A Holistic Paradigm for Schema Matching*

Bin He, Kevin Chen-Chuan Chang
Computer Science Department
University of Illinois at Urbana-Champaign
binhe@uiuc.edu, kcchang@cs.uiuc.edu

Abstract

Schema matching is a critical problem for integrating heterogeneous information sources. Traditionally, the problem of matching multiple schemas has essentially relied on finding pairwise-attribute correspondence. In contrast, we propose a new matching paradigm, *holistic schema matching*, to holistically match many schemas at the same time and find all the matchings at once. By handling a set of schemas together, we can explore their *context* information that reflects the semantic correspondences among attributes, which is not available when schemas are matched only in pairs. As the realizations of the holistic paradigm, we developed two alternative approaches recently. This article takes an initial step to unify those two approaches and further contrasts their strength and weakness. Specifically, we develop two alternative methods for realizing holistic schema matching: *global evaluation* and *local evaluation*. Global evaluation exhaustively assesses all the possible models, where a *model* expresses all attribute matchings. In particular, we propose the MGS framework for such global evaluation with the hypothesis of the existence of generative models. On the other hand, local evaluation independently assesses every single matching to incrementally construct the model. In particular, we develop the DCM framework for such local evaluation with the observation that co-occurrence patterns across schemas often reveal the complex relationships of attributes. We apply our approaches on matching Web query interfaces on the deep Web. The result shows the effectiveness of both the MGS and DCM approaches, which together demonstrate the promise of the holistic paradigm for schema matching.

1 Introduction

Schema matching is fundamental for enabling query mediation and data exchange across information sources [1, 12]. This article proposes a new matching paradigm, *holistic schema matching*, which unifies two alternative approaches we developed recently as its realizations. Traditionally, schema matching has been approached mainly by finding *pairwise-attribute correspondence*, to construct an integrated schema for two or some (small number of) n sources. We observe that there are often challenges (and certainly also opportunities) to deal with large numbers of sources. In such scenarios, the challenge of large scale can itself be an opportunity for new approaches – We can take a holistic view of all the input schemas and find all the matchings at once.

Such scenarios arise, in particular, for integrating databases across the Internet, or the so-called "deep Web." Our recent survey [3] in December 2002 estimated between 127,000 to 330,000 deep Web sources. With the virtually unlimited amount of information, the deep Web is clearly an important frontier for data integration. On this deep Web, numerous online databases provide data via their *query interfaces*, instead of static URL links. Each query interface accepts queries over its *query schemas* (e.g., author, title, subject, ... for *amazon.com*). *Schema matching* (i.e., discovering semantic correspondences of attributes) across Web interfaces is essential for mediating queries across deep Web sources.

However, existing schema matching works mostly focus on small scale integration by finding pairwise-attribute correspondence between two sources. Traditionally, schema matching relies on matchings between pairwise attributes before integrating multiple schemas. For instance, traditional binary or n-ary [10] schema integration methodologies (as [1] surveys) exploit pairwise-attribute correspondence assertions (mostly manually given) for merging two or some n sources. Further, recent works on automatic schema matching mostly focus on matchings between two schemas (e.g., [9, 8]). Based on this fact, the latest survey [11] abstracts schema matching as pairwise similarity mappings between two input sources.

To tackle the challenge of large scale matching, as well as to take advantage of its new opportunity, we propose a new paradigm, *holistic schema matching*, to match many schemas at the same time and find all the matchings at

*This material is based upon work partially supported by NSF Grants IIS-0133199 and IIS-0313260. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

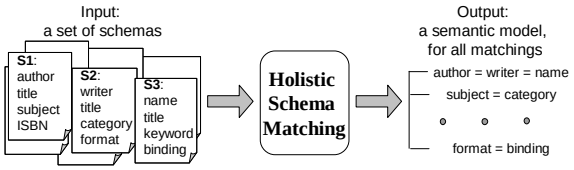

Figure 1: The holistic schema matching paradigm.

once, as Figure 1 shows. In particular, holistic schema matching takes a set of schemas as input and outputs a semantic *model*, which contains all the matchings among the input schemas (e.g., a model of book schemas may contain author = writer = name, subject = category, ...). Such a holistic view enables us to explore the *context* information beyond two schemas (e.g., similar attributes across multiple schemas; co-occurrence patterns among attributes), which is not available when schemas are matched only in pairs.

Compared with traditional approaches, we believe the holistic approach has several advantages: First, *scalability*: By unifying a large number of input schemas holistically rather than matching attributes pairwise, it addresses the scale of matching required in the new frontier of networked databases, such as our motivating goal of the deep Web. Second, *solvability*: In fact, the large scale can itself be a crucial leverage to make schema matching more solvable– in particular, it enables effective exploration of the context information. Such context information will be more sufficient as more sources are exploited. Intuitively, we are building upon the "peer context" among schemas. Being context-based, the holistic matching will benefit from the scale: the accuracy will "scale" with the number of sources. For instance, our specific MGS and DCM approaches are both statistical methods, which will thus benefit from more "observations."

With the holistic paradigm, this article takes an initial step to unify two alternative approaches we developed recently as its realizations. Specifically, to realize holistic schema matching, we develop two different methods with respect to how the semantic model (as Figure 1 introduced) is evaluated: *global evaluation* and *local evaluation*. Global evaluation assesses a model as a whole, while local evaluation incrementally constructs the model.

On one hand, global evaluation exhaustively evaluates all possible models and selects the best one among them. The best model contains the set of matchings with the highest overall confidence to assemble the correct model.

In particular, we develop the MGS framework [5] for such global evaluation by hypothesizing the existence of a hidden generative model for each domain (e.g., Books, Movies) (Section 2). Under this hypothesis, a schema can be viewed as an instance generated from the model with some probabilistic behavior. Schema matching is thus transformed into the discovery of the hidden model, given a set of schema instances. To realize such hidden model discovery, we develop the MGS framework, which discovers matchings with statistical hypothesis testing.

On the other hand, local evaluation independently assesses every single matching and then incrementally constructs the model. Instead of exhaustively enumerating all the possible models, local evaluation approximately searches for the best model by constructing it incrementally. For instance, among all the potential matchings in book schemas, we may first select the most confident matching author = writer = name and consider it as part of the best model. Then we iteratively select the next most confident matching under this partial model result, toward eventually completing the best model.

In particular, we develop the DCM framework [7] for such local evaluation with the observation that co-occurrence patterns across schemas often reveal the complex relationships of attributes (Section 3). Specifically, we observe that *grouping attributes* (e.g., {first name, last name}) tend to be co-present in query interfaces and thus positively correlated. In contrast, *synonym attributes* are negatively correlated because they rarely co-occur. This insight motivates us to develop the DCM framework, which greedily discovers complex matchings with a dual mining of positive and negative correlations.

We compare global evaluation and local evaluation in Section 4. First, we qualitatively discuss their advantages and disadvantages. Second, we apply the MGS and DCM approaches respectively on matching deep Web query interfaces in the same domain (e.g., Books and Movies) and compare their matching accuracy.

The rest of the article is organized as follows: Section 2 briefly presents the MGS framework and Section 3 the DCM framework. Section 4 qualitatively compares global evaluation and local evaluation and their experimental results on matching real query interfaces. Section 5 discusses some open issues that warrant further research and then concludes the paper.

2 Global Evaluation: Matching as Hidden Model Discovery

To realize the global evaluation which finds an overall best model, we hypothesize the existence of the hidden generative behavior of a model. This hidden-model hypothesis provides a principled statistical method, hypothesis testing [2], to evaluate the confidence of a model (as a statistical hypothesis), given a set of schemas as observations. We thus abstract the schema matching problem as hidden model discovery and develop the MGS framework [5] to realize the global evaluation.

In particular, our hidden-model hypothesis is based on two observations: First, we observe *proliferating sources*: As the Web scales, many data sources exist to provide structured information in the same domains, as our survey [3] shows. Second, we also observe *converging vo-*

cabularies: The aggregate schema vocabulary of sources in the same domain tends to converge at a relatively small size. Figure 2 shows, for each domain, the growth of vocabularies as sources increase in numbers. The curves clearly indicate the convergence of vocabularies. Since the vocabulary growth rates (i.e., the slopes of these curves) decrease rapidly, as sources proliferate, their vocabularies will tend to stabilize. This observation indicates that homogeneous sources (in the same domain) share some "concerted" vocabulary of attributes.

These observations lead us to hypothesize the existence of a hidden schema model that probabilistically generates, from a finite vocabulary, the schemas we observed. Intuitively, such a model gives the "structure" of the vocabulary to constrain how instances can be generated. The hypothesis sheds new light on a different way for coping with schema matching: If a hidden model does exist, its *discovery* would reveal the vocabulary structure. Such model-level unification of all attributes in the same domain will subsume their pairwise correspondence (as used in traditional schema matching). We thus propose the hidden model discovery paradigm as the global evaluation for holistic schema matching.

To realize such hidden model discovery, we propose a general framework, MGS, consisting of hypothesis modeling, generation, and selection. We believe the MGS framework is important in its own rights: In principle, by application-specific hypothesis modeling, MGS can be applied to capture different types of semantic relationships. Specifically,

1.*Hypotheses Modeling*: To guide the seeking of a hypothetical model, or a *hypothesis*, we start by defining the general structure of such models. Such modeling should essentially capture specific semantics we want to discover. For instance, if we want to find synonyms, a model should explicitly express the relationship of "synonyms." Such modeling will also specify a generative behavior of how schemas can be generated. Such behavior is mainly *probabilistic* (e.g., attributes will be drawn randomly by their "popularity"), although it can also partially be *deterministic* (e.g., no synonyms can be selected together). Effectively, the model forms a statistical distribution, which generates a particular schema with some *instantiation probability*.

2.*Hypotheses Generation*: We then enumerate concrete hypotheses (in the specified abstract model) that are consistent with the observed schemas (with non-zero probabilities). Note that, even with a parameterized structure, there will be a large space of candidate hypotheses to search, for a vocabulary of reasonable size. This generation step helps to focus the search to only those promising hypotheses that are likely to generate the observed schemas.

3.*Hypotheses Selection*: Finally, we select hypotheses that are consistent with the observed schemas with sufficient statistical significance. There are various statistical devices for such hypothesis testing [2]. For instance, we use χ^2 testing in our MGS$_{ac}$ algorithm.

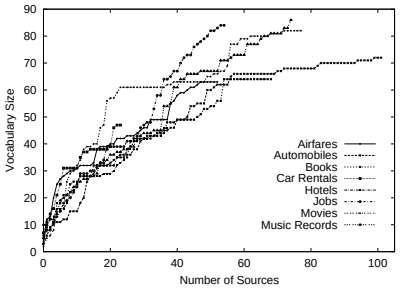

Figure 2: Schema vocabularies over 8 domains.

Guided by the general MGS framework, we develop Algorithm MGS$_{ac}$, specifically for discovering 1:1 synonym matchings. Because of the space limitation, we omit the concrete modeling and algorithm of MGS$_{ac}$ in this article. Please refer to [5] for details.

3 Local Evaluation: Matching as Correlation Mining

Our realization of the local evaluation deals with a more general type of matchings: *complex matching*. In particular, for our focus of the "deep Web," query schemas generally form complex matchings between attribute groups. In contrast to simple 1:1 matching, complex matching matches a set of m attributes to another set of n attributes, which is thus also called m:n *matching*. For instance, in Books domain, author is a synonym of the grouping of last name and first name, i.e., {author} = {first name, last name}; in Airfares domain, {passengers} = {adults, seniors, children, infants}.

As local evaluation aims at "greedily" finding individual matchings (e.g., {author} = {first name, last name}), its realization relies on discovering some properties that indicate such matchings– We pursue a correlation mining approach by exploiting the *co-occurrence* patterns of attributes. Specifically, the holistic view provides the co-occurrence information of attributes across many schemas, which reveals the semantics of complex matchings. For instance, we may observe that last name and first name often co-occur in schemas, while they together rarely co-occur with author. More generally, we observe that *grouping attributes* (i.e., attributes in one group of a matching e.g., {last name, first name}) tend to be co-present and thus positively correlated across sources. In contrast, *synonym attributes* (i.e., attribute groups in a matching) are negatively correlated because they rarely co-occur in schemas.

These dual observations motivate us to develop a correlation mining abstraction of the schema matching problem. Specifically, we view a schema as a *transaction*,

a conventional abstraction in association and correlation mining. In data mining, a transaction is a set of items; correspondingly, in schema matching, we consider a schema as a set of *attribute entities*. An attribute entity contains attribute name, type and domain (i.e., instance values). We develop a dual correlation mining framework, DCM, for mining complex matchings, consisting of three steps: mining positive correlations as groups, mining negative correlations as complex matchings and matching selection as model construction. In this section, we briefly discuss each step. Please refer to [7] for more details.

First, group discovery: We mine *positively correlated attributes* to form potential attribute groups. A potential group may not be eventually useful for matching; only the ones having synonym relationship (i.e., negative correlation) with other groups can survive. For instance, if all sources use last name, first name, and not author, then the potential group {last name, first name} is not useful because there is no matching (to author) needed.

Second, matching discovery: Given the potential groups (including singleton ones), we mine *negatively correlated attribute groups* to form potential complex matchings. A potential matching may not be considered as correct due to coincidental correlations. Specifically, as a statistical approach, correlation mining can discover true semantic matchings and, as expected, also false ones due to the existence of coincidental correlations. For instance, in Books domain, the mining result may have both {author} = {first name, last name}, denoted by M_1 and {subject} = {first name, last name}, denoted by M_2. We can see M_1 is correct, while M_2 is not. The reason for having the false matching M_2 is that in the collected schema data, it happens that subject does not often co-occur with first name and last name.

Third, matching selection for model construction: We develop an iterative selection algorithm to incrementally construct the model by choosing the most confident matching in each iteration. Specifically, the existence of false matchings may cause matching conflicts. For instance, M_1 and M_2 conflict in that if one of them is correct, the other one will not. If both of them are correct, we should be able to also find the matching M_3: {author} = {subject} by the transitivity of synonym relationship. Since our mining algorithm does not find M_3, M_1 and M_2 cannot co-exist in the same model and thus they conflict. Based on this observation, we develop an iterative selection strategy to construct the model: In each iteration, we select the most confident matching as part of the best model and remove the conflicting matchings. By this iterative selection process, we incrementally construct a model with a set of consistent complex matchings.

Intuitively, between conflicting matchings, we want to select the more negatively correlated one because it indicates higher confidence to be synonyms. For example, our experiment shows that, as M_2 is coincidental, it is indeed that M_1 is more negatively correlated than M_2, and thus we select M_1 and remove M_2. With larger data size, semantically correct matching is more possible to be the winner. The reason is that, with larger size of sampling, the correct matchings are still negatively correlated while the false ones will remain coincidental and not as strong.

4 Comparisons

To better understand the characteristics of the MGS framework for global evaluation and the DCM framework for local evaluation, we compare these two approaches in both qualitative and experimental aspects, as Section 4.1 and Section 4.2 will discuss respectively.

4.1 Qualitative Analysis

The global and local evaluation methods both have their pros and cons. On one hand, global evaluation is a more systematic and principled way to evaluate models since it exhaustively evaluates all possible models with statistic basis. In particular, in the MGS framework, the statistical hypothesis testing can report matchings with respect to a given theoretical *significance level*. Also, the discovered model can naturally be employed as a unified schema to mediate queries to specific sources. However, global evaluation can be expensive. The exploration of all the possible models can be generally exponential, as shown in [5]. Further, modeling can be a difficult task, depending on specific target semantics to be discovered. In particular, it is unclear how to extend the modeling in [5] to accommodate complex matchings, which the DCM framework copes with (Section 3).

On the other hand, local evaluation adopts a greedy strategy to incrementally construct a potentially suboptimal model. The greedy selection is not as systematic as the exhaustive enumeration in the global evaluation. Also, as the core of correlation mining, we need to choose an appropriate correlation measure for our application scenario. Since correlation measure is often empirically designed based on heuristics, the mining result may be lack of principled justification for its confidence. However, it does have some advantages. First, the computation of local evaluation is very efficient, since instead of exhaustively exploring every model as a whole, we select one matching at a time as part of the best model. Second, it is easier to accommodate complex matchings in local evaluation since it does not require formal statistical modeling. In particular, the DCM framework [7] supports complex matchings by considering both positive and negative correlations. Finally, our experiments show that the matching accuracy of local evaluation is empirically close to global evaluation in discovering 1:1 matchings.

domain	*the MGS framework*	*the DCM framework*
Books	{author} = {last name} (P) {author} = {first name} (P) {subject} = {category} (Y)	{author} = {last name, first name} (Y) {publisher} = {last name} (N) {subject} = {category} (Y)
Movies	{artist} = {actor} = {star} (Y) {genre} = {category} (Y)	{artist} = {actor} (Y) {genre} = {category} (Y) {rating} = {keyword} (N) {price} = {format} (N)
MusicRecords	{title} = {album} (Y) {artist} = {band} (Y) {genre} = {soundtrack} (N) {keyword} = {catalog} (N)	{title} = {album} (Y) {artist} = {band} (Y) {genre} = {label} (N)
Automobiles	{style} = {type} = {category} (Y) {state} = {mileage} (N) {zip code} = {color} (N)	{style} = {type} = {category} (Y) {state} = {mileage} (N)

Figure 3: Experimental results of the two approaches on the BAMM dataset.

Domain	*Final Output After Matching Selection*	*Correct?*
Airfares	{destination (string)} = {to (string)} = {arrival city (string)}	Y
	{departure date (datetime)} = {depart (datetime)}	Y
	{passenger (integer)} = {adult (integer), child (integer), infant (integer)}	P
	{from (string), to (string)} = {departure city (string), arrival city (string)}	Y
	{from (string)} = {depart (string)}	Y
	{return date (datetime)} = {return (datetime)}	Y
Hotels	{check in (date), check out (date)} = {check in date (date), check out date (date)}	Y
	{check in (date)} = {check in date (date)}	Y
	{check out (date)} = {check out date (date)}	Y
	{type (string)} = {country (string)}	N
	{guest (integer)} = {adult (integer), child (integer), night (integer)}	P

Figure 4: Part of the experimental result of the DCM approach on the TEL-8 dataset.

4.2 Experimental Analysis

We apply the MGS and DCM approaches in our motivating application: matching Web query interfaces on the deep Web, which is a special type of schema matching. Specifically, we choose two datasets, the BAMM dataset and the TEL-8 dataset, of the UIUC Web Integration Repository [4] as the testbed of our work. The BAMM dataset contains manually extracted attribute names over 211 sources in 4 domains (with around 50 sources per domain). The TEL-8 dataset contains raw Web pages over 447 deep Web sources in 8 popular domains. Each domain has about 20-70 sources.

Before matching, we clean the schemas by merging syntactically similar attributes (e.g., "title of book" is merged to "title"). In particular, we conduct a manual syntactic merging for the BAMM dataset and further fully automate this process for the TEL-8 databset by exploiting the syntactic similarity of both attribute names and instance values [7]. This cleaning action serves for two purposes: First, it shows that syntactic merging cannot discover all the matchings, especially the "semantically difficult" ones. For instance, in Movies domain, star and actor are synonyms, but they bear essentially no syntactic similarity in names. Also, both of them are only associated with input boxes and thus have no instance values. As our experiment shows, many popular matchings are indeed "semantically difficult." Second, syntactic merging, by increasing the frequency of merged attributes, can enhance the accuracy of holistic matching approaches. The reason is that as statistical methods, these approaches rely on "sufficient observations" of attribute occurrences and thus they are likely to perform more favorably for frequent attributes.

We run the MGS and DCM approaches on the BAMM dataset, to compare their ability in discovering simple 1:1 matchings. Also, to show the discovery of complex matchings, we test the DCM approach on the TEL-8 dataset. To illustrate the effectiveness of the holistic approaches, in this article, we only list and count the "semantically difficult" matchings discovered by the holistic algorithms, not the "semantically simple" ones by the syntactic merging.

Results on the BAMM Dataset: We report the experimental results of the two approaches on the BAMM dataset, as Figure 3 shows. In particular, Figure 3 lists the discovered matchings for each of the four domains: Books, Movies, MusicRecords and Automobiles. The matching followed by "Y" means a correct matching, "P" a partially correct one and "N" an incorrect one. As we can see from the result, both approaches can discover correct matchings in each domain (e.g., {subject} = {category} in the Books domain). Also, as statistical approaches, they may output some incorrect matchings due to the accidental bias of the data. Further, the DCM approach can find the complex matching {author} = {last name, first name} in the Books domain, while the MGS

approach currently only supports simple 1:1 matching and thus outputs two partially correct ones {author} = {last name} and {author} = {first name}. Therefore, the results empirically show that the local evaluation can achieve closely to the global evaluation with respect to the simple 1:1 matching.

Results on the TEL-8 Dataset: In the BAMM dataset, only one complex matching is observed (i.e., {author} = {last name, first name}). However, in other domains such as Airfares, Hotels, CarRentals, more complex matchings can be found. To show the ability that the DCM approach can really discover complex matchings, we execute it on the 8 domains in the TEL-8 dataset, which contains more complex matchings. Because of the space limitation, we only show the discovered matchings in domains Airfares and Hotels, as Figure 4 shows. (The complete result can be found in [6].) The results show that the DCM approach can find complex matchings in many domains. For instance, in Airfares domain, we find 5 fully correct matchings, e.g., {destination (string)} = {to (string)} = {arrival city (string)}. Also, {passenger (integer)} = {adult (integer), child (integer), infant (integer)} is partially correct because it misses senior (integer). Note that since we incorporate type recognition in [7], the attribute names are followed by their data types in the matchings.

In summary, we can see that both approaches are effective in discovering "semantically difficult" matchings in Web query interfaces, which shows the promise of the holistic way of schema matching.

5 Concluding Discussion

In our study for holistic schema matching, we also observed some open issues that warrant further research. First, we plan to perform more thorough and systematic comparison for the two approaches. In this article, we present the matching result on the BAMM dataset. In the future, we plan to investigate a more systematic comparison. In particular, since the BAMM dataset only covers four domains with 50 sources in each domain, which may not be sufficient for thoroughly comparing the two approaches, a larger dataset with more domains and sources can be considered as the testbed. In addition to accuracy, it is also interesting to compare the two approaches on various other aspects, such as robustness to data noises (e.g., how accurate is the matching result if the query interfaces are not perfectly extracted).

Second, given the respective pros and cons of the global and local evaluations, we wonder if a hybrid of the two approaches will achieve the strength of both without the weakness of either. In particular, our goal is to design a hybrid approach with systematic modeling, efficient execution and expressive semantics. For instance, we can use the result of the local evaluation to prune the search space of the global evaluation. Or, we can use the global evaluation to help the model construction of the local evaluation. Specifically, in each iteration of greedy selection, instead of independently evaluating each potential matching, we can evaluate the confidence of incorporating each potential matching into the existing partial model.

In summary, for large scale integration, our experience indicates high promise for moving the traditional pairwise-attribute correspondence toward a new holistic paradigm. This approach is well suited for the new frontier of massive networked databases, such as the deep Web. This article has proposed this holistic paradigm for schema matching, which unifies the MGS and DCM frameworks as the realization of the global and local evaluations respectively.

References

[1] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323–364, 1986.

[2] P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Prentice Hall, 2001.

[3] K. C.-C. Chang, B. He, C. Li, and Z. Zhang. Structured databases on the web: Observations and implications. Technical Report UIUCDCS-R-2003-2321, Department of Computer Science, UIUC, Feb. 2003.

[4] K. C.-C. Chang, B. He, C. Li, and Z. Zhang. The UIUC web integration repository. Computer Science Department, University of Illinois at Urbana-Champaign. http://metaquerier.cs.uiuc.edu/repository, 2003.

[5] B. He and K. C.-C. Chang. Statistical schema matching across web query interfaces. In *SIGMOD Conference*, 2003.

[6] B. He, K. C.-C. Chang, and J. Han. Automatic complex schema matching across web query interfaces: A correlation mining approach. Technical Report UIUCDCS-R-2003-2388, Dept. of Computer Science, UIUC, Dec. 2003.

[7] B. He, K. C.-C. Chang, and J. Han. Discovering complex matchings across web query interfaces: A correlation mining approach. In *SIGKDD Conference*, 2004.

[8] Y. Lee, A. Doan, R. Dhamankar, A. Halevy, and P. Domingos. imap: Discovering complex mappings between database schemas. In *SIGMOD Conference*, 2004.

[9] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB Conference*, 2001.

[10] S. Navathe and S. Gadgil. A methodology for view integration in logical data base design. In *VLDB*, 1982.

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

[12] L. Seligman, A. Rosenthal, P. Lehner, and A. Smith. Data integration: Where does the time go? *Bulletin of the Tech. Committee on Data Engr.*, 25(3), 2002.