Why is Schema Matching Tough and What Can We Do About It?

Avigdor Gal
Technion – Israel Institute of Technology
avigal@ie.technion.ac.il

ABSTRACT

In this paper we analyze the problem of schema matching, explain why it is such a "tough" problem and suggest directions for handling it effectively. In particular, we present the monotonicity principle and see how it leads to the use of top-K mappings rather than a single mapping.

1. INTRODUCTION

Schema matching is the task of matching between concepts describing the meaning of data in various heterogeneous, distributed data sources (*e.g.* database schemata, XML DTDs, HTML form tags, *etc.*). Schema matching is recognized to be one of the basic operations required by the process of data integration [4], and thus has a great impact on its outcome. Schema mappings (the outcome of the matching process, as will be defined in Section 2) can serve in tasks of generating global schemata, query rewriting over heterogeneous sources, duplicate data elimination, and automatic streamlining of workflow activities that involve heterogeneous data sources. As such, schema matching has impact on numerous applications. It impacts business, where company data sources continuously realign due to changing markets. It also impacts life sciences, where scientific workflows cross system boundaries more often than not.

Despite two decades of research in this area, summarized in surveys (*e.g.*, [21, 7, 23]) and various online lists (*e.g.*, OntologyMatching[1], Ziegler[2], DigiCULT[3], SWgr[4]) schema matching still seems to involve ad-hoc solutions with only a few works that involve foundational principles of schema matching [4, 17, 15, 11, 2].

In 2001, Maurizio Lenzerini claimed that "Data Integration Is Harder Than You Thought" [14]. Here we shall analyze the problem of schema matching, which is a subproblem of data integration, explain why it is such a "tough" problem and suggest directions for handling it effectively.

2. SCHEMA MATCHING BASICS

Due to its cognitive complexity [6], traditionally schema matching has been performed by human experts [13]. As the process of data integration has become more automated, the ambiguity inherent in concept interpretation has become one of the main obstacles to effective schema matching. For obvious reasons, manual concept reconciliation in dynamic environments (with or without computer-aided tools) is inefficient and at times close to impossible. Introduction of the Semantic Web vision [3] and shifts toward machine-understandable Web resources and Web services have made even clearer the vital need for automatic schema matching.

Although these tools comprise a significant step towards fulfilling the vision of automated schema matching, it has become obvious that the user must accept a degree of imperfection in this process [11]. A prime reason for this is the enormous ambiguity and heterogeneity of data description concepts: It is unrealistic to expect a single mapping engine to identify the correct mapping for any possible concept in a set. Another (and probably no less crucial) reason is that "the syntactic representation of schemas and data do not completely convey the semantics of different databases" [19]; *i.e.*, the description of a concept in a schema can be semantically misleading. Therefore, managing uncertainty in schema matching has been recognized as the next issue on the research agenda in the realm of data integration [15].

As a basis for our discussion we next layout a generic model for schema matching that serves the needs of this paper. However, the reader should not consider this model to be complete or unique. Let S_1 and S_2 be two schemata, defined using some data model (*e.g.*, relational or ontological), with n_1 and n_2 attributes, respectively. We set no particular constraints on the nature of attributes. Therefore, attributes can be as simple as relational schema attributes, or complex, *e.g.*, sub-trees in an XML schema. A common representation of the schema matching problem in the literature uses a bipartite graph $G = (V, E)$, where nodes represent attributes, each side of the graph represents a different schema (*i.e.*, $V = S_1 \cup S_2$), and an edge (v, u) represents a possible mapping between attributes. A schema mapping in this setting is a subset $E' \subseteq E$.

3. MODELING UNCERTAINTY IN SCHEMA MATCHING

Melnik and Bernstein [4, 17] have proposed the *Match* abstraction as a basic tool for model management. Match operates on schemata and returns a mapping $E' \subseteq E$.

To represent the uncertainty inherent in the matching process, one can extend the bipartite graph to use labeled edges, where a label of an edge can have the semantics of a "level of certainty." Therefore, a label can be a function $\omega : E \rightarrow [0, 1]$ where a label of 1 between two nodes represents the highest level of certainty regarding the attribute mapping.

It is common to compute the uncertainty of a schema

[1] http://www.ontologymatching.org/

[2] http://www.ifi.unizh.ch/˜pziegler/IntegrationProjects.html

[3] http://www.digicult.info/pages/resources.php?t=10

[4] http://www.semanticweb.gr/modules.php?name=News&file=categories&op=newindex&catid=17

matching from the level of uncertainty of its components. Therefore, given a mapping (E', one can define a schema mapping level of certainty as a function $\Omega : 2^E \rightarrow [0, 1]$. Examples of Ω include weighted average, dice [8], and others (*e.g.*, by comparing top-2 mappings [5]).

So far, no limitations were set on the set of edges in E'. A typical cardinality constraint of 1 : 1 sets a constrains as follows: $\forall v \in V, (v, u) \in E' \rightarrow \forall w \in V \backslash \{u\}, (v, w) \nrightarrow E'$. It is worth noting that only limited research is currently devoted to other types of cardinality constraints, such as $1 : n$, $n : 1$, and $n : m$ (*e.g.*, [24]). The interested reader is referred to [9] for a discussion and analysis of this phenomenon.

While modeling uncertainty in schema matching provides a nice formal framework, many questions remain unresolved. How can one choose a good heuristic for determining the labeling of edges (ω)? Are there "good" and "bad" Ω functions? The next section presents the monotonicity principle as a mechanism for providing insights into the selection process of the ω and Ω functions.

4. THE MONOTONICITY PRINCIPLE

The evaluation of schema mappings (the outcome of schema matching) is typically performed with respect to some "golden rule" mapping, as given (at least conceptually) by a domain expert. We denote such a mapping to be the *exact mapping*. Clearly, such expert opinions are not readily available when matching schemata (otherwise, one can simply use the expert opinion). Therefore, empirical evaluation is typically performed on a limited set of schemata to "get the feeling" on the performance of a matching algorithm. Two metrics (and their combinations), borrowed from the area of Information Retrieval, namely *precision* and *recall*, were used for the empirical evaluation of performance. Assume that out of the $n_1 \times n_2$ attribute mappings, there are $c \leq n_1 \times n_2$ correct attribute mappings, with respect to the expert mapping. Also, let $t \leq c$ be the number of mappings, out of the correct mappings, that were chosen by the matching algorithm and $f \leq n_1 \times n_2 - c$ be the number of incorrect such attribute mappings. Then, precision is computed to be $\frac{t}{t+f}$ and recall is computed as $\frac{t}{c}$. Clearly, higher values of both precision and recall are desired. From now on, we shall focus on the precision measure, denoting by $p(E')$ the precision of a schema mapping E'. Extensions that include the recall measure as well are left open for future research.

We observe that precision takes its values from a discrete domain in $[0, 1]$. Therefore, one can create equivalence schema mapping classes on 2^E, the power set of G's edges. Two mappings E' and E'' belong to a class p if $p(E') = p(E'') = p$, where $p \in [0, 1]$. Let us consider now two mappings, E' and E'', such that $p(E') < p(E'')$. For each of these two mappings we can compute their schema mapping level of certainty, $\Omega(E')$ and $\Omega(E'')$. We say that a matching algorithm is *monotonic* if for any two such mappings $p(E') < p(E'') \rightarrow \Omega(E') < \Omega(E'')$.

Clearly, a monotonic matching algorithm can easily identify the exact mapping. Let E^* be the exact mapping, then $p(E^*) = 1$. For any other mapping E', $p(E') \leq p(E^*)$, since p takes its values in $[0, 1]$. Therefore, if $p(E') < p(E^*)$ then from monotonicity $\Omega(E') < \Omega(E^*)$. All one has to do then is to devise a method for finding a mapping E^* that maximizes Ω.[5] In fact, this is one of the two most common methods for identifying the exact mapping nowadays [8, 11, 5]. The other common method, adopted in [18, 12] and others, is to only determine the values of ω automatically, allowing the user to identify the exact (schema) mapping from the individual attribute mappings.

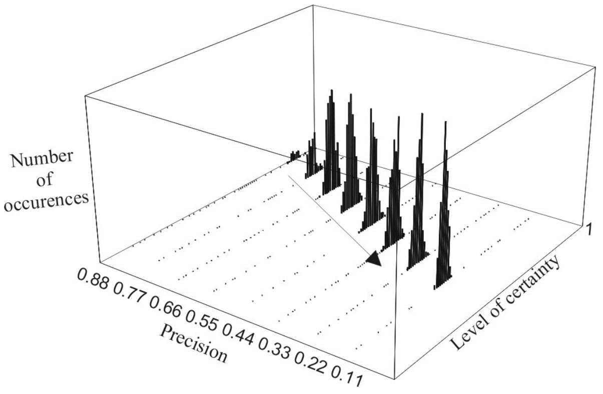

Figure 1: Illustration of the monotonicity principle

Figure 1 provides an illustration of the monotonicity principle using a mapping of "Absolute Agency" with "Adult Singles," both taken from the dating and matchmaking domain, using the combined algorithm, as provided by OntoBuilder.[6] Given a set of mappings, each value on the x-axis represents a class of schema mappings with a different precision. The z-axis represents the level of certainty. Finally, the y-axis stands for the number of schema mappings from a given precision class and with a given level of certainty.

Two main insights can be derived from Figure 1. First, the level of certainty of mappings within each schema mapping class form a "bell" shape, centered around a specific level of certainty. Such a behavior indicates a certain level of robustness of a schema matcher, assigning similar certainty levels to mappings within each class. Second, the "tails" of the bell shapes of different classes overlap. Therefore, a schema mapping from a class of a lower precision may receive a higher level of certainty than a mapping from a class of a higher precision. This, of course, contradicts the monotonicity definition. In fact, our experience with various schema matching algorithms and various real-world schemata shows that no matcher we have encountered is shown (even empirically) to be monotonic. However, the first observation serves as a motivation for a definition of a statistical monotonicity, first introduced in [11]:

Definition 1 (Statistical monotonicity). *Let $\mathcal{E} = \{E_1, E_2, ..., E_m\}$ be a set of mappings over schemata S_1 and S_2 with n_1 and n_2 attributes, respectively, and define $n = \max(n_1, n_2)$. Let $\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_{n+1}$ be subsets of $\mathcal{E}$ such that for all $1 \leq i \leq n+1$, $E \in \mathcal{E}_i$ iff $\frac{i-1}{n} \leq p(E) < \frac{i}{n}$. We define M_i to be a random variable, representing the level of certainty of a randomly chosen mapping from $\mathcal{E}_i$. $\mathcal{E}$ is* statistically monotonic *if the following inequality holds for any $1 \leq i < j \leq n+1$:*

$$\bar{\Omega}(M_i) < \bar{\Omega}(M_j) \tag{1}$$

[5] In [11], where the monotonicity principle was originally introduced, it was shown that while such a method works well for fuzzy aggregators (*e.g.*, weighted average) it does not work for t-norms such as min.

[6] http://ie.technion.ac.il/OntoBuilder

where $\bar{\Omega}(M)$ stands for the expected value of M.

Intuitively, a schema matching algorithm is statistically monotonic with respect to given two schemata if the expected certainty level increases with precision. Statistical monotonicity can assist us in explaining certain phenomena in schema matching and also to serve as a guideline in finding better ways to use schema matching algorithms.

5. FROM STATISTICAL MONOTONICITY TO TOP-K SCHEMA MAPPINGS

Consider the set of all possible mappings between two schemata, ranked in a decreasing order of $\Omega(\cdot)$ of some (statistically) monotonic matcher. Assume that this set of mappings is statistically monotonic. Looking at the top-ranked mapping (the best mapping E'), is it the exact mapping E^*? Not necessarily. Although $\bar{\Omega}(M_i) < \bar{\Omega}(M_n)$ for all $1 \leq i < n$, for specific instances $\Omega(E') > \Omega(E^*)$ is a valid option, given the statistical nature of our definition. Therefore, an immediate observation from this analysis is that if one limits oneself to a matching task that identifies the best mapping, one may not identify the exact mapping. This is a well-known fact in the schema matching research area, since rarely one has a precision and recall of 1, indicating that $E' = E^*$. Therefore, this model serves as a justification, in a retrospect to a tough reality (recall that the first part of the title of this paper is "Why is Schema Matching Tough").

An additional observation goes as follows. Given the monotonic nature of our matcher, it seems likely that the number of mappings E' satisfying $\Omega(E') > \Omega(E^*)$ is small with respect to the total number of possible mappings. To motivate this observation, consider once more Figure 1 and note that a mapping E' for which $\Omega(E') > \Omega(E^*)$ must come from the "upper tail" of the mapping groups with lower precision. The chance that such a mapping will indeed receive a higher similarity measure decreases with group precision. Therefore, the exact mapping is likely to be found in the top-K mappings, where K depends on the distribution of similarity values of the precision groups, but is likely to be much smaller than the set of all possible mapping.

Rank	Combined algorithm
0	71%
1-10	19%
11-99	10%
>100	0%
Average rank	7

Table 1: Exact mapping positioning with respect to the best mapping

Table 1 presents an empirical analysis, taken from [11], summarizing the positioning of the exact mapping using a statistically monotonic matcher. A rank of 0 means that the algorithm was successful in identifying the exact mapping as the best mapping. Other ranks show the positioning within all possible mappings ($9! = 362,880$). Even if this matcher fails to identify the exact mapping as the best mapping, it was still ranked high, saving one the need to possibly iterate over all permutations.

Following this observation, a matching process can be devised iteratively, where in each iteration a schema mapping is tested for correctness (*e.g.*, by sending a form to a server). In case of a failure, the next best mapping is being constructed. An algorithm for generating top-K mappings was presented in [1], based on Murty's assignment algorithm [20].

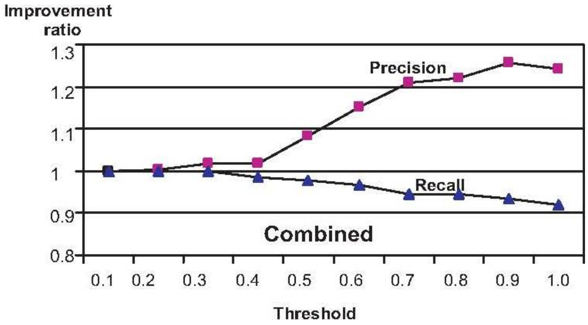

Figure 2: Precision and Recall for Stability analysis with $K = 10$

In [10], another use of top-K mappings was suggested. Previously, we have assumed that a matcher can find the exact mapping. Such an assumption may not be valid. For example, assume that our matching algorithm solves a maximum weight bipartite graph problem to identify mappings. Such an algorithm aims at adding as many attributes as possible to a mapping. However, in many real-world cases, some attributes cannot be mapped. Such a scenario is typically handles by setting a threshold so that attribute mappings with low certainty level cannot be included in a schema mapping. Alas, thresholds are not easily tuned [22]. Therefore, the algorithm proposed in [10] suggests to generate a dynamic threshold by analyzing **simultaneously** a set of top-K mappings. According to our observations, monotonic matchers tend to rank high schema mappings with many correct attribute mappings. Therefore, the algorithm keeps attribute mappings only if they appear a sufficient number of times (a threshold t) in the top-K mappings.

Figure 2 presents the average change to precision and recall for different thresholds over 43 real data pairs, as presented in [10]. K was set to 10. For this data set, precision increases (in general) up to $t = 0.9$ with the increased threshold. Recall demonstrates a monotonic decrease with the increased threshold. Such a phenomenon accords with our initial intuition and is expected for monotonic matchers. A closer look at the amount of improvements reveals that the matcher provides an increase of 25.6% (with $t = 0.9$). As for recall, it decreases by a maximum of 8%.

6. DISCUSSION AND CONCLUSIONS

In this paper we have introduced the principle of monotonicity as a theoretical principal in schema matching and showed the relationships between matcher monotonicity and its ability to generate useful mappings. A few attempts at setting theoretical foundation for schema matching exist in the literature. The seminal work of Melnik and Bernstein [4, 17] has been discussed already in Section 3.

A recent work on representing and reasoning about mappings between domain models was presented in [15]. This work provides a model representation and inference analysis. Managing uncertainty was recognized as the next step on the research agenda in this area and was left open for a future research. Our work fills this gap in providing a model that represents the uncertainty (as an imprecision measure) in the matching process outcome.

Soundness of schema matching methods was discussed in [2]. There, matching correctness was defined using *pragmatic competence*, the ability to make decisions that are sound with respect to the semantics of the problem. The monotonicity principle can be viewed as a method for refining pragmatic competence using quantified methods.

Studying the uncertainty inherent to the schema matching process (which is the "tough" part of schema matching) is an ongoing research task (see, *e.g.*, [16]). More matchers are needed, utilizing top-K mappings. Also, a more refined classification of monotonic matchers is needed, based on aspects such as application domain and the amount of variance of certainty level values within each precision group.

Acknowledgments

Many thanks to Phil Bernstein, George Fletcher, and Amit Sheth for useful comments and discussions.

7. REFERENCES

[1] A. Anaby-Tavor. Enhancing the formal similarity based matching model. Master's thesis, Technion-Israel Institute of Technology, May 2003.

[2] M. Benerecetti, P. Bouquet, and S. Zanobini. Soundness of schema matching methods. In *Proceedings of ESWC 2005*, pages 211–225, 2005.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic Web. *Scientific American*, May 2001.

[4] P.A. Bernstein and S. Melnik. Meta data management. In *Proceedings of the IEEE CS International Conference on Data Engineering*. IEEE Computer Society, 2004.

[5] A. Bilke and F. Naumann. Schema matching using duplicates. In *Proceedings of the IEEE CS International Conference on Data Engineering*, pages 69–80, 2005.

[6] B. Convent. Unsolvable problems related to the view integration approach. In *Proceedings of the International Conference on Database Theory (ICDT)*, Rome, Italy, September 1986. In *Computer Science*, Vol. 243, G. Goos and J. Hartmanis, Eds. Springer-Verlag, New York, pp. 141-156.

[7] H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proceedings of the 2nd Int. Workshop on Web Databases (German Informatics Society), 2002.*, 2002.

[8] H.H. Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of the International conference on very Large Data Bases (VLDB)*, pages 610–621, 2002.

[9] A. Gal. On the cardinality of schema matching. In *IFIP WG 2.12 and WG 12.4 International Workshop on Web Semantics (SWWS)*, pages 947–956, 2005.

[10] A. Gal. Managing uncertainty in schema matching with top-k schema mappings. *Journal of Data Semantics*, 6:90–114, 2006.

[11] A. Gal, A. Anaby-Tavor, A. Trombetta, and D. Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *VLDB Journal*, 14(1):50–67, 2005.

[12] B. He and K.C.-C. Chang. Making holistic schema matching robust: an ensemble approach. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 429–438, 2005.

[13] R. Hull. Managing semantic heterogeneity in databases: A theoretical perspective. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 51–61. ACM Press, 1997.

[14] M. Lenzerini. Data integration is harder than you thought. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, editors, *Cooperative Information Systems, 9th International Conference, CoopIS 2001, Trento, Italy, September 5-7, 2001, Proceedings*, volume 2172 of *Lecture Notes in Computer Science*, pages 22–26. Springer, 2001.

[15] J. Madhavan, P.A. Bernstein, P. Domingos, and A.Y. Halevy. Representing and reasoning about mappings between domain models. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, pages 80–86, 2002.

[16] M. Magnani, N. Rizopoulos, P. McBrien, and D. Montesi. Schema integration based on uncertain semantic mappings. In *ER*, pages 31–46, 2005.

[17] S. Melnik. *Generic Model Management: Concepts and Algorithms*. Springer-Verlag, 2004.

[18] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the IEEE CS International Conference on Data Engineering*, pages 117–140, 2002.

[19] R.J. Miller, L.M. Haas, and M.A. Hernández. Schema mapping as query discovery. In A. El Abbadi, M.L. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, and K.-Y. Whang, editors, *Proceedings of the International conference on very Large Data Bases (VLDB)*, pages 77–88. Morgan Kaufmann, 2000.

[20] K.G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[21] E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[22] M. Sayyadian, Y. Lee, A. Doan, and A. Rosenthal. Tuning schema matching software using synthetic scenarios. In *Proceedings of the International conference on very Large Data Bases (VLDB)*, pages 994–1005, 2005.

[23] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal of Data Semantics*, 4:146 – 171, December 2005.

[24] W. Su, J. Wang, and F. Lochovsky. Aholistic schema matching for web query interfaces. In *Advances in Database Technology - EDBT 2006, 10th International Conference on Extending Database Technology, Munich, Germany, March 26-31, 2006, Proceedings*, pages 77–94, 2006.