Engineering Complex Data Integration, Harmonization and Visualization Systems

Iman Avazpour[a], John Grundy[b], Liming Zhu[c]

[a]*School of IT, Deakin University, Burwood, VIC 3125, Australia*
[b]*Faculty of IT, Monash University, Clayton, VIC 3800, Australia*
[c]*Software & Computational Systems, Data61, CSIRO, Australia*
[d]*School of Computer Science and Engineering, University of New South Wales, Sydney, Australia*

Abstract

Complex data transformation, aggregation and visualization problems are becoming increasingly common. These are needed in order to support improved business intelligence and end-user access to data. However, most such applications present very challenging software engineering problems including noisy data, diverse data formats and APIs, challenging data modeling and increasing demand for sophisticated visualization support. This paper describes a data integration, harmonization and visualisation process and framework that we have been developing. We discuss our approach used to tackle complex data aggregation and harmonization problems and we demonstrate a set of information visualizations that can be developed from the harmonized data to make it usable for its target audience. We use a case study of Household Travel Survey data mapping, harmonization, aggregation and visualization to illustrate our approach. We summarize a set of lessons that we have learned from this industry-based software engineering experience. We hope these will be useful for others embarking on challenging data harmonization and integration problems. We also identify several key directions and needs for future research and practical support in this area.

Email addresses: iman.avazpour@deakin.edu.au (Iman Avazpour), john.grundy@monash.edu (John Grundy), Liming.Zhu@data61.csiro.au (Liming Zhu)

Accepted to Journal of Industrial Information Integration *August 27, 2019*

1. Introduction

One of the most common problems in computing is the need to integrate multiple sources of information represented in disparate data representations in order to leverage the combined information i.e. to "harmonize" the disparate data into a single, consistent form [1, 2, 3]. Once systems are created, changing the data formats is very expensive due to engineering and change impact propagation. When integrating data sources with a diverse set of federated owners, changing them can be impossible due to ownership and even legal issues in government datasets.

Various research and industrial applications have been working on developing such data mapping and aggregation solutions in order to make transitioning from one data format to another less expensive and more user-friendly format (e.g. [4, 5, 6, 7]). This is becoming an increasingly common problem with the increase in availability of large and open datasets and demand for such data integration, ongoing updates, analysis and visualization [8], while addressing privacy and security concerns [9]. Developing an automated end-to-end process to support data wrangling and harmonisation will potentially result in a data product of limited quality [10]. On the other hand, it can be argued that without suitable visualizations, understanding and using such integrated large data in these aggregated systems quickly becomes very cumbersome [11]. Many users are not familiar with the low-level representation of data that is often targeted to specific technical audiences and applications [5]. To address this limitation, standard and familiar visualizations need to be incorporated that use the integrated data sub-systems. Key Example application domains for these approaches include healthcare – integrating Electronic Medical Records from disparate systems and providers; "smart city" transport systems – integrating traffic, road usage and control data from multiple systems; and land information systems – integrating GIS, land usage, and agricultural data.

This paper reports on our efforts to develop a software engineering process for complex data integration, harmonization and visualization problems. We describe the process and a prototype toolset, Harmonizer+, that we implemented to achieve such diverse data integration and harmonization. Our approach involves analysis of disparate source data models and then the design of a harmonized, consistently aggregated schema that can represent the diverse source datasets. Model mappings from each source schema are devised and used to generate complex code to transform source data into

the new harmonized schema format. Data is aggregated in a consistent way to allow querying and combining but respect different source data privacy and anonymity requirements. Finally, we have developed an interactive data visualization support tool for the harmonized data set.

As a concrete industrial case study using our approach, we use a new data integration and visualization example into the Australian Urban Research Infrastructure Network (AURIN). AURIN is an Australian initiative to make a wide range of demographic, economic, social, cultural and geographic datasets available to geographers, sociologists, government agencies, businesses and ultimately citizens for multi-dataset querying, aggregation and visualization [12]. An example of such multi-dataset information are Household Travel Surveys (HTS). For example, planners would like to know how people currently travel to and from work or school; if there are socio-economic, demographic or other impacts on these travel choices, and emergent behaviors over changing time, demographic and economic conditions. AURIN envisions to provide a consistent data querying and web-based information visualization architecture. We used our Harmonizer+ process and toolset to harmonize, integrate and visualize several State government HTS datasets into the AURIN framework.

The rest of this paper is organized as follows: in Section 2 we outline our motivation and problem statement. Section 7 briefly outlines key related work. We describe our approach to realizing a consistent, integrated HTS data schema, aggregated data and data visualization support in section 3. In the following section we describe our harmonized data model development and modeling of source data to target harmonized schema transformation. Section 5 provides details of our Harmonizer+ architecture and implementation. Section 6 provides a summary of strengths and weaknesses of our approach and key lessons learned from its industrial case study application . It lists some areas for future research and practice.

2. Motivation and problem statement

The Australian Urban Research Infrastructure Network (AURIN) is a national institute aiming to gather data from participating Australian states. It provides a framework for researchers to access, investigate and use a wide range of data from across Australia [12]. Data includes census results (e.g. demographic and socio-economic profiles), geographic data (e.g. location of roads, rail, and other infrastructure), and organizational data (e.g. Com-

monwealth, State, Local organizational structures, businesses, and hospitals) among others.

Household Travel Surveys (HTS) are an example dataset that provide insights into mobility patterns and utilization of public and private transport. Across Australian states, a number of diverse HTS have been conducted by different government agencies to find out the travel behaviors of citizens. Unfortunately all states use vastly differing data formats to record survey results. Many aggregate these results using different street, locale, suburb, demographic or other categorizations. The systems supplying the data are diverse - data comes in CSV, XML and relational formats. Some systems support interactive querying while others only batch export. None provide effective visualization capabilities especially when combining the HTS data with other data.

The AURIN project wanted to integrate HTS data seamlessly into the wider project resources, including a single harmonized data model, regular data updates, multi-dataset querying, and integrated and effective visualization support. HTS data integrated with other AURIN data would enable researchers to explore and discover new knowledge around Australian's mobility patterns. It would allow planners to investigate for example, how transport infrastructure could be improved, discover relationships between travel choices, determine how travel choices are influenced, and might even allow for improvement of travel outcomes.

The above exemplar illustrates the diverse range of data sets, data integration challenges, harmonization issues and visualization needs our work is addressing. Given the diversity of data collection instruments and vastly different data aggregation methods, we need to take source datasets and integrate them into a harmonized and consistent form to be useful. Various different aggregations of the source data often need to be made available. This enables users to access data collected by different organizations and compare their information patterns. The harmonized data could also be queried with other datasets e.g. such as other demographic data available in the AURIN framework. The results of these queries could be exported as common data formats (e.g. CSV, Spreadsheets) or visualized to better enable user investigation and analysis.

The following points summarize some of the key problems that an effective data integration, harmonization and visualization solution needs to address:

- Data harmonization and integration:

  - Data is sourced from a diverse range of repositories
  - Data repositories use a range of technologies to provide access
  - Data has differing levels of aggregation, often done to preserve information privacy or reduce data storage and querying costs
  - Access to disaggregated data is limited. Many systems provide batch data export to CSV, XML, Excel or relational formats rather than providing "live" access
  - Data represents different categories and classifications across data providers that need to be agreed.
  - A single, harmonized data model must be produced that is capable of representing all State HTS datasets, at disaggregated and multiple levels of orthogonal aggregation.

- Data query and visualization:
  - An interactive visualization capability is needed to allow end user access to and exploration of the harmonized data
  - Some harmonized data can be drilled down into, some cannot
  - User specific visualizations e.g. research analysts from diverse domains, government planners and ultimately business and citizen end users need to be supported

Here, we primarily focus on addressing data harmonization and integration requirements, while also describing example solutions for harmonized data query and information visualization.

3. Our Approach

Our overall approach to harmonization aims to use data transformations to make disparate available datasets *usable*, as stated by Kandel et al. [13]. The harmonized data will then be queried and a set of visualizations for the queried data will be generated to support end users (data consumers) in their analysis of the integrated data. Figure 1 outlines the approach we take and its main steps: Cleansing, Wrangling and Usage. In the following, we describe these steps used within our Harmonizer+ framework.

Data cleansing involves three main tasks: reading data documents, examining data formats provided from each state (1); identifying shared data e.g.

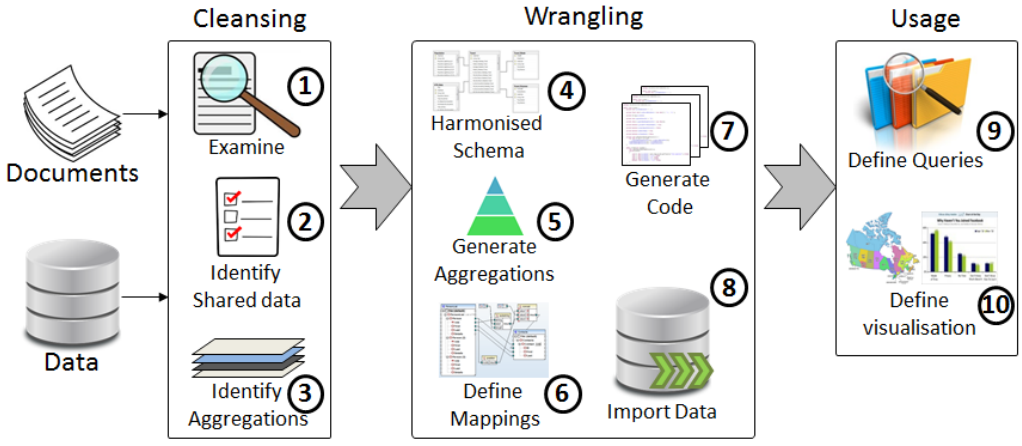


Figure 1: Outline of our approach.

locale, demographics and missing data fields (2); and identifying commonalities, differences, and aggregation levels (3). Our wrangling step includes defining harmonized schema and possible aggregation levels (4); generating aggregators for imported raw data to match other existing data (5); defining mappings from each source dataset to harmonized dataset (6); generating code based on the mapping and transformation specifications (7); and applying the mappings and transformation while importing data from various datasets (8). The final stage of our approach defines how the harmonized data is going to be used and it includes querying the harmonized data and other available datasets (9); and developing visualizations (10). In the following, we describe each step in more detail and in the context of our HTS data harmonization case study.

4. Case Study: Household Travel Survey Data Integration and Harmonization

4.1. HTS Integration Process

The problem of taking data in its initial form and transforming it into a desired form is known as data integration [14]. Success and failure of data integration frameworks is dependent on understanding the data's context-sensitive meaning and the quality of the data [15]. Kendal et al. provide an example were a data analyst is faced with datasets provided by three states and has to perform much trial and error to cleanse the data before being able to use it [13]. Each source data set needs to be carefully understood from

available documents, schemas and example data (1). In our experience this can be challenging due to the variable quality and availability of information and often needs experimentation and domain knowledge. In the context of HTS data integration and harmonization, we first spent considerable time to understand each state's HTS data via their documents, schemas and example data (step (1) in Figure 1). Unfortunately much of the documentation and schemas we had to work with on this project were incomplete or inaccurate. We then used a variety of tools to aid us in exploring the data and schemas including SQL Server, Altova MapForce, and Excel. This contributed to the bulk of our time (almost 60%). However, it paved the way for the rest of the harmonization to produce an effective result.

We then identified shared information, even if in different formats, between HTS schema and datasets e.g. geographic location, local government organizational units, and demographic information (step (2) in Figure 1). We identified commonalities and differences in the data item formats used to represent the same information. For example, a data field representing "Tram" as a mode of transport in one dataset may be represented as "Light Rail", or via a nominal value (e.g. 20) in another. Differences in record structure, differences in foreign keys, differences in hierarchical structure, and differences in aggregation levels across records and hierarchies (3). For example, we were given access to raw HTS data by one state and data aggregated to household level by another state, and data aggregated to local government area (i.e. county in USA context) by another. Such issues are becoming very common; for example due to varying privacy policies across states.

Using steps (2) and (3) and end user inputs, we then designed an integrated data model broad enough to represent all data structures and fields in the source HTS datasets and their schema (4). As we are designing for a very wide range of potential end users with a goal of preserving the original data as much as possible, end user inputs did not play a major role in designing the integrated data model for this project. We then generated the aggregation procedures required to transform each HTS source data to the same aggregation level required by the harmonized schema (5).We tried using several (semi-)automated data aggregation tools, but none could meaningfully automate the aggregation in our case. We were forced to use SQL queries to manually aggregate the datasets.

The mappings between each data model's corresponding elements and groups, and the harmonized model were defined next (6). We had to design in house methods to keep record of and document these mapping specifica-

tions. We then generate implementations of data import, aggregation, and mapping specifications (7). This can ideally be done using automated code generation facilities provided by tools like Altova MapForce. These code generation facilities usually take source and target schema, and specifications of import, aggregation and mappings and automatically generate the required code in multiple programming language syntax. This is particularly useful if the integration project is to be embedded inside another framework. The generated code needs to then run on each source system dataset and be carefully checked for errors, correcting mappings and regenerating code where necessary (8). Data import needs to run every time new data becomes available and depending on source of the data, can be batch execution or applied on stream data. Any major changes in new data formats may trigger the repeat of early tasks. This was a highly iterative process.

Final stage of the harmonization and integration is to define queries over harmonized data and other externally-stored data (9). The data was then used to design and generate interactive visualizations with CONVErT [16, 17] data mapping and visualization toolset (10). Through following sub-sections we will describe more details on the approach and its usage on the case study example of HTS data harmonization.

4.2. Harmonized HTS Data Model

Household Travel Surveys (HTS) are conducted to provide record of how people travel in a specific geographic area. These surveys are generally designed to investigate "how people utilize public transport or personal vehicles", "what are the main purposes of their travel", and some "demographics" for example what is the structure of households (household size, occupations, etc.).

At the starting point of this project, we had received four datasets from three Australian states namely New South Wales (NSW), Victoria and Western Australia. NSW had provided two sets of data with different aggregation levels. Ideally, there would be one single, consistent, agreed and detailed HTS data model that all Australian states use - or even an international standard allowing cross-country comparison. Unfortunately, such a model does not exist. In practice the states all use vastly different models, designed for different purposes, different data collection strategies and tools, different databases, and different underlying data formats. This will be a recipe for many inconsistencies across the datasets. In the following subsection, we

provide details of the inconsistencies we encountered and how we addressed them.

4.2.1. Major Inconsistencies Encountered

Given that the travel surveys were conducted using different survey instruments and by separate organizations, we were faced with many inconsistencies in the data. We have grouped these inconsistencies into five categories described below.

Different types of data access points. The data access points each state provided were very different. Some states provided web service and/or database access, others batch query results in form of XML or CSV file dumps.

Different high level data structure. With different data access points, data samples also came with different high level data structure. For example one sample used a relational database with various tables, while another sample was one CSV file that could be represented as a table in any type of database system.

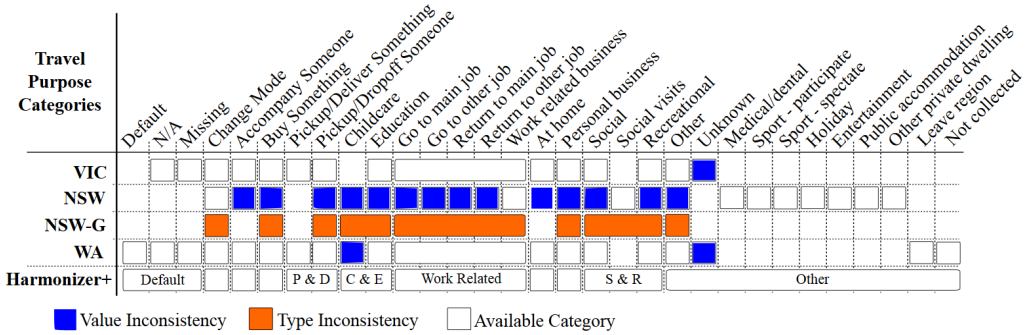
Low level data item formats. Many states had different low-level data item formats that would have to be transformed to a common representation e.g. times, dates, addresses, locations, transport modes, and purpose fields. Many numbered fields had different types as well. For example, trip distances were recorded as float, long, double, or in case of CSV files as strings.

Different coding and categorization structures. How categories are recorded were also different. For example, modes of transport could be recorded as nominal values (i.e. numbers represent modes, e.g 2 = vehicle, 4 = public transport), as text (e.g. "vehicle"), or in separate columns (e.g. a column representing how much of distance is traveled by public transport, a column for distance by vehicle, and another distance by bike, and so on).

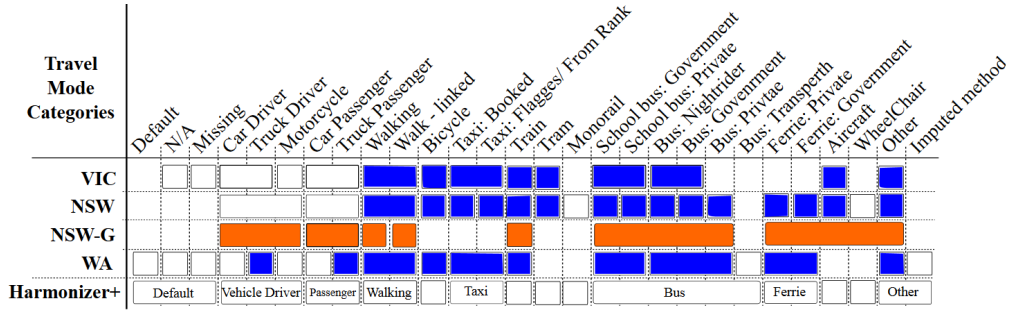
Missing data types, categories, or information. Since the surveys were conducted in isolation, our datasets represented many fields that were missing. This could be due to unavailability of a certain facility, lack of importance of recording an item, or different routines and procedures. For example, one state did not record how many bikes are available in each household. In another example, a state did not have Tram as a means for public transport so it was not included in the list of travel modes.

Figure 2 shows two examples highlighting inconsistencies in the categories used for *travel purpose* (e.g. work, school, leisure, health treatment) and *travel mode* (e.g. Bus, Car, Walking) across different states. Each column in the table reflects a category. Top row of the tables list all categories that

are used across all datasets. The middle four rows list different state provided datasets. Since the state of New South Wales had provided a sample aggregated data and a sample of their raw data, there are two rows associated to this state's categories (NSW and NSW-G). The bottom row lists the categories we used for our final harmonized data model.



(a) Categories used for travel purpose.



(b) Categories used for travel mode.

Figure 2: Samples of inconsistencies encountered within categories used in the datasets.

The tables of Figure 2 represent available categories by boxes. For example, the dataset provided by state of Victoria includes a travel purpose for Accompanying Someone, as a result a box is put in the corresponding **VIC** row and *Accompany Someone* column. Same is true for **NSW** and **WA**. Where the category is missing in the dataset, the representative cell in the table is missing. For example **NSW-G** does not provide information for *Accompany Someone* and therefore does not have the box. Tables of Figure 2 use colors to reflect different types of inconsistencies discovered in these datasets. Low level inconsistencies are shows by *Orange*, Categorical and coding inconsistencies are represented by *Blue*. For example, categories of **NSW-G** are represented by strings while others use integer nominal values.

As a result, categories of **NSW-G** are represented by Orange. Since most of the datasets provided nominal values, there is also the possibility of different values representing different or similar categories. For example VIC and WA datasets represent *Accompany Someone* by 2 while NSW represents it by 20, as a result the box representing *Accompany Someone* in **NSW** is blue. These inconsistencies should all be considered in the design of data mappings.

Similarly, there are accumulated categories (we call them multi category). For example NSW has multiple categories indicating different types of work related purposes while VIC, NSW and WA consider an accumulative field for all work related categories. These accumulative fields can be spotted with the spanning boxes across multiple columns. We had to develop such detailed data analysis and comparison tables to aid us in determining a suitable harmonized data model that could represent all of the combined data in a single manner.

4.2.2. Data Aggregation

In addition to the encountered data inconsistencies, we were faced with data at quite different aggregation levels in the source datasets. For example, we had access to the data collected by surveys, as well as data aggregated into geographical areas, for the state of New South Wales (NSW), who could provide two sets of data. One is raw data collected from user surveys that has strict privacy requirements and cannot be released. The other is aggregated to Local Government Area (LGA) as defined by the state government. Some states provided raw data only. Other states provided aggregated data to differing levels e.g. locale, street, street groups, LGA, Statistical Area (SA2), or combinations.

For this aggregation process, AURIN partners agreed that the preferred level of aggregation be at SA2 level. However, some providers do not provide data down to this level and hence we had to compromise to the LGA level. We chose to have two sets of data: one based on available raw data and/or SA2 level (when available), and another based on aggregation of the raw data to LGA level. This would give AURIN users the opportunity to see the aggregated data first and compare it. They would then investigate the raw data below this level to drill down to the more detailed data based on differing individual surveys. In addition, unharmonized data can still be exposed directly through state-specific original datasets. Thus our compromise in using LGA vs. SA2 or removing information in our harmonized data sets, are not very detrimental or irreversible since the source data is still accessible

if needed.

We used SQL querying and Microsoft SQL Server (MS-SQL) to develop our aggregations. This decision was due to availability of information about the data and the databases structure. A temporary database was defined in MS-SQL and the raw data was imported into this temporary database. Then the required queries were defined to calculate aggregations and save as new datasets. For example, a query was written to select all travels grouped by their specific LGAs and calculate sum of the distance traveled by vehicle. Differing queries were applied to raw data and aggregates below the LGA level. LGAs were then normalized to a consistent set across Australia. The process was performed once and the required queries for load, transform and export where registered as batch operations for future reuse.

4.2.3. Harmonized Data Model Design

We needed to design a new, harmonized data model for our canonical and intermediate database. This database was to be used for storing harmonized dataset where data queries for visualization would be executed on. Additionally, it would play a role in bringing together the different available datasets providing a unified data schema to map the original data to.

For the design of this harmonized data model, we needed to consider some important data limitations. Since the provided multi-state data had a variety of inconsistencies, we had three options for designing the data model: accepted majority; available in depth; or a combination of both. Accepted majority indicates the data that is available in most provided datasets. This would have helped to simplify implementation of the required data mappings. However, it would also mean that we had to remove some information provided by different states. Available in depth on the other hand, would allow us to keep all provided information, but at the expense of some incomplete datasets. Although available in depth information would not reduce the information, it would not provide a dependable platform for comparison of the data provided by different states. Additionally it would pose problems for visualization frameworks as they then need to cope with missing data. We chose the third approach which is a combination of both.

To determine how to best organize the harmonized dataset, we interviewed a sample of potential end users for the Harmonizer+ platform to see what their needs would be for this harmonized dataset and its usage. We interviewed three social science researchers who were among the end users of the final system and asked them a range of questions about their needs.

This included: what they would like to have/see in the data? what data aggregations and data elements they particularly need in their research? what platforms and data they are currently using? and what they would find useful features to have in the harmonized dataset? We also asked them about their information querying and visualization needs. Some key findings of these interviews regarding *data availability* and *data usage* are summarized below.

Data availability issues:

- Higher release frequency of data from per year to per quarter or half year - the researchers need updated HTS data more frequently than a per-year release. Some surveys are conducted quarterly or even monthly. Our Harmonizer+ solution must support update of the HTS datasets at least quarterly.

- Future-proofed support for other units of data collection beyond household - researchers see the need for data collected from other societal groups in the future, such as by organization or by venue. These need to be provided by design in our solution even though not currently available, to avoid major re-engineering efforts later.

- Targeted in-depth profiling of areas (such as disadvantaged areas and individuals) - as social scientists, these end users want to tag data, where possible, with demographic, socio-economic and other information, but with privacy considerations in mind i.e. individual or family privacy must not be breached.

- More context information about the data, including historical data - researchers want to be able to query and visualize information relating to e.g. long term changes in travel behavior due to suburb development, infrastructure development, changes in demographics etc. This means we must keep more contextual information with HTS data items as well as historic survey data and be able to link it with newer survey data.

- Common data format and data model from different states - a key goal of the project. However, additional states must be able to be added that themselves come with different HTS datasets, thus forward-looking design is essential to enable incorporation of foreseeable data not in the current models.

- Integrating with per-state data privacy policies - states have differing data privacy legislation and care must be taken not to breach any or collectively when making aggregated data available to Australia-wide researchers.

- Configurable data sets to allow different types of access from different types of users such as free or paid subscribers, privacy-compliance agreement signed or not signed. For example, some data attributes or operations on attributes may be removed for different user groups.

- Ability to accommodate continuous surveys in future. For example, user mobile devices that automatically track travel patterns and users supply additional information on purpose of travel and travel mode choice.

Data Usage issues:

- Highly customizable reports including visual reports - researchers need much more customizable reports than the existing AURIN system which incorporate primarily visual abstractions. End user specification of reports would be ideal.

- Visual graph for representing data set relationships - including support for recommending possible combinations and extensions of one data set to another e.g. suggesting combining specific locale, socio-economic or other filter/group-by.

- Mechanisms for easy linking to other attributes available in other data sets - this includes information such as income, house price, school support, access to services and goods transportation. Ideally, we would allow these users to arbitrarily integrate additional information.

- "Playground functionality" - this includes supporting what-if questions around how a data change would affect some other related attributes without deep domain knowledge. The ARUIN portal provides some basic statistical tools such as regression or descriptive statistics to allow explorative studies. The playground would allow users to include/exclude certain attributes or adjust weightings for regression analysis or descriptive results. For example, a HTS data user may want to try to do a quick examination of an attribute in another data

set (e.g. house price, walkability metrics of a suburb) to see if it has impact on total trips, before deciding to download that data set.

- Support for widely differing user group needs - this includes social science research experts, citizens, journalists, rare and daily users.
- Live on-line support for data end users - this includes pre-packaged and reusable queries and visual reporting.
- Better support of zoom-in functionalities in visualization - The default AURIN visualizations provide aggregated visualizations in charts and heat maps. These visualization, although necessary for abstract insights, does not provide detailed information of the surveys.

Based on our analysis of the base data sets commonalities and differences, and the findings from our end user interviews, we chose the following features and information to keep and to omit respectively from the datasets. Our approach was to define two separate data models and intermediate databases, one for raw data and one for aggregated data. Parts of our intermediate aggregate and harmonized database are shown in Figure 3. The main tables are HTS_Data and Travel. HTS_Data provides basic information about the surveyed LGAs including total households, average people per each household and total vehicles. Travel table provides more fine grained information regarding types of trips, time, distance, purpose and mode. The values in these tables are estimated according to surveyed participants responses.

Tables Travel, HTS_Data and Population are essentially linked by a one to one relationship, which is not a usual database design practice. We made the decision to separate them for the following reasons: HTS_Data in its current form provides generic information regarding LGAs that satisfies most researches needs. Travel table would provide more in depth information for interested researchers. This separation would allow faster processing for generic purpose querying in our on-line architecture. Only one state records participants' age as age groups. As a result, this information is recorded in separate tables. Similar to Population table, information regarding Vehicles including types, fuel, make and models, Work including occupations and income were not provided by states specially due to privacy concerns. We have included them in the design for future proofing the model.

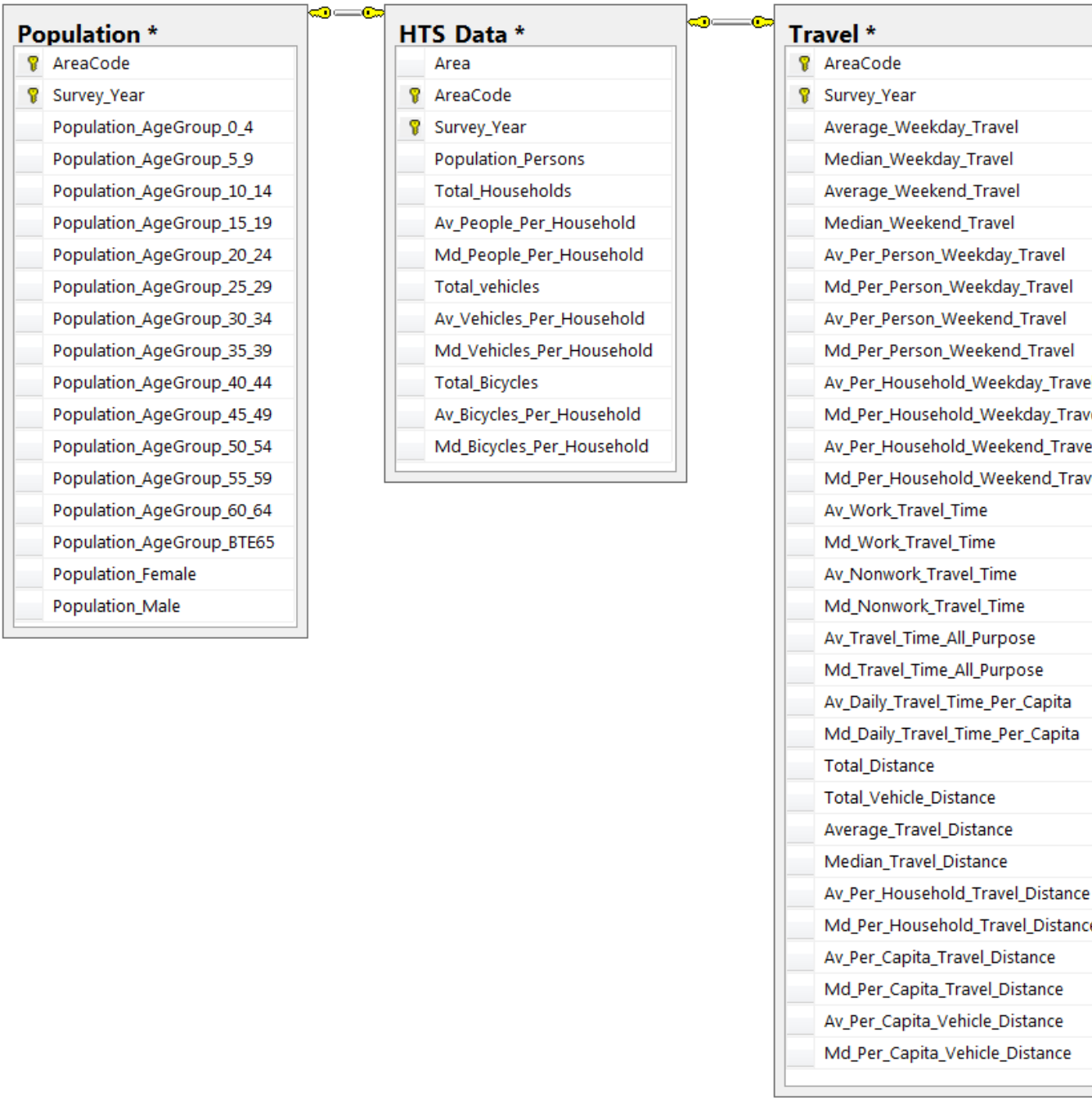
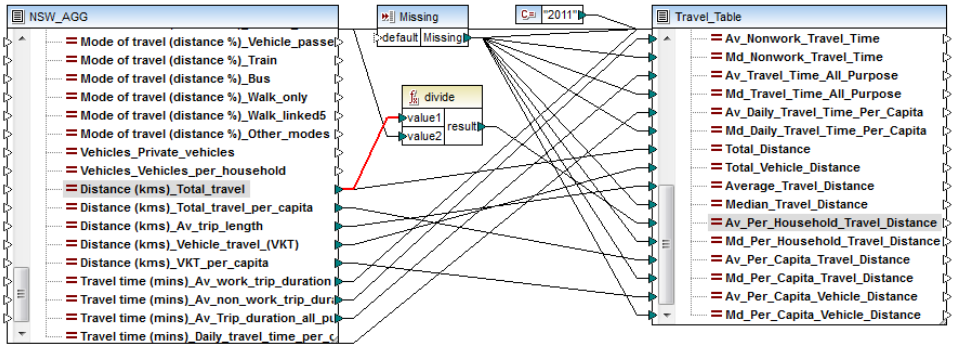
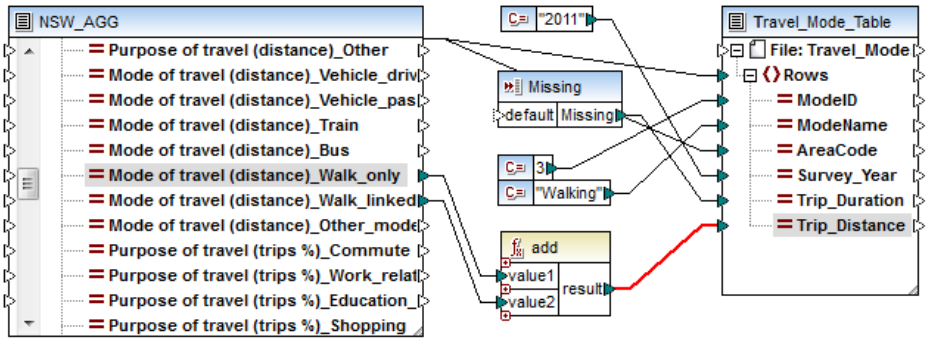


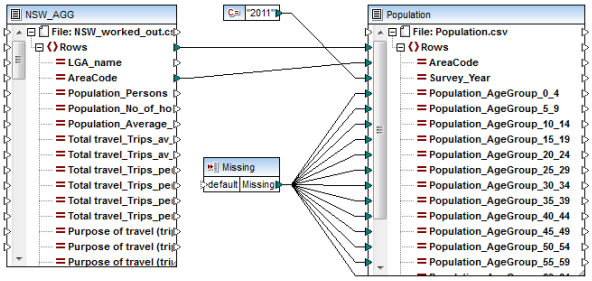
Figure 3: Part of the harmonized data model showing base travel data entities.



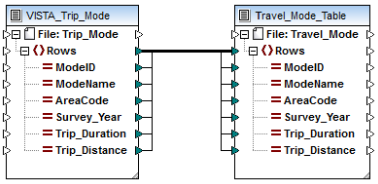
(a) Using a divide function to calculate average travel distance per household for an LGA



(b) Using add function to combined two walking related values of NSW-G dataset



(c) Providing defaults missing values



(d) Mapping calculated aggregated values.

Figure 4: Sample of mapping datasets to portion of the harmonized data model using Altova MapForce.

4.3. Defining Schema Mappings with Altova MapForce

With our harmonized data model design completed, our next step was to define the data mappings. These mappings would import the collected

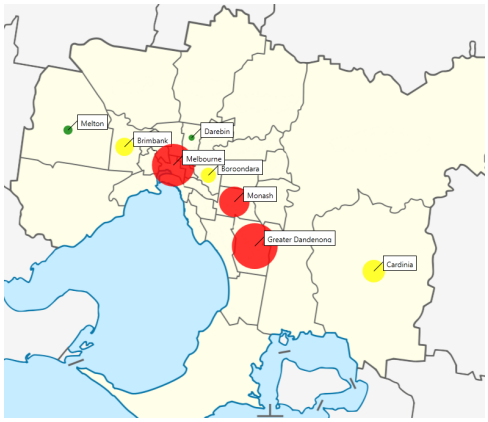
data to the new data model. We used Altova Mapforce for this data mapping task. Mapforce provides a powerful, flexible and relatively user friendly framework for complex data mapping. It provides the necessary data connectors to connect to various data sources that eliminates separate coding of the connectors. MapForce automatically generates schemas for imported data and allows viewing source and target schemas side by side. Mapping correspondences can be defined by drag and dropping elements of source and target schemas. Complex 1 to many, many to one and many to many transformations can be specified. Some such mappings we used are shown in Figure 4. From these mappings, skeleton codes for mapping and transformation implementations can be generated that are included in our Harmonizer+ framework.

The mappings depicted in Figure 4 provide various examples of fixing inconsistencies or calculating required values. For example, a divide function is used to calculate average travel distance per household for an LGA, a value that was not provided with the dataset (Figure 4(a)). Or an add function is used to combined two walking related values of NSW-G dataset (Figure 4(b)). The figure also depicts how default missing values are provided to population table where the necessary data is not available by the dataset (Figure 4(c)).

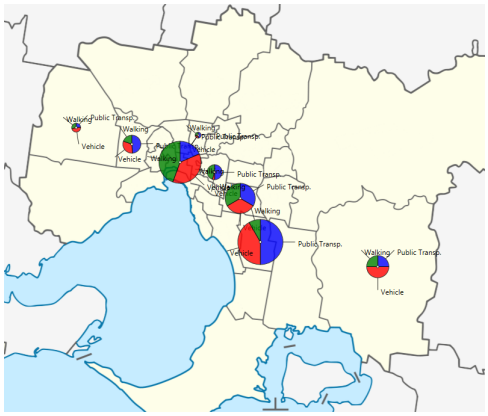
When generating aggregations of provided raw data, we had our data model in mind. That is, the raw values were aggregated to LGA level, and structured to the harmonized data model at the same time. This would provide easier integration of aggregated datasets and the harmonized data model. For example, Figure 4(d) shows how aggregated trip mode values of the dataset provided by state of Victoria is mapped to harmonized travel mode table.

From the MapForce specifications we generated a set of Java programs that extract data from each state's provided HTS dataset and import it into a single, integrated SQL Server database based on our harmonized HTS data model. The data extraction can be carried out in bulk e.g. a complete reload; or it can be done incrementally e.g. query for updated data (if supported by source system) or data in a specified area e.g. LGA (again, if supported).

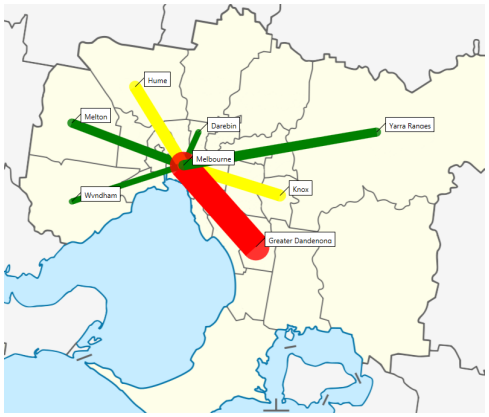
We tested each data extraction program extensively to ensure consistent, harmonized data was resulting. This was a partly manual process. We had to select parts of the harmonized dataset using SQL Server queries. Then compare query results to source datasets to check for correct data extraction, data format transformation and entity/attribute mappings. During



(a) Bubble map showing total trips for a selection of Melbourne suburbs.



(b) Distribution of primary mode of transport, for a selection of Melbourne suburbs.



(c) Total trips using public transport, vehicles and walking.

Figure 5: Example of Harmonizer+ visualizations.

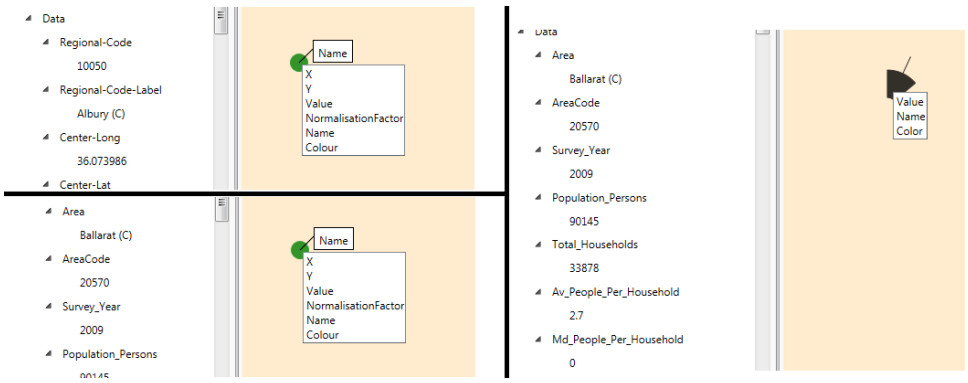
the conversion of the final data, we also did some domain-knowledge and anomaly-detection based checks on the results. For example, whether the data fits into a common-sense range, whether particular data stands out from the rest. We built some JUnit tests to semi-automate these checks so that if the data mappings and thus extractor programs are modified, many consistency checks can be repeated automatically.

4.4. Data Querying and visualization

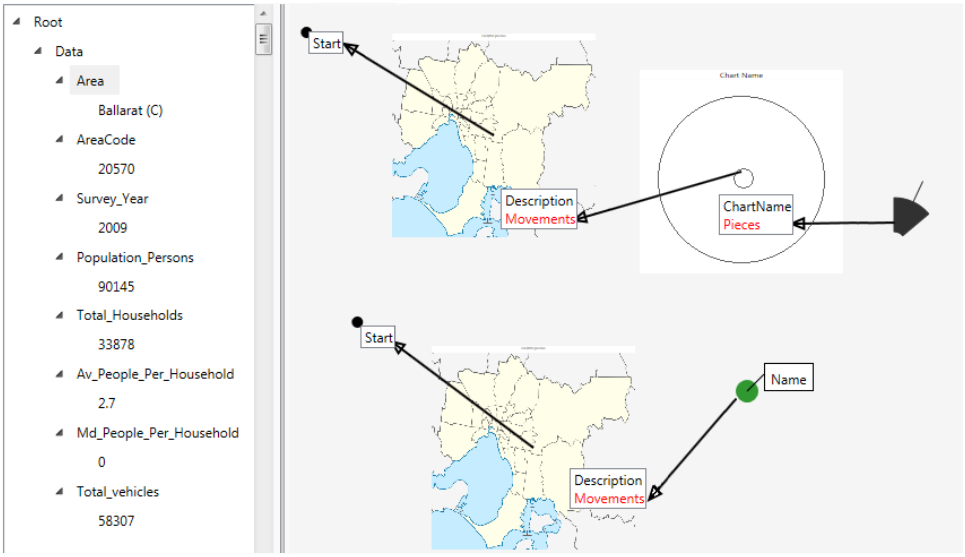
The harmonized data available in Harmonizer+ is not that useful unless powerful and easy-to-use information visualizations are provided to end users. The current AURIN framework provides a set of default visualizations including geographic highlighting, some basic charts, and heat maps. However, the extent to which data can be explored very much depends on how many dimensions of the data can be visualized. AURIN also provides facilities for querying and exporting data. Users can select range of attributes to be included using provided GUI. A query will be generated and executed on available data and its results can be exported. This way, once the harmonized data is available in the framework, harmonized HTS data can be queried and combined with existing AURIN data e.g. household and LGA demographic and income data. However, we found the existing AURIN visualization tools very limited when trying to incorporate our harmonized HTS data. For example, while we could show total trips per suburb (or LGA) using a heat map or bubble map (see for example Figure 5(a)), it does not give any information regarding what is the mode of transport used for those trips.

Accordingly, we used an existing visualisation tool, CONVErT [17], to design set of new, more powerful and expressive visualizations for HTS and associated datasets retrieved from AURIN. CONVErT provides a by-example approach to visualization, i.e. users can import their data samples to the framework, and use available (or design new) visual notations in CONVErT and map their data using a by-example drag and drop approach. The toolset allows different notations to be composed to form complex visualizations. Returning to the mode of transport example above, we can represent a selection of transport modes by a pie chart visualization with modes as pie pieces proportional to total trips (see Figure 5(b)). Users can then compose visualization of pie charts instead of just simplistic bubbles. This way, the radius of the pie chart still represents the total trips, but its internal pie pieces will represent proportions of individual trip mode categories.

Figure 6 demonstrates the required steps for an end user for generating such a visualization. Users map their data to provided visual notations (Figure 6(a)), compose the defined notations (Figure 6(b)), and the framework will generate the required codes to generate the visualizations similar to Figures 5. These visualizations can be exported as web-enabled visualization (XAML or SVG) or as PNG images.



(a) Mapping data values to visual notations.



(b) Designing a visualisation by composing visual notations.

Figure 6: Designing a visualization in the CONVErT framework.

5. Architecture and Implementation

The architecture of Harmonizer+ solution is an ensemble of multiple components as illustrated in Figure 7. We had multiple Household Transport Survey datasets provided to us (Figure 7 (1)). Since the agreed format of use in the approach was CSV, all provided data were converted to CSV before inclusion in the approach. Depending on the level of aggregation of the provided datasets, they were sent directly to our data mapping module or through a data aggregator. Where the provided data was aggregated, the

resulting aggregated dataset was then fed to the data mapping module. Our data aggregator module (Figure 7 (2)) imports the raw state HTS data into a temporary Microsoft SQL server database, applies set of data dependent SQL queries to aggregate the data, and then exports the data as CSV files.

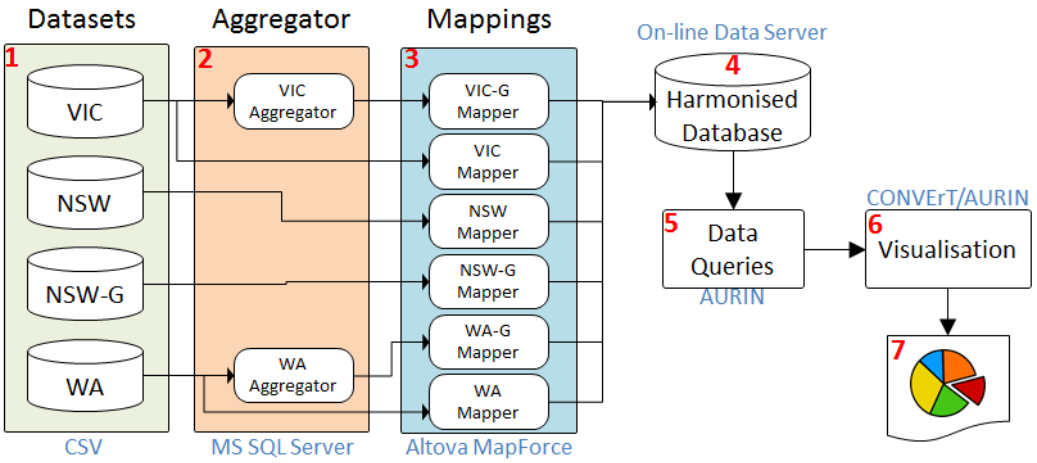


Figure 7: Data transfer and mapping procedure. Arrows indicate data flow.

Once the input data are all at same aggregation level, they are sent to our data mapping module (Figure 7 (3)). Here the set of data mappings were used to map the imported data to the harmonized data model. This module was implemented using Altova MapForce. The initial dataset specific mappings between imported datasets and intermediate schema were implemented in MapForce. The automatically generated mapping code in Java was augmented by hand to implement particularly complex transformations, and then registered as Data mapping module. The results of these data mappings are exported into the harmonized HTS database available on-line within the AURIN framework Figure 7 (4).

The AURIN framework provides facilities to query data according to its available attributes (Figure 7 (5)). Users can design and save these queries and execute them when required. The results of these queries can be used inside AURIN as spreadsheets or by set of predefined visualizations provided by the AURIN framework. Alternatively, then can be exported to CONVErT for visualization generation (Figure 7 (6)). New surveys are periodically conducted and when available their data is fed into the system. New data can be batch imported or incrementally added, depending on the capabilities to obtain it from the particular source state system.

6. Discussion and Lessons Learned

This section discusses strengths and weaknesses of our approach and user feedback on the AURIN HTS solution that we developed. We then list the lessons learned from this project and provide some pointers for future research in data harmonization approaches for complex software systems.

6.1. Strengths and weaknesses of our approach

Considering the requirements laid out in Section 2, our Harmonizer+ has met these requirements for our case study application. We have developed a forward-looking, harmonized data model able to incorporate all important aspects of the disparate current State HTS survey data. This has served as the source of a single, aggregated HTS dataset that has been incorporated into the AURIN portal. AURIN queries can be run across this integrated dataset combining with other AURIN datasets. Our CONVErT-implemented visualizations provide user-friendly, extensible visualization capabilities. However, the framework can be complex at times for generating complex visualizations.

Feedback from AURIN researchers was highly positive. All key HTS data from each state is accessible in a single, integrated form. It can be queried and visualized in highly useful and effective ways. New state data - including Queensland and Western Australia - is expected to be integrated with the existing harmonized data schema. Improved visualization using CONVErT-style by-example specification is attractive to AURIN end users.

It is challenging, even with domain experts available to the data integration team, to understand complex source system data, schema, access support (querying), and underlying technologies, especially for legacy systems. We spent a large amount of time in the AURIN HTS project trying to understand source data meaning, especially some of the complex classifications and attribute values we had to harmonize. A lack of tools to support this in the data wrangling led us to develop some support in our Harmonizer+ platform to explore source datasets and source system querying, export and API interfaces.

In terms of the completeness of the harmonized HTS data model, we had to make a trade-off between having too many missing or questionably-mapped values in a union-of-all-fields style and omitting some important data. For the most part, we were able to achieve an acceptable balance between these for AURIN end users. However, all of the individual state data is still available in its original form and original (dis-)aggression level

if really needed i.e. via its source data portal interface. Expert users may have access to unharmonized data and the detailed mapping functions so we opted for removing some information to enable a wider-level of users, including ultimately citizens and journalists, to better understand the data and combine and query HTS data with other AURIN data.

In general, developers of integration systems will face similar data harmonization problems. Particularly challenging issues include handling source data that uses very different identifiers, very different classifications and taxonomies, different grouping and repeating structures for records, and different aggregation levels. Data can not usually be disaggregated, meaning some systems may provide data at vastly different levels of aggregation, limiting the complete set of harmonized and integrated data. Different schema designs may handle repeating records i.e. multiple instances of records or sub-records. These can be very challenging to map into a single harmonized format.

There were interesting and important privacy concerns that arose during our work. States need to ensure dis-aggregated or small locale area data does not compromise citizen privacy, and different states use different concepts of privacy. This presents a challenge when trying to harmonize the disparate source data. Removing some fields or aggregating data to highly levels to preserve privacy has an impact on research using the harmonized dataset: removing columns (attributes) may make less research possible for AURIN end users. On the other hand, removing some rows due to privacy concerns may significantly skew research results. This issue has also been noted in recent research relating to releasing MOOC data [18]. In general, data provenance, privacy and security issues are extremely important to end users, organisations and often have legislative constraint. We wanted to integrate support for data provenance and privacy management into our Harmonizer+ toolset.

6.2. Summary of lessons learned

In this section we provide a list of lessons we learned from our harmonization project and hope to draw set of future research directions in similar data harmonization cases.

Documentation: A large part of our time in the AURIN HTS project was spent on reading and understanding data documents. Where these documents were not provided, we had to reverse engineer or generate them by investigating the datasets. Often these investigations forced us to conduct

multiple question sessions with data providers. Additionally, when documents are not specifically designed for software engineers and data experts, it is very hard to understand them from the technical point of view. In one example, we were provided with a set of user manuals and data collection procedures rather than data documentation and we had to relate the provided dataset to the manuals. This proved to be a big challenge in understanding and integration of data. Additionally, once the data mappings and harmonization process was finished, we had no acceptable and agreed method of documenting our data mappings. Given the importance of understanding data mappings in similar projects, standard data mapping documentation must be available for future maintenance.

Tool support: Most of the tools we tried had very specific and limited functionalities in comparison to the full life-cycle of our data harmonization project. Most visualization tools, for example, assume data is clean and data wrangling tools mostly do not provide flexible visualizations. It is necessity to have easy and accessible to use harmonization tools. Learning the available tools to perform data aggregation and data wrangling proved a very long learning curve. As a result, our decision was to use our available expertise, and invest more time on understanding the data. Research in more user-centric approaches for performing both tasks will help the data analyst community and other harmonization projects.

Raw data: When it come to the notion of *raw data*, different stakeholders have their own interpretations. For example, we had data provided to us in form of text (e.g. csv), processed statistical files (e.g. SPSS), or as exported databases (e.g. Access DBs). Our decision was to use the lowest level of the data, i.e. text files (csv). While transforming to lowest level is most of the times possible, it might be beneficial to use the data in higher levels specially when dealing with large databases. When collecting information, it is essential for organizations to consider as fine-grained data as possible. It is very hard to disaggregate information if not impossible. When access to fine-grained data is provided, aggregations can be generated according to the problem at hand.

Use of Models: Many areas of software engineering are benefiting from the use of model based approaches, e.g. data transformers and visualization. This can provide better testing facilities, less need for implementation in low level coding, better scalability and validation, to name a few. We hope to see more use of model-based approaches defined as round-tripping processes. This could benefit documentation i.e. use models to document the process

(e.g. data aggregation and mappings) and generate part of the final code automatically. In our example, use of automatic code generation facilities of Altova MapForce greatly improved productivity.

In our example, the automatically generated mapping code we used to transform source HTS data from states into our integrated Harmonizer+ repository proved to be highly effective. The use of Altova MapForce greatly enhanced our ability to specify complex data mappings predominantly declaratively and generate highly efficient Java programs to carry out the data transformation and integration. Maintenance effort is relatively low for these mappings and the generated mapping code as we are able to regenerate by far the majority of the translator components from MapForce.

Privacy: We have identified an interesting new research area of dynamically integrating privacy policy (for specifying which rows/columns or operations are not allowed and for what usage situation) with data access, query and analytics support. Any data filtering or removal (especially at the row level) should be communicated clearly to inform researchers using the harmonized data of the possible impact on their research results (such as correlation studies).

Visualizations: Our observations revealed that finding inconsistencies within datasets is a crucial step in data wrangling and cleansing. With large variety of datasets, it is very hard to track inconsistencies. As a result, we chose to use visualizations to help us track these inconsistencies. The visualizations of Figure 2 are samples of these visualizations using Gant chart metaphor. More research in developing such visualizations is required in conjunction with research on clustering approaches.

Future applications: We are applying the approach presented in this paper on healthcare system integration problems. Our health provider partners have diverse systems where complex patient, diagnostic, treatment and monitoring data are stored in varied formats. These need to be brought together, harmonized and visualised in much the same way as the travel survey data case study in this paper. We also plan to apply these techniques to integrating complex traffic data sourced from several systems with our roading partners, and smart home and building sensor data.

7. Related Work

Various approaches and tools have been developed and used to perform complex data mappings for data integration and mapping scenarios. For ex-

ample, a form-based mapper was introduced to help business analyst users perform data mapping using a concrete form-based metaphor [5]; Transformations to support data integration within multiple views of source and target models [19, 20]; Mapping agents to generate automated mappings between multiple source and targets [7]; And support for mapping and transformation generation using concrete visualizations [21]. However, all of these frameworks were targeted at specific domain data mapping problems, none fulfilling examples like our AURIN HTS data harmonization needs.

Clio is an early attempt by IBM to provide data transformation and mapping generator for information integration applications [6]. Clio provided declarative mappings to be specified between source and target schemas and supported mapping generation in XQuery, XSLT, SQL, and SQL/XML queries. In our project, we were provided with raw text-base data. As a result approaches that use abstractions or schema mappings were not applicable. Although we could have reverse engineered the schemas, but that would introduce multiple problems for example incompleteness of the reverse engineered schema.

Multiple approaches exists for data wrangling and cleansing with text-based datasets including Toped++ [22], Potluck [23], Karma [24], and Vegimite [25]. These approaches however do not provide all necessary mapping facilities (e.g. reshaping data layout, aggregation, and missing value manipulation) and only support a subset of the needed transformations [13]. Scripting languages alike Ajax and Potter's Wheel provide data mapping and manipulation facilities [26, 27]. These however are restricted in few set of commands and introduce difficulties in programming directly with scripting languages that is not feasible for our intended end users. More recently, new frameworks have emerged to fill the gap for data integration. Examples include Talend studio[1] (specifically Talend Open Studio for Data Integration), Altova[2] (specifically MapFroce), and Informatica[3]. These frameworks provide facilities for data integration that reduce the need to engage in low level coding of various data loaders, transformations and preparation procedures. While the end results of most such frameworks for candidate projects may be the same, working with each framework, users would be exposed to different routines and data integration life cycles. Hence, we are embarking on developing a more uniform software engineering process to cater for most data integration and harmonization projects.

Given the extend to which the data processing is being used across different domains, data wrangling community is moving towards using more

modern tools that demand less technical knowhow to perform various steps of data cleaning and integration. An example of approaches that are targetted to less technical users is Trifacta Wrangler [28]. Wrangler provides a framework where users interactively manipulate data and the system inferres the relevant data transformations [29]. It provides natural language description of data mappings intended for less technical users. In our project however, we were interested in batch processing of the transformations and a complete wrangling life cycle from data cleaning to visualizations, hence we chose to use

Usability of our approach was also dependant on generating understandable data visualizations. AURIN framework by default provides a set of predefined visualizations for unsers to query and see sample data through standard visualisations like barchart, and heat maps. However, additional visualizations cannot be defined inside the framework. We investigated integration of other visualization tools (e.g. Protovis [30], Lyra [31]). We found these tools very sensitive to uncleansed data (for example missing data fields) and could not be well integrated in the framework. Similarly, powerful visualization scripting like D3 [32], and Vega [33] require scripting knowledge and were not suitable for use in our approach.

8. Conclusions

We have described a process and supporting framework for engineering complex data integration, harmonization and visualization systems. Our approach focuses on data understanding, development of a harmonized schema, design and generation of integration mappings and mapping support, and canonical, harmonized dataset querying for end user-oriented visualizations. We have presented an industry-based project using our harmonization approach of integrating multiple household travel surveys into an intermediate canonical database. It incorporates complex multi-source data aggregation, data mapping and transformation, and information visualization. Our approach is practical for industrial usage in such domains. We have learned a number of important lessons from this experience and have identified set of key directions for future research to better-support such challenges. We are

[1] www.talend.com
[2] www.altova.com
[3] www.informatica.com

applying this approach to healthcare, traffic and smart home data integration, harmonization and visualisation.

Acknowledgments

The authors would like to acknowledge the support form AURIN and Data61/CSIRO for this research.

References

[1] Y. Zheng, Methodologies for cross-domain data fusion: An overview, IEEE Transactions on Big Data 1 (1) (2015) 16–34.

[2] R. Lämmel, E. Meijer, Mappings make data processing go 'round, in: International Conference on Generative and Transformational Techniques in Software Engineering, GTTSE'05, Springer-Verlag, 2006, pp. 169–218.

[3] J. Grundy, J. Hosking, R. Amor, W. Mugridge, Y. Li, Domain-specific visual languages for specifying and generating data mapping systems, Journal of Visual Languages and Computing 15 (34) (2004) 243 – 263.

[4] S. Kandel, A. Paepcke, J. Hellerstein, J. Heer, Wrangler: Interactive visual specification of data transformation scripts, in: ACM Conference on Human Factors in Computing Systems, CHI '11, ACM, 2011, pp. 3363–3372.

[5] Y. Li, J. Grundy, R. Amor, J. Hosking, A data mapping specification environment using a concrete business form-based metaphor, in: IEEE Symposia on Human Centric Computing Languages and Environments, 2002, pp. 158–166.

[6] R. Fagin, L. M. Haas, M. Hernández, R. J. Miller, L. Popa, Y. Velegrakis, Conceptual modeling: Foundations and applications, Springer-Verlag, 2009, Ch. Clio: Schema Mapping Creation and Data Exchange, pp. 198–236.

[7] S. Bossung, H. Stoeckle, J. Grundy, R. Amor, J. Hosking, Automated data mapping specification via schema heuristics and user interaction, in: 19th International Conference on Automated Software Engineering, 2004, pp. 208–217.

[8] G.-D. Sun, Y.-C. Wu, R.-H. Liang, S.-X. Liu, A survey of visual analytics techniques and applications: State-of-the-art research and future challenges, Journal of Computer Science and Technology 28 (5) (2013) 852–867.

[9] J. P. Daries, J. Reich, J. Waldo, E. M. Young, J. Whittinghill, D. T. Seaton, A. D. Ho, I. Chuang, Privacy, anonymity, and big data in the social sciences, Queue 12 (7) (2014) 30:30–30:41.

[10] M. Koehler, A. Bogatu, C. Civili, N. Konstantinou, E. Abel, A. A. Fernandes, J. Keane, L. Libkin, N. W. Paton, Data context informed data wrangling, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 956–963.

[11] I. Avazpour, Towards user-centric concrete model transformation, Ph.D. thesis, Swinburne University of Technology (2014).

[12] R. Sinnott, G. Galang, M. Tomko, R. Stimson, Towards an e-infrastructure for urban research across australia, in: 7th IEEE International Conference on E-Science, 2011, pp. 295–302.

[13] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, P. Buono, Research directions in data wrangling: Visuatizations and transformations for usable and credible data, Information Visualization 10 (4) (2011) 271–288.

[14] A. M. Cabrera, C. J. Faber, K. Cepeda, R. Derber, C. Epstein, J. Zheng, R. K. Cytron, R. D. Chamberlain, Dibs: A data integration benchmark suite, in: Companion of the 2018 ACM/SPEC International Conference on Performance Engineering, ACM, 2018, pp. 25–28.

[15] M. Janssen, E. Estevez, T. Janowski, Interoperability in big, open, and linked data–organizational maturity, capabilities, and data portfolios, Computer 47 (10) (2014) 44–49.

[16] I. Avazpour, J. Grundy, L. Grunske, Specifying model transformations by direct manipulation using concrete visual notations and interactive recommendations, Journal of Visual Languages & Computing 28 (0) (2015) 195 – 211.

[17] I. Avazpour, J. Grundy, CONVErT: A framework for complex model visualisation and transformation, in: IEEE Symposium on Visual Languages and Human-Centric Computing, 2012, pp. 237–238.

[18] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, D. E. Pritchard, Who does what in a massive open online course?, Communications of the ACM 57 (4) (2014) 58–65.

[19] H. Stoeckle, J. Grundy, J. Hosking, A framework for visual notation exchange, Journal of Visual Languages and Computing 16 (3) (2005) 187–212.

[20] H. Stoeckle, J. Grundy, J. Hosking, Approaches to supporting software visual notation exchange, in: IEEE Symposia on Human Centric Computing Languages and Environments, 2003, pp. 59–66.

[21] J. Grundy, R. Mugridge, J. Hosking, P. Kendall, Generating edi message translations from visual specifications, in: Proceedings of 16th Annual International Conference on Automated Software Engineering, 2001. (ASE 2001), 2001, pp. 35–42.

[22] C. Scaffidi, B. Myers, M. Shaw, Intelligently creating and recommending reusable reformatting rules, in: 14th International Conference on Intelligent User Interfaces, IUI '09, ACM, 2009, pp. 297–306.

[23] D. Huynh, R. Miller, D. Karger, Potluck: Semi-ontology alignment for casual users, in: The Semantic Web, Vol. 4825 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 903–910.

[24] R. Tuchinda, P. Szekely, C. A. Knoblock, Building mashups by example, in: 13th International Conference on Intelligent User Interfaces, IUI '08, ACM, 2008, pp. 139–148.

[25] J. Lin, J. Wong, J. Nichols, A. Cypher, T. A. Lau, End-user programming of mashups with vegemite, in: 14th International Conference on Intelligent User Interfaces, IUI '09, ACM, 2009, pp. 97–106.

[26] H. Galhardas, D. Florescu, D. Shasha, E. Simon, Ajax: An extensible data cleaning tool, SIGMOD Rec. 29 (2) (2000) 590–.

[27] V. Raman, J. M. Hellerstein, Potter's wheel: An interactive data cleaning system, in: 27th International Conference on Very Large Data Bases, VLDB '01, Morgan Kaufmann, 2001, pp. 381–390.

[28] T. W. Enterprise, Trifacta wrangler (2015) (2016).

[29] J. M. Hellerstein, J. Heer, S. Kandel, Self-service data preparation: Research to practice., IEEE Data Eng. Bull. 41 (2) (2018) 23–34.

[30] M. Bostock, J. Heer, Protovis: A graphical toolkit for visualization, IEEE Transactions on Visualization and Computer Graphics 15 (6) (2009) 1121–1128.

[31] A. Satyanarayan, J. Heer, Lyra: An interactive visualization design environment, in: Computer Graphics Forum, Vol. 33, Wiley Online Library, 2014, pp. 351–360.

[32] M. Bostock, V. Ogievetsky, J. Heer, D3: Data-driven documents, IEEE Transactions on Visualization and Computer Graphics 17 (12) (2011) 2301–2309.

[33] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, J. Heer, Vega-lite: A grammar of interactive graphics, IEEE Transactions on Visualization and Computer Graphics 23 (1) (2017) 341–350.