Data Wrangling for Big Data: Towards a Lingua Franca for Data Wrangling

Tim Furche, Georg Gottlob, Bernd Neumayr, and Emanuel Sallinger

University of Oxford

1 Introduction

We are dealing with ever growing amounts of data, or as some like to call it, we are at the beginning of the era of big data. The value gained from the analysis of big data is expected to be huge. Yet data analytics needs well-organized, clean data to work on. The process of transforming raw data from various sources into a form suitable for data analytics is called *data wrangling* [15, 12]. It is estimated that up to 80% of the time spent gaining value from big data is spent on data wrangling as opposed to data analytics itself [20]. Hence, there is a clear need for data wrangling to be more effective.

While the term *data wrangling* is relatively new, transforming data from one format into another has been a focus of the data management community for many years now. There are systems that handle *data extraction* very well [11]. *Data integration* [19] and *data exchange* [9] have been studied in detail. We have gained a deep understanding of *data quality*, as well as *querying* and *reasoning* over this data. All of these components are needed for successful data wrangling.

Yet, with the advent of big data, new challenges have arrived, sometimes characterized by the 4 V's of big data: volume (the scale of data), velocity (speed of change), variety (different forms and formats) and veracity (uncertainty). These pose challenges for each component of data wrangling itself, but also for the whole system.

So far, the challenges that big data poses for data wrangling have mostly been met at the level of individual components (such as data extraction or integration). Yet it is sharing knowledge between these components that has the most potential of improving the data wrangling process, e.g. by allowing the data management system to optimize execution based on such shared knowledge [12]. In this paper, we describe the vision and design principles of a Datalog-based language for data wrangling that facilitates such sharing of knowledge.

Overall, data wrangling today is an area highly demanded by industry, but without a clearly defined research community or clearly defined research objectives. The data management community is a natural fit for taking up this task and shaping this new field. The VADA (Value-Added Data Systems) programme, with the University of Oxford, the University of Manchester and the University of Edinburgh at its core, seeks to significantly advance data wrangling, both in practice by providing an architecture and prototype implementation for data wrangling, and in academic research by solving the challenges of wrangling big data in collaboration with the data management community.

2 A Lingua Franca for Wrangling Big Data

The need to share data and knowledge between components of a data wrangling system makes it clear that a common language is needed to express such knowledge, and enable reasoning over it. In this paper, our goal is to describe the vision of a *lingua franca for data wrangling*. Such a language should provide a uniform way to address the different needs in the data wrangling process:

- expressing knowledge in a shared knowledge-base
- reasoning about data and transformation of data within the components
- specifying the workflow between the components

The language should address these features by providing a uniform view of data (independent of its source), while supporting the components by being suited to data extraction, integration and exchange. At the same time, it must allow for efficient processing and scalability when confronted with big data.

One of the best-established languages in the data management community for knowledge-based reasoning is *Datalog*. Over the years, it has been studied in great detail and extended in various ways [6]. In the area of data exchange [9, 16] and data integration [14, 19], extended Datalog rules called *tuple-generating dependencies* (tgds; sometimes also called *existential rules* [5]) are used to specify *schema mappings* [9]. They have been successfully applied in IBM's Clio system and form the core of products offered by companies such as LogicBlox to empower data analytics. Complementing that, their theoretical properties are well-studied [1], including their management [7, 3], composition [2, 4], optimization [21, 18] and reasoning [22, 17], in particular the computational limits of reasoning [10, 13]. The family of languages often called Datalog$^{\pm}$ [8] seeks to add to Datalog's expressive power, yet not by sacrificing efficiency and scalability.

A key challenge to such a language is large volume of data on the one hand, and requirements for highly expressive reasoning on the other hand. Clearly, meeting both requirements at the same time is hard. Yet there is a full spectrum of possibilities in between:

- small volume of data: complex reasoning
- large volume of data: simple processing
- very large volume of data: parallel processing

The challenge of offering both expressiveness and scalability in a single system poses particular design challenges to a language for data wrangling. A single monolithic – highly expressive – language is not enough to meet the requirement of scalability in the presence of big data.

2.1 Design Principles

In this section, we present our vision for VADALOG, our proposed language for data wrangling. We will do this by following the major features, themes and properties of the language. For space reasons, we will focus on the design principles of the language, and not go into the details of its syntactic representation.

Solid Foundation: Datalog. The language is based on Datalog, extended by features that are well-known in the data management community: in particular existential quantification (as in tgds, existential rules or Datalog$^{\pm}$) as well as numerous other features motivated by the theoretical and practical needs of data wrangling. Having Datalog at its foundation gives VADALOG a well-understood core that has been the topic of research for many years now.

Family of Languages. One single, all-encompassing language cannot meet at the same time the goal of being highly expressive as well as having low computational complexity. VADALOG therefore consists of *profiles* of the language (in the same meaning as the profiles of languages such as OWL), each providing a specific subset suited to a particular purpose. This allows simple computations to be efficiently and scalably executed over big data, while at the same time allowing complex reasoning over a smaller knowledge-base.

Combining Strengths. There exist a number of powerful knowledge-based systems, database systems, and systems that are able to deal with big data that often combine the expertise of large groups of researchers and engineers. The language is thus designed with the intent of making use of systems specifically suited for particular tasks. For example, if a VADALOG programme is formulated in a profile of the language that is particularly suited to an existing knowledge-based system, then this system is used as a backend-engine. This design choice does not come for free – many interesting research challenges have to be addressed to deal with multiple engines working in a unified system.

Handling Volume. Volume may be coped with by using an engine that is suited for handling huge amounts of data. Yet this is not always the most efficient approach. The language contains as "first-class citizens" the support for partitioning the data into *dataspheres* which may be parameterized using domain-dependent or data-dependent parameters. This language-design principle of being able to handle volume by allowing clever partitioning of the data, combined with using engines that can handle big amounts of data when necessary, allows VADALOG to efficiently deal with huge amounts of data.

Modularity. Reasoning and transformation tasks are organized into self-contained modules – based on the concept of a data *transducer* which receives data from different dataspheres and produces data in different dataspheres. Such transducers modularly encapsulate their dependencies (which dataspheres they require to be present) and their guards (what conditions must be met to be executed). Defining all of these parts of the transducer is done using VADALOG in a single, maintainable module for each such transducer.

Dynamic Orchestration. Defining single modules – transducers – is only one part of a data wrangling system. A key part of such a system is that all its components are able to share data and knowledge between them and, importantly, react to such knowledge by dynamically selecting which next steps to take. For example, as a result of quality analysis, the system may choose to redo data extraction with different background knowledge, or adding a data source. A key part of VADALOG is thus a profile for specifying such *transducer networks* that dynamically orchestrate components of the data wrangling system.

Extensibility. A rule-based language based on Datalog is clearly suited for knowledge-based reasoning tasks. Yet, while a data wrangling system has reasoning-intensive tasks, it also has tasks which are better suited to be implemented in other languages. To harness components implemented in other languages, VADALOG allows extensibility at a number of levels: At the transducer level, which gives the components wide-ranging freedom in how to approach its task (such as an existing component analysing data quality); at the level of *actions*, which are middle-scale tasks (such as navigating web pages), and at the level of external *functions*, which can add small-scale functionality not supported in the language (such as non-supported string or number functions).

3 Conclusion

Data wrangling is an important challenge in practice and for research. The data management community is in a good position to take on this challenge, giving this growing field a scientific community and shaping its research objectives. The data management community should take an active role in this area.

Developing a language for data wrangling is an important step that is clearly needed for that purpose. In this paper, we described the design principles of VADALOG, our proposed language for data wrangling. While a number of design principles are related to solving the challenges of data wrangling itself, two important ones promote a collaborative approach to this: It allows to harness the power of existing knowledge-based reasoning systems, and it promotes extensibility by allowing integration of components.

We invite the data management community to collaborate with us both in using the power of currently-developed systems for data wrangling, as well as to shape the future of VADALOG.

Acknowledgements. This work was supported by the EPSRC programme grant EP/M025268/1. Bernd Neumayr receives funding from a habilitation grant of the state of Upper Austria. Emanuel Sallinger was partially supported by the Austrian Science Fund projects (FWF):P25207-N23 and (FWF):Y698.

References

1. Arenas, M., Barceló, P., Libkin, L., Murlak, F.: Foundations of Data Exchange. Cambridge University Press (2014)
2. Arenas, M., Fagin, R., Nash, A.: Composition with target constraints. Logical Methods in Computer Science 7(3) (2011)
3. Arenas, M., Pérez, J., Reutter, J.L., Riveros, C.: Foundations of schema mapping management. In: PODS. pp. 227–238. ACM (2010)
4. Arenas, M., Pérez, J., Reutter, J.L., Riveros, C.: The language of plain so-tgds: Composition, inversion and structural properties. JCSS 79(6), 763–784 (2013)
5. Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: On rules with existential variables: Walking the decidability line. Artif. Intell. 175(9-10), 1620–1654 (2011)
6. Barceló, P., Pichler, R. (eds.): Datalog in Academia and Industry - Second International Workshop, Datalog 2.0, LNCS, vol. 7494. Springer (2012)
7. Bernstein, P.A., Melnik, S.: Model management 2.0: manipulating richer mappings. In: SIGMOD Conference. pp. 1–12. ACM (2007)
8. Calì, A., Gottlob, G., Lukasiewicz, T.: A general datalog-based framework for tractable query answering over ontologies. In: PODS. pp. 77–86. ACM (2009)
9. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theor. Comput. Sci. 336(1), 89–124 (2005)
10. Feinerer, I., Pichler, R., Sallinger, E., Savenkov, V.: On the undecidability of the equivalence of second-order tuple generating dependencies. Inf. Syst. 48 (2015)
11. Furche, T., Gottlob, G., Grasso, G., Guo, X., Orsi, G., Schallhart, C., Wang, C.: DIADEM: thousands of websites to a single database. PVLDB 7(14) (2014)
12. Furche, T., Gottlob, G., Libkin, L., Orsi, G., Paton, N.W.: Data wrangling for big data: Challenges and opportunities. In: EDBT (2016)
13. Gottlob, G., Pichler, R., Sallinger, E.: Function symbols in tuple-generating dependencies: Expressive power and computability. In: PODS. pp. 65–77. ACM (2015)
14. Halevy, A.Y., Rajaraman, A., Ordille, J.J.: Data integration: The teenage years. In: VLDB. pp. 9–16. ACM (2006)
15. Kandel, S., Heer, J., Plaisant, C., Kennedy, J., van Ham, F., Riche, N.H., Weaver, C., Lee, B., Brodbeck, D., Buono, P.: Research directions in data wrangling: Visualizations and transformations for usable and credible data. Information Visualization 10(4), 271–288 (2011)
16. Kolaitis, P.G.: Schema mappings, data exchange, and metadata management. In: PODS. pp. 61–75 (2005)
17. Kolaitis, P.G., Pichler, R., Sallinger, E., Savenkov, V.: Nested dependencies: structure and reasoning. In: PODS. pp. 176–187. ACM (2014)
18. Kolaitis, P.G., Pichler, R., Sallinger, E., Savenkov, V.: Limits of schema mappings. In: ICDT. LIPIcs, vol. 48, pp. 19:1–19:17 (2016)
19. Lenzerini, M.: Data integration: A theoretical perspective. In: Popa, L., Abiteboul, S., Kolaitis, P.G. (eds.) PODS. pp. 233–246. ACM (2002)
20. Lohr, S.: For big-data scientists, 'janitor work' is key hurdle to insights. The New York Times (2015), http://nyti.ms/1Aqif2X
21. Pichler, R., Sallinger, E., Savenkov, V.: Relaxed notions of schema mapping equivalence revisited. Theory Comput. Syst. 52(3), 483–541 (2013)
22. Sallinger, E.: Reasoning about schema mappings. In: Data Exchange, Information, and Streams, Dagstuhl Follow-Ups, vol. 5, pp. 97–127 (2013)