A Model for Schema Integration in Heterogeneous Databases

Avigdor Gal*
Technion — Israel Institute of Technology, Israel
avigal@ie.technion.ac.il

Alberto Trombetta
Università dell'Insubria, Italy
alberto.trombetta@uninsubria.it

Ateret Anaby-Tavor
Technion — Israel Institute of Technology, Israel
ateret@techunix.technion.ac.il

Danilo Montesi†
Università di Camerino, Italy
danilo.montesi@unicam.it.

Abstract

Schema integration is the process by which schemata from heterogeneous databases are conceptually integrated into a single cohesive schema. In this work we propose a modeling framework for schema integration, capturing the inherent uncertainty accompanying the integration process. The model utilizes a fuzzy framework to express a confidence measure, associated with the outcome of a schema integration process. In this paper we provide a systematic analysis of the process properties and establish a criterion for evaluating the quality of matching algorithms, which map attributes among heterogeneous schemata.

*Partially supported by Technion V.P.R. Fund - New York Metropolitan Research Fund and the IBM Faculty Award for 2002/2003 on "Self-Configuration in Autonomic Computing using Knowledge Management." Also, supported by the Ministry of Science, Culture, and Sport in Israel and by the CNR in Italy.

†Partially funded by MURST Project "Algorithms to index and query semistrucrured data" RSO ex-60% and by MIUR as part of the SAHARA Project. Also, partially supported by the Ministry of Science, Culture, and Sport in Israel and by the CNR in Italy.

1 Introduction and motivation

Schema integration is the process by which schemata from heterogeneous databases are conceptually integrated into a single cohesive schema. In this work we propose a modeling framework for schema integration, capturing the inherent uncertainty accompanying the integration process. We assert that the proposed formal model provides a solid foundation for analyzing the quality of a schema integration process. To substantiate our claim, we provide a systematic analysis of the process properties and establish a criterion for evaluating the quality of matching algorithms, which map attributes among heterogeneous schemata. This criterion (dubbed *monotonicity*) demonstrates the usefulness of the model and can serve in a com-

parative empirical analysis of various algorithms. Our research is motivated by the shift from manual schema integration, as was proposed in [23, 15] to semiautomatic schema integration [12] and fully automatic schema integration [18]. The latter is of particular importance in supporting the reasoning capabilities of software agents in the Semantic Web. The proposed model, to be given in details in Section 2, utilizes a fuzzy framework to model a confidence measure, associated with the outcome of a schema integration process. For example, given two attribute sets $\mathcal{A}$ and $\mathcal{A}'$, the model associates a similarity measure, normalized between 0 (total dissimilarity) and 1 (equivalence) with any mapping among attributes of $\mathcal{A}$ and $\mathcal{A}'$. Therefore, given two attributes $A \in \mathcal{A}$ and $A' \in \mathcal{A}'$, A and A' are *μ-similar* (denoted $A \backsim_{\mu_{att}} A'$), specifying a confidence measure for the mapping. We assume that a manual matching is a perfect process, resulting in a *crisp* mapping, with $\mu_{att} = 1$. As for automatic matching, a hybrid of algorithms, such as those presented in [7, 18, 21] or adaptation of relevant work in proximity queries (*e.g.*, [5, 2]) and query rewriting over mismatched domains (*e.g.*, [6]) can determine the level of μ_{att}. Identifying a similarity measure μ, in and by itself, is insufficient for matching purposes. One may claim, and justly so, that the use of syntactic means to identify semantic equivalence, may be misleading in that a mapping with a high μ can be less precise, as conceived by an expert, than a mapping with a lower μ. We therefore propose a family of "well-behaved" mappings (termed *monotonic mappings*), for which one can safely interpret a high similarity measure as a good semantic mapping. An immediate consequence of this result is the establishment of a corroboration for the quality of mapping techniques, based on their capability to generate monotonic mappings.

Despite a vast body of research on heterogeneous schemata matching (MOMIS [3], DIKE [22], Clio [20], Cupid [18], and OntoBuilder [21], to name a few), there is sparse academic literature on appropriate evaluation tools for proposed algorithms and matching methods in this area. A recent work on representing mappings between domain models was presented in [17]. 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 framework that models and enables reasoned analysis of uncertainty. In [19], a model for estimating information loss in a matching process was introduced. The model computes precision and recall of substitutions of terms in a generalization-specialization hierarchy. The proposed metrics (and their combination, as suggested in [19]) serve as alternatives to the μ-similarity measure we propose in this paper. However, no evaluation of the correspondence of these measures with the "goodness" of the mapping, as perceived by an expert, are available. Our work shows that μ-similarity can be correlated with mapping quality. Our approach was inspired by works of Fagin [10], who proposed a method of combining answers to queries over different data sources using simple fuzzy set theory concepts and a method for allowing users to weight different parts of their queries. This work extends im-

precision to metadata and identifies a family of mappings for which imprecision calculations is meaningful. An alternative to the fuzzy sets framework exists in the form of probabilistic methods (*e.g.*, [9]). A probabilistic-based approach assumes that one has an incomplete knowledge on the portion of the real world being modeled. However, this knowledge can be encoded as probabilities about events. The fuzzy approach, on the other hand, aims at modeling the intrinsic imprecision of features of the modeled reality. Therefore, the amount of knowledge at the user's disposal is of little concern. Our choice, in addition to philosophical reasoning, is also based on pragmatic reasoning. Probabilistic reasoning typically relies on event independence assumptions, making correlated events harder to assess. Our approach is supported by the results presented in [8], where a comparative study of the capabilities of probability and fuzzy methods is presented. This study shows that although probabilistic analysis is intrinsically more expressive than fuzzy sets, fuzzy methods demonstrate higher computational efficiency.

The rest of the paper is organized as follows. Section 2 introduces the proposed schema integration model. We formally define similarity relations (primitive and compound) as fuzzy relations and demonstrate these concepts by defining similarities among data values, domains, individual attributes, and mappings. We next define a class of monotonic mappings in Section 3, for which we show that fuzzy matching reflects the precision of the mapping itself. In Section 4 we analyze some properties of compound similarity relations. In particular, we provide a justification, in retrospect, for the common use of weighted bipartite matching in identifying the best mapping. The paper is concluded in Section 5 with a discussion of the model applicability and directions for future research.

2 The model

In this section we provide a formal model for computing similarities among attribute sets, based on fuzzy relations [4], as follows. A *fuzzy set* A over a domain $\mathcal{D}$ is a set, characterized by a membership function $\delta_A : \mathcal{D} \rightarrow [0, 1]$, where $\delta_A(a) = \mu$ is the fuzzy membership degree of the element a in A. In what follows we use $\mu^{A,a}$ to specify the elements of interest whenever it cannot be clearly identified from the context. Given domains $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ and their Cartesian product $\mathbf{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_n$, a *fuzzy relation* R over the domains $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ is a fuzzy set of elements (tuples) of $\mathbf{D}$. We next introduce two types of similarity relations. Primitive similarity relations are introduced in Section 2.1. Section 2.2 introduces compound similarity relations.

2.1 Primitive similarity relations

Given domains $\mathcal{D}$ and $\mathcal{D}'$, a *primitive similarity relation* is a fuzzy relation over $\mathcal{D} \cup \mathcal{D}'$, denoted $\sim_\mu$, where μ is a membership function such that the following properties hold:

- (ref) For every $d \in \mathcal{D} \cup \mathcal{D}'$, $d \sim_\mu d$ (using an infix notation) with $\mu = 1$.
- (sym) For $d \in \mathcal{D}$, $d' \in \mathcal{D}'$, $d \sim_\mu d' \rightarrow d' \sim_\mu d$.

- (trin) For $d \in \mathcal{D}$, $d' \in \mathcal{D}'$ and $d'' \in \mathcal{D}''$ (where $\mathcal{D}''$ is a third domain and the similarity relation is defined over $\mathcal{D} \cup \mathcal{D}' \cup \mathcal{D}''$), $(d \sim_{\mu} d' \wedge d' \sim_{\mu'} d'') \rightarrow d \sim_{\mu"} d''$ such that $\mu" \leq \mu + \mu'$.

A primitive similarity relation is a fuzzy relation (over $\mathcal{D}, \mathcal{D}'$) whose membership degree is computed using some distance metric among domain members. We can also require the partition of the domain such that for $d \neq d'$, if $\{d, d'\} \subseteq \mathcal{D}$ or $\{d, d'\} \subseteq \mathcal{D}'$, then $d \sim_{\mu} d'$ with $\mu = 0$. Such partitioning is natural in our case, given that our aim is to match elements of different domains. We annotate by $\mu^{d,d'}$ the similarity between d and d'. As an example, consider two non-negative numeric domains $\mathcal{D} = \{0, 15, 30, 45\}$ and $\mathcal{D}' = \{0, 10, 20, 30, 40, 50\}$, both representing a fraction of an hour in which a car will be picked up. Assume that the similarity of elements $d \in \mathcal{D}$ and $d' \in \mathcal{D}'$ is measured according to their Euclidean distance, normalized between 0 and 1:

$$\mu^{d,d'} = 1 - \frac{|d - d'|}{\max_{d_i, d_j \in \mathcal{D} \cup \mathcal{D}'} \{|d_i - d_j|\}} \qquad (1)$$

Therefore, the similarity score between 15 (in $\mathcal{D}$) and 30 (in $\mathcal{D}'$) is 0.7. $\mu^{d,d'}$, as defined in Equation 1, is a primitive similarity relation.

The properties of primitive similarity relations are desirable properties when it comes to schema integration. Reflexivity ensures that exact matching receives the highest possible score (as in the case of two attributes with the same name). Symmetry ensures that the order in which two schemata are compared has no effect on the final outcome. Finally, the triangular property enables the generation of similarity classes, sets of attributes (one of each schema) that are synonymical. This last property enables a desirable learning feature for automatic schema integration.

2.2 Compound similarity relations

Compound similarity relations use similarity measures (either primitive or compound) to compute new similarity measures. In this section we introduce compound similarity relations via an example. We defer the formal analysis of such relations to Section 4. As an example, we can compute the similarity of two numeric domains, based on the similarity of their values. Let $\mathcal{D}$ and $\mathcal{D}'$ be the domains. Let μ_{dom} be a function, termed the *domain similarity measure*. Then, $\sim_{\mu_{dom}}$ is a *domain similarity* relation (over a set of domains) and $\mathcal{D} \sim_{\mu_{dom}} \mathcal{D}'$ is the domain similarity of the domains $\mathcal{D}$ and $\mathcal{D}'$. μ_{dom} is a function of the similarities of every pair of elements from $\mathcal{D}$ and $\mathcal{D}'$. For example, one may compute μ_{dom} as:

$$\mu_{dom}^{\mathcal{D},\mathcal{D}'} = \min_{d \in \mathcal{D}, d' \in \mathcal{D}'} \left(\mu^{\mathcal{D},d'}, \mu^{\mathcal{D}',d}\right) \qquad (2)$$

where for all $d' \in \mathcal{D}'$, $\mu^{\mathcal{D},d'} = \max_{d \in \mathcal{D}} \left(\mu^{d,d'}\right)$ and for all $d \in \mathcal{D}$, $\mu^{\mathcal{D}',d} = \max_{d' \in \mathcal{D}'} \left(\mu^{d,d'}\right)$. That is, each value in $\mathcal{D}$ is matched with the "best" value in $\mathcal{D}'$, and vice versa, and the strength of μ_{dom} is determined by the strength of the "weakest link." Our use of min and max is in line with fuzzy logic conventions, where max is interpreted as disjunction and min is interpreted as conjunction. We shall discuss alternative operators in Section 4, providing constraints on the possible operator selection. As a concrete example, consider $\mathcal{D}$ and $\mathcal{D}'$ given above. Computing $\mu_{dom}^{\mathcal{D},\mathcal{D}'}$ according to Equation 2

yields a matching of 0 with 0, 10 and 20 with 15, *etc.* $\mu_{dom}^{\mathcal{D},\mathcal{D}''} = 0.9$, since each element in $\mathcal{D}'$ has a corresponding element in $\mathcal{D}$ which is at most 5 minutes apart (and $1 - \frac{5}{50} = 0.9$). It is worth noting that the similarity measure given by Equation 2 is both reflexive and symmetric.

3 Monotonic mappings: measuring matching quality

In this section we aim at modeling the relationship between a choice of a mapping, based on similarity of attributes, and a choice of a mapping, as performed by a human expert. The more correlated these mappings are, the more effective would an automatic mapping process be. In order to compare the effectiveness of various choices of mappings and operators, we introduce the notion of mapping *imprecision*, which follows common IR practice for retrieval effectiveness (*e.g.*, [11]). Assume first that among all possible mappings between two attribute sets of cardinality n ($n!$ such mappings for $1:1$ matching), we choose one and term it the *exact mapping* (denoted $\bar{F}$). Intuitively, the exact mapping is the best possible mapping, as conceived by a human expert. Having selected the exact mapping between $\mathcal{A}$ and $\mathcal{A}'$, we measure the imprecision of any other mapping G simply by counting how many arguments of $\bar{F}$ and G do not coincide. We next present a formal definition of mapping imprecision.

Definition 1 *Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ and $\mathcal{A}' = \{A'_1, \ldots, A'_n\}$ be attribute sets of cardinality n. Also, let F and G be two mappings over $\mathcal{A}$ and $\mathcal{A}'$ and $A_i \in \mathcal{A}$ an attribute. F* discord *with G over A_i if $F(A_i) \neq G(A_i)$. $\mathcal{D}^{F,G}$ denotes the set of attributes of $\mathcal{A}$ over which F discord with G .*

Definition 2 *Let $\bar{F}$ be an exact mapping over $\mathcal{A}$ and $\mathcal{A}'$ and let G be a mapping over $\mathcal{A}$ and $\mathcal{A}'$ such that there are $m \leq n$ attributes in $\mathcal{A}$ over which $\bar{F}$ discord with G. Then G is* m-imprecise *(with respect to $\bar{F}$). We denote by i_G the imprecision of G.*

Definition 3 *Let F and G be mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$. F and G are* similarity preserving *on an attribute $A \in \mathcal{A}$ if $i_F < i_G$ implies $\mu_{att}^{A,F(A)} > \mu_{att}^{A,G(A)}$. $\mathcal{M}^{F,G}$ denotes the set of attributes of $\mathcal{A}$ on which F and G are similarity preserving.*

Definition 4 *Let $\mathcal{F} = \{F_1, F_2, ..., F_m\}$ be a set of mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$, and let $\bar{\varpi} = (\varpi_1, ..., \varpi_n)$ be a weight vector that sums to unity, associating with each attribute $A_i \in \mathcal{A}$ a weight ϖ_i. $\mathcal{F}$ is* monotonic *if the following inequality holds for any pair $\{F_i, F_j\} \subseteq \mathcal{F}$ such that $i_{F_i} < i_{F_j}$:*

$$\sum_{A_k \in \mathcal{D}^{F_i,F_j} \cap \mathcal{M}^{F_i,F_j}} \left(\varpi_k \left(\mu_{att}^{A_k,F_i(A_k)} - \mu_{att}^{A_k,F_j(A_k)} \right) \right) > \sum_{A_k \in \mathcal{D}^{F_i,F_j} \setminus \mathcal{M}^{F_i,F_j}} \left(\varpi_k \left(\mu_{att}^{A_k,F\,j(A_k)} - \mu_{att}^{A_k,F_i(A_k)} \right) \right)$$

The sum on the left represents the benefit of switching from F_j to F_i. $\mathcal{D}^{F_i,F_j} \cap \mathcal{M}^{F_i,F_j}$ represents those attributes over which F_i discord with F_j, yet are similarity preserving. Since $i_{F_i} < i_{F_j}$, each term in the sum adds to the overall similarity. The sum on the right represents the loss involved in switching from F_j to F_i. $\mathcal{D}^{F_i,F_j} \setminus \mathcal{M}^{F_i,F_j}$ represents those attributes over which F_i discord with F_j, and that are not similarity preserving. These

attributes lower the overall similarity by switching from F_j to F_i. If the benefit of switching from F_j to F_i surpasses the cost for all pairs $\{F_i, F_j\} \subseteq \mathcal{F}$ such that $i_{F_i} < i_{F_j}$, we consider the set to be monotonic. If the exact mapping is chosen among monotonic mappings, then the following holds: if $\bar{F} \in \mathcal{F}$ and $\mathcal{F}$ is monotonic then $\bar{F}$'s overall similarity measure is greater than the overall similarity degrees of i-imprecise mappings in $\mathcal{F}$, even if such mappings yield better similarity degrees on some pairs of domain elements and on some pairs of attribute names. If all one wishes to obtain is the ability to identify the exact mapping through the use of similarity, one needs a weaker notion of monotonicity, as defined next.

Definition 5 *Let $\mathcal{F} = \{F_1, F_2, ..., F_m\}$ be the set of all possible mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$. $\mathcal{F}$ is* monotonic with respect to the exact mapping *$\bar{F} \in \mathcal{F}$ if the following inequality holds for any $F_i \in \mathcal{F}$:*

$$\sum_{A_k \in \mathcal{D}^{F_i,\bar{F}} \cap \mathcal{M}^{F_i,\bar{F}}} \left(\varpi_k \left(\mu_{att}^{A_k,\bar{F}(A_k)} - \mu_{att}^{A_k,F_i(A_k)} \right) \right) > \sum_{A_k \in \mathcal{D}^{F_i,\bar{F}} \setminus \mathcal{M}^{F_i,\bar{F}}} \left(\varpi_k \left(\mu_{att}^{A_k,F_i(A_k)} - \mu_{att}^{A_k,\bar{F}(A_k)} \right) \right)$$

4 Compound similarity properties

Having defined the framework for expressing schema mapping similarity, we turn our attention to compound similarity properties. In Section 4.1, we discuss the set of alternative operators we have at our disposal and their inter-relationship. Section 4.2 presents some interesting properties of monotonic mappings. The main result of this section states that, under appropriate hypotheses made explicit in Section 4.2, a monotonic set of mappings orders mappings according to their imprecision level.

4.1 Similarity operators

In this section we present two families of similarity operators, namely triangular norms and fuzzy aggregate operators, and compare their properties. Operators from both families are typically used in fuzzy-based applications to combine various fuzzy membership degrees. Since the study of different ways of combining similarities is crucial to this work, we provide a brief introduction of their main properties.

The min operator was introduced in Section 2.1 for computing the similarity degree of two domains. This operator is the most well-known representative of a large family of operators called *triangular norms* (t-norms, for short), routinely deployed as interpretations of fuzzy conjunctions. In the following, we define t-norms and discuss their relevant properties. We refer the interested reader to [16] for an exhaustive treatment of the subject.

A *triangular norm* $T : [0,1] \times [0,1] \to [0,1]$ is a binary operator on the unit interval satisfying the following axioms for all $x, y, z \in [0,1]$:

- (boundary condition) $T(x,1) = x$,
- (monotonicity) $x \leq y$ implies $T(x,z) \leq T(y,z)$,
- (commutativity) $T(x,y) = T(y,x)$,
- (associativity) $T(x,T(y,z)) = T(T(x,y),z)$.

Examples of t-norms that are typically used as interpretations of fuzzy conjunctions include mini-

mum ($Tm(x,y) = \min(x,y)$), product ($Tp(x,y) = x \cdot y$), and the Lukasiewicz t-norm ($Tl(x,y) = \max(x+y-1,0)$). It is worth noting that Tm is the only idempotent t-norm. That is, $Tm(x,x) = x$. This becomes handy when comparing t-norms with fuzzy aggregate operators. Also, it can be easily proven (see [14]) that $Tl(x,y) \leq Tp(x,y) \leq Tm(x,y)$ for all $x, y \in [0,1]$.

The *average* operator that is typically used for the computation of the similarity of attribute sets does not satisfy the t-norm axioms. Rather, it belongs to another large family of operators termed *fuzzy aggregate operators* [16]. A fuzzy aggregate operator $H : [0,1]^n \rightarrow [0,1]$ satisfy the following axioms for every $x_1, \ldots, x_n \in [0,1]$:

- (idempotency) $H(x_1, x_1, \ldots, x_1) = x_1$,
- (increasing monotonicity) for every $y_1, y_2, \ldots, y_n \in [0,1]$ such that $x_i \leq y_i$, $H(x_1, x_2, \ldots, x_n) \leq H(y_1, y_2, \ldots, y_n)$,
- H is a continuous function.

Let $\bar{x} = (x_1, \ldots, x_n)$ be a vector such that for all $1 \leq i \leq n$, $x_i \in [0,1]$ and let $\bar{\varpi} = (\varpi_1, ..., \varpi_n)$ be a weight vector that sums to unity. Examples of fuzzy aggregate operators include the *average* operator $Ha(\bar{x}) = \frac{1}{n}\sum_1^n x_i$ and the *weighted average* operator $Hwa(\bar{x}, \bar{\varpi}) = \bar{x} \cdot \bar{\varpi}$. Clearly, *average* is a special case of the *weighted average* operator, where $\varpi_1 = \cdots = \varpi_n$. It is worth noting that Tm (the min t-norm) is also a fuzzy aggregate operator, due to its idempotency (its associative property provides a way of defining it over any number of arguments). However, Tp and Tl are not fuzzy aggregate operators. T-norms and fuzzy aggregate operators are comparable, using the inequality $\min(x_1, \ldots, x_n) \leq H(x_1, \ldots, x_n)$ for all $x_1, \ldots, x_n \in [0,1]$ and function H satisfying idempotency, increasing monotonicity and continuity axioms.

4.2 Monotonic mappings revisited

In this section we present some relevant properties of compound similarity operators. In particular, we show that for a monotonic set of mappings, the use of a weighted average to compute mapping similarity orders mappings according to their imprecision level.

Theorem 1 *Let $\mathcal{F}$ be a monotonic set of mappings and let $\{F_i, F_j\} \in \mathcal{F}$ be mappings over attribute sets $\mathcal{A}$ and $\mathcal{A}'$ with imprecision i_{F_i} and i_{F_j}, respectively, such that $i_{F_i} < i_{F_j}$. If the corresponding similarity measures are combined using the Hwa (weighted average) operator yielding respectively μ^F and μ^G, then $\mu^F > \mu^G$.*

Theorem 1 requires that similarities are combined using the Hwa (weighted average) operator. We now show that the use of weighted average is preferred over any t-norm operator to compute mapping similarity. For simplicity sake, we restrict our discussion to similarity among attribute pairs and their combination into similarities among schemata. The following result can be easily generalized to any similarity measure method. We denote by X_1X_2 a particular selection of operators for computing attribute similarity (X_1), and mapping similarity (X_2). We next show that, in most cases, a selection of type X_1Ha is superior to any selec-

tion of type X_1T, where T stands for any t-norm operator.

Definition 6 *Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ and $\mathcal{A}' = \{A'_1, \ldots, A'_n\}$ be attribute sets of cardinality n. $\mathcal{A}$ and $\mathcal{A}'$ are* closely related *if for any mapping F over $\mathcal{A}$ and $\mathcal{A}'$, if $(A, A') \in F$, then $\mu_{att}^{A,A'} > 0$.*

Closely related attribute sets consist of attributes that may map well in various combinations. Our experience show that this is hardly ever the case, since attributes tend to vary in names and domains. We next present a proposition arguing that t-norms are not suitable for modeling attribute sets that are not closely related.

Proposition 1 *Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ and $\mathcal{A}' = \{A'_1, \ldots, A'_n\}$ be attribute sets of cardinality n. If $\mathcal{A}$ and $\mathcal{A}'$ are **not** closely related, any selection of operators of type X_1T yields a non monotonic mapping set.*

An immediate corollary to Proposition 1 relates to mappings using weighted bipartite graph matching. Given two attribute sets, $\mathcal{A}$ and $\mathcal{A}'$, one may construct a weighted bipartite graph $G = (V, E)$, such that $V = \mathcal{A} \cup \mathcal{A}'$, and $(v_i, v_j) \in E$ if $v_i \in \mathcal{A}$, $v_j \in \mathcal{A}'$. The weight function $\varpi : \mathcal{A} \times \mathcal{A}' \rightarrow [0, 1]$ is defined to be $\varpi(v_i, v_j) = \mu_{att}^{v_i,v_j}$. The weighted bipartite graph matching algorithm yields a $1:1$ mapping F with maximum weight $\Omega^F = \sum_{(v_i,v_j)\in F} \varpi(v_i, v_j)$. Given that $\mathcal{A}$ and $\mathcal{A}'$ are attribute sets of cardinality n, that are not closely related, and assuming a selection of operators of type X_1Ha, such mapping yields $\mu^F = \frac{1}{n}\Omega^F$. Therefore, the use of weighted bipartite graph matching is equivalent to a selection of operators of type X_1Ha, which yields results as good as any selection of operators of type X_1T, and possibly better.

5 Conclusion and future work

We have presented a formal model for schema matching, capturing the inherent uncertainty of the outcome of automating the process. The model presentation is followed by an analysis of the model properties and the identification of a sufficient condition for the correlation of the automatic process outcome with that of a manual process. The formal model borrows from fuzzy set theory in modeling uncertainty. The theoretical analysis of the model have yielded that for monotonic mappings one may correlate similarity measure with precision, as conceived by a human expert. While monotonicity is a strong notion, weaker notions, such as monotonicity with respect to an exact match suffices for practical purposes (such as identifying the exact mapping within a small number of iterations). Therefore, matching algorithms that generate monotonic mappings (in any form) are well suited for automatic semantic reconciliation. Unless attributes in schemata are closely related, mapping similarity cannot utilize any t-norm as its computation vehicle. A preferred operator would come from the fuzzy aggregate operator family, *e.g.*, the *average* operator. This result provides a theoretical support for the use of variations of the weighted bipartite graph matching for computing schema mapping.

We have performed initial experiments, aiming at verifying empirically the correlation between a similarity measure (generated by a given algorithm) on

the one hand and monotonicity on the other hand, using imprecision level as the experimentation tool. A full report of the experiments is available in [13].

We envision multitude of applications for automatic schema matching. For example, the research is likely to aid in the design of smart agents that will negotiate over information goods using schema information and provide them with some practical tools to combat schema heterogeneity. Towards this end, we shall conduct a thorough analysis of schema usability, to enable a realistic evaluation of the outcomes of a top-K algorithm [1] on a practical level. The top-K algorithm, presented in [1], enables an efficient identification of K mappings with the highest similarity measure. The outcome of the analysis would be the development of robust methods for assessing the usability of mappings to a user. Using these methods, an agent performing on behalf of a user will be able to filter out non-usable matchings from the top-K group, so that the remaining results, that are to be presented to the user, would be of the best quality.

References

[1] A. Anaby-Tavor, A. Gal, and A. Moss. Efficient algorithms for top-k matchings. Submitted for publication. Available upon request from avigal@ie.technion.ac.il, 2003.

[2] W.G. Aref, D. Barbará, S. Johnson, and S. Mehrotra. Efficient processing of proximity queries for large databases. In P.S. Yu and A.L.P. Chen, editors, *Proceedings of the IEEE CS International Conference on Data Engineering*, pages 147–154. IEEE Computer Society, 1995.

[3] S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration of heterogeneous information sources. *Data & Knowledge Engineering*, 36(3), 2001.

[4] P. Ciaccia, D. Montesi, W. Penzo, and A. Trombetta. Imprecision and user preferences in multimedia queries: A generic algebraic approach. In *Lecture Notes on Computer Science, 1762*, pages 50–71. Springer, 2000.

[5] L.S. Davis and N. Roussopoulos. Approximate pattern matching in a pattern database system. *Information systems*, 5(2):107–119, 1980.

[6] L. G. DeMichiel. Performing operations over mismatched domains. In *Proceedings of the IEEE CS International Conference on Data Engineering*, pages 36–45, Los Angeles, CA, February 1989.

[7] A. Doan, P. Domingos, and A.Y. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In Walid G. Aref, editor, *Proceedings of the ACM-SIGMOD conference on Management of Data (SIGMOD)*, Santa Barbara, California, May 2001. ACM Press.

[8] J. Drakopoulos. Probabilities, possibilities and fuzzy sets. *International Journal of Fuzzy Sets and Systems*, 75(1):1–15, 1995.

[9] T. Eiter, T. Lukasiewicz, and M. Walter. Extension of the relational algebra to probabilistic complex values. In B. Thalheim K.-D. Schewe, editor, *Lecture Notes on Computer Science, 1762*, pages 94–115. Springer, 2000.

[10] R. Fagin. Combining fuzzy information from multiple systems. *J. of Computer and System Sciences*, 58:83–99, 1999.

[11] W.B. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structures & Algorithms*. Prentice Hall, Englewood Cliffs, NJ 07632, 1992.

[12] N. Fridman Noy and M.A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 450–455, Austin, TX, 2000.

[13] A. Gal, A. Trombetta, A. Anaby-Tavor, and D. Montesi. A framework for evaluating similarity-based schema matching. Submitted for publication. Available upon request from avigal@ie.technion.ac.il, 2003.

[14] P. Hajek. *The Metamathematics of Fuzzy Logic*. Kluwer Acad. Publ., 1998.

[15] 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.

[16] G.J. Klir and B. Yuan, editors. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, 1995.

[17] 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.

[18] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proceedings of the International conference on very Large Data Bases (VLDB)*, pages 49–58, Rome, Italy, September 2001.

[19] E. Mena, V. Kashayap, A. Illarramendi, and A. Sheth. Imprecise answers in distributed environments: Estimation of information loss for multi-ontological based query processing. *International Journal of Cooperative Information Systems*, 9(4):403–425, 2000.

[20] R.J. Miller, M.A. Hernàndez, L.M. Haas, L.-L. Yan, C.T.H. Ho, R. Fagin, and L. Popa. The clio project: Managing heterogeneity. *SIGMOD Record*, 30(1):78–83, 2001.

[21] G. Modica, A. Gal, and H. Jamil. The use of machine-generated ontologies in dynamic information seeking. 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 433–448. Springer, 2001.

[22] L. Palopoli, L.G. Terracina, and D. Ursino. The system DIKE:towards the semi-automatic synthesis of cooperative information systems and data warehouses. In *PADBIS-DASFAA*, pages 108–117, 2000.

[23] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.