V for Variety: Lessons Learned from Complex Smart Cities Data Harmonization and Integration

Iman Avazpour*, John Grundy* and Liming Zhu†‡
*Software Innovation Laboratory, Faculty of Science, Engineering and Technology
Swinburne University of Technology, Hawthorn 3122, VIC, Australia
Email: {iavazpour, jgrundy}@swin.edu.au
†Software and Computational Systems, Data61, CSIRO
‡School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
Email: Liming.Zhu@data61.csiro.au

***Abstract*—With emerging trends for Internet of Things (IoT) and Smart Cities, complex data transformation, aggregation and visualization problems are becoming increasingly common. These tasks support improved business intelligence, analytics and end-user access to data. However, in most cases developers of these tasks are presented with challenging problems including noisy data, diverse data formats, data modeling and increasing demand for sophisticated visualization support. This paper describes our experiences with just such problems in the context of Household Travel Surveys data integration and harmonization. We describe a common approach for addressing these harmonizations. We then discuss a set of lessons that we have learned from our experience that we hope will be useful for others embarking on similar problems. We also identify several key directions and needs for future research and practical support in this area.**

I. Introduction

One of the most common problems in computing is the need to integrate multiple sources of information presented in disparate data formats [1], [2]. Such integration would allow leveraging the combined information i.e. to “harmonize” the disparate data into a single, consistent form. This data integration is indeed a key point in success of smart cities' applications. On the other hand, when integrating data sources with a diverse set of federated owners, changing them can be impossible due to ownership and legal issues in government data sets.

This paper discusses our experience in addressing common data integration, aggregation and harmonization tasks. It focuses on addressing *Variety* among the four Vs of big data (Volume, Velocity, Veracity, and Variety). We provide a case study of Household Travel Survey (HTS) harmonization. Using this case study, we draw a set of lessons learned during implementation of this and similar applications. We hope the lessons help novice data analytics developers in their harmonization task and to better incorporate resources. We also identify areas that have the potential and could benefit from further research.

This paper is organized as follows: We start with background from the HTS harmonization case study in section II. Section III briefly outlines key related work. We describe an approach to perform harmonization in section IV and describe how this approach was applied on HTS data harmonization. Section V provides a discussion, summary of strengths and weaknesses of the approach, key lessons learned, and key areas for future research.

II. Background

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 [3]. 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. Commonwealth, State, Local organizational structures, businesses, and hospitals) among others.

Household Travel Surveys (HTS) as an example dataset 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. The AURIN project wanted to integrate HTS data seamlessly into the wider project resources through a single harmonized data model. 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.

III. Related work

While data management approaches like Hadoop, Spark, Pig and Hive have been recently center of attention by the community, they are not particularly designed with focus on data integration. Rather, they provide platforms for processing, accessing and data from multiple sources. In contrast, various research and industrial applications have been working on addressing data mapping and aggregation solutions in order to make transitioning from one data format to another less expensive and more user-friendly (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].

We have previously developed and used complex data mappers for various data integration and mapping scenarios. For example, 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 [10], [11]; Mapping agents to generate automated mappings between multiple source and targets [7]; And support for mapping and transformation generation using concrete visualizations [12], [13]. Each of these frameworks was targeted to address a specific domain, audience or data mapping problem.

A more generic approach to handling data transformations is Clio. Clio provided data transformation and mapping generator for information integration applications [6]. It provided declarative mappings to be specified between source and target schemas and supported mapping generation in XQuery, XSLT, SQL, and SQL/XML queries. Use of schemas however, limited the use of raw text-base data which is often provided in data integration problems, as the data schema are not generally provided by the data custodians.

Multiple approaches exist for data wrangling and cleansing with text-based datasets including Toped++ [14], Potluck [15], Karma [16], and Vegimite [17]. 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 for the fully fledged data integration system [18].

More recently, Wrangler was designed as a framework where users interactively manipulate data and the system inferred the relevant data transformations [4]. It provided natural language descriptions of data mappings intended for less technical users. Additionally, it utilized a programming by demonstration approach were the actions taken by users were translated to data processing queries to be applied on the data. Wrangler since has been merged with Trifacta[1] as a framework for data processing and manipulation. At the time of writing this manuscript, a free beta version of Trifacta Wrangler is set to be launched and made available to public audience. Tamr[2] is another application that seeks to address the problem of data identification. It is based on the idea that in data intensive environments, it is often the case where various data gets collected in storages (called data lakes). These data lakes can get massive, resulting in analysts overlooking certain available characteristics or data points. Tamr uses machine learning to provide a system of reference (rather than records) as a centralized data catalog. This catalog can then be used by data analysts familiar with a certain data structure or format to make them more productive in their analysis.

IV. Approach

The life cycle for performing many data integration tasks fallows a set of common steps. We call this life cycle Common Open Data hArmonization approach or CODA for short. A CODA approach generally has 5 steps.

1) **Understand your data**: Read data documentations. List data file types and formats and determine aggregation levels. Understand specific data types used in the datasets for example categorical and nominal values.
2) **Design target data models**: A canonical place for all data formats to be harmonized.
3) **Implement data importers**: Data may be available online, off-line, on local machines, on dedicated servers, on paper, or storage disks among others. It may also be recorded using varying tools, databases, or storage formats.
4) **Implement data transformers**: Implement data transformers and mappers. Extract Transform Load (ETL) techniques may be used at this stage.
5) **Use the data**: This is the stage where visualizations are generally used to understand, explore and search for more details.

These steps are reflected on Figure 1 and are categorized into three different tasks: Data familiarity, Wrangling, and Usage. To better see these steps, we demonstrate them by our industry case study of HTS harmonization.

A. Case Study: Household Travel Survey Harmonization

In this section, we describe how we used the CODA approach to solve data harmonization and integration for the HTS datasets of section II. To perform this harmonization project, we started by investigating provided data documentation to identify nature of the available disparate data, see if there are any commonalities in the data e.g. locale and demographics, what are the collection methods and how they differ, find any missing data fields, examine aggregation levels, and investigate if there is any difference in the terminology used for each dataset (step 1 in Figure 1). This step contributed to the bulk of our data preparation for the cleansing task, i.e. identify and remove any inconsistencies. It was an important step as success and failure of data integration frameworks is dependent on understanding the data's context-sensitive meaning and the quality of the data [21]. This cleansing process can sometimes take up to 80 percent of the work [22].

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

[1] www.trifacta.com

[2] www.tamr.com

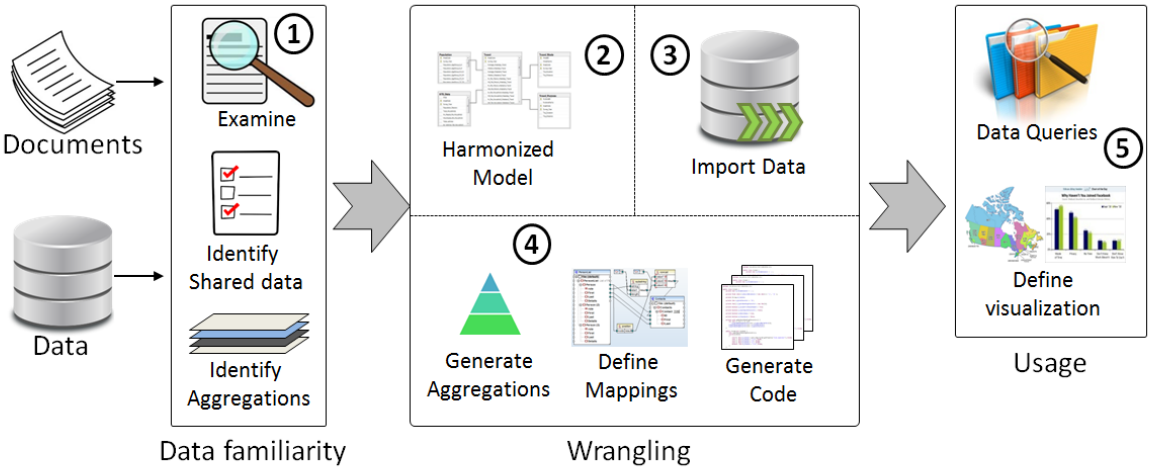


Fig. 1. Outline of the CODA approach.

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, some data samples also came with different high level data structure. For example one sample used a relational database with various tables as a Microsoft Access database. While another sample was as one CSV file that could be represented as a table in a relational 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 etc. 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 in some datasets could be recorded as nominal values (i.e. numbers represent modes, e.g 2 = vehicle, 4 = public transport), or 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 traveled by vehicle, and another for 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 (light rail) as a means for public transport so it was not included in the list of travel modes.

Figure 2 shows an example highlighting inconsistencies in the categories used for *travel purpose* (e.g. work, school, leisure, health treatment). 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. In some cases we had data at multiple levels of aggregation. For example state of New South Wales had provided a sample aggregated data and a sample of their raw data (hence NSW and NSW-G). The bottom row lists the categories we used for our final harmonized data model.

Figure 2 represents available categories by boxes. For example, the dataset provided by state of Victoria includes a travel purpose as 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 empty. For example **NSW-G** does not provide information for *Accompany Someone*. Figure 2 also uses 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. Additionally, 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-G 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 and design the required data mappings.

We then designed a target model for our integration. This target model (here we call it harmonized model) would play the role of a canonical data model that all other datasets can be mapped to (step 2 in Figure 1). For the design of this harmonized 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

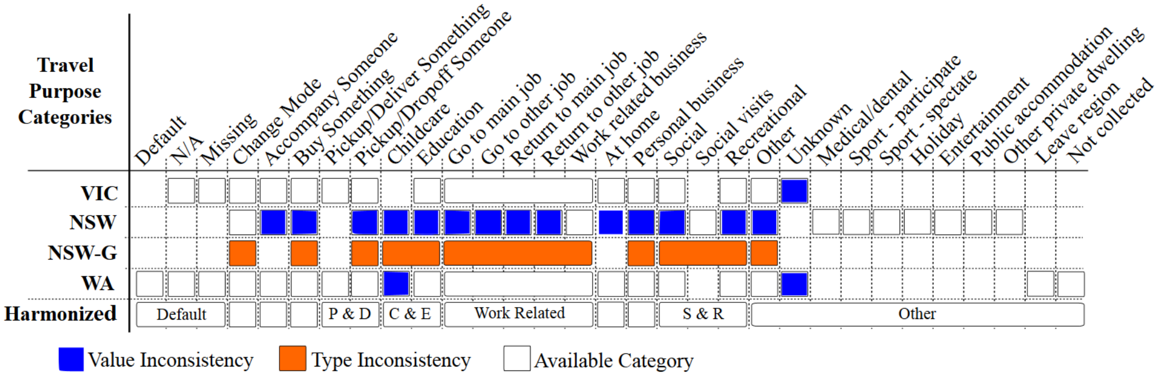


Fig. 2. Sample of inconsistencies encountered within categories used for *Travel Purpose*.

combination of both. Accepted majority using indicates the data that is available in most provided datasets and discarding the rest. This would have helped us 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 cannot tolerate missing data. We chose the third approach which is a combination of both, i.e. we chose which fields to merge, which fields to keep, and which fields to discard. For example in Figure 2, six travel purpose categories of the NSW dataset have been merged to “Other”, as the rest of the dataset did not provide a corresponding purpose category.

Given that data sets in our case were provided from different technologies and formats, we needed to develop data importers to import various sources into our harmonization framework (step 3 in Figure 1). The selection of suitable technology for importing various data sources depends very much on the available skill sets of the integration team and the data mapping and transformation technology to be used. In our case the technology used for data mapping and transformation development (Altova Mapforce[3]) provided facilities for importing range of different data sources. It provided the necessary data connectors to connect to various data sources that eliminates separate coding of the connectors.

Once the data is imported, we moved to develop data processing and integration. This step would include aggregating and disaggregating (if possible) datasets at different levels of abstraction, defining data mapping from various sources to her harmonized model, and generating the mapping transformation code (step 4 in Figure 1).

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.

Our next step was to define the data mappings. These mappings would insert 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. MapForce automatically generates schemas for imported data and allows viewing source and target schemas side by side. Mapping correspondences can then be defined by drag and dropping elements of source and target schemas. From these MapForce specifications a set of Java programs are automatically generated that extract data from each dataset and import it into a single, integrated SQL Server database based on our harmonized data model.

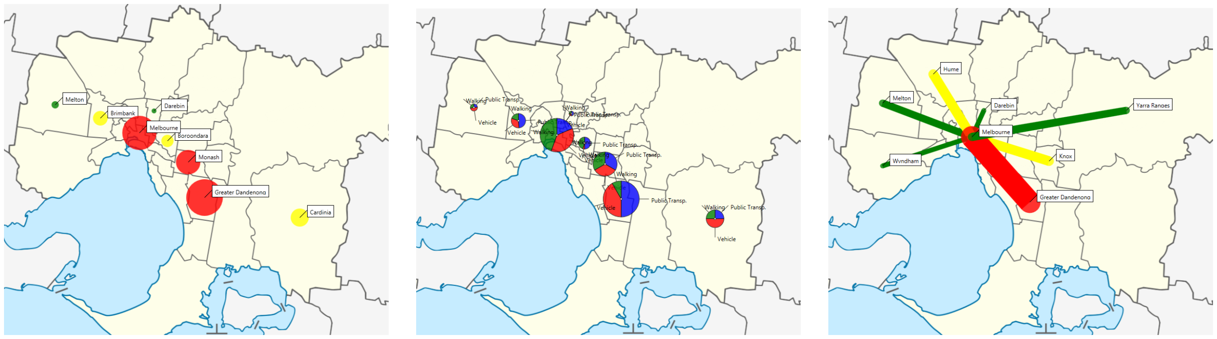
The output of harmonization is set of polished and ready to be used data. Usage could be in form of data queries, or visualization (step 5 in Figure 1). The current AURIN framework provides a set of default visualizations including geographic highlighting, some basic charts, and heat maps. However, the extent to which data can be explored very much depends on how many dimensions of the data can be visualized. Accordingly, we used our CONcrete Visual assistEd Transformation (CONVErT) framework [23], to design set of new, more powerful and expressive visualizations for HTS and associated datasets retrieved from AURIN. CONVErT allows different notations to be composed to form complex visualizations. Examples of such visualizations are depicted on Figure 3.

AURIN also provides facilities for querying and exporting data. Users can select range of attributes to be included using provided GUI. This way, the harmonized data can be queried and combined with existing AURIN data e.g. household, demographic and income data. Detailed description of this HTS project and the implementation can be found in the project technical report [24].

V. Discussion and Lessons Learned

This section discusses strengths and weaknesses of the CODA approach for data harmonization and integration. We

[3] www.altova.com



(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.

Fig. 3. Example of CONVErT generated visualizations.

then list key lessons learned from this project and provide some directives for future research.

The approach has satisfied the requirements laid out by the project i.e. we have developed a forward-looking, harmonized data model able to incorporate all important aspects of (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 mixed approach for the harmonized HTS data model has had some consequences. 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. However, all of the individual state data is still available in its original form and original (dis-)aggression level if really needed. 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.

There are interesting and important privacy concerns that arose during the project. 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.

In the following we list key 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 this 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 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. 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 the text format (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, the automatically generated mapping code we used to transform source HTS data from states into our integrated 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 policies (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.

VI. Summary and Conclusion

We have described an industry-based project on harmonization of multiple household travel surveys into an intermediate canonical database. It incorporates complex multi-source data aggregation, data mapping and transformation, and information visualization. The describe common 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 key directions for future research to better-support such challenges.

Acknowledgments

We acknowledge the support form National Information Communication Technology Australia (NICTA). NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Center of Excellence Program.

References

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

[2] J. Grundy, J. Hosking, R. Amor, W. Mugridge, and Y. Li, "Domain-specific visual languages for specifying and generating data mapping systems," *Journal of Visual Languages and Computing*, vol. 15, no. 34, pp. 243 – 263, 2004.

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

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

[5] Y. Li, J. Grundy, R. Amor, and 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, and Y. Velegrakis, "Conceptual modeling: Foundations and applications," A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu, Eds. Springer-Verlag, 2009, ch. Clio: Schema Mapping Creation and Data Exchange, pp. 198–236.

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

[8] G.-D. Sun, Y.-C. Wu, R.-H. Liang, and 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*, vol. 28, no. 5, pp. 852–867, 2013.

[9] J. P. Daries, J. Reich, J. Waldo, E. M. Young, J. Whittinghill, D. T. Seaton, A. D. Ho, and I. Chuang, "Privacy, anonymity, and big data in the social sciences," *Queue*, vol. 12, no. 7, pp. 30:30–30:41, Jul. 2014.

[10] H. Stoeckle, J. Grundy, and J. Hosking, "A framework for visual notation exchange," *Journal of Visual Languages and Computing*, vol. 16, no. 3, pp. 187–212, Jun. 2005.

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

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

[13] I. Avazpour, J. Grundy, and L. Grunske, "Specifying model transformations by direct manipulation using concrete visual notations and interactive recommendations," *Journal of Visual Languages & Computing*, vol. 28, no. 0, pp. 195 – 211, 2015.

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

[15] D. Huynh, R. Miller, and D. Karger, "Potluck: Semi-ontology alignment for casual users," in *The Semantic Web*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4825, pp. 903–910.

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

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

[18] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono, "Research directions in data wrangling: Visualizations and transformations for usable and credible data," *Information Visualization*, vol. 10, no. 4, pp. 271–288, Oct. 2011.

[19] H. Galhardas, D. Florescu, D. Shasha, and E. Simon, "Ajax: An extensible data cleaning tool," *SIGMOD Rec.*, vol. 29, no. 2, pp. 590–, May 2000.

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

[21] M. Janssen, E. Estevez, and T. Janowski, "Interoperability in big, open, and linked data–organizational maturity, capabilities, and data portfolios," *Computer*, vol. 47, no. 10, pp. 44–49, Oct 2014.

[22] D. Patil, *Data Jujitsu*. " O'Reilly Media, Inc.", 2012.

[23] I. Avazpour and 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.

[24] I. Avazpour, R. Sadauskas, J. Grundy, and L. Zhu, "Australian household travel survey data integration, harmonization and visualization," 2015.