estimation schemes to achieve results which can be used for comparisons over time. As a consequence, when dealing with aggre-

[1] By way of the project *Life Courses in Context* which started in 2003.

gate data we do not know beforehand which harmonizations are the most optimal choices. It is a process of trial and error, exploring the data gradually. For this reason, it is necessary to have flexible systems which enable us to create different harmonizations on the same variables in an iterative way. Current census harmonization practices and models are not designed in such a way and usually do not (easily) allow different views on the data. These, mostly micro data practices, result in only one version of a newly categorized and classified dataset. Current efforts therefore lean more towards 'goal-oriented' methods, where the users of the data are bounded to choices and interpretations which have been set before (Cameron and Richardson 2005; Thaller 1993). According to Greenstein's (1989) definition, the source-oriented approach should allow two main requirements. Namely that, the same source can be handled differently in various stages of historical research and that the uses of sources may vary over time.

The source-oriented approach is the preferred and preeminent method in historical research. Being able to refer to the original sources and allowing different interpretations on the same data is an important requirement in this field of research. As we will show in this article, in order to compare the aggregate Dutch historical censuses over time, harmonization of the data is an inevitable process. However, thus far the harmonization of historical census data is based on goal-oriented methods. We strongly believe that the principles of the *source-oriented* approach in *historical research* should also be applied when *harmonizing* these types of data. Source-oriented data processing methods will not force the historian to make a decision on which methods to be applied at the time of the database creation (Boonstra, Breure and Doorn 2006; Thaller 1993). With aggregate data we need a different and more flexible bottom-up approach which allows a learning experience, to iteratively test and provide different harmonizations in order to deal with the ambiguous nature of aggregated data. It is an approach that we refer to as 'source-oriented harmonization.'

In this article we explain how we[2] implemented a source-oriented, structured harmonization approach with the Dutch aggregated historical census data, using Semantic Web technologies. We propose an iterative harmonization workflow in RDF (Resource Description Framework) which makes different interpretations possible without losing track of the original aggregated data. We provide tangible links from the harmonized data to the original sources upon which the harmonizations are based. In doing so, accountability is guaranteed. In the following sections, we start by looking at census harmonization in general and describe its challenges. Next, we go into the details of our workflow and explain each step of our suggested approach and how we used this to gradually build harmonized tables in the context of a pilot project. We end with a

[2] This work is done in the context of the CEDAR Project, and is part of the Computational Humanities Group of the KNAW, see <https://www.cedar-project.nl> and <http://www.ehumanities.nl/projects>.

discussion of the results and the wider impact of our source-oriented harmonization system.

2. Harmonization

Historical data harmonization, such as the unification of formats, structures and content of historical data, is a knowledge intensive task which highly depends on expert decisions and choices. Knowledge about the source data is an essential aspect of historical data harmonization (Mandemakers and Dillon 2004). Accordingly, formal descriptions of the data can only be provided by advanced users of the data or those involved in the data creation itself. Expert knowledge about the source data and its underlying model is therefore essential in understanding the problems to be addressed during the harmonization process.

This holds even more true when harmonizing historical censuses (Esteve and Sobek 2003). It is not a one-try process; it is an iterative process of trying and learning how the classified and harmonized data interacts with the original data. Currently, there are no clear definitions or guidelines explaining which steps need to be taken in order to make the data comparable over time. Even when users are interested in the *same* data their motivations and goals may diverge, meaning that *different interpretations* on the data are an essential aspect (Greenstein 1989, Thaller 1993). To allow these different interpretations, it is essential to follow a source-oriented harmonization approach. Although there are no clear guidelines, we *can* identify a *set of practices* which are currently applied by researchers in order to make census data comparable across time. These different practices *together* are what constitutes census data harmonization in our view. In the following sections, we describe the practices we have developed for the harmonization of the Dutch censuses through the workflow we have introduced. So, while current approaches lack a defined harmonization workflow, the problems and challenges we face are notorious and have been described and documented thoroughly.

Building on these practices we identify the following four topics as key terms averting the harmonization of historical censuses: (1) integrating dissimilar data sources and formats, (2) dealing with changing variables, values, structures and classifications, (3) constructing a database which can be queried across the years and last but not least, (4) the existence of a practical and generic harmonization workflow for aggregate data. Taking these ('needs') into consideration we define source-oriented historical (census) data harmonization as:

> An accountable process of creating an unified and unambiguous version of the dataset, which is flexible enough to deal with the changing characteristics of the data, whilst not committing to a predefined interpretation, by gradually applying a combination of known harmonization practices.

We adhere to the source-oriented approach of the digitization process and gradually build semi manual bottom-up harmonizations, following a structured harmonization workflow. Our harmonization definition is accompanied by a practical workflow and technological backbone to support our methods (Meroño-Peñuela, et al. 2015a). We provide a structured, generic and repeatable workflow in order to make the harmonization process more explicit. We do this both in practical and technical terms, we aim to be as transparent as possible and stimulate similar efforts on other datasets (Meroño et al. 2016a).

In our efforts, we also explore the suitability of using RDF for longitudinal historical census data harmonization. The use of RDF for census data publication and harmonization is not a novelty. Several attempts have already been undertaken (Ashkpour, Meroño-Peñuela and Mandemakers 2015). We distinguish our census harmonization approach in three different ways. First, we see harmonization across time and space as the most important step to make the data more usable, after publishing the data. Many current efforts merely aim to convert and publish historical datasets (such as census data) into RDF, with the anticipation of gaining Semantic Web benefits such as extending and enriching the data with other systems. Conversion into RDF simply represents the data with all its faults and problems in another format. The harmonization part is usually absent in these practices so far, except in some cases where censuses were harmonized to make all data comparable *for a given* census year by harmonizing over regions and levels of abstraction. Second, these efforts mostly use micro data as a point of take-off. A third significant difference is that these projects harmonize *contemporary* censuses and not *historical* ones.

3. The Harmonization Workflow

In order to explore the possibilities of publishing the original and harmonized data of the Dutch historical censuses in the Semantic Web, using RDF, we developed a pilot to test our methods and workflow. For this pilot we focus on a subset of the censuses containing the number of inhabitants and dwellings for each locality and municipality. We selected these so-called *Local Division* tables for the census years 1859, 1869, 1879, 1889, 1899, 1909 and 1920. If we succeed in providing a harmonized version of this data, the state of the nation can be studied on abstract levels such as the total number of inhabited houses, houses under construction, houseboats or the number of males/females. It will also be possible to ask detailed questions such as "the total number of people counted in monasteries in the centers of small towns for each province, across the years."

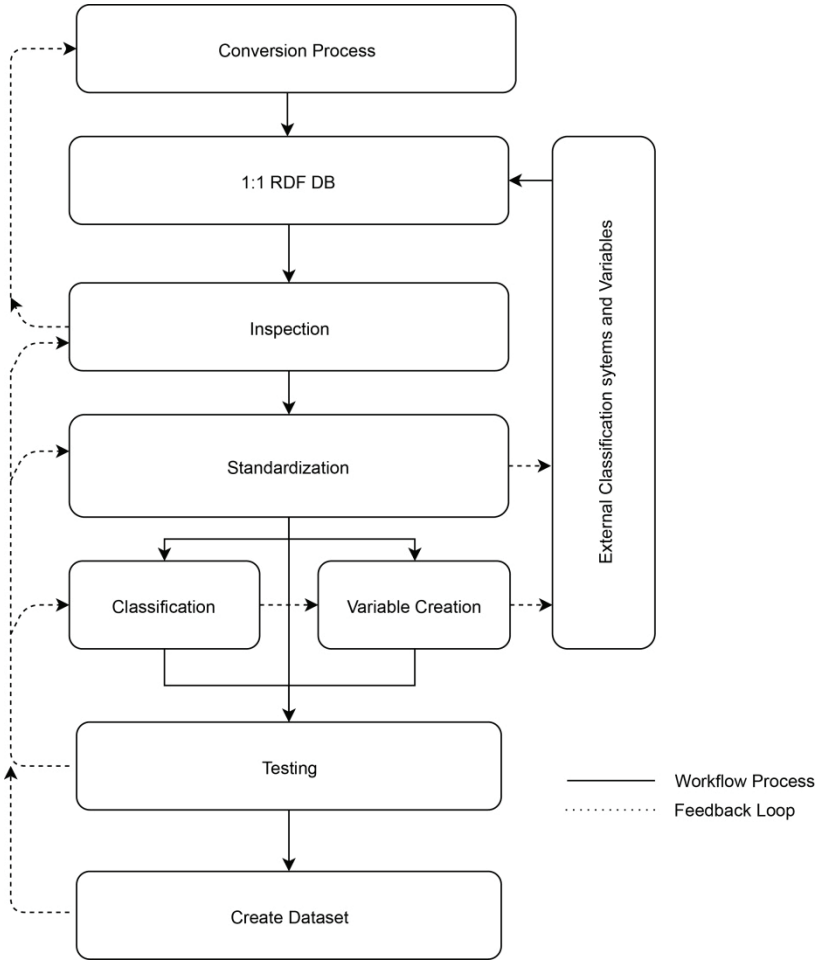
The data of these seven census years are currently stored in 60 different Excel tables. The number of tables and measure of detail differs widely between the different censuses while containing different types of variables on different

geographical levels. For some years there are large Excel files containing different tables (sheets) per province, for other years we have smaller Excel files (tables) for each province separately.

Figure 1 presents a scheme of the source-oriented harmonization workflow we have developed. The points of departure for the construction of this scheme were the following principles: 1) The workflow is applicable to other datasets, 2) The workflow allows systematic testing and feedback loops in all stages, 3) The raw data may never be changed, 4) Since data as complex as historical censuses cannot be harmonized in one try, the workflow must follow an iterative processes of trial and error, 5) Different interpretations on the data are allowed.

The first step in the workflow (Figure 1) consists of the *conversion* of the original Excel sheets into RDF. Once the data is converted, we have a *1:1 database* which we use to build our harmonizations on. The second step is the *inspection* stage of the data. During this stage we try to get a better understanding of our variables and values by directly querying the newly created RDF database. The first feedback loop of our workflow starts here (note that there is no feedback loop to the original data). The third step of the workflow is the *standardization stage* where we actually make harmonization decisions on how to define the different variables and values uniformly over the years. After standardization we move on to the *classification* stage in which variables and values are put into meaningful groups. During this stage, we create internal bottom-up classifications and make use of external classification systems wherever possible, whilst enriching the web with our census specific systems (see feedback loops to "external classification systems and variables"). The next part of the workflow is the *variable/value creation section*, where we actually create (missing) variables and values to fill in the gaps in our tables and bridge between the different censuses. Depending on the needs after standardization, this stage could be applied prior, after or simultaneous with the classification stage. For example, it may be that some variables need to be grouped into other variables to make meaningful classifications. The finishing touch of the workflow is the *testing* of all procedures. We 'test' the produced data extensively after each stage of our workflow in an iterative manner by querying the database and creating intermediate tables until a certain degree of quality is reached. Now harmonized and tested, in the *create dataset* stage we produce different types of tables for researchers. In the following sections we go into the details of each stage and how we worked towards producing a harmonized dataset for the Dutch *Local Division* tables of the census.

Figure 1: Source-Oriented Harmonization Workflow for Aggregate (Historical) Data



3.1 Census Data in RDF: Conversion and 1:1 Model

The first step in our harmonization workflow is to convert the data and its original hierarchies and structures from the Excel sheets, in which they were stored in 1997, into a RDF database. The premise of our source-oriented approach builds on the notion that the underlying dataset should be converted into an RDF database *without* making *any* decisions on how to model the data beforehand. This means that we represent the historical data sources as one to one copies in RDF. By converting the data to RDF we gain the advantage that we

are now able to query the census tables as a whole. We are able to explore, discover, try, fail and try again in order to learn the data with all its peculiarities before committing to a certain interpretation.

Currently, the application of Semantic Web technologies is being advocated in different historical fields (Meroño-Peñuela, et al. 2015). Different types of historical data are being converted to RDF using a variety of tools and methods. In order to move towards a database in RDF, an appropriate *RDF model* should be used. Depending on the *type* of data (textual, statistical, structured etc.) different models are available. In the case of *census* data we used *RDF DataCube*, the *standard* in the Semantic Web to model "multi-dimensional statistical data" (Cyganiak, Reynolds and Tennison 2016). This model is based on the *SDMX* cube model and ISO standards for the exchange of statistical data. In order to convert our Excel tables (see Figure 2) we used a very straightforward tool (TabLinker[3]) to convert the excel tables into RDF DataCube compliant data (Ashkpour, Meroño-Peñuela and Mandemakers 2015).

Figure 2: Original Excel Table with the Number of Inhabitants and Houses per Geographical Unity for the Census Year of 1889

Municipality	Local Division					Houses in the Municipality				Temporary Present Ships	POPULATION						Temporary present	
						Residential Houses			Inhabited Ships		Present during count		Temporary absent		Total			
	Inside / Outside the center	District	Type of place	Name	Housing	Inhabited	Uninhabited	Under construction			M	F	M	F	M	F	M	F
Achtkarspelen			Village Agustinusga															
	Inside																	
					Houses	76					147	156	1	3	148	159	1	
					Ships	15	1				36	38		1	36	39		
	Ouside								3		8	4			8	4		
			Hamlet	Blaauwverlaat		8					21	17		1	21	18		
			Hamlet	Kleiweg		12					23	20			23	20		
			Hamlet	Rohel		1					5	5			5	5		
			Hamlet	Poldermolen		1					3	1			3	1	1	
			Hamlet	Tjoele		1					2	3			2	3		1
			Hamlet	Nieuwkerk		3					8	7			8	7		
			Hamlet	Turflaan		2					5	7			5	7		
			Hamlet	West		42	1				95	70	1	1	96	71		
			Hamlet	Koodeschuur	Houses	2					5	3			5	3		
					Temp. pres. Ships	9					26	19	1		27	19		1
			Hamlet	Heide						2							6	1
						6					20	17			20	17		
			Village Buitenpost															
	Inside				Houses													
					Ships	140	1				248	286	12	12	260	298	13	14
					Temp. pres. Ships				2		4	2			4	2		
	Outside									2							6	6

Note: Column and row headers are translated from Dutch.

Figure 2 is an example of an Excel table to illustrate the structure and contents of our source data. This table contains the number of inhabitants and houses per geographical unity for the census year of 1889. The figure shows how the different numbers in the tables are connected to *multiple* row and column headers. These headers contain different types of *hierarchical* variables and values. For example the highlighted cell with the number "42" is connected to a '*hamlet* called '*West'* belonging to a '*village*' called '*Augustinusga*' in the municipality

[3] <https://github.com/Data2Semantics/TabLinker>.

of '*Achtkarspelen,*' in a place '*outside the center*' *of that hamlet*, presenting the number of '*inhabited residential houses.*'

In our conversion of the raw data from Excel into RDF we preserve the original structure and dependencies between the different variables. We use the definitions provided by TabLinker to manually prepare the different tables in order to convert them into RDF (Ashkpour, Meroño-Peñuela and Mandemakers 2015). Using this tool, we can take advantage of the structured layout of the Excel tables and define the different *areas* where the numbers and variables/values are contained. Figure 3 shows an example of the same table as in Figure 2, but now styled using the definitions provided by Tablinker. We have colored and numbered the different data areas to illustrate the styling process. To do the styling we first import the different styling options of Tablinker into Excel. Next, we open the Excel tables to color the different data areas. The colors in Figure 3 represent the various Tablinker styles which are applied to this specific table, namely: Row Property (1), Column Header (2), Row Header (3) and Data (4). Using these styles Tablinker is able to distinguish several 'data areas' in multi-dimensional tables.

Figure 3: The Same Table as in Figure 2 but now Styled with our Conversion Tool. 1=orange 2=blue 3=light blue and 4=light brown

Municipality	Local Division 1					Houses in the Municipality				Temporary Present Ships	POPULATION						2	
						Residential Houses					Present during count		Temporary absent		Total		Temporary present	
	Inside / Outside the center	District	Type of place	Name	Housing	Inhabited	Uninhabited	Under construction	Inhabited Ships		M	F	M	F	M	F	M	F
Achtkarspelen			Village	Agustinusga														
	Inside				Houses	76					147	156	1	3	148	159		
					Ships	15	1				36	38		1	36	39		
	Ouside								3		8	4			8	4		
			Hamlet	Blaauwerlaat		8					21	17		1	21	18		
			Hamlet	Kleiweg		12					23	20			23	20		
			Hamlet	Rohel		1					5	5			5	5		
			Hamlet	Poldermolen		1					3	1			3	1	1	
			Hamlet	Tjoele		1					2	3			2	3		1
			Hamlet	Nieuwkerk		3					8	7			8	7		
			Hamlet	Turflaan		2					5	7			5	7		
			Hamlet	West		42	1				95	70	1	1	96	71		
			Hamlet	Roodeschuur	Houses	2					5	3			5	3		
					Temp. pres. Ships	9					26	19	1		27	19		1
			Hamlet	Heide						2							6	1
						6					20	17			20	17		
			Village	Buitenpost														
	Inside				Houses													
					Ships	140	1				248	286	12	12	260	298	13	14
					Temp. pres. Ships				2		4	2			4	2		
	Outside									2							6	6

After styling the tables we create a RDF database using the Tablinker scripts. This 1:1 RDF database contains the same content and structures as the original Excel files. We call this the 'raw data' layer. The raw data layer together with several queries and scripts are used to *assist* the harmonization process in the next stages of our workflow.

It is very important to keep in mind that during this stage of the workflow, under no circumstances the original data should be touched, even when obvious mistakes are spotted. By doing so, we are always able to reproduce the prove-

nance (W3C 2015) of our actions, back to the original source material. Data errors and ambiguities will be dealt with later in the process, by structurally going through the different steps of our workflow. Errors made in the conversion are dealt with by improving and running our system again. We provide in depth technical descriptions for those interested in understanding and setting up a similar workflow (Meroño-Peñuela et al. 2016b).

3.2 Inspection

The next step is the inspection of the data. This is done in a semi-automatic way. Out of the enormous pile of raw data in RDF format, we need to identify all classifications, variables, values etc. contained in the original censuses. In other words, before we can define the variables we first need to know what we have. At this preliminary stage, we can already analyze the raw data as a whole to provide *insights* for the harmonization itself. So, while staying true to the source-oriented approach (Boonstra, Breure and Doorn 2006; Thaller 1993; Cameron and Richardson 2005), we have created a database which we can use to query in order to get statistics about the landscape of the historical censuses. We can now ask questions such as e.g. what are the different variables and their values, which ones are the most frequent used, how are these variables related, can we find similar classification systems, do we need all literals to define a variable? etc.

During this stage we clearly need to define the scope of the data which we want to harmonize. Although changing definitions is a known hindrance to historical census harmonization, there *are* certain periods in which the censuses share *common* characteristics such as the same classifications, variable, values, structures etc. By starting with harmonization of censuses which share similar characteristics, we create general rules and practices which can be extended to the entire dataset. For instance, in the case of the *Local division* tables we can identify that there are three subgroups of censuses which use similar classifications i.e. 1859-1879, 1889-1899 and 1909-1930. The harmonization input itself is heavily dependent on expert knowledge and human input. Therefore, not exposing the data to the experts as one big dump, makes it easier to get a better grasp on the data when analyzing it as a whole (Slavakis, Giannakis and Mateos 2014).

After similar subgroups have been identified it is time to start looking at its content. The first major step in the inspection process is to make frequency distributions of the variables and values to see what we have across the years. To get a clear idea we have to look at this in twofold. First, we make univariate frequency lists of the raw variables and values in order to create data driven vocabularies. Second, we create hierarchical frequency lists to understand the mutual connections between variables and how these are hierarchically situated in the tables. As we will illustrate further on, the context and *relationship* of the

variables are key to the understanding and creation of formal descriptions of the data. For example, where a frequency list (Table 1) would merely give us an overview of the variables and values which occur most often, a multivariate hierarchical frequency table (Table 2) shows how the terms are connected in the original tables. This helps us to understand its context, and by this the nature of the variables and its values.

Table 1: Sample of a Frequency List of 'Raw Terms' in the Original Tables and Directly Generated by Querying the RDF Graph

Literal	#
Males	8981
Females	8721
M.	654
F.	607
Temp. Present	4506
Temporary Present	2151
Pop	1015
Population.	9647
Population	2458
Legal Present	2412
Leg. Present	894
Legally Present	2452
Factual Present	5853
Total.	2545
HouseBoats	5482

Table 2: Flattened List Example of the Hierarchies among the Variables in a Census Table, Directly Generated from the RDF Graph

Year	Variable 1	Variable 2	Variable 3
1869	Temporary Present	F	
1869	Temporary Present	M	
1889	Temp. Present	Males	
1889	Population	Males	Legally Present
1879	Population	Males	Total.
1899	HouseBoats	Temporary Present	
1879	Population	Females	Total.
1899	Population	F.	Legally Present
1899	Population	M.	Legally Present

The examples in Table 1 and Table 2 show the results of the data inspection stage (note these are samples for illustration purposes only). Table 1 is a simple frequency list of the literals used most often in the census. Table 2 presents the same terms as in Table 1 but now in relation with each other. In this *hierarchical* view we have flattened for example the variable combination 'Temporary Present' and 'Males/Females' (to represent the original hierarchy, see Figure 2). Simply looking at the frequency list (Table 1) makes it difficult to make sense of the meaning and context of the variables; by considering the

original hierarchies of the variables, we now see for example that term 'Temporary Present' is connected to both 'Sex' (a demographic variable) and 'HouseBoats' (a housing type variable). By providing this information to the expert user, we assist them in the process of creating distinct formal definitions. For the queries we have used to extract these literals from the RDF tables, see our website <www.censusdata.nl>.

Thus, during the inspection stage we focus on identifying subgroups of censuses which share similar characteristics first. Within these subgroups, we focus on the most frequent and/or important variables in the censuses. After this is done, we focus on the details and specificities of less frequent variables. This workflow allows us to create, in a semi-manual way, a variable overview across the years which will serve as the *input* for the next stage of the harmonization process, i.e. standardization. During the subsequent stages of our source-oriented workflow we systematically come back to the inspection stage to identify new problems and to improve the process of standardization and variable creation.

3.3 Standardization

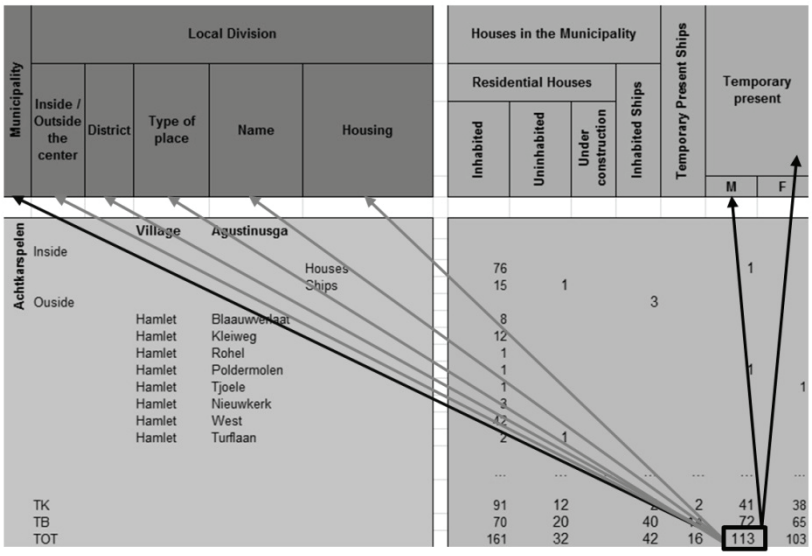
In our source-oriented harmonization approach we first have converted everything 'as is' into one RDF system. This means that the variables are still only accessible by their own literals. To allow longitudinal analysis we still have to standardize each and every single variable and value in this new RDF database. *Standardization* is the first harmonization stage in our workflow where we have to decide on how to make the data uniformly accessible over the years. During this process expert knowledge about the source data is key in assigning meaningful definitions and mappings. In this section, we describe the four different elements of our standardization process. We start with a *selection* of variables and values to standardize; next we *formally define* the identified variables and values. Once defined, we describe the *grouping* of them and we finish with illustrating the importance of maintaining valid variable *mappings*. This standardization procedure enables us to access all the different variables and values uniformly over the tables and extract all relevant data.

3.3.1 Understanding the Data Structure: A First Selection of Variables

Figure 4 presents a table which describes the number of inhabitants and houses for a given year. Each data cell (number) in this Excel table is connected to various column and row headers of the census table. These headers represent the multiple dimensions of our RDF model. During this stage we have to determine the meaning of all literal values such as 'Achtkarspelen,' 'Uninhabited,' 'Males,' 'Houses,' 'Temporary Present' etc. and all their variations. We use the input from the inspection stage to build 'bottom-up' standardizations.

Based on this, we *first* select those variables that are sufficient to define a number in the table. This is what we call the '*minimum required definitions.*' We then gradually define the more fine grained variables and values as we progress. In Figure 4, we see an example where eight different row and column headers are connected to the (highlighted) number we are interested in, i.e *113*. These headers or dimensions are indicated by arrows in the table. The number 113 refers to the *total* number of *temporary present males* in the municipality of Achtkarspelen, so for this number only three out of the eight dimensions are minimally required to define the number, i.e. *Municipality*, *Temporary Present* and *M* (the three black arrows in Figure 4).

Figure 4: Excel Table Highlighting the Different Dimensions which are Related to the Bold Number



We therefore (first) provide standardizations for:

- Municipality – 'Achtkarspelen'
- Residence Status – 'Temporary Present'
- Males – 'M'

By standardizing these three variables, *in combination*, we are able to retrieve this *specific* number from our tables. We transfer the *totals* (TK, TB, TOT) to RDF and standardize them for comparison purposes but purposefully ignore these values in the query process in order to avoid over-counting as we create our own totals. This is needed because the original totals are not always correct and do not add additional value according to the principle of the *minimum required definitions*. Moreover, by creating our own totals using all the lower

sub-values, we can break down a total in case of wrong values and i.e. identify the specific cell which is wrongly standardized. The more we define and standardize, the more specific we can target a data cell in the tables. For example, to get the total number of '*males*' which are '*temporary present*' in a specific '*district*,' '*outside the center*' *or in a certain* '*housetype*' of that 'municipality,' the lower geographical areas need to be defined in addition to the municipality. The iterative nature of our workflow allows us to start the standardization at more abstract levels and focus on the specificities and details in later stages. This is necessary to rise above the data deluge problem so that experts providing the formal definitions do not get overwhelmed with literals to define. To keep track of our progress we frequently produce statistics[4] to see how much of any given table is defined and what is still left.

3.3.2 Providing Formal Definitions

Building on the input from the inspection stage and by identifying the *minimum required definitions*, we provide standardized terms for the given literals in a structured way. During this process, we enrich the literals with standardized terms. By doing so we are able to access the data across time and space using a *common* vocabulary. This means that we consistently assign standard definitions or codes to all possible variations of a given variable or value. See Table 3 for an example of how we use the input from the inspection stage to standardize the terms in a structured way.

Table 3: Using the Frequency List and Flattened Hierarchical View Formal Definitions are Given by Expert Users of the Data

1869 Population Census Table				
Original String	Standardized	Original String	Standardized	Formal Expert Definition
Total	Legally Present	M	Males	Legally Present Males
Total	Legally Present	F	Females	Legally Present Females
Present during count	Actually Present	M	Males	Actually Present Males
Present during count	Actually Present	F	Females	Actually Present Females

Each line in this table has to be seen as a possible variable combination (based on the original hierarchies, whereby the original terms are translated into English in Table 3). In order to query for all the dimensions and their combinations, the variables first need to be defined separately. The columns entitled "Original String" represent the original string/literals in the tables (extracted during the

[4] <http://lod.cedar-project.nl/cedar/stats.html>.

inspection stage). The last column is the *formal definition* given by the expert user and the columns entitled "Standardized" the standardized terms given by us, based on the formal definition. We follow this approach to structurally standardize all the literals in our raw RFD dataset.

3.3.3 Putting Values into Standardized Variables: Grouping

At this stage of the standardization process, the literals are formally defined and standardized, but they still are not grouped into meaningful variables or domains. For example, the values Male and Female are now standardized and accessible uniformly across the tables but what are males and females? What do 'Temporary Present,' 'Factual Present,' 'Legally Present' or 'Houseboats' etc. mean? In order to give them meaning we need to put them into standardized variables, i.e. variables which have been created by ourselves. In our example, we assigned our values to three standardized variables. For example, we attached the Male and Female values to the standardized variable *Sex*. The different statuses given to persons or housing types are defined as *ResidenceStatus*. Finally we also created a standardized variable for the different *Housingtypes* (houses, wagons, houseboat). These standardized variables *together* are what allow us to reconstruct the original variables and values during the querying process when combined and reshuffled, as will be explained below.

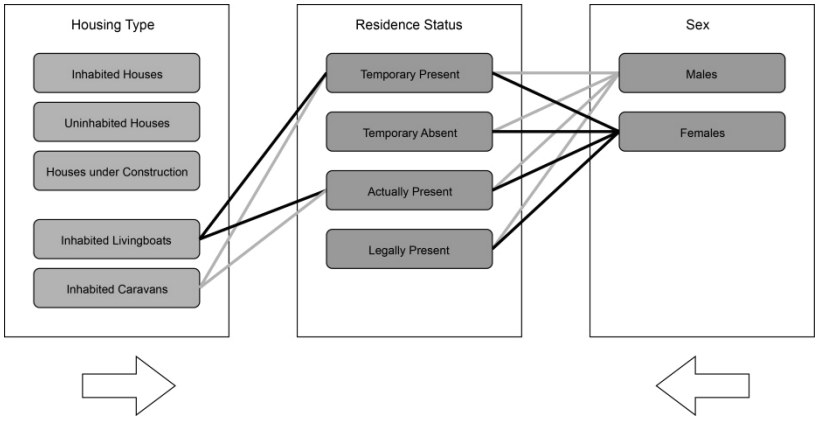
3.3.4 Mappings

In order to (correctly) use the standardized variables and query the data, one last important step remains. An expert user may know which combinations of variables are possible, but others may not. Merely providing a query endpoint where users can enter queries does not work if they do not know what to query for *and* which combinations are possible. In order to guide users in making queries, we provide 'variable mappings' (as showed as connections in Figure 5). This figure shows the standardized variables, the standardized values *and* how they are related to one another.

Mappings are important to avoid invalid questions on the data. For example, *without* looking at our standardization model below, users which are interested in a very basic demographical statistic such as '*the total number of males in a certain city*' will get the wrong number back when they simply query for *exactly* that. If we look at Figure 5 we see that the variable value *males* is connected to *four* different 'ResidenceStatus' values. So merely asking for the total number of males *without* a 'ResidenceStatus' *restriction* would give us the total number of *males* which are:

- Temporary present
- Temporary absent
- Actually present and
- Legally present

Figure 5: Variable Mappings – Overview of the Created Variable Groups, their Values and Mappings



What the user really needs is the combination of the standardized values *males* and *legally present* (to avoid major over counting). Using these mappings and documentation about the meaning of the standardized variables and values, users will be able to construct valid queries on the harmonized data and produce sound statistics when querying the data themselves. Therefore, it is the *combination* of standardized variables and values which allow us to reconstruct the data of the original tables and make *sensible* queries in the RDF database.

3.4 Classification

Once the data has been standardized and tested, we move on to the next stage of our harmonization workflow, i.e. classification of the data values. In this stage, all variables which contain numerous different values are grouped together into meaningful classes (Beghtol 2010). In the censuses there are many variables which have more than a few possible values. The variable municipality contains around twelve hundred municipalities, and there are over a hundred different lower-level municipal areas, thousands of literals referring to different religions, hundreds of occupations and occupational classes and almost two thousand different housing types. They all need to be classified.

In the case of the Dutch historical population censuses, we have three main classification systems (see 3.3 Inspection). Bridging the gap between the different classification systems, using aggregate data, is not always possible without creating our own classifications *or* variables (see Section 3.6).

Our classification approach is a twofold one (see Figure 1 with feedback loops to external classifications and variables). First of all, we try to reap the benefits from having everything exposed in the Semantic Web and connect to existing classifications systems wherever possible. Next to that, we create our

own bottom-up systems to accommodate the lack of standard variables and classifications in the Semantic Web. As we are aiming to harmonize *historical statistical* data, and a very *specific* one as the *historical censuses*, we found that the majority of the variables we are interested in are just not in the Semantic Web, yet. Except for relatively simple variables such as Sex and Marital Status, provided by the *SDMX (Statistical Data and Metadata eXchange)* vocabulary. Consequently, during the initial inspection stage we already realized that almost all of the classification systems and standardizations we were interested in had to be made by ourselves. By doing this in RDF we are not only harmonizing our dataset but also enriching the web with our definitions and variables.

What we need are census-specific (bottom-up) harmonizations, starting with a frequency list of all the different values for municipalities, religious denominations, residence statuses, housing types etc. As we have defined the location of the variables during the conversion of the Excel files into RDF, we are able to query for a specific variable with *all* its values to create semi-manual classifications. The classifications described below are created and based on our harmonization needs, using the expertise of frequent data users. In the following we give examples of bottom-up classifications and connections to external systems (see Figure 1).

Housing types such as barracks, wagons, ships, institutions, hospitals, prisons, etc. are used throughout the population census in different degrees of detail. Whereas in some years we have detailed information such as "the asylum of Saint Paul" or "the abbey of Berne" in other years we have only information on the aggregated level of such housing types, i.e. asylum or monastery. The former detailed cases, although interesting for local historians, would not be of much use for researchers interested in longitudinal analysis. As said, we need to put these (detailed) variables into usable groups based on the function they perform (hospital, military buildings, mental institutions, etc.). By doing so we have created a bottom-up classification system which for the first time allows us to analyze the evolution of the different housing types in the Netherlands over time with marginal effort. From the number of care homes for the elderly, to mental institutions, to the number of monasteries, a variety of interesting house types are now standardized and classified using automated frequency lists and expertise of knowledge users in the project. This classification resulted in the grouping of 2000 unique literals into fifteen classes of housing types.

Municipalities are the most-used geographical level after provinces in the census. The census is one – and sometimes the only systematic – of the historical sources for researchers, providing comprehensive geographic coverage and broad chronologic scope (Ruggles and Mennard 1995). However, boundaries of municipalities may change over time, as well as their names, severely hampering longitudinal studies. Historically, the municipalities in the Netherlands underwent major changes. Between 1812 and 2006 there were only six munici-

palities which did *not* experience changing boundaries in the Netherlands (van der Meer and Boonstra 2006). In order to track these changes several (external) classifications have been developed (Amsterdam Code, CBS Code, Wageningen Code etc.) to allow comparisons over time and space. We use the AMCO (Amsterdam Code) as the main classification to harmonize the municipalities in our dataset. Not only does this classification cover the entire time span of our dataset, it is also built on the principle of *minimum varying* codes. In other words, municipalities get fixed codes and the system does not take changing names, composition or spelling variants into consideration.

Sub-municipal areas such as districts, neighborhoods and streets have been recorded from 1849 onwards throughout the Dutch historical population censuses. As we showed in Figure 4, municipalities are among the minimum required variables which need to be defined in order to get a total for a specific year and place. However, this total is made up of data from the sub-municipal levels. It would be interesting to be able to zoom in on these data. These lower-level areas in the historical censuses have been neglected by researchers for comparisons over time, mainly due to serious inconsistencies. Different years use different levels of detail and ways of organizing the sub-municipal levels, hampering longitudinal analysis. We build on the work of Boonstra (2007) and identify "Kom" (the Center of a village) and "Wijk" (Quarter or District of a town) as the only two sub-municipal variables which are usable for comparisons over time. These two are the most frequent lower level variables and available for almost the entire range of our harmonized subset of the data, i.e. 1859-1899.

Although the data on Kom and Wijk level are present, they have been poorly transcribed (Boonstra 2007; Ashkpour, Meroño-Peñuela and Mandemakers 2015) making it difficult to identify and utilize them. By querying the data as a whole and using basic NLP techniques and scripts, we identify each and every cell where a certain Quarter or Village center occurs and use these frequency lists for bottom-up classification purposes. To standardize these, we built on formal definitions provided by expert users of the data. These definitions were responsible for 90 percent of the "koms" being identified by our scripts and got standardized accordingly. However, the last 10% of the koms were not identified by the formal definitions, leaving us with around 10,000 exceptions. For example, during the first testing stages we noticed missing data for 1859. The problem here was that for the tables of 1859, transcriptions errors were made which resulted in the literal "Kom" being used in one line (string) next to the other variables instead of having its own column. To deal with this particular example, we used specific rules and scripts to identify whether a cell contained the term *Binnen* (inside) or *Buiten* (outside) de Kom and marked them as exceptions to include the 10.000 missing values for 1859. The final dozen literals which were not identified by our scripts where standardized manually.

3.5 Variable/Value Creation

One of the final stages of the harmonization workflow is the 'variable/value creation' stage. The main imperative of this section is based on the need for *bridging* and *filling* gaps in the final dataset. By *bridging*, we create new variables to make comparisons possible across the tables and over the years; by *filling*, we create solutions for value gaps in our data. During the previous stages of the workflow, we did not apply any harmonization where the *values (numbers)* are actually affected, the focus of this stage however is *exactly* that. In the case of harmonizing micro data, these are typically unnecessary steps, since there is always the possibility to (re)create variables and values according to one's needs. We have created this stage of the workflow in order to bridge between the different census years and compensate the lack of micro data by creating our own variables and values.

Bridging is done because we are interested in creating new *variables,* to make data comparable over the years, or to make implicit data explicit. We identify different types of variable creations. First we create variables from implicit data contained in other variables. Examples of these are the creation of variables for totals of *provinces*, *population* and the creation of values such as *temporary absent* from the *ResidenceStatus* variable. *Provinces* are not always explicitly defined in the tables but can be constructed by summing the values of the municipalities. The *Population Total* can be constructed by adding up the total number of *females* and *males*. The *value* 'Temporary Absent' from the *ResidenceStatus* variable was only provided for certain census years. By looking at the difference between the *Legal* and *Actual Population* size, we are able to provide an estimation of the number of 'Temporary Absent' individuals for years where there is no explicit data. However, in the case of dealing with different *age groups* over the years or changing *occupational classes* we have to use statistical computations to create new variables and values which *cannot* be derived from the census. This can be done by using various statistical techniques such as aggregation, estimation, interpolation etc. when required. For example, age groups can be regrouped to make e.g. 11-16, 17-22, 23-28 comparable with 11-16, 17-28 (by adding 17-22 with 23-28) or 11-18, 19-28 (by interpolating the group of 17-22). We *flag* the newly created variables all as *interpretations* in our dataset. The flag indicates what the change encompasses, tracing the harmonized data to the original sources.

Filling gaps refers to creating *values (numbers)* which are missing in the harmonized RDF database because of errors occurring in the workflow or simply because these values are not included in the original tables. Basically there are four reasons for occurring gaps: (1) data entry mistakes (2) mistakes in the construction of the styling, used to convert the Excel data into RDF (3) mistakes in the RDF syntax and (4) missing data from original tables. However much we try to harmonize everything and deal with all the peculiarities and

exceptions, we will always have some exceptions which do not comply with general rules. These exceptions result in empty cells or 'holes' throughout our harmonized dataset (see examples in Table 4). In order to fill these holes, we do not write specific exception rules or dive into the sources to manually identify a mistake for each and every random exception in the tables. To deal with these exceptions, we first apply different rules and scripts to *identify* and *estimate* the missing values. For example we found cases where a given variable, which is available for several consequent years, suddenly disappears and then returns. Or in other cases, we have data for six consequent years, except the last or first year (see gaps in Table 4). We use these *characteristics* as *detection rules and* write *generic* scripts to identify and fill in the gaps in a separate table called GapFiller.

Table 4: Example of Produced Harmonized Table with an Illustration of Different Types of Gaps

	AMCO	1859	1869	1879	1889	1899	1909	1920
Municipality	10002	335	390	.	442	539	672	.
	10071	252	275	283	.	320	364	458
	10072	223	273	305	.	.	405	474
	10073	209	268	367	378	345	.	470
	10035	.	251	314	410	545	654	699

Following this approach, we store these corrections in a separate file and never make changes to the raw data itself. See Table 5 for an illustration of the GapFiller table, providing different types of corrections to fill in the gaps and correct the data. Table 6 provides the GapFiller content in the same structured way as in Table 4 for illustration purposes.

Table 5: Example of Corrected or Estimated Values in the GapFiller Table

Census info	Original Value	New Value	Flag Nr
VT_1859_K234-s0	0	195	F2
VT_1879_T147-s0	0	420	F2
VT_1889_H437-s7	0	299	F1
VT_1889_T428-s7	0	342	F2
VT_1899_F317-s0	0	378	F2
VT_1909_01_T_h189	0	397	F2
VT_1920_01_T_h213	0	723	F2

Note: *F= Flag... F1 = no value, corrected manually. F2= no value, estimated.

Table 6: Structured Table View of the GapFiller Corrections to Illustrate the Filling of Gaps

	AMCO	1859	1869	1879	1889	1899	1909	1920
Municipality	10002			420 F2				723 F2
	10071				299 F1			
	10072				342 F2	378 F2		
	10073						397 F2	
	10035	195 F2						

The GapFiller table (Table 5) is based on four fields, i.e. (1) the definition of the table of the census and the cell number (2) the original value (or 0 in case of missing data), (3) the new value and (4) a flag number (description of the type of change according to our flag classification system). GapFiller contains all the corrections (i.e. estimations) which have been spotted by way of scripts or entered manually by users of the data. This file can be used by the correction department (in our case the archivists at DANS) to improve the raw data and by the software developers to improve the software (e.g. using the exception found during testing to build better vocabularies and data linking methods). Using this approach in conjunction with automatic estimation, we allow users to improve on the estimated numbers and overall quality of the data.

3.6 Testing

The source-oriented harmonization workflow we propose puts lots of emphasis on testing and positions it as the gateway to the final result, i.e. a harmonized dataset. In our workflow, each major data transformation process is directly connected, in an iterative way, to the testing stage. It is one of the most important stages in the entire process and the most time consuming part. The goal of testing is to eliminate any noise added during the conversion and different stages of standardization, classification and variable creation. This entails that we systematically compare the harmonized output to the original source files in order to make sure that the numbers we produce are correct.[5] By exploiting the structured nature of our Excel tables, we are able to test our results using only a part of the data. Once we have tested the results of a harmonized variable for a specific province of a certain year, the other tables (provinces) for that year are also accounted for. This is because the tables mostly share the same structure per census year for the different provinces.

Testing entails mainly the construction of longitudinal (SPARQL) queries, using the *standardization, classification and variable creation* outcomes with the *mappings* we assigned earlier. The goal is to produce exactly the same

[5] During this process we do not activate the Gapfiller table to prevent wrong comparisons because of improvement of the original data, GapFiller is used to deal with exceptions found in the final output.

numbers as found in the original Excel tables, but now harmonized over the years. To test our data we begin with querying for totals in the tables and use queries which return a single number, e.g. the *total* of *inhabited houses* in *Amsterdam*. In case of suspicious numbers, we use 'detailed' queries producing all the numbers in the Excel tables which made up that specific total. By doing so, we structurally investigate and identify mistakes in the data which we subsequently improve. Furthermore, an additional (complementary) way to inspect our harmonizations is by producing new versions of the Excel files, now ingested[6] with the standardizations we applied earlier. This allows us to map our new data in the original Excel tables. In these enriched tables, all the literals contained in the original column and row headers are enriched with the standardized terms provided by us. This allows us to *visually* inspect the mappings by just opening the file and hovering over a cell to see the associated standardizations. For example, in case of suspicious numbers one of the first steps is to look if the number we are looking for has all the correct mappings and standardizations assigned to it.

By structurally testing our data after each section of our workflow we have identified several typical mistakes such as; mistakes in the conversion from Excel to RDF, mistakes in the harmonization itself (i.e. wrong standardizations, classifications, etc.), issues regarding exceptions, the importance of creating preliminary tables to spot mistakes which otherwise would have been easily overlooked and dealing with software (preservation) related issues. The following subsections give an overview of the most common mistakes we have dealt with:

The Conversion: Update RDF Input

Mistakes in the data could be the result of mistakes in the conversion of the data from the Excel tables into the RDF structure. The conversion of the Excel tables to RDF requires manual input which was defined in so-called stylings. Decisions are made on the basis of the table layout and knowledge about the data. Poorly styled tables, tables with a specific layout which were not supported (yet) by our tool, forgotten ones or just certain styling choices of which the justification was not easily known beforehand, all resulted in incorrect or missing output in RDF. Some of these mistakes can be spotted directly after converting the data by just looking at the logs, others only when they are compared with the original data. Every time we find a case where a new styling is required we produce new versions which directly renew the existing ones in our online repository (GitHub).[7] We refer to this whole process as the Integrator. The CEDAR Integrator[8] is an integration workflow (set of scripts) that automa-

[6] <https://github.com/CEDAR-project/DataDump-mini-vt/tree/master/enriched-source>.
[7] <https://github.com/CEDAR-project/DataDump-mini-vt/tree/master/source-data>.
[8] <https://github.com/CEDAR-project/Integrator>.

tizes the semantic publication process. The integrator uses the outcome of the workflow to connect our harmonizations to the raw RDF graph.

Harmonization: Update Standardizations

The bottom-up approach is one which is coupled with iteration. Harmonization of aggregate historical data should not be a definitive commitment but a learning process. Our flexible approach is built exactly for this. Where we first started with defining a general set of variables, we (at the end of the various iterations) have developed quite specific mappings to deal with the many exceptions and peculiarities which are in the census. This meant that we often had to update our mappings, i.e. add new or correct current standardizations and update the classification codes.

For example, after standardization we directly test and analyze the results. After one of the first runs, we found that we were missing many municipalities. The problem was that we were missing certain combinations for municipalities because of spelling variants. To address this we wrote a repeatable script which produces mappings by setting a certain threshold for the Levenshtein distance, using the standard vocabulary we have built for the tables which *do* work. Once we set a new threshold and ran the mappings we went from 10,000 missing standardizations to just 20. To make sure no wrong mappings were applied we manually inspect a sample of the results. The remaining 20 mappings were later coded manually.

Dealing with Exceptions

Already during the first step of the harmonization process (standardization) we encountered the ambiguous nature of many variables and values. In other words, how to handle literals which have multiple meanings? The literal 'Huizen' for example could refer to a municipality in the province of North Holland but it could also simply refer to houses since that is the literal meaning of *huizen,* all in the same table. In this case we know by expert knowledge that 'Huizen' in the column headers *always* mean 'houses' and the ones in the rows are always municipalities. We created RDF queries to extract all the 'Huizen' literals and their specific locations in the excel tables to mark them as exceptions. To apply these exceptions we just added an extra column next to the original and standardized term in our harmonization input file. In this new column, we mentioned the specific location of the exceptions (on three different levels: *table*, *sheet* or *cell* level) and provide the appropriate standardization for that specific case.

Create Preliminary Tables

During the first harmonization rounds, we produce many versions of preliminary and intermediate harmonized tables. When the data is still being tested, the 'creating data' stage proves very useful to identify common mistakes. Having the end result in tables such as Excel or another (relational) table system is especially needed when dealing with RDF data because this kind of data are not meant to be visually inspected or read. The difficulty here is especially that we cannot know what we are missing by looking at the RDF graph database. In order to actually see what we have harmonized and test our result we query the RDF database and produce structured tables to spot certain mistakes in our data.

For example, we know by expert knowledge that the classifications and variables for 1859 and 1869 are quite similar and that there were no major changes in the municipal boundaries. By presenting the data in a tabular and readable form we could clearly spot that the first version we produced had too many changes between those two years, which was unexpected. Upon closer inspection we found that we needed to introduce more standardization variations and add missing municipalities. Other examples where we clearly saw many gaps in our constructed tables were for the tables of 1909 and 1920. These tables diverge from the rest with regard to how they were transcribed. The tables do not have any clear structural hierarchies, i.e. all variables are contained in one row instead of separate cells and columns (with no clear order, i.e. sometimes separating values with a dot, sometimes with a comma, or in other cases no separator at all). In order to include these years, we built custom repeatable scripts to identify all separate values which were contained in one single string based on expert input.

Processing: Update Software

Next to testing the different elements of the harmonization, we also acknowledge the need to keep developing and testing our tools, scripts and RDF output. Different scripts and automated processes make sure that our harmonization efforts are translated into RDF DataCube compliant data and made interlinkable. Problems occurring at this stage mostly concern server side issues such as crashing during the conversion process, outdated software resulting in processes not working, versioning of the software, conversion rules (scripts) which need to be changed or improved on by implementing more Semantic Web standards etc. Although rather rare these glitches can be prevented by regularly testing and updating the pipeline software. In the long term, organization commitment is needed to maintain the software and make sure the system stays up and running in the future.

3.7 Create (Final) Dataset

Once we have followed all the workflow steps several times and are satisfied with the quality of our data we actually make the data available for the scientific community and other end users. We do this in three different ways, putting the user needs at the forefront: querying, creating tables which can be downloaded and using a semi-automatic extraction system.

First of all, the harmonized data is available for querying via a so-called SPAQL endpoint. To help the user, we provide as many query examples[9] as possible, document it and emphasize how to use the correct mappings. All this aside, we acknowledge that the core users of this dataset (historians, sociologist, demographers but also the public) are not waiting to write SPARQL queries when accessing the data. Therefore, next to dissemination via querying we provide 'harmonized data dumps.' Users would like to have immediate access to the tables by simply having a link to download the data instead of query interfaces. These users have more knowledge of the data itself, are used to working with (big) tables and want to incorporate these files into their *own* workflows and tools with which they are familiar. We create the following *harmonized* output:

- Flat table in Excel and CSV format (the result of the query output: use this as the input for your workflow and tools)
- Structured Excel tables (hierarchical harmonized view on the data in Excel format, provides an intuitive overview across years in an eye glance)
- SPSS file (ready to use SPSS file with variables already defined)

We first start with producing the *flat table* which is the direct result from querying the RDF graph. This flat table is ideal for researchers to use as an input for their own workflow and tools such as Excel, SPSS or GIS tools. However, this flat table is not very intuitive for other users to inspect visually. In order to provide a table which shows the evolution and differences of the variables over time we create *structural tables* similar to the (hierarchical) structure of the original Excel tables. To build these more intuitive tables, we import the flat tables into tools such as SPSS, define the variables and build a structured (hierarchical) table. It was also this format that we used to (visually) spot mistakes or gaps in the final dataset. Moreover, users who do not want to be bothered by all the intermediary steps in creating their own structured tables and just want to analyze the harmonized data in a glance of the eye can use these tables to do so.

The third option in our data dissemination focuses on the more general users, which are just interested in looking at specific variables or just want to explore the data without being presented the entire set of variables. To allow

[9] <http://lod.cedar-project.nl/cedar/data.html>.

this we provide a 'guided variable query' option where users select the variables and values they are interested in and build (valid) tables.

This article is accompanied with an interface in the form of a website, <www.censusdata.nl>, including links to the harmonized data, RDF output, RDF query examples, mappings, documentation, GIS visualizations and more. We aim not only to suggest a workflow or a method but also to show the practical outcomes of the steps we have presented, providing tangible outcomes which are open for all to access (from the images to the harmonized data). We aim to stimulate greater use of the censuses which up until now were seen more as an 'interesting' dataset rather than a practical research asset. For example, using standard templates users can now simply query the harmonized database and ask for the total number of 'inhabited houses' across the seven harmonized years of our pilot case effortlessly. Prior to our efforts users had to consult 60 different tables and over 80,000 data cells in the original Excel tables to answer this question and end up spending more time on data integration than analysis.

4. Accountability

Documentation alone is not sufficient to account for the different data transformations. In order to provide accountability, we track and provide the trail of sources (provenance) on two levels. First, we describe the way our results are realized and give detailed information on the different harmonization practices applied to make the data accessible over the years. Second, we provide the trail to the underlying sources, linking the harmonized outcomes back to the original data, i.e. the Excel tables and the scanned images from the original books.

Provenance of the Harmonized Outcomes (The 'Source Trail')

Besides describing our variables, providing valid mappings, documentation, etc., we want to account for each and every number we produce in the final harmonized dataset. The software we developed for the integration pipeline keeps track of *all* the transformations made during the harmonization stages of our workflow. When the data is harmonized we produce different tables and are able to account for each individual harmonized number (a key requirement in historical research). For example, the query '*number of Occupied Houseboats across all the years and municipalities and sublevels*' produces thousands of harmonized results, for which we can provide the complete provenance. We can pick any number from this list and see how this specific value is created and which harmonizations were applied. According to our harmonization results the total number of 'Occupied Houseboats,' 'Outside the Center,' in '1889' for the municipality of 'Achtkarspelen' is 40, see Table 8.

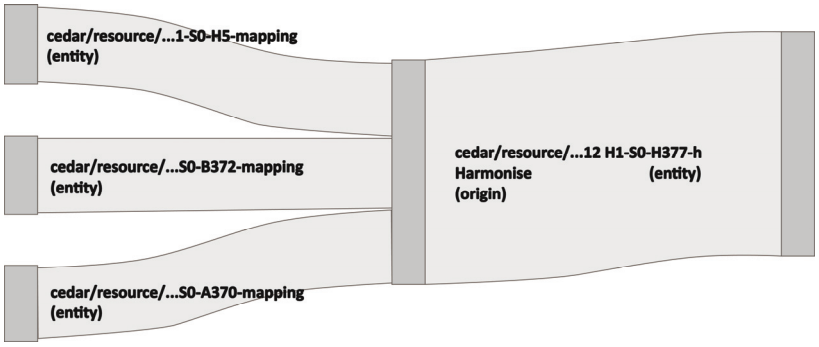
Table 8: Provenance Trail of the Harmonized Outcomes of the Number of Occupied Houseboats outside the Center of Achtkarspelen, 1899

Source	Municipality	Year	Pop
http://.../VT_1889_04_H1-S0-K132-h	http://.../amco/10199	1889	3
http://.../VT_1889_04_H1-S0-K79-h	http://.../amco/10199	1889	12
http://.../VT_1889_04_H1-S0-K11-h	http://.../amco/10199	1889	3
http://.../VT_1889_04_H1-S0-K118-h	http://.../amco/10199	1889	1
http://.../VT_1889_04_H1-S0-K40-h	http://.../amco/10199	1889	4
http://.../VT_1889_04_H1-S0-K56-h	http://.../amco/10199	1889	3
http://.../VT_1889_04_H1-S0-K69-h	http://.../amco/10199	1889	4
http://.../VT_1889_04_H1-S0-K72-h	http://.../amco/10199	1889	4
http://.../VT_1889_04_H1-S0-K101-h	http://.../amco/10199	1889	6

As shown in Table 8, we are able to trace back the harmonized RDF output to the original Excel files on a cell level. Using standard queries we are able to reconstruct how the total of the variable *pop* (it sums up to a number of 40) is generated, which file(s) and specific cells (e.g K132, K79, K11 etc.) are used to do so. To trace this back even further to the original sources, we make use of information already contained in the Excel tables and provide the necessary (meta)data to link these to the scanned *images* and *books*, presenting the *year* (e.g. 1889) and *type* of the census (e.g. VT, which stands for the Population Census), the *table* (e.g. 04_H1), *page* (e.g. 4) and *image number* (e.g. 03-0176). Next to providing the trail of the original sources, we can visualize the entire trail of the *harmonization practices* such as standardizations, classification, mappings etc. which were applied to retrieve this specific number, see Figure 6.

Figure 6 shows the visualization of the harmonizations we used for the query 'Occupied Houseboats,' 'Outside the Center,' in '1889' for the municipality of 'Achtkarspelen.' For example in this case, the mappings (harmonizations) used for cell H1-S0-H377h were from the S0-H5-mapping, S0-B372-mapping and S0-A370-mapping files. Using this information, users can trace back the specific harmonizations and see which standardizations and classifications values were applied in this specific case. For example, the corresponding mappings show that the classification code '10199' is used to harmonize the municipality of Achtkarpselen, the ResidenceType variable for the standardization of the value 'Occupied Houseboats' and the lower geographical value standardized as 'Outside the Center' etc.

Figure 6: Visualization of the Provenance Trail



By applying provenance at each stage of our workflow, we are able to point to the original sources at all times. With this information at hand, researchers can consult the original source data and actually see where the data comes from. Moreover, being able to connect the harmonized outcomes to the harmonization practices applied leaves room open for researchers to make their own interpretations when needed.

5. Conclusion

Harmonization of historical census data, especially in aggregated form and in comparisons over time, has been a relatively vaguely defined concept so far. By providing a source-oriented harmonization definition and workflow, we aim to make the concept of harmonization more concrete for researchers facing similar issues. The source-oriented approach is the preferred method in historical research; however this is not reflected in current harmonization efforts. In this article we appeal for *more* source-oriented harmonization efforts and provide a workflow to guide researchers in the harmonization process. We truly believe that the process of harmonizing the data itself can be made more explicit by following a structured and iterative approach which combines sets of known harmonization practices. Next to that, being able to connect the harmonized outcomes to the original *sources* (provenance) leaves room open for researchers to check the original data and make their own interpretations when needed.

Although the challenges, requirements and specific methods of census data harmonization have been thoroughly described, the lack of a structured workflow when dealing with these complex data prevents further development and use of many valuable datasets. In order to make the harmonization itself more

reproducible and explicit we have developed an iterative and structured workflow which builds on the source-oriented paradigm.

The workflow we described is based on the necessity of having a system which allows us to iteratively explore the peculiarities of our data. Flexibility is something which is usually not associated with harmonization of aggregate historical (census) data. Our harmonization workflow puts lot of emphasis on flexibility, accountability and allowing a learning curve when going through this process. Following a source-oriented approach is especially important in the case of aggregated data since interpreting and harmonizing this kind of data introduces more ambiguity compared to the harmonization of micro data. We have used our source-oriented harmonization workflow to test our methods and harmonized seven historical census years, spanning from 1859 to 1920. The goal of our effort was to build a census specific workflow, source-oriented harmonization methods, rules and tools which could easily be extended to include other years. For example, in order to harmonize the seven years of our pilot we already went outside of the scope of this particular subset. The harmonizations we provide on municipalities, lower-level areas, housing types, various demographical variables, residence statuses etc. are all present in some way or the other in the other tables and can be (re)used seamlessly. In fact, adding additional years to the data after defining previous years is a marginal effort in our system. As a result, the iterative nature of our workflow allows us to easily extend the data with additional years. Future work therefore mainly consists of adding more years, harmonizations and further enrichment of the data with other sources. We have explored the possibilities of *source-oriented harmonization* of *historical* censuses over *time*. By making the harmonization process more graspable in the form of a structured workflow, we make it easier for others to work with similar types of data. We provide structured and accountable harmonization solutions which are not bound to our specific dataset. Moreover, the final products of our efforts (the software we used, our scripts, tools, harmonized tables, harmonization rules, mappings etc.) are all deposited in online repositories in order to ensure its longevity and to stimulate further use by the public and researchers outside of the realm of our project. We provide these files in different formats to eliminate any intermediary step by the researcher and allow easy direct access (where no knowledge of RDF is needed to access the data). Accordingly, this article is accompanied with a website and data to show the tangible outcomes of our results <www.censusdata.nl>. We aim to inspire more *source-oriented harmonization* efforts and revive similar datasets in becoming more useable for historical research. We aim to provide a bedrock on which further re-interpretation of ambiguous, abstract, heterogeneous and disconnected (historical) data can be carried out. By doing this in RDF and using Semantic Web technologies, the data and harmonizations produced will become instantly accessible for other to re-use.

References

Ashkpour, Ashkan, Albert Meroño-Peñuela, and Kees Mandemakers. 2015. The Aggregate Dutch Historical Censuses: harmonization and RDF. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 48 (4): 230-45.

Beghtol, Claire. 2010. Classification Theory. *Encyclopedia of Library and Information Science*, 3rd ed., 1045-60. Taylor and Francis.

Boonstra, Onno. 2007. Buurten en wijken in de volkstellingen van de negentiende eeuw. In *Twee eeuwen Nederland geteld*, ed. Onno Boonstra, Peter Doorn and Rene van Horik, 455-70. The Hague: DANS.

Boonstra, Onno, Leen Breure, and Peter Doorn. 2004. Past, present and future of historical information science. *Historical Social Research* 29 (2): 4-132.

Boonstra, Onno, Leen Breure, and Peter Doorn. 2006. *Past, Present and Future of historical information science.* The Hague: DANS.

Cameron, Sonja, and Sarah Richardson. 2005. *Using Computers in History*. London: Palgrave Macmillan.

Cyganiak, R., D. Reynolds, and J. Tennison. 2016. The RDF data cube vocabulary. *World Wide Web Consortium.*

den Dulk, Kees, and Jacques van Maarseveen. 1999. The population cenuses in the Netherlands. In *A century of statistics. Counting, accounting and recounting the Netherlands*, ed. Jacques van Maarseveen and B. M. G. Gircour, 303-34. Amsterdam: CBS.

Doorn, Peter, Jan Jonker, and Tom Vreugdenhil. 2001. Digitalisering van de Nederlandse volkstellingen 1795-1971: met een nadere beschouwing van de gedigitaliseerde telling van 1899. In *Nederland een eeuw geleden geteld: een terugblik op de samenleving rond 1900*, ed. Jacques van Maarseveen and Peter Doorn, 41-64. Amsterdam: Stichting beheer IISG.

Esteve, Albert, and Matthew Sobek. 2003. Challenges and methods of international census harmonization. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 36 (2): 37-41.

Greenstein, D. I. 1989. A source-oriented approach to history and computing: The relational database. *Historical Social Research* 14 (51): 9-16 <http://www.ssoar.info/ssoar/handle/document/5189>.

Higgs, Edward. 1996. *A clearer sense of the census: Victorian censuses and historical research*, vol. 28. London: Stationery Office Books.

Mandemakers, Kees, and Lisa Dillon. 2004. Best practices with large databases on historical populations. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 37 (1): 34-8.

Meroño-Peñuela, Albert, Ashkan Ashkpour, Marieke van Erp, Kees Mandemakers, Leen Breure, Andrea Scharnhorst, Stefan Schlobach, and Frank van Harmelen. 2015. Semantic technologies for historical research. *Semantic Web – Interoperability, Usability, Applicability* 6 (6): 539-64.

Meroño-Peñuela, Albert, Ashkan Ashkpour, Christophe Guéret, and Stefan Schlobach. 2016a. The Dutch Historical Censuses as Linked Open Data. *Semantic Web – Interoperability, Usability, Applicability* 0: 1-14.

Meroño-Peñuela, Albert, Ashkan Ashkpour, Christophe Guéret, and Stefan Schlobach. 2016b, under submission. An Ecosystem for Integrating and Web-Enabling Messy Spreadsheet Collections. *Knowledge Based Systems*.

Muurlings, Sanne, and Kees Mandemakers. 2012. MOSAIC Census Inventory of the Netherlands. *IISG.* Max Planck Institute for Demographic Research <http://www.iisg.nl/hsn/documents/mosaic-wp-2012.pdf> (Accessed May 10, 2016).

Ruggles, Steven, and Russel R. Mennard. 1995. The Minnesota Historical Census Projects. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 28 (1): 6-10.

Slavakis, Konstantinos, Georgios B. Giannakis, and Gonzalo Mateos. 2014. Modeling and optimization for big data analytics: (Statistical) learning tools for our era of data deluge. *IEEE Signal Processing Magazine*: 18-31.

St-Hilaire, M., B. Moldofsky, L. Richard, and M. Beaudry. 2007. Geocoding and Mapping Historical Census Data: The Geographical Component of the Canadian Century Research Infrastructure. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 40 (2): 76-91.

Thaller, Manfred. 1993. What is 'source oriented data processing'; what is a 'historical information science'. In *Istoriia i comp'iuter. Novye informatsionnye tekhnologii v istoricheskikh issledovanii akh i obrazovanii,* ed. Leonid I Borodkin and Wolfgang Levermann, 5-18. St. Katharinen.

van de Putte, Bart, and Andrew Miles. 2005. A social classification scheme for historical occupations. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 38 (2): 61-92.

van der Meer, Ad, and Onno Boonstra. 2006. *Repertorium van Nederlandse gemeenten vanaf 1812-2011.* The Hague: DANS.

van Leeuwen, Marco H. D., Ineke Maas, and Andrew Miles. 2004. Creating a Historical International Standard Classification of Occupations: An Exercise in Multinational Interdisciplinary Cooperation. *Historical Methods A Journal of Quantitative and Interdisciplinary History* 37 (4): 186-97.

van Maarseveen, Jacques. 2003. *De virtuele volkstelling en het sociaal statistisch bestand: een verslag van de conferentie gehouden in Amsterdam op 11 november 2003.* Amsterdam: CBS.

van Maarseveen, Jacques. 2008. *Dutch Occupational Censuses 1849-1971/2001. A component of the Population Census.* The Hague: CBS.

van Maarseveen, Jacques. 2002. Intrekking Volkstellingenwet. Registertellingen; op weg naar volkstellingen nieuwe stijl 1979-2000. In *Algemene tellingen in de twintigste eeuw*, ed. Jacques van Maarseveen, 89-114. Voorburg: CBS.

W3C. 2015. *What is provenance* <https://www.w3.org/2005/Incubator/prov/wiki/What_Is_Provenance#A_Working_Definition_of_Provenance> (Accessed May 10, 2016).