Towards A Unified Framework For Schema Merging

Xiang Li
supervised by Matthias Jarke
Informatik 5 (Information Systems)
RWTH Aachen University
52056 Aachen, Germany
lixiang@dbis.rwth-aachen.de

ABSTRACT

Merging schemas to create a mediated view is a recurring problem in applications related to data interoperability. The task becomes particularly challenging when the schemas are highly heterogeneous and autonomous. Classical data integration systems rely on a mediated schema created by human experts through an intensive design process. Automatic generation of mediated schemas is still a goal to be achieved.

We present a novel logical framework for merging multiple relational schemas related via a collection of mapping constraints in the form of tuple-generating dependencies (tgds), to address the challenge of creating a mediated query interface for data integration systems. The semantics of schema merging is characterized as query answering properties of the output mapping system between the union of the source schemas and the mediated schema. We then focus on a class of mediated schemas that result from transformations of the source schemas and present preliminary results and future work for searching minimal mediated query interfaces.

Our work is a first step towards a unified framework of logical schema merging.

1. INTRODUCTION

Schema merging is the process to consolidate multiple related heterogeneous schemas to provide a unified user view called the *mediated schema*. In order to support data loading from the sources to the mediated schema (e.g., in data warehousing), or to enable querying of sources through the mediated schema (e.g., in data integration [23] or dataspaces [34]), mappings revealing the relationship between the mediated schema and the source schemas have to be established in the merging process. Classical data integration systems [23] nowadays still rely on a mediated schema created by an intensive manual design process by human experts, which is costly and inflexible in a dynamic evolving environment such as dataspaces.

In vision of the importance of schema merging, Merge is proposed as one of the major operators in *Model Management* [6]. As retrospected by Bernstein and Melnik in [8], the original vision of Model Management 1.0 is not semantic but structural, i.e., not relating schema and data. We believe a semantic formalism for schema merging is in need, so that the capabilities and liabilities of a merging process can be assured. Furthermore, a formal foundation is a prerequisite for understanding the expressiveness and tractability of merging, which is inevitable for realizing a model management engine to address real-world data programmability problems. Last but not least, a formalism is also vital for clarification of the semantics of the merging process for different data intensive applications. For example, semantics differ quite a lot in data integration and view integration, since the former aims at creating a mediated query interface while the latter opts for a storage schema.

In this thesis, we present a novel logical framework for merging multiple relational schemas related via a collection of mapping constraints in the form of tuple-generating dependencies (tgds), to address the challenge of creating a mediated query interface for data integration systems. The semantics of schema merging is characterized as query answering properties of the output mapping system between the union of the source schemas and the mediated schema. We then focus on a class of mediated schemas that result from transformations of the source schemas and present preliminary results and future work for searching minimal mediated query interfaces.

The remaining parts of the paper are structured as follows. Our approach is presented in Section 3, after describing preliminaries in Section 2. We present our research agenda in Section 4. State of the art of schema merging is discussed in Section 5. Finally, concluding remarks are given in Section 6

2. PRELIMINARIES

We now describe the language of our mapping formalism and then explain its semantics in the concrete application scenario of virtual data integration systems.

A *tuple generating dependency (tgd)* [1], is a query containment constraint in the form of:$\forall\vec{x}[\exists\vec{y}\phi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}\psi(\vec{x},\vec{z})]$, where ϕ and ψ are conjunctions of atoms and $\vec{x}$, $\vec{y}$ and $\vec{z}$ are mutually disjoint variables. It is *full*, if there are no existential variables on the right hand side, otherwise it is an *embedded* tgd. When the two sides of tgds are defined over two distinct schemas, it is a natural language for binary schema mappings. For a source schema S and a target schema T, a *source-to-target tgd (s-t tgd)* is a tgd such that the antecedent contains only atoms from S and the consequent contains only atoms from T.

The chase procedure [14] is an indispensable tool for reasoning with data dependencies. With Σ being a set of tgds and egds with terminating chase, we use $chase_{\Sigma}(I)$ to denote the result of chase using Σ over database I. When Σ is a set of s-t tgds and target dependencies, we also use $chase(I,\Sigma)$ to denote the target instance J such that $(I,J) = chase_{\Sigma}(I,\emptyset)$. We say the input of schema merging is *with terminating chase* if the union of the data dependencies in the input mapping and in the source schemas always have a terminating chase.

A schema S is a set of relation schemas including key and foreign key constraints. The constraints are formulated as a set of dependencies Σ_S (tgds and equality-generating dependencies (egds) [1]). A mapping between two schemas S_1 and S_2 is a triple $\mathscr{M} = (S_1, S_2, \Sigma)$ where Σ is a set of dependencies, i.e., egds and tgds. The semantics of a binary mapping is a binary relation with instances of the source schema as the domain and instances of the target schema as the range, i.e., $Inst(\mathscr{M}) = \{(I,J) : I \in Inst(S_1) \wedge J \in Inst(S_2) \wedge (I,J) \models \Sigma\}$. The semantics of the composition of two mappings $\mathscr{M}_{13} = \mathscr{M}_{12} \circ \mathscr{M}_{23}$ is then defined as $Inst(\mathscr{M}_{13}) = \{(I,K) : \exists J\ (I,J) \in Inst(\mathscr{M}_{12}) \wedge (J,K) \in Inst(\mathscr{M}_{23})\}$. For a mapping $\mathscr{M}$ from S to T, the possible worlds, called *solutions*, of T wrt. an instance I of S are $Sol_{\mathscr{M}}(I) = \{J \in Inst(T) : (I,J) \in Inst(\mathscr{M})\}$. We denote it by $Sol(I)$ when the mapping involved is clear from context.

An incomplete database I with schema S and dependencies Σ is a representation of a set of possible worlds. Given an instance I of schema S, the *semantics* of I wrt. a set of data dependencies Σ over S is $Sem_\Sigma(I) = \{I' : I' \in Inst(S) \wedge I \subseteq I' \wedge I' \models \Sigma\}$. When the dependencies are clear from context, we simply write $Sem(I)$ for brevity.

The *certain answer* of a query q wrt. a set of database instances P over the same schema is: $certain(q,P) = \bigcap_{I \in P} q(I)$. Equivalence of two sets of databases is then defined using certain answers. Two sets of database instances P_1 and P_2 under the same schema are said to be *L-equivalent* wrt. a query language class L, denoted by $P_1 \equiv_L P_2$, if for any query in L they have the same certain answer. We are interested in equivalence wrt. conjunctive queries, i.e., CQ-equivalence.

Given a collection of schemas S_1, S_2, ..., S_n without repeating relation names, the *joint source schema* is then $S = \bigcup_i S_i$. Let I_i be the extension of S_i, the *joint source instance* is then a concatenation $I = (I_1, I_2, \ldots, I_n)$. Given a collection of schemas S_1, S_2, ..., S_n, local constraints Σ_C and mapping constraints Σ_i among the sources, the semantics of a joint source instance I is $Sem_{\Sigma_C \cup \Sigma_i}(I)$. Therefore, a *merge input* is a pair (S, Σ), where S is the joint source schema and Σ is the union of source ICs Σ_C and the input mapping constraints Σ_i. The schema merging is stated as: $\mathscr{M}_o = (S, G, \Sigma_o) = Merge(S, \Sigma)$, where G is the mediated schema, Σ_o is a set of data dependencies, and $\mathscr{M}_o$ is the output mapping between S and G.

3. A LOGICAL FRAMEWORK

In this section, we describe our framework for schema merging. Semantic characterization of schema merging is discussed in Section 3.1. We introduce in Section 3.2 a main component of our algorithm, i.e., deciding whether the desired properties are retained after schema transformation. The schema minimization procedure using the property retainment test is then presented in Section 3.3. Finally, an illustrative example is provided in Section 3.4.

3.1 Characterizing Mediated Query Interfaces

The first issue of merging schemas is what we would like to use the merged schema for. We set up our problem in the scenario of creating a mediated query interface for users so that they can query all the data in the sources in a seamless way.

3.1.1 Retaining Certain Answers

It is natural to require an automatically merged schema to retain all the information in the sources, called completeness in [5]. Most of the approaches that do not emphasize generating output mappings fulfill completeness by retaining all attributes of source schemas [5, 35, 29, 22, 30, 32]. Few [10, 31] achieve this goal by ensuring that all source data are retained via the output mapping supporting the mediated schema.

For incomplete data sources with mapping constraints, retaining only extensional data is not enough, since more query answers may result from the constraints. Therefore, we deem it necessary that the output mapping system retain all certain answers.

Definition 1. Given a mapping language L, source schemas S with data dependencies Σ, and a mediated schema G with output mapping $\mathscr{M}_o = (S, G, \Sigma_o)$, a *witness mapping* is a mapping $\mathscr{M}_w = (G, S, \Sigma_w)$ with Σ_w specified in L such that for any source instance $I \in Inst(S)$, we have: $Sem_\Sigma(I) \equiv_{CQ} Sol_{\mathscr{M}_o \circ \mathscr{M}_w}(I)$.

Completeness of a mediated schema is defined using the witness mapping.

Definition 2. An output mapping $\mathscr{M} = (S, G, \Sigma_o)$ is *complete* wrt. a mapping language L if there exists a witness mapping $\mathscr{M}_w = (G, S, \Sigma_w)$ with Σ_w specified in L.

In contrast, completeness in [31] is requiring extensions of source relations are retained, while we require all certain answers of CQs are retained.

3.1.2 Mix Equivalent Data

Input mapping in the form of query containment constraints are guiding clues how the schema and the underlying data should be merged. [26] capture the intuition by enforcing the composition of $\mathscr{M}_{13}$ and the inverse of $\mathscr{M}_{23}$ is equivalent to the input mapping constraints, when merging M_1 and M_2 to M_3 and generating supporting mappings M_{13} and M_{23} from each source schema to the mediated schema. Unfortunately, it is known that the inverse of mappings in even s-t tgds usually does not exist [18, 21].

We formulate the requirement of reflecting knowledge encoded in the input mapping constraints in term of logical properties of the output mapping system.

Definition 3. An output mapping system $\mathscr{M}$ is *integrated*, if for any joint source instance I, $Sol_{\mathscr{M}}(I)$ and $Sol_{\mathscr{M}}(Sem(I))$ are CQ equivalent.

The integratedness property requires that an incomplete source behaves the same as its semantics regarding query answering. For arbitrary schema mappings, the property is undecidable. As described later, integratedness is ensured in our algorithm in a model theoretic manner.

3.1.3 Smaller Is Better

A main goal of schema merging is to create a unified query interface for multiple data sources and hence reduce metadata chaos [13]. We take the assumption that a mediated schema with less redundancy is better, i.e., smaller is better. In order to avoid comparing heterogeneous schemas supported by various output schema mappings, we focus on minimality wrt. a given output mapping.

Minimality states that the output mapping system cannot be transformed via a family of mappings to achieve another output mapping with a smaller mediated schema without losing a given property. In schema merging, we are interested in properties such as completeness and integratedness raised earlier.

Definition 4. Let $\mathscr{P}$ be a class of output mapping systems determined by some property, an output mapping system $\mathscr{M} = (S, G, \Sigma)$ is *minimal* wrt. a family of transformation mappings $\mathscr{T}$ if there does not exist a mapping $\mathscr{M}_t \in \mathscr{T}$ such that the composition of $\mathscr{M}$ and $\mathscr{M}_t$ is in $\mathscr{P}$.

The family of mappings $\mathscr{T}$ determine the search space for candidate output mappings. For creating smaller query interface, we are interested in transformations that strictly reduce the size of the signature of the mediated schema.

3.2 Property Retainment via Transformations

It is easy to show that completeness and integratedness are undecidable for arbitrary schema mappings via reduction from datalog boundedness and datalog containment, respectively. Therefore, it does not make sense to insist on creating arbitrary output mappings for schema merging. We follow an approach in which an initial output mapping satisfying the desired properties is created, and then successive testing whether the properties are retained after transformation is performed.

3.2.1 Canonical Mediated Schema

It is always possible to construct a particular output schema mapping that is both complete and integrated, which we call the *canonical output mapping*. For a given merge input (S, Σ), we create the canonical output mapping as follows:

1. Create target schema G as a replica of S, and denote by Σ_{copy} the copy s-t tgds from S to G.

2. Let ρ be the renaming of predicates from S to the copied ones in G, construct the target dependencies over G as $\Sigma_G = \rho(\Sigma_i \cup \Sigma_S)$.

3. The canonical output mapping is $\mathscr{M}_o = (S, G, \Sigma_{copy} \cup \Sigma_G)$.

PROPOSITION 1. *The canonical output mapping is complete wrt. full s-t tgds and integrated.*

Taking the canonical output schema as a starting point is quite natural, since it incorporates all the information in the joint source schema. We have to point out that almost all the existing approaches to schema merging start from the original source schemas and then carry out transformations.

Interestingly, Melnik describes in [25] a straightforward algorithm creating a mediated schema for view integration, which differs from our canonical mediated schema only in the direction of the output mapping. However, signature size does not matter for view integration and hence they do not head for minimization of the mediated schemas.

3.2.2 Retainment of Properties

Given a schema mapping $\mathscr{M}$ with a property P, the problem of property retainment for a given transformation mapping $\mathscr{M}_t$ is to decide whether $\mathscr{M} \circ \mathscr{M}_t$ still has the property P.

The following proposition states that when the transformation mappings are specified by a set of s-t tgds, target egds, and target tgds with terminating chase, retainment of integratedness is always satisfied.

PROPOSITION 2. *Let $\mathscr{M}$ be an integrated schema mapping that admits universal solutions and $\mathscr{M}_t$ be a transformation mapping specified in a finite set of s-t tgds and a set of target tgds and egds with terminating chase, then $\mathscr{M} \circ \mathscr{M}_t$ is also integrated.*

As preliminary work, we studied in [24] the retainment testing for completeness, when the transformation mappings are confined to projections. We show that completeness retainment is equivalent to preserving answers to a finite set of queries after projection. This is due to the fact that any ground solution of the canonical output mapping is essential, i.e., the core of some source instance.

For a given projection mapping $\mathscr{M}_p$ and a given query class L, a query q is recoverable wrt. a set of dependencies Σ, if there exists another query $q' \in L$ so that for each legal database I wrt. Σ, $q'(\mathscr{M}_p(I)) = q(I)$. Given a projection $\mathscr{M}_p$ of the canonical mediated schema G with dependencies Σ_G and a witness mapping $\mathscr{M}_w$, the completeness retainment test returns true if for each source relation $R \in S$, unfolding of its identity query $q(\vec{x}) \leftarrow R(\vec{x})$ against $\mathscr{M}_w$ is recoverable wrt. Σ_G and UCQs. It is shown in [24] that completeness is retained after projection if and only if the test returns true.

The above implies the following result

THEOREM 1. *Retainment of completeness is decidable wrt. projections for the canonical output mapping.*

It remains open how retainment of completeness can be extended to allow more expressive transformation mappings. For instance, collapsing is another popular transformation employed in existing approaches to schema merging.

3.3 Minimization

With the property retainment test at hand, we can describe the algorithmic framework of our approach. First, we create the canonical output mapping and add it to a processing queue. For each candidate on the processing queue, we enumerate all mediated schemas reachable by transformations and put those that retain the desired properties to the processing queue. If no transformation can be applied to a candidate, then report it as a minimal mediated schema. The termination of the above algorithm relies heavily on the nature of the family of transformation mappings. In the scenario of creating mediated query interfaces, we head for a minimal signature. Therefore, we use the transformations that strictly reduce the size of the mediated schema. The monotony of the transformations ensures termination of the minimization process.

In [24], we have dealt with the minimization wrt. projection. This is effectively achieved due to the fact that composition of consecutive projections is equivalent to a single projection. Therefore, we are able to find minimal output mapping for a given starting point by exploiting the projection dropping the largest number of attributes.

However, it remains open how to minimize a mediated schema when collapsing is admitted. The challenges lie in several aspects. First, the test for completeness retainment has to be extended to allow for collapsing. Second, intermediate output mappings resulting from composition of transformations are no longer expressible as standard schema mappings, since composition of embedded s-t tgds is involved. A possible solution for the second challenge is to use s-t second order dependencies (s-t SO dependencies) [4] as the mapping language, which admits a chase procedure. Therefore, the second challenge reduces to extend the completeness retainment test to s-t SO dependencies.

Moreover, it is of interest how to efficiently enumerate or how to prune the candidate space, so as to avoid unnecessary logical reasoning. In [24], we propose an A-priori variant to avoid duplicate testing of projections. A challenge to be addressed is how to efficiently prune the search space, when both projection and collapsing are in play.

3.4 An Illustrative Example

We describe in Fig. 1 a simple merging scenario adapted from [31]. The corresponding foreign key constraints for these relationships are indicated by $\subseteq$-lines, while keys are underlined. In addition, we show the value correspondences of the attributes of the schemas in dashed lines. The mapping constraint can be stated as

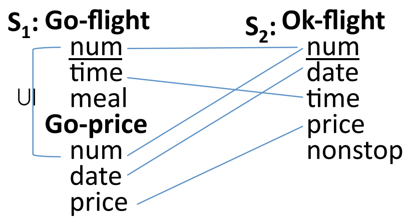

Figure 1: An Illustrative Example

$$\text{Go-flight}(N,T,M), \text{Go-price}(N,D,P) \leftrightarrow \text{Ok-flight}(N,D,T,P,N)$$

We write "$\leftrightarrow$" as a shorthand indicating equivalence of two queries, i.e., two query containment constraints.

As the starting point the canonical mediated schema is created, resulting in a replica of the source relations (Go-flight$_c$, Go-price$_c$ and Ok-flight$_c$), while all data dependencies are taken as target constraints on the mediated schema. We then perform schema minimization wrt. projections. First, we find all size-one projections that can be performed without losing query answers. These include all attributes of Go-price$_c$ and *date*, *time* and *price* of Ok-flight$_c$. Larger possible projections are generated bottom-up in an a-priori style to avoid unnecessary reasoning. A candidate projection is feasible if it retains completeness (see below). The above example has four minimal mediated schemas, each corresponding to a maximal projection. One mediated schema with minimal signature size includes two relations $Go-flight_m(num, time, meal)$ and $Ok-flight_m(num, date, price, nonstop)$, with $Go-price_c$ and the *time* attribute of $Ok-flight_c$ projected. The output mapping is then the composition of the canonical output mapping and the projection, which we omit here for brevity.

We now show how to test retainment of completeness against projections, considering the projection removing the whole relation Go-price$_c$. The retainment test succeeds if each relations's identity query is recoverable after projection. We showcase testing whether the identity query of Go-price$_c$ is still recoverable. We first chase the query against the dependencies Σ, resulting in a query $q = \pi_{num,date,price}(Go-flight_c \bowtie Go-price_c \bowtie Ok-flight_c)$. Then we freeze the body of q as a database. We perform the given projection on the frozen database and then expand the frozen database wrt. the projection, i.e., introducing a fresh variable for each appearance of the projected attributes. The expanded instance is then chased again against Σ. Finally, we test whether the frozen head of q is contained in the answer of evaluating q over the result instance obtained earlier. The query under test is recoverable as the above test succeeds.

4. RESEARCH AGENDA

The research agenda addresses four areas: 1) extending the approach to include collapsing; 2) extending the framework to take inconsistent databases into consideration; 3) building a prototype; and 4) evaluation.

In the near future, we will incorporate collapsing of relations into our framework, which together with projection provides a fairly large candidate space covering almost all the possibilities studied in existing approaches.

- The output mapping language need to be extended to s-t SO dependencies [4], which is shown to be the right language for composition of standard schema mappings.
- A completeness retainment test for collapsing has to be established.
- An efficient algorithm enumerating candidates is to be created.

We are building a schema merging prototype, which is able to merge n-ary schema interrelated by weakly acyclic tgds.

- Since we are using n-ary mappings among schemas, we are developing viewers for illustrating n-ary mappings intuitively.
- Query answering over the mediated schema is also under development in the prototype. Query answering using a materialized approach as in data exchange [19] always suffices for egds and tgds with terminating chase.
- Query rewriting over the mediated schema is more intricate. When the input mapping is weakly acyclic, we can extend the inverse rule algorithm to obtain rewritings. Without the presence of egds, we can rewrite a CQ over the mediated schema into a datalog program without functions, using an extension of the predicate-split technique described in [17]. In presence of egds and full tgds, the technique using datalog rules for simulating chasing of egds [17] is still applicable. For the most general case, when embedded tgds and egds are involved, to the best of our knowledge, there is no rewriting technique known. When the input mapping is not weakly acyclic (but with terminating chase, e.g., stratified), negation might be necessary.

In all the previous sections, we assume that the data sources are consistent, i.e., there exists a finite model for the given merge input. However, real world data is dirty. In absence of egds (e.g., keys), termination of chase guarantees the existence of a finite model for the joint source. It becomes more intricate when egds are present in the sources. When egds are violated, there are basically two strategies. The first one is to take source constraints as local and hence do not make use of them in merging. After deciding which source constraints are to be excluded, the input can be simply fed to the merging algorithm. The second strategy is to follow consistent query answering [9].

The evaluation plan includes the data set selection and metrics to be measured.

- Data set: the Illinois Semantic Integration Archive (`http://pages.cs.wisc.edu/~anhai/wisc-si-archive/summary.type.html`) contains a collection of real world schemas with matches, which can be used for testing effectiveness and expressiveness of our merging algorithm over real world data. As a complement, STBenchmark [2] configurable mapping scenario generation tool which can be used for generating sample inputs for our framework.
- We will investigate the expressiveness of input mappings of our merging algorithm relative to real world data sets. Query answering and/or rewriting feasibility is to be tested to prove

the usefulness in data integration. Reduction ratio measures how much duplication is reduced on the schema level, and is taken as an indication of a quality metric. For those inputs with a referential mediated schema, comparison is to be carried out to reveal the effectiveness over real world data. We are also going to perform experiments on the scalability of our algorithm over variously sized input.

5. RELATED WORK

Classical view/database integration approaches make up a rich line of work [5, 11, 35, 29, 22]. As the integration tasks are usually carried out in a schema design scenario, schemas schemas are usually represented in a variant of ER model or Object-Oriented model. The approaches differ quite a lot on how inter-schema relationships between source schemas are represented. So called inter-schema assertions [35, 32] are a popular language specify set-based relationships (e.g., inclusion, disjoint, and equal) between possible extensions of concepts in different schemas. The approaches usually undergo two steps: first collapse equivalent elements in the source schemas and then resolve the conflicts arising in collapsing. Spaccapietra et al. [35] propose a well known representative algorithm, which is able to handle a wide class of structural conflicts. Model management 1.0 [6] has seen several approaches operating on general structures [30, 32], independent of the modeling languages. A common shortcoming of this line of work is that no output mapping in the form of logical view definitions are generated, although attribute correspondences between source schemas and target schemas are an implicit result.

Schema merging using expressive logical mappings are considered to be largely unexplored [15, 7]. [10] and [12] are pioneers using logical constraints in merging. Similar to us, they consider source integrity constraints and head for a minimal mediated schema. However, their input mapping language is a special class of mappings in the form of one-to-one relation-wise implications and keys are required to be present in each implication. Our approach can be deemed as an extension of their work in the sense that we consider tuple generating dependencies which is much more expressive. Another distinction is that we consider source incompleteness, while they do not.

In [31], Pottigner and Bernstein extend their early work [30] to a merging algorithm working with relational schemas and generating output mappings. Similar to us, they also head for a minimal signature for mediated schemas in data integration. Our approach differs from theirs in several ways. First, their input mapping language is a special class of GLAV mappings using conjunctive queries to specify overlap between schemas. In contrast, we consider arbitrary query containment constraints in the form of tuple generating dependencies with the only restriction that they admit a terminating chase. Second, their merging semantics is based on preserving source information and overlap. Since we do not have the concept of overlap, there is no overlap preservation in our requirements for schema merging. They assume that the extensions of the sources do not conform to any direct constraints and hence their completeness requirement is based on preserving all extensionally stored data. In contrast, we consider source incompleteness as a basic assumption and take the input mapping as expected constraints over the integrated global database. Therefore, our completeness requires preserving not only extensional data but also inferred data in the form of certain answers. Third, the queries used in their approach to witness the complete preserving of source information do not contain joins, i.e., over a single relation, while our approach uses conjunctive queries over the mediated schema to reconstruct source information, which probably leads to smaller mediated schemas. Last but not least, source integrity constraints made use of in our approach are not exploited in theirs.

As clarified in [27], view integration is a closely related but semantically different problem from data integration. View integration aims at creating a backend storage schema supporting source the schemas as views, which results in quite different requirements on the merging algorithm. Melnik [25] proposes a straightforward algorithm for view integration of logical schemas. The mediated schema is taken to be a disjoint union of the source schemas, with source dependencies and input mapping encoded as constraints. Output mappings are identity mappings copying part of the mediated schema to a corresponding source schema. Arenas et al. [3] extend the work to achieve a smaller instance for the mediated schema by adding denial constraints. The two works differ fundamentally from our work in that they head for creating a backend storage schema to support the views satisfying the input mapping. That's why their output mapping is from the mediated schema to the source schemas while we create a mapping from the sources to the mediated schema. They are more concerned with creating a smaller mediated instance of the mediated schema while the signature's size is insignificant. To the contrary, we aim at generating a minimal query interface instead of a minimal instance.

It is widely observed that multiple plausible mediated schemas may co-exist for a given input [13, 34, 33]. Chiticariu et al. [13] propose an interactive schema merging approach using schema matches as input. Concepts are extracted from logical schemas and each possible configuration of concept collapsing results in a plausible mediated schema. The space of plausible collapsing of concepts is then navigated by the user in an interactive manner. Since each extracted concept has a particular join path in the source schemas, two concepts and value correspondences between them comprise an implicit GLAV mapping. Following this point of view, a schema match is a representation of a collection of uncertain mapping constraints. The work is extended in [33] to generate only most desired top-k mediated schemas, which reduces the cost in the interactive exploration of the candidates. A mediated schema is considered more desired if it collapses concepts with higher similarity or sub-element coverage. Sarma et al. [34] is another uncertain schema merging approach taking schema matches as input. They represent alternative mediated schemas as a probability distribution over different clustering of attributes. A probabilistic mapping [16] is produced for each possible mediated schema. In our approach, the input mapping language is in logical constraints and bears no uncertainty. However, we still capture the uncertainty that the same piece of information can be structured in different ways by admitting multiple minimal mediated schemas.

Our approach is heavily influenced by the work in data exchange, e.g., set based semantics for mapping composition, focused on weakly acyclic tgds, and considering certain answers and solutions of schema mappings. However, the witness mapping raised in our approach is not the same as an inverse mapping considered for inverting schema mapping. In fact, inverse mappings do not exist for all weakly acyclic tgds. The witness mapping used in our work to verify the property of the mediated schema may better be analyzed using the theory of schema mapping equivalence [20]: the composition of the output mapping and the witness mapping is CQ-equivalent to the mapping (S, T, Σ) where T is a replica of S, and Σ is all the data dependencies.

Another related area is view determinacy [28], which decides whether a given set of views is able to answer a query. However, we study for a given set of source schemas with integrity constraints and inter-schema mappings, whether a mediated schema is able to retain all certain answers of all possible queries for the possible worlds of the source.

6. CONCLUSION

In this proposal, we have made a first step towards a unified logical framework for schema merging. We have presented a general logical framework for schema merging taking tuple generating dependencies and source integrity constraints as input. We have provided a characterization of the semantics of schema merging as properties of the output mapping system under Open World Assumption in a concrete scenario, namely creating a mediated query interface for data integration systems. We present an approach which takes an initial complete and integrated output mapping as a starting point, and then minimize the mediated schema while retaining desired properties of the output mapping system. Our preliminary result when transformations are projections is described. A research agenda for the dissertation is also presented.

Acknowledgements: The work is supported by the Research Cluster on Ultra High-Speed Mobile Information and Communcation UMIC (`www.umic.rwth-aachen.de`). The author thanks the anonymous reviewers, Christoph Quix and Sandra Geisler for providing helpful comments on improving the paper.

7. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
[2] B. Alexe, W. C. Tan, and Y. Velegrakis. STBenchmark: towards a benchmark for mapping systems. *PVLDB*, 1(1):230–244, 2008.
[3] M. Arenas and et al. Foundations of schema mapping management. In *PODS*, 2010.
[4] M. Arenas, R. Fagin, and A. Nash. Composition with target constraints. In *ICDT*, 2010.
[5] 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.
[6] P. A. Bernstein, A. Y. Halevy, and R. Pottinger. A vision for management of complex models. *SIGMOD Record*, 29(4):55–63, 2000.
[7] P. A. Bernstein and H. Ho. Model management and schema mappings: Theory and practice. In *Proc. VLDB*, pages 1439–1440, 2007.
[8] P. A. Bernstein and S. Melnik. Model management 2.0: Manipulating richer mappings. In *SIGMOD*, pages 1–12, 2007.
[9] L. E. Bertossi. Consistent query answering in databases. *SIGMOD Record*, 35(2):68–76, 2006.
[10] J. Biskup and B. Convent. A formal view integration method. In *SIGMOD*, pages 398–407, 1986.
[11] P. Buneman, S. Davidson, and A. Kosky. Theoretical aspects of schema merging. In *Proc. EDBT*, volume 580 of *LNCS*, pages 152–167. Springer, 1992.
[12] M. A. Casanova and V. M. P. Vidal. Towards a sound view integration methodology. In *PODS*, pages 36–47, Atlanta, GA, 1983. ACM.
[13] L. Chiticariu, P. G. Kolaitis, and L. Popa. Interactive generation of integrated schemas. In *Proc. SIGMOD*, pages 833–846, 2008.
[14] A. Deutsch, A. Nash, and J. Remmel. The chase revisited. In *PODS*, pages 149–158, 2008.
[15] A. Doan and A. Y. Halevy. Semantic integration research in the database community: A brief survey. *AI Magazine*, 26(1):83–94, 2005.
[16] X. L. Dong, A. Y. Halevy, and C. Yu. Data integration with uncertainty. *VLDB J.*, 18(2):469–500, 2009.
[17] O. M. Duschka, M. R. Genesereth, and A. Y. Levy. Recursive query plans for data integration. *Journal of Logic Programming*, 43(1):49–73, 2000.
[18] R. Fagin. Inverting schema mappings. *ACM Transactions on Database Systems*, 32(4), 2007.
[19] R. Fagin, P. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. *Theoretical Computer Science*, 336:89–124, 2005.
[20] R. Fagin, P. G. Kolaitis, A. Nash, and L. Popa. Towards a theory of schema-mapping optimization. In *PODS*, pages 33–42, 2008.
[21] R. Fagin, P. G. Kolaitis, L. Popa, and W. C. Tan. Quasi-inverses of schema mappings. *ACM Trans. Database Syst.*, 33(2), 2008.
[22] J. A. Larson, S. B. Navathe, and R. Elmasri. A theory of attribute equivalence in databases with application to schema integration. *IEEE Trans. Softw. Eng.*, 15(4):449–463, 1989.
[23] M. Lenzerini. Data integration: A theoretical perspective. In *PODS*, pages 233–246, 2002.
[24] X. Li, C. Quix, D. Kensche, and S. Geisler. Merging schemas using mapping constraints over incomplete sources. Submitted for publication.
[25] S. Melnik. *Generic Model Management: Concepts and Algorithms*, volume 2967 of *LNCS*. Springer, 2004.
[26] S. Melnik, P. A. Bernstein, A. Y. Halevy, and E. Rahm. Supporting executable mappings in model management. In *Proc. SIGMOD Conf.*, pages 167–178. ACM Press, 2005.
[27] R. J. Miller, Y. E. Ioannidis, and R. Ramakrishnan. The use of information capacity in schema integration and translation. In *Proc. VLDB*, pages 120–133. Morgan Kaufmann, 1993.
[28] A. Nash, L. Segoufin, and V. Vianu. Determinacy and rewriting of conjunctive queries using views: A progress report. In *Proc. ICDT*, volume 4353 of *LNCS*, pages 59–73, 2007.
[29] C. Parent and S. Spaccapietra. Issues and approaches of database integration. *Communications of the ACM*, 41(5):166–178, 1998.
[30] R. Pottinger and P. A. Bernstein. Merging models based on given correspondences. In *VLDB*, pages 826–873, 2003.
[31] R. Pottinger and P. A. Bernstein. Schema merging and mapping creation for relational sources. In *Proc. EDBT*, 2008.
[32] C. Quix, D. Kensche, and X. Li. Generic schema merging. In *Proc. CAiSE'07*, volume 4495 of *LNCS*, pages 127–141, 2007.
[33] A. Radwan, L. Popa, I. R. Stanoi, and A. A. Younis. Top-k generation of integrated schemas based on directed and weighted correspondences. In U. Çetintemel, S. B. Zdonik, D. Kossmann, and N. Tatbul, editors, *SIGMOD Conference*, pages 641–654. ACM, 2009.
[34] A. D. Sarma, X. Dong, and A. Y. Halevy. Bootstrapping pay-as-you-go data integration systems. In *SIGMOD*, pages 861–874, 2008.
[35] S. Spaccapietra, C. Parent, and Y. Dupont. Model independent assertions for integration of heterogeneous schemas. *VLDB Journal*, 1(1):81–126, 1992.