SMART: A tool for analyzing and reconciling schema matching networks

Nguyen Quoc Viet Hung[*], Nguyen Thanh Tam[*], Zoltán Miklós[†], Karl Aberer[*], Avigdor Gal[#], Matthias Weidlich[+]

[*]*École Polytechnique Fédérale de Lausanne,* [†]*Université de Rennes 1,* [#]*Technion - Israel Institute of Technology,* [+]*Imperial College London*

***Abstract*—Schema matching supports data integration by establishing correspondences between the attributes of independently designed database schemas. In recent years, various tools for automatic pair-wise matching of schemas have been developed. Since the matching process is inherently uncertain, the correspondences generated by such tools are often validated by a human expert. In this work, we consider scenarios in which attribute correspondences are identified in a network of schemas and not only in a pairwise setting. Here, correspondences between different schemas are interrelated, so that incomplete and erroneous matching results propagate in the network and the validation of a correspondence by an expert has ripple effects. To analyse and reconcile such matchings in schema networks, we present the Schema Matching Analyzer and Reconciliation Tool (SMART). It allows for the definition of network-level integrity constraints for the matching and, based thereon, detects and visualizes inconsistencies of the matching. The tool also supports the reconciliation of a matching by guiding an expert in the validation process and by offering semi-automatic conflict-resolution techniques.**

I. Introduction

Schema matching is the process of establishing correspondences between the attributes of database schemas for data integration purposes. The outcome of the schema matching process is a set of attribute correspondences, which are the basis for the definition of attribute mappings, i.e. the actual transformations applied to data instances.

Tools for schema matching support this process by taking two schemas as input and creating correspondences between their attributes based on heuristic methods. There is a large body of research on heuristic matching algorithms, see [1], [2] for surveys, and several commercial and academic matching tools have been developed, e.g., COMA++ [3] or AMC [4]. However, given the inherent uncertainty of the matching process, the correspondences created by schema matching tools are often incomplete and sometimes erroneous. Hence, the created correspondences are typically at least partially validated by a human expert [5].

Matching in Schema Networks. For many recent applications, data integration is no longer limited to a single pair of data sources. Most prominently, Web Tables emerged as a paradigm to publish structured data on the Web [6]. Given the sheer number of publicly available Web Tables, it is important to link them by creating correspondences between their schemas [7]. Matching in schema networks, however, is not limited to integration of Web Tables, but also beneficial for enterprise applications [8] and in mashup contexts [9]. In principle, all data integration scenarios, for which the creation of a monolithic, mediated data schema would be too costly or even infeasible can benefit from matching in schema networks.

Integrity constraints. Our work on analysis and reconciliation of a matching in a network of schemas is grounded on integrity constraints. Such constraints are defined over attribute correspondences in a network and ensure a high overall quality of the matching. The presence of integrity constraints creates numerous dependencies between correspondences, which complicate the validation by an expert especially in large-scale networks. However, dependencies between correspondences also represent an opportunity to guide the expert's work by defining the order in which feedback is sought.

Consider enterprises that intend to exchange information on purchase orders, but apply different schemas to capture this information. To facilitate integration scenarios, a schema matching network is created by establishing pairwise matchings between the schemas, cf., Figure 1. However, the resulting network should obey certain integrity constraints. An example would be the *one-to-one constraint* that requires each attribute to be matched to at most one attribute in another schema. In Figure 1, this constraint is violated by the correspondences $\{c_2, c_4\}$. Another example is the network-level *circle constraint*: If correspondences form a circle at the schema level, then the correspondences should form a closed circle. In the given example, this constraint is violated, e.g., by the correspondences $\{c_1, c_2, c_5\}$.

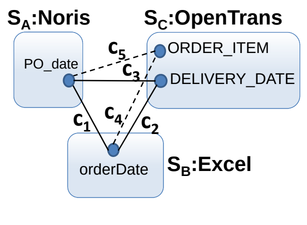

Fig. 1. A schema matching network

SMART. To realize constraint-based analysis and reconciliation of schema matching networks, we present the Schema Matching Analyzer and Reconciliation Tool (SMART). SMART has three main features, see also Figure 2.

- *Network analysis.* The result of automatic tools for matching is typically uncertain in the sense that the generated correspondences violate various integrity constraints. SMART helps an expert to detect and analyze constraint violations and assesses the correctness probability of correspondences and the overall uncertainty of the network.

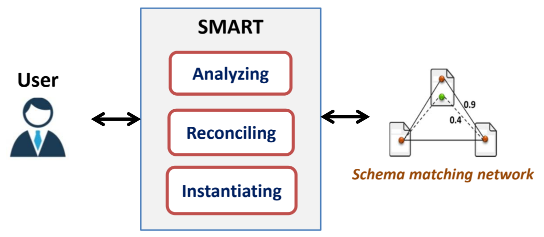

Fig. 2. Simplified architecture of SMART

- *Reconciliation.* Reconciliation of a matching is a pay-as-you-go process, in which the quality of a matching network is incrementally improved by seeking expert input. As the cost of expert effort is expensive, SMART selects and ranks candidate correspondences for validation based on their information gain, which is measured as the amount of uncertainty reduction induced by expert input for a certain correspondence.
- *Instantiation.* To implement reconciliation as a pay-as-you-go process, at any point in time, SMART allows to instantiate a set of correspondences based on the input of automatic tools for matching and the expert input collected so far. SMART chooses a subset of correspondences that satisfy all integrity constraints and are likely to be correct (according to our model [10]).

The remainder of this paper is structured as follows. Section II summarizes the main concepts from our earlier work [10], [11], underlying SMART. Section III outlines the demonstration of SMART, before we conclude in Section IV.

II. Implementation

Below, we discuss the network analysis, reconciliation, and instantiation techniques offered by SMART, followed by performance considerations.

A. Network Analysis

Detecting constraint violations. To detect violations of integrity constraints over attribute correspondences, SMART uses a model based on Answer Set Programs (ASP). That is, the matching in the network of schemas as well as integrity constraints are formalized as ASPs, so that constraint violations are identified by logical reasoning as implemented in state-of-the-art ASP solvers, see [11].

Assessing the correctness of correspondences. Common tools for automatic pair-wise matching of schemas assign confidence values to candidate correspondences [1]. A confidence value may be interpreted as a probability for the correctness of a correspondence. Yet, confidence values are not normalized, often unreliable, and unrelated to the application goals [2].

SMART takes a different approach to assess the correctness of candidate correspondences. It relies on a probabilistic model, assigning a correctness probability to each correspondence [10]. Here, the idea is that a correspondence is more likely to be correct, if it appears in many *matching instances*, maximal sets of correspondences that contain no constraint violations and respect the user input (i.e., contain only approved correspondences or those for which no input has been sought). Since the computation of the actual probabilities is intractable given the exponential number of possible matching instances, SMART uses a sampling method based on random-walk and simulated annealing for probability estimation [10].

Measure the network uncertainty. Our approach to measuring the network uncertainty relies on the aforementioned probabilistic model and a set of binary random variables that model the decision whether to include a correspondence in a matching instance. Then, we compute the overall uncertainty of the network as the Shannon entropy over these random variables, see [10]. Following this line, a network uncertainty value of zero means that all probabilities are equal to one or zero; i.e., there is no 'uncertain' correspondence remaining and the matching has been fully reconciled.

B. Reconciliation

The set of candidate correspondences generated by automatic tools for matching typically violates the integrity constraints defined for a network of schemas. SMART treats reconciliation as an iterative process, such that, in each iteration, an expert asserts the correctness of a single correspondence. Hence, an iteration comprises (1) selecting an attribute correspondence not yet validated; (2) eliciting expert input on the selected correspondence; (3) computing the consequences of the feedback. Reconciliation stops if the goal of reconciliation is reached, e.g., elimination of all constraint violations.

Minimizing the number of iterations to reach the reconciliation goal requires effective solutions for step (1), the attribute selection, and step (3), the computation of feedback consequences. SMART strives to reduce the effort needed for reconciliation by applying effective heuristics for both steps.

Effort minimization by ordering. When seeking expert feedback on correspondences, the order in which the correspondences are selected influences the number of iterations needed to reach the reconciliation goal. SMART ranks correspondences for which feedback shall be elicited using a decision theoretic approach [12]. That is, the potential benefit of knowing whether a correspondence is correct is measured as the *information gain* of the correspondence. Technically, the information gain is defined in terms of the expected uncertainty reduction in the network, computed as the difference between the network uncertainty before and after incorporating feedback on a particular correspondence.

Effort minimization by reasoning. When integrating the feedback of an expert, SMART does not only update the correctness probability for the respective correspondence, but also exploits the integrity constraints to achieve a more effective computation of the consequences of the input. The representation of the matching network and the integrity constraints as ASPs allows for concluding on the incorrectness of certain correspondences by logical reasoning. Consider, for instance, the schemas S_b and S_c of the example given in Figure 1, for which a matching tool generated two candidate correspondences c_2 and c_4. Assuming that the one-to-one constraint is applied, an approval of c_2 by an expert allows for immediate falsification of correspondence c_4.

C. Instantiation

SMART supports a pay-as-you-go reconciliation process by offering the possibility to instantiate a valid matching, i.e., select one of the matching instances, at any point in time. To assess which of the possible matching instances best approximates the actual set of correspondences, SMART solves an optimisation problem that is defined along two dimensions: the *repair distance* and the *likelihood* of a matching instance.

- The *repair distance* measures the difference between the correspondences of a matching instance and the set of correspondences generated by an automatic matching tool in the first place. As such, it is a means to quantify the amount of information loss of deriving the matching instance by removing correspondences to guarantee satisfaction of the integrity constraints.
- The *likelihood* indicates the correctness of a matching instance. Technically, it is the product of the probabilities of the respective correspondences.

Using these measures, instantiation is an optimization problem: we are interested in a matching instance with minimal repair distance and maximal likelihood. In the context of schema matching, we consider the repair distance to be more important than the likelihood, since information about correspondences should be preserved as much as possible. Since this optimization problem is NP-hard, SMART applies a heuristic algorithm to approximate the solution [10].

D. Performance considerations

Finally, we turn to performance considerations. To cope with a high number of very large schemas, SMART employs partitioning and caching strategies.

Partitioning. To handle an overwhelming number of correspondences, SMART allows for partitioning a network of schemas into smaller parts, which can be reconciled by an expert efficiently. The decomposition implemented in SMART is driven by two criteria:

- *Size equality:* The network should be divided into partitions of (nearly) equal size.
- *Informativeness:* Decomposition may lead to information loss regarding constraint violations that involve correspondences between attributes of schemas in different partitions. To avoid this phenomenon, SMART aims at preserving correlations between constraints, so that the constraints can be harnessed per network partition.

Based on these criteria, SMART solves the decomposition problem by tracing it back to hypergraph partitioning, see [13].

Caching. SMART also implements two view-maintenance [14] liked techniques to cache intermediate results. The rationale behind these techniques is that reconciliation is an incremental process where the input of a single iteration affects only a small part of the network.

First, we use caching in the sampling-based computation of network uncertainty [10]. Instead of re-sampling in each reconciliation iteration, a set of sampled matching instances is maintained and updated upon the arrival of expert input.

Second, SMART exploits caching for the ASP reasoning to derive consequences of expert input [13]. Instead of sending the whole ASP-encoded network to an ASP solver in each iteration of the reconciliation process, SMART maintains the previous reasoning result and performs reasoning only over the affected network region.

III. Demonstration

To demonstrate the application of SMART, we first present its user interface before we turn to an exemplary scenario.

User interface. SMART offers a rich user interface (see Figure 3) for schema matching networks, which consists of multiple views, including a schema-link diagram and a matrix-based representation.

- A *schema-link diagram* is an interaction graph between schemas in the network. Each node represents a schema and, thus, a set of attributes, while edges represents attribute correspondences. The latter are also assigned a confidence value that is generated by a tools for automatic matching. A user can control the visualization, for example, by sorting the attributes of schema or filtering correspondences by their confidence values. In addition, the layout of the diagram is computed such that similar schemas and attributes are positioned close to each other.
- A *matrix-based representation* provides the user with overview of the matching in a schema network. We use the affinity matrix model [15] in which each column represents a schema and each row represents the associated attributes, such that there are correspondences between them. Existing matrix ordering techniques [16] are leveraged to help a user to identify network clusters.

The schema-link diagram provides a natural representation of the correspondences between schemas and is an effective visualisation for small networks. The matrix-based representation, in turn, is best-suited to manipulate large or dense networks and reveals the high-level network structures (e.g. clusters) by the ordering rows and columns [17]. We combine both representations, so that a user may use the matrix to identify clusters of schemas, which can then be explored in more detail using the schema-link diagram. Both representations are synchronized, i.e., data filtered in one representation is also hidden in the other representation. SMART further provides a wide range of interaction features for both representations, such as moving matrix rows or columns, sorting rows or columns by criteria, drag&drop for schemas or attributes, modifying correspondences, and zooming into network regions.

Demonstration scenario. Taking up the example from Figure 1, we demonstrate how to use SMART to analyze and reconcile schema matching networks (a screencast of the demonstration is publicly availabl[1]).

First, SMART detects violations of the aforementioned one-to-one and cycle constraints. That is, all correspondences $\{c_1, \ldots, c_5\}$ in Figure 1 would be highlighted. Then, instantiation would consider two possible matching instances, $I_1 = \{c_1, c_2, c_3\}$ and $I_2 = \{c_1, c_4, c_5\}$. Since, for our simple

[1]http://youtu.be/T9T6sK5RqjU

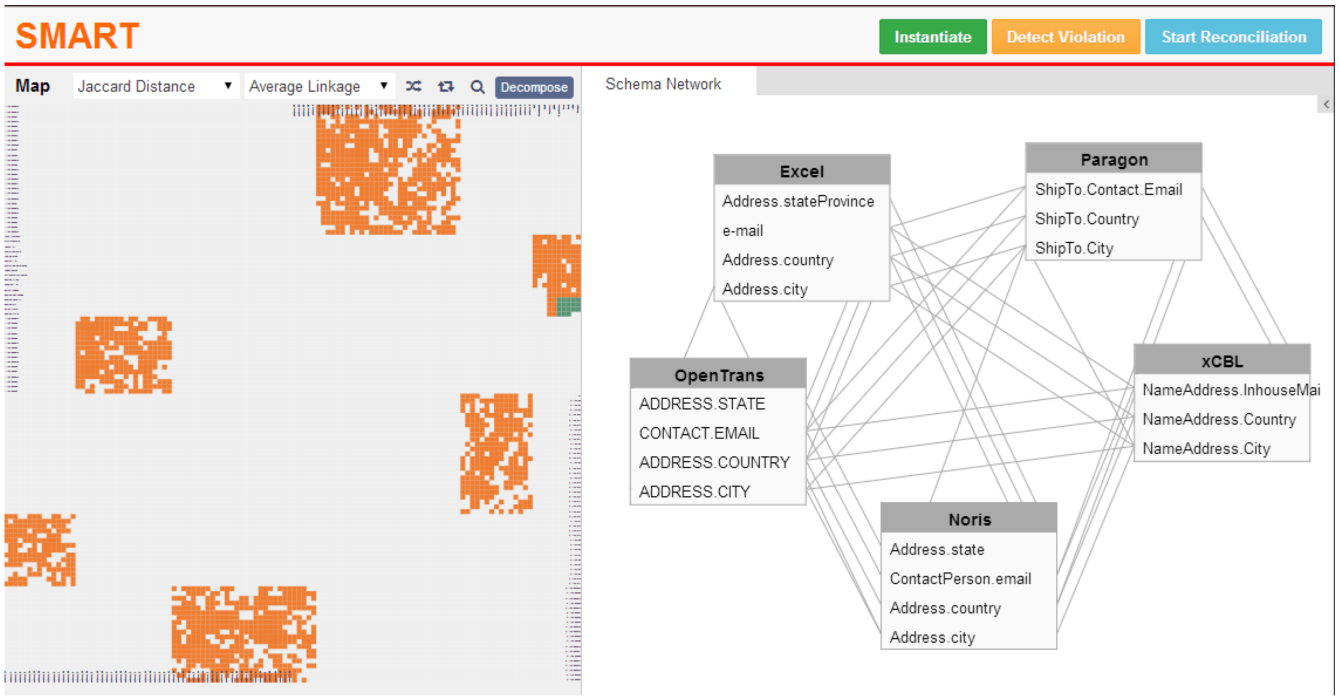

Fig. 3. The SMART user interface for the analysis and reconciliation of schema matching networks

example, both instances have equal likelihood and repair distance, SMART selects one randomly as a result. To improve the quality of the matching by reconciliation, SMART computes the network uncertainty and the correctness probability of correspondences based on the constraint violations, i.e., $p_{c_1} = 1$, $p_{c_2} = p_{c_3} = p_{c_4} = p_{c_5} = 0.5$. Based on the information gain computed for each of the correspondences, SMART suggests to seek expert feedback for c_2. Indeed, if c_2 is approved by the expert, I_1 remains the only possible matching instance, so that the network uncertainty is reduced to 0.

IV. Conclusion

In this paper, we presented SMART, a tool for analyzing and reconciling schema matching networks in the context of data integration. Unlike traditional tools for schema matching, our focus is on scenarios in which a large number of schemas is matched. SMART supports the reconciliation of a matching in a network of schemas with a pay-as-you-go process that leverages the integrity constraints defined for a matching. Based on the output of tools for automatic matching and expert input, a valid matching can be instantiated at any point in time. The matching quality is improved by seeking expert feedback on correspondences. SMART analyses network level conflicts and, based thereon, guides the expert in the conflict resolution. As such, SMART serves as a decision-support system for database developers, who can identify problems of existing schemas, and data integration experts, who benefit from the construction of a valid matching.

Acknowledgment

The research received funding from the following projects: EU-FP7 NisB (grant no. 256955), EU-FP7 EINS (grant no. 288021) and EU-FP7 PlanetData (grant no. 257641).

References

[1] E. Rahm and P. A. Bernstein, "A Survey of Approaches to Automatic Schema Matching," *JVLDB*, pp. 334–350, 2001.
[2] P. Bernstein, J. Madhavan, and E. Rahm, "Generic Schema Matching, Ten Years Later," in *VLDB*, 2011.
[3] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm, "Schema and ontology matching with coma++," in *SIGMOD*, 2005, pp. 906–908.
[4] E. Peukert, J. Eberius, and E. Rahm, "AMC - A framework for modelling and comparing matching systems as matching processes," in *ICDE*, 2011, pp. 1304–1307.
[5] K. Belhajjame, N. W. Paton, A. A. A. Fernandes, C. Hedeler, and S. M. Embury, "User feedback as a first class citizen in information integration systems," in *CIDR*, 2011, pp. 175–183.
[6] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "Webtables: exploring the power of tables on the web," in *PVLDB*, 2008, pp. 538–549.
[7] J. Fan, M. Lu, B. C. Ooi, W.-C. Tan, and M. Zhang, "A hybrid machine-crowdsourcing system for matching web tables," in *ICDE*, 2014, pp. 976–987.
[8] K. P. Smith, M. Morse, P. Mork, M. Li, A. Rosenthal, D. Allen, L. Seligman, and C. Wolf, "The role of schema matching in large enterprises," in *CIDR*, 2009.
[9] G. Di Lorenzo, H. Hacid, H.-y. Paik, and B. Benatallah, "Data integration in mashups," *SIGMOD Rec.*, pp. 59–66, 2009.
[10] Q. V. H. Nguyen, T. T. Nguyen, Z. Miklós, K. Aberer, A. Gal, and M. Weidlich, "Pay-as-you-go reconciliation in schema matching networks," in *ICDE*, 2014, pp. 220–231.
[11] Q. V. H. Nguyen, T. K. Wijaya, Z. Miklos, K. Aberer, E. Levy, V. Shafran, A. Gal, and M. Weidlich, "Minimizing Human Effort in Reconciling Match Networks," in *ER*, 2013.
[12] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial intelligence: a modern approach*. Prentice hall, 1995.
[13] Q. V. H. Nguyen, "Reconciling schema matching networks," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2014.
[14] J. A. Blakeley, P.-A. Larson, and F. W. Tompa, "Efficiently updating materialized views," in *SIGMOD*, 1986, pp. 61–71.
[15] N. Q. V. Hung, D. S. Thanh, N. T. Tam, K. Aberer, "Privacy-preserving schema reuse," in *DASFAA*, 2014, pp. 234–250.
[16] I. Liiv, "Seriation and matrix reordering methods: An historical overview," *Statistical analysis and data mining*, 2010.
[17] M. Ghoniem, J.-D. Fekete, and P. Castagliola, "On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis," *Information Visualization*, pp. 114–135, 2005.