The Data Mapping Problem: Algorithmic and Logical Characterizations

George H. L. Fletcher
Computer Science, Indiana University, Bloomington, USA
gefletch@cs.indiana.edu

Abstract

Technologies for overcoming heterogeneities between autonomous data sources are key in the emerging networked world. Our doctoral research investigates technologies for alleviating structural heterogeneity between relational data sources. At the heart of structural heterogeneity is the data mapping problem*. The data mapping problem is to discover effective mappings between structured data sources. These mappings are the basic "glue" for facilitating large-scale ad-hoc information sharing between autonomous peers in a dynamic environment. Automating their discovery is one of the fundamental unsolved challenges for data interoperability. Our research on solutions to the data mapping problem has two main components: (1) a general algorithmic approach to automating the discovery of mappings and (2) a general formal approach to understanding the data mapping problem. We outline our progress on each of these fronts and discuss directions for future research.*

1. Introduction

The emerging networked world promises new opportunities and possibilities for information dissemination, wide-scale collaboration, and knowledge construction. These opportunities will be fostered in large part by technologies which bring together autonomous data sources. Since these data sources were created and have evolved in isolation, they are each maintained differently according to local constraints and usage patterns. Consequently, facilitating technologies must bridge a wide variety of heterogeneities. These heterogeneities encompass differences at the system level, differences in the structuring of data, and semantic pluralism in the interpretation of data.

We are investigating technologies for overcoming structural heterogeneity between relational data sources. Although research in databases has lead to great practical successes in the storage and management of structured data, it has made limited progress on technologies for alleviating structural heterogeneity. At the heart of overcoming structural heterogeneity is the *data mapping problem*: automating the discovery of mappings between structured data sources [4]. These mappings are the basic "glue" for building information sharing systems [7], and automating their discovery is one of the fundamental unsolved challenges for data interoperability, integration, and sharing. This problem arises in almost any information system with multiple data sources. Consequently, the problem has many manifestations: schema matching [16] and schema mapping [15] in databases, ontology mapping on the Semantic Web [10], schema mediation in peer-to-peer systems [7], and matching of information models [14], to name a few.

Example 1 *To illustrate the data mapping problem, consider the three databases containing student grade information in Figure 1. In this example, each database contains the same information. As shown, there are many natural ways to organize even the simplest datasets. For example,* `G1` *and* `G2` *maintain the information in a single relation, while* `G3` *contains a relation for each assignment. To move between these representations of student data, schema matchings and both data-data and data-metadata transformations must be performed. For example, mapping data from* `G1` *to* `G2` *involves promoting the values in column* `Assignment` *in* `G1` *to column names in* `G2` *and "matching" the* `Name` *and* `Student` *attributes. To move information from* `G3` *to* `G1`*, relation names must be demoted to data values.*

Any solution for overcoming structural heterogeneity must consider the full data mapping problem space for relational data sources. Both schema matchings ("traditional" metadata-metadata mappings between schema elements [16]) and data-metadata mappings (where data elements in one structure serve as metadata components in the other, or vice versa [12]) must be expressible. It is important to note that consideration of the full mapping space blurs the distinction between schema matching and schema mapping, since data-metadata mapping encompasses schema matching as a special case. When metadata itself is seen as data, both schema matching and schema mapping are encompassed in *data mapping*.

Our investigation of the data mapping problem is part of the *Modular Integration of Queryable Information Sources*

Proceedings of the 21st International Conference on Data Engineering (ICDE '05)
1084-4627/05 $20.00 © 2005 IEEE

IEEE COMPUTER SOCIETY

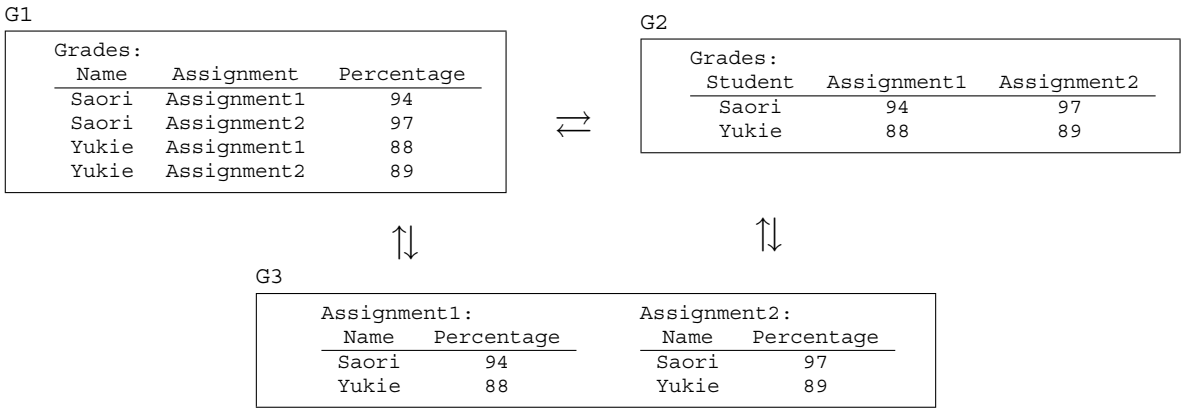

Figure 1. Mappings between student grade representations.

(MIQIS) project at Indiana University [19], a formal framework for data interoperability on the Semantic Web and in peer-to-peer data management systems. Among the distinguishing features of MIQIS is a focus on the *modular* nature of information systems, encompassing XML, relational, text, and other data sources. The framework fully respects the autonomy of peers to manage locally their schemata and concepts. The two major components of MIQIS are a formal investigation of data interoperability and a practical implementation of modules driven by this investigation.

We are focusing on the relational module of MIQIS in our doctoral research. In this paper we present our initial results in this research, which is joint work with Catherine M. Wyss. In Section 2 we present a novel algorithmic solution to the data mapping problem [4, 5]. Our approach views the data mapping problem as *search* in the full space of transformations on relational data. Our solution, which utilizes heuristic search, includes data to metadata transformations (and vice versa), allowing a generalization of previous solutions such as token-based schema matching. In Section 3 we present our initial results on a formal study of data interoperability [6]. We introduce a novel data model and calculus for expressing data mappings between relational data sources, laying the ground for a better understanding of the data mapping problem. In Section 4 we briefly discuss related works. We give concluding remarks and indications for future work in Section 5.

2. Data Mapping as a Search Problem

Can the discovery of the mappings between databases in Figure 1 be (semi) automated? A very general statement of the data mapping problem is as follows:

Definition 1 (Data Mapping Problem) *Given source data schema $\mathcal{S}$, target data schema $\mathcal{T}$, and query language $\mathcal{L}$, find a transformation $\tau \in \mathcal{L}$ (if it exists) such that for any instance s of $\mathcal{S}$ and corresponding instance t of $\mathcal{T}$, $s \overset{\tau}{\longmapsto} t$.*

This definition encompasses all variants of the data mapping problem discussed in Section 1. Note that $\mathcal{S}$ and $\mathcal{T}$ are not assumed to be schemas of the same data model. It is not immediately clear how to automate a solution to the *general* problem. In MIQIS, we focus on subcases of the problem by following a modular approach to information exchange. In our doctoral research we are developing the MIQIS module to generate SQL compatible transformations when $\mathcal{S}$ and $\mathcal{T}$ are both *relational* schemas.

We consider a fixed set of simple transformations on relational data. All structural transformations between the databases in Figure 1 can be performed using compositions of the following simple, compositional, structural transformations: $\downarrow$ for demoting metadata to data, $\rightarrow$ for dereferencing attributes, $\uparrow$ for promoting data to metadata, $\wp$ for partitioning relations, ɯ for dropping attributes, the standard relational rename operator ρ, and a simple merge operator $\oplus$. We omit a complete formal definition of the operators in this paper; these operators mimic algebraic operators developed elsewhere for federated relational systems [18].

Example 2 *Consider the basic transformations involved in restructuring the information in* `G1` *into the format of* `G2`*:*

$R_1 :=$ $\uparrow$(`G1`, `Assignment`, `Percentage`)
Promote assignments to metadata with corresponding "`Percentage`" values.
$R_2 :=$ ɯ(R_1, `Assignment`)
Drop column "`Assignment`"
$R_3 :=$ ɯ(R_2, `Percentage`)
Drop column "`Percentage`"
$R_4 :=$ ρ(R_3, `Name`, `Student`)
Rename attribute "`Name`" to "`Student`"
$R_5 :=$ $\oplus$(R_4, `Student`)
Merge assignment grades for students.

The output relation R_5 is exactly `G2`*.*

Given illustrative "canonical" source and target instances (s, t) as input, we view data mapping discovery as an exploration of the transformation space of these operators on the source instance s. Search terminates when the transformed

IEEE COMPUTER SOCIETY

source instance s becomes a superset of the target instance t. At this point, the transformational path is translated to a parameterized map between instances of the source schema and instances of the target schema. This process is illustrated in Figure 2. Approaching the data mapping problem as a search problem allows us to leverage existing artificial intelligence search techniques. In this approach, no assumptions of common domains, global schema, underlying generative ontology, or other simplifications are made. Data are treated simply as opaque objects; the search process is purely syntactically and structurally driven [1, 11]. The user-provided source and target instances provide the initial matches which drive the search process.

A prototype search module for relational data mappings has been implemented in Scheme [4, 5]. We have tested this module with simple search heuristics and shown that it is effective for both schema matching and full data mapping.

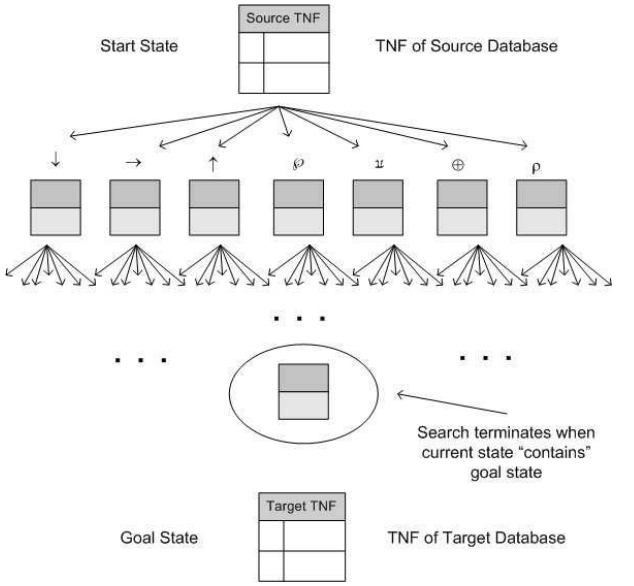

Figure 2. The search for data mappings.

3. Data Mapping Calculus

The second contribution of our research is a simple declarative data mapping calculus (DMC) for reasoning about the data mapping problem. Our calculus is a minimal extension of the standard relational model for cleanly expressing structural transformations for data mapping. This logical formalism complements our work on automating solutions to the data mapping problem [4, 5]. This is joint work with Edward Robertson and Dirk Van Gucht [6].

In the data model for the DMC, databases are named finite sets of tuples. Tuples, in turn, are named finite sets of ordered pairs over some infinite domain of atoms. Intuitively, the name of a tuple is its relation name and the ordered pairs are the attribute-value pairs of the tuple.

Example 3 *In this data model, the single database* `G2` *of Figure 1 has the following representation:*

$$\begin{array}{lll} \texttt{G2}, & \langle \texttt{Grades}, & \{\langle \texttt{Student}, \texttt{Saori}\rangle, \\ & & \langle \texttt{Assignment1}, 94\rangle, \\ & & \langle \texttt{Assignment2}, 97\rangle\}\rangle, \\ & \langle \texttt{Grades}, & \{\langle \texttt{Student}, \texttt{Yukie}\rangle, \\ & & \langle \texttt{Assignment1}, 88\rangle, \\ & & \langle \texttt{Assignment2}, 89\rangle\}\rangle \end{array}$$

During data mapping, the output schema of a query is dynamic and not known at runtime. Hence null values have typically been introduced as special values indicating missing or inapplicable information. The data model for the DMC avoids the problems associated with this approach by viewing a null value as *absence*. For example, if a single student in `G2` has the attribute-value pair $\langle$`extra-credit`, `83`$\rangle$, it is not necessary to explicitly have null values for the attribute `extra-credit` in the tuples of all the other students.

The DMC is essentially the standard relational calculus extended with simple syntax for seamlessly manipulating metadata; in particular, it is easy in DMC to express each of the algebraic transformations used in the data mapping solution presented in Section 2. We will present the language by examples which illustrate its expressive power.

Example 4 *We begin with a standard query on database* `G1` *to return the student names in the* `Grades` *relation in an output relation named* `Students`*:*

$$\{x : t \mid (\exists r : s \in \texttt{G1})\ r = \texttt{Grades} \wedge x = \texttt{Students} \wedge t.\texttt{Name} = s.\texttt{Name}\}$$

Intuitively, the new syntax "$\exists r : s \in$ `G1`*" means "there exists a tuple* s *in relation* r *in database* `G1`*." The output is all tuples* t *with relation name* x *satisfying the query.*

Next, we give two data-metadata queries. The query to map `G1` *to* `G2` *(as in Example 2) must promote data values to attribute values:*

$$\begin{array}{l}\{x : t \mid (\exists r : s \in \texttt{G1}) \\ \quad x = \texttt{Grades} \wedge r = \texttt{Grades} \wedge t.\texttt{Student} = s.\texttt{Name} \\ \wedge\ (\forall r' : s' \in \texttt{G1})(\forall a)\ r' = r \wedge t.\texttt{Student} = s'.\texttt{Name} \\ \qquad \Rightarrow (a = s'.\texttt{Assignment} \wedge t.a = s'.\texttt{Percentage})\}\end{array}$$

Note that the DMC *syntax cleanly captures the dynamic output schema.*

The query to map `G2` *to* `G3` *must promote attribute names to relation names:*

$$\begin{array}{l}\{x : t \mid (\exists r : s \in \texttt{G2})(\exists a)(\exists v)\ r = \texttt{Grades} \\ \qquad \wedge\ a \neq \texttt{Student} \wedge s.a = v \wedge x = a \\ \qquad \wedge\ t.\texttt{Name} = s.\texttt{Student} \wedge t.\texttt{Percentage} = v\}\end{array}$$

It turns out that the full calculus is unsafe in several respects. Hence we are working on syntactic restrictions to make DMC equivalent to tractable languages for relational data-metadata querying [6, 18].

IEEE COMPUTER SOCIETY

4. Related Work

Overcoming structural heterogeneity is a long standing problem in database research [12, 16]. We briefly discuss research related to our algorithmic solution and DMC.

The approaches to data mapping most closely related to our data mapping solution are the works of Bilke and Naumann [1] and Kang and Naughton [11] on schema matching, and the Clio project [15] on schema mapping. To our knowledge, these works have not considered the full space of data-metadata transformations, with only the Clio [15] project considering any aspects of such mappings. Our work complements and extends these works with a new perspective on the data mapping problem and a novel solution to this problem for the complete relational transformation space. Our solution can be considered as a useful addition to a multi-strategy data mapping approach [3].

Our work on DMC is motivated by relational languages for database interoperability. The DMC is a descendant of the Uniform Calculus developed by Jain et al. [9], specialized to investigate relational data mapping. Our calculus is a novel development of Jain's language that clearly captures a very minimal extension of the standard relational model for structural transformations for data mapping. Other influential research for our work is Chen et al. [2] on higher order logic programming, Krishnamurthy et al. [12], and Lakshmanan et al. [13] on multidatabase languages, and Sattler et al. [17] on languages for database federations. Our work complements and extends this research with a *logical* characterization of the full space of relational data mapping transformations.

5. Conclusions and Future Work

In this paper we presented doctoral research on the data mapping problem for relational data sources. We discussed a novel solution to the data mapping problem addressing the full space of data-metadata transformations. This solution was founded on a new perspective on the data mapping problem as a search problem. We also discussed ongoing work on a formal underpinning for the data mapping problem. We developed DMC, a calculus for data mapping, and illustrated its appropriateness for expressing data mapping transformations

We are currently enhancing the performance of our prototype algorithmic solution, incorporating improvements developed in the search literature and more sophisticated heuristics [4, 5]. We are also applying our approach to data mapping on the "deep web" [8]. Many thorny research issues remain to be explored for DMC. We are currently tackling safety issues and pinning down a fragment of DMC which is equivalent to tractable languages for data mapping [18]. In addition, with DMC in hand we are now in a position to clearly formalize the data mapping problem in terms of DMC decision problems. Our next major step in this research is to formally state these problems and establish their complexity and/or decidability.

References

[1] Bilke, Alexander and Felix Naumann, "Schema Matching using Duplicates," in *Proc. IEEE ICDE*, Tokyo, Japan, 2005.

[2] Chen, Weidong, Michael Kifer, and David Scott Warren, "HILOG: A Foundation for Higher-Order Logic Programming," *J. Logic Programming* 15(3): 187-230, 1993.

[3] Do, Hong-Hai, and Erhard Rahm, "COMA - A System for Flexible Combination of Schema Matching Approaches," in *Proc. VLDB Conf.*, pp. 610-621, Hong Kong, China, 2002.

[4] Fletcher, George H.L. and Catharine M. Wyss, "Mapping Between Data Sources on the Web," in *Proc. IEEE ICDE Workshop WIRI*, Tokyo, Japan, 2005.

[5] Fletcher, George H.L. and Catharine M. Wyss, "Relational Data Mapping in MIQIS," Demo to appear in *Proc. ACM SIGMOD*, Baltimore, Maryland, USA, 2005.

[6] Fletcher, G.H.L., C.M. Wyss, E.L. Robertson, and D. Van Gucht, "A Calculus for Data Mapping," in *Proc. COORDINATION Workshop InterDB*, Namur, Belgium, 2005.

[7] Halevy, A. Y., Z. G. Ives, D. Suciu, and I. Tatarinov, "Schema Mediation in Peer Data Management Systems," in *Proc. IEEE ICDE*, pp. 505-516, Bangalore, India, 2003.

[8] He, Bin and Kevin Chen-Chuan Chang, "Statistical Schema Matching Across Web Query Interfaces," in *Proc. ACM SIGMOD*, pp. 217-228, San Diego, CA, USA, 2003.

[9] Jain, M., A. Mendhekar, and D. Van Gucht, "A Uniform Data Model for Relational Data and Meta-Data Query Processing," in *Proc. COMAD*, Pune, India, 1995.

[10] Kalfoglou, Y. and M. Schorlemmer, "Ontology Mapping: the State of the Art," *Knowledge Eng. Review* 18(1):1-31, 2003.

[11] Kang, Jaewoo and Jeffrey F. Naughton, "On Schema Matching with Opaque Column Names and Data Values," in *Proc. ACM SIGMOD*, pp. 205-216, San Diego, CA, 2003.

[12] Krishnamurthy, Ravi, et al., "Language Features for Interoperability of Databases with Schematic Discrepancies," in *Proc. ACM SIGMOD*, pp. 40-49, Denver, CO, USA, 1991.

[13] Lakshmanan, L.V.S., F. Sadri, and I.N. Subramanian, "Logic and Algebraic Languages for Interoperability in Multidatabase Systems," *J. Logic Prog.* 33(2):101-149, 1997.

[14] Melnik, Sergey, "Generic Model Management: Concepts and Algorithms," *Springer Verlag LNCS 2967*, 2004.

[15] Miller, Renée J., Laura M. Haas, and Mauricio A. Hernández, "Schema Mapping as Query Discovery," in *Proc. VLDB Conf.*, pp. 77-88, Cairo, Egypt, 2000.

[16] Rahm, E. and P. Bernstein, "A Survey of Approaches to Automatic Schema Matching," *VLDB J.* 10(4):334-350, 2001.

[17] Sattler, Kai-Uwe, et al., "Interactive Example-Driven Integration and Reconciliation for Accessing Database Federations," *Information Systems* 28(5):393-414, July 2003.

[18] Wyss, Catharine M. and Edward Robertson, "Relational Languages for Metadata Integration," *ACM Transactions on Database Systems*, to appear June 2005.

[19] Wyss, Catharine M., George H.L. Fletcher, Fulya Erdinc, and Jeremy T. Engle, "MIQIS: Modular Integration of Queryable Information Systems," in *Proc. VLDB Workshop IIWeb*, pp. 136-140, Toronto, Canada, 2004.