Relational Data Mapping in MIQIS

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

Catharine M. Wyss
Computer Science and Informatics
Indiana University at Bloomington
cmw@cs.indiana.edu

ABSTRACT

We demonstrate a prototype of the relational data mapping module of MIQIS, a formal framework for investigating information flow in peer-to-peer database management systems. Data maps constitute effective mappings between structured data sources. These mappings are the 'glue' for facilitating large scale ad-hoc information sharing between autonomous peers, and automating their discovery is one of the fundamental unsolved challenges for information interoperability and sharing. Our approach to automating data map discovery utilizes heuristic search within a space delineated by basic relational transformation operators. A novelty of our approach is that these operators include data to metadata transformations (and vice versa). This approach leverages new perspectives on the data mapping problem, and generalizes previous approaches such as token-based schema matching.

1. INTRODUCTION

The vision of peer-to-peer database management systems (P2PDBMS) brings promise of ad-hoc dynamic information sharing, with support for richer semantics than the current breed of simple file-sharing peer-to-peer systems [7, 10]. The complementary vision of the Semantic Web also holds promise for intelligent complex information exchange on the Web [2, 7]. It is important to note that these systems cannot and should not be built from scratch. A significant portion of data on the Web resides in non-Semantic-Web-enabled structured sources [4, 7]. The participation of these sources in P2PDBMS information sharing scenarios requires new enabling technologies which respect source autonomy.

A fundamental unsolved challenge in information sharing is the *Data Mapping Problem*: automating the discovery of effective mappings between structured representations of data. These mappings are the basic 'glue' for facilitating ad-hoc information exchange between autonomous peers [7]. This central problem is encountered in many information management settings. Consequently, many variants of the problem have been identified and investigated: schema mapping [17], schema matching [5, 18], and data translation [19] during data integration [12], ontology mapping [8] on the Semantic Web, and model matching [15], to name a few.

SIGMOD 2005 June 14-16, 2005, Baltimore, Maryland, USA.

We are investigating the data mapping problem in the *Modular Integration of Queryable Information Sources* (MIQIS) project at Indiana University [21], a formal framework for investigating many aspects of information flow in P2PDBMS. Among the distinguishing features of MIQIS are its generality and focus on the *modular* nature of information systems. This approach naturally encompasses XML, relational, text, deep web [4], and other data sources. The framework fully respects the autonomy of peers to manage locally their schemata and concepts. On the Semantic Web and P2PDBMS, global consensus and monolithic architectures are infeasible. MIQIS fully accommodates this heterogeneity of data sources in large scale ad-hoc information sharing scenarios.

2. RELATIONAL DATA MAPPING

Can the discovery of data mappings be (semi) automated? That is, can the discovery of an appropriate mapping between data structured under two distinct schemas be facilitated with minimal user input? A very general statement of the *Data Mapping Problem* is as follows:

DEFINITION 1 (DATA MAPPING PROBLEM). *Given source schema $\mathcal{S}$, target 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 $\mathcal{T}$, $s \overset{\tau}{\longmapsto} t$.*

Note that in this most general problem statement, we do not assume $\mathcal{S}$ and $\mathcal{T}$ are schemas of the same data model. It is not immediately clear how to automate a solution to this problem. In this demonstration we report on a prototype implementation of the MIQIS module to generate τ when $\mathcal{S}$ and $\mathcal{T}$ are both *relational* schemas.

EXAMPLE 1. *Suppose that peers contain student grade information within a larger 'e-learning' network for managing student information. In this example, suppose there are three schools managing separate relational databases for this data as illustrated in Figure 1. As shown, there are many natural ways to organize even the simplest datasets such as these. Note that to move between these representations of student data, data $\leftrightarrow$ metadata transformations must be performed. We will use this setting as our running example in this demonstration.*

2.1 Data Mapping as Search

We view the data mapping problem as a *search problem*. A key component of our solution is the *Rosetta Stone Principle*: user provided small 'canonical' instances of $\mathcal{S}$ and $\mathcal{T}$ can be effectively used to guide the discovery of τ in a transformation space. These instances are elicited in a manner similar to the interactive building of extraction patterns in the Lixto visual wrapper system [6]. In our approach, we also explicitly consider the full data mapping problem space for relational DBs: we consider both schema matchings

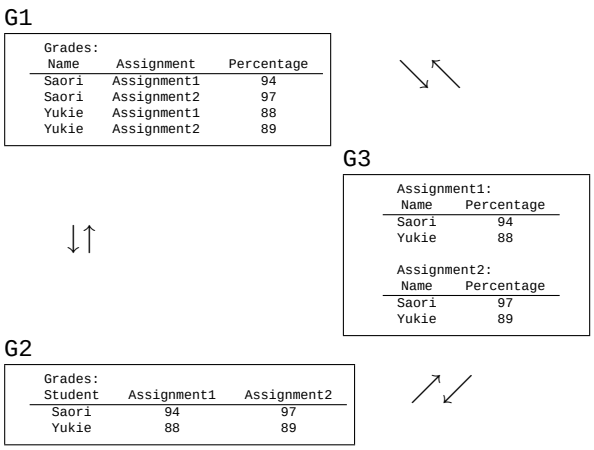

Figure 1: Mappings between student grade databases.

Operation	Effect
$\downarrow (R)$	*Demote Metadata*. Cartesian product of relation R with binary table containing the metadata of R.
$\rightarrow (R, A, B)$	*Dereference Column A on B.* $\forall t \in R$, append a new column B with value $t[t[A]]$.
$\uparrow (R, A, B)$	*Promote Column A to Metadata.* $\forall t \in R$, append a new column $t[A]$ with value $t[B]$.
$\wp(R, A)$	*Partition on Column A.* $\forall v \in \pi_A(R)$, create a new relation named v, where $t \in v$ iff $t \in R$ and $t[A] = v$.
$\text{ɯ}(R, A)$	*Drop* column A from relation R.
$\oplus(R, A)$	*Merge* tuples in relation R based on compatible values in column A.
$\rho(R, R', A, A')$	*Rename* relation R to R' and column A to A'.

Table 1: Basic transformations defining relational search space.

(ie, traditional metadata $\leftrightarrow$ metadata mappings between schema elements) and data $\leftrightarrow$ metadata mappings where data elements in one structure serve as metadata components in the other (or vice versa) [11, 16]. It is important to note that consideration of the full mapping space blurs the distinction between schema matching [18] and schema mapping [17]. Data-metadata 'matching' encompasses schema matching as a special case; when metadata itself is seen as data, the entirety of schema matching and schema mapping is encompassed in *data mapping*. The output of our relational data mapping module is a data-to-data transformation that is parameterized by schema information.

2.2 Tuple Normal Form

Another key technical component of our approach is a normal form for relational data, *Tuple Normal Form* (TNF), first introduced by Litwin et al. [13]. This standardized format for representing relational data allows us to seamlessly manipulate metadata alongside data using standard SQL. Furthermore, multiple input relations are represented in a single TNF relation; thus TNF enables data mappings where the source and/or target information may be split over more than one relation (such as the transformations involving database G3 in Figure 1).

For a given relation, r, we compute TNF of r (denoted r^*) as follows. First, every tuple in the relation is given a unique identifier. Then, r^* is a four-column relation with attributes TID, REL, ATT, VALUE containing the data in r in a piecemeal fashion. For an input database d, d^* is simply the union of r^* for all $r \in d$.

EXAMPLE 2. *We illustrate this with the TNF of database G3.*

*G3** :

TID	REL	ATT	VALUE
t_1	Assignment1	Name	Saori
t_1	Assignment1	Percentage	94
t_2	Assignment1	Name	Yukie
t_2	Assignment1	Percentage	88
t_3	Assignment2	Name	Saori
t_3	Assignment2	Percentage	97
t_4	Assignment2	Name	Yukie
t_4	Assignment2	Percentage	89

Note that TNF of a database can be computed in SQL with access to system tables.

2.3 Relational Transformation Space

Ideally we would like our mapping language $\mathcal{L}$ to be practical. In the context of relational data sources, this means mappings must be SQL compatible queries so that we can maximize the use of underlying RDBMS technology. In our solution, we consider a fixed set of simple SQL compatible transformations on data in TNF (Table 1). This allows us to consider data mapping discovery as an exploration of the transformation space of these operators on the input Rosetta Stone source instance. Search terminates when the TNF representation of the source instance becomes a superset of the TNF representation of the input Rosetta Stone target instance. At this point, the transformational path is translated to a parameterized map between instances of the source schema and instances of the target schema.

In the approach, we make no assumption of common domains, global schema, underlying generative ontology, or other simplifications. We treat data simply as opaque objects; the search process is purely syntactically and structurally driven [3, 9]. As per the Rosetta Stone Principle, the user-provided canonical source and target instances provide the initial matches which drive the search process.

All structural transformations between student databases in Figure 1 can be performed using compositions of the simple, compositional, invertible transformations shown in Table 1. These operators mimic algebraic operators developed elsewhere for federated relational systems [20]. These operators are easily implemented in SQL on the TNF representations of relational databases.

EXAMPLE 3. *Consider the basic transformations involved in restructuring the information in G1 into the format of G2.*

$R_1 :=\uparrow (G1^*, \texttt{Assignment}, \texttt{Percentage})$
Promote assignments to metadata.
$R_2 := \text{ɯ}(R_1, \texttt{Assignment})$
Drop column Assignment.
$R_3 := \text{ɯ}(R_2, \texttt{Percentage})$
Drop column Percentage.
$R_4 := \rho(R_3, \_, \texttt{Name}, \texttt{Student})$
Rename column Name *to* Student.
$R_5 := \oplus(R_4, \texttt{Student})$
Merge assignment grades for students.

The output TNF relation R_5 is exactly $G2^*$.

Note that the user is responsible for providing post-filters such as "Drop all students with grades less than 70", if desired. The operators presented focus on bulk structural transformations rather than selections. In fact, selection conditions cannot in general be uniquely determined [10].

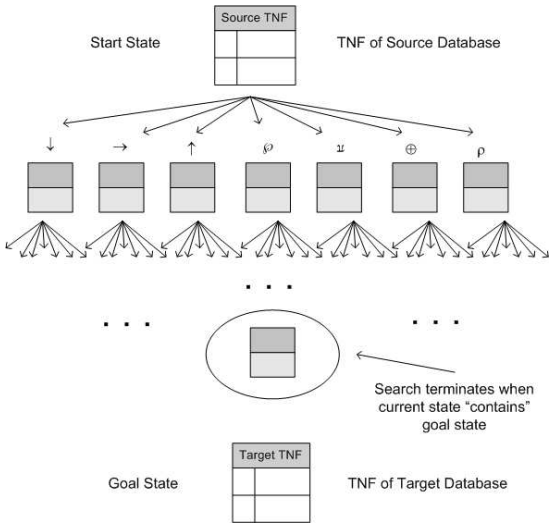

Figure 2: The search for relational data mappings.

Approaching the data mapping problem as discovery of a transformation from TNF to TNF database representations in a relational search space delineated by basic transformations allows us to leverage existing artificial intelligence search techniques [14]. The search is depicted in Figure 2. The branching factor of the search must take into account the active domain in the Rosetta Stone instances, which means the (unoptimized) branching factor is proportional to $|s| + |t|$. In spite of this daunting search space size, our initial experiments have shown that the search performs well in practice. In particular, metadata to metadata mappings (i.e., token-based schema matching [18]) are possible for hitherto unrealized scenarios, such as multi-relation mappings between wide source and target instances.

3. RELATED WORK

Due to space constraints, we omit a comprehensive discussion of related work. Most closely related to our data mapping solution are the works of Bilke et al. [3] on using duplicate values to guide schema matching, Barbançon et al. [1] and Doan et al. [5] on leveraging machine learning techniques for schema integration, Kang et al. [9] on treating data as opaque objects during schema matching, and the Clio project [17] on semi-automating the discovery of schema 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 transformation space for relational sources.

4. DEMONSTRATION

A prototype semi-automatic search module for relational data mappings has been fully implemented in Scheme. The search routine takes as input Rosetta Stone source and target instances, translates them into TNF, and performs the search for a transformation from the source to the target as outlined above. The purpose of the demonstration is to illustrate the approach on a variety of real and synthetic input databases for schema matching and full data-metadata mapping.

In particular, we will compare two search procedures and three heuristic measures governing the search process in the demonstration. First, a comparison between between *A** and *Iterative Deepening A** (IDA*) search methods [14] will be illustrated. Although IDA* does redundant work, it has performance asymptotic to A* with significantly less memory usage and hence performs well in practice.

A comparison between three search heuristics will also be given. Each of the three measures gives an idea of the relative "containment" of the target database in the current search database. Search is terminated when the target is fully contained in the transformed input instance. The path taken is then "backed up" to produce the generating transformations. For a given search state x and target state t, heuristic h_1 measures the number of relation, column, and data values in the target state t which are missing in state x, heuristic h_2 measures the minimum number of promotions ($\uparrow$) and demotions ($\downarrow$) needed to transform x into target t, and heuristic h_3 is simply $\mathbf{max}\{h_1(x), h_2(x)\}$.

During the demonstration, we will allow participants to graphically select search method, heuristic measure, and source/target inputs from a large body of examples. We will then illustrate the progress of the search in real time.

5. REFERENCES

[1] Barbançon, F. and D.P. Miranker. Interactive Schema Integration with Sphinx, in *Proc. FQAS*, pp. 175-190, Lyon, France, 2004.

[2] Berners-Lee, Tim, James Hendler, and Ora Lassila. The Semantic Web, *Scientific American* 284(5):34-43, May 2001.

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

[4] Chang, Kevin C.-C., et al. Structured Databases on the Web: Observations and Implications, *SIGMOD Record* 33(3):61-70, 2004.

[5] Doan, AnHai, Pedro Domingos, and Alon Halevy. Learning to Match the Schemas of Databases: A Multistrategy Approach, *Machine Learning* 50(3):279-301, March 2003.

[6] Gottlob, Georg, C. Koch, R. Baumgartner, M. Herzog, and S. Flesca. The Lixto Data Extraction Project – Back and Forth between Theory and Practice, in *Proc. ACM PODS*, pp. 1-12, Paris, France, 2004.

[7] Ives, Zachary G., et al. Piazza: Mediation and Integration Infrastructure for Semantic Web Data, *J. Web Sem.* 1(2):155-175, 2004.

[8] Kalfoglou, Yannis and M. Schorlemmer. Ontology Mapping: the State of the Art, *Knowledge Engineering Review* 18(1):1-31, 2003.

[9] 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, USA, 2003.

[10] Kementsietsidis, Anastasios, et al. Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues, in *Proc. ACM SIGMOD*, pp. 325-336, San Diego, CA, USA, 2003.

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

[12] Lenzerini, Maurizio. Data Integration: A Theoretical Perspective, in *Proc. ACM PODS*, pp. 233-246, Madison, WI, USA, 2002.

[13] Litwin, Witold, et al. First Order Normal Form for Relational Databases and Multidatabases, *SIGMOD Record* 20(4):74-76, 1991.

[14] Luger, George F. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5th Ed.*, Addison-Wesley, London, 2005.

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

[16] Miller, Renée J. Using Schematically Heterogeneous Structures, in *Proc. ACM SIGMOD*, pp. 189-200, Seattle, WA, USA, 1998.

[17] Miller, Renée J., et al. Schema Mapping as Query Discovery, in *Proc. VLDB Conf.*, pp. 77-88, Cairo, Egypt, 2000.

[18] Rahm, Erhard and Philip A. Bernstein. A Survey of Approaches to Automatic Schema Matching, *VLDB Journal* 10(4):334-350, 2001.

[19] Torlone, Riccardo and Paolo Atzeni. A Unified Framework for Data Translation over the Web, in *Proc. IEEE WISE*, pp. 350-358, Kyoto, Japan, 2001.

[20] Wyss, Catharine M. and Edward Robertson. Relational Languages for Metadata Integration, *ACM TODS*, to appear June 2005.

[21] Wyss, Catharine M., et al. MIQIS: Modular Integration of Queryable Information Systems, in *Proc. VLDB Workshop IIWeb*, pp. 136-140, Toronto, Canada, 2004.