Applying Cognitive Principles of Similarity to Data Integration – The Case of SIAM

Joerg Evermann
Memorial University of Newfoundland, St. John's, NF, Canada., jevermann@mun.ca

Applying Cognitive Principles of Similarity to Data Integration – The Case of SIAM

Joerg Evermann
Memorial University of Newfoundland
jevermann@mun.ca

ABSTRACT

Increasingly, modern system design is concerned with the integration of legacy systems and data. Consequently, data integration is an important step in many system design projects and also a prerequisite to data warehousing, data mining, and analytics. The central step in data integration is the identification of similar elements in multiple data sources. In this paper, we describe an application of principles of similarity based in cognitive psychology, specifically the theory of Similarity as Interactive Activation and Mapping (SIAM) to the problem of database schema matching. In a field that has been dominated by a multitude of ad-hoc algorithms, cognitive principles can establish an appropriate theoretical basis. The results of this paper show initial success in matching applications and point towards future research.

Keywords (Required)

Databases, Database systems, Data models, Data Integration, Cognition, Similarity, SIAM.

INTRODUCTION

With the proliferation of information systems in business, and an ever increasing number of new and legacy data sources, an important aspect of system design and development is that of systems integration. Few systems are developed in a complete vacuum. An important part of systems development and systems integration is therefore the integration of the data. Data integration is also necessary to support business analytics and the need to have an integrated view of data is ever increasing to support better decision making. Data warehousing and analytics draw on integrated data, typically from multiple transactional data sources, to provide business insight. As much of the corporate information is stored in multiple relational databases, their integration is critical if organizations are to derive business value from business intelligence applications and be efficient and effective in data and information management. In summary, the quality of data integration during a system development effort has a direct influence on the business decisions that are made based on data warehousing and analytics applications on the integrated data.

One of the main steps in database integration is the identification of database elements that are similar. When considering structured data in relational databases, this is called schema matching and is the focus of this paper. Despite its importance to business intelligence and analytics, schema matching remains a difficult problem, which has given rise to a multitude of heuristics (Rahm and Bernstein, 2001; Doan and Halvey, 2005), but to date, there has not been a sound theoretical basis for approaching this problem.

Here, we suggest that a schema matching algorithm is considered useful or successful by its human users when the algorithm produces matches between data elements in just such a way in which humans would make the matching decisions. Thus, in line with previous suggestions (Evermann, 2008a, b; Lukyanenko and Evermann, 2011), we propose to look to cognitive psychology for an appropriate theoretical foundation. Theories of how humans judge the similarity of objects, which in turn are related to theories of analogical reasoning and metaphors, have been investigated for some time by cognitive psychologists. In this paper, we examine one such theory and apply it to the database context to identify similarities between schema elements for matching purposes.

This paper makes two contributions. First, we highlight the importance and the relevance of cognitive theories of similarity to the business intelligence context. Second, we demonstrate how one specific, well researched theory is applied to the problem of schema matching and provide initial results of its applicability.

The remainder of the paper is structured as follows. We first present a brief overview over the problem of schema matching and existing approaches to the problem. The subsequent section argues for the importance of a cognitive approach to this problem. We then provide a description of the SIAM model (Similarity as Interactive Activation and Mapping) and the adaptations to the context database that we have made. This is followed by a demonstration of the algorithm to example

problems in order to illustrate its properties. The paper concludes with a general discussion of the SIAM model and an outlook to future research.

SCHEMA MATCHING

Schema matching is the identification of database schema elements that are similar. Development of software algorithms for schema matching has a long tradition in computer science and has led to a multitude of different schema matching approaches (Evermann, 2008b; Doan and Halevy, 2005; Rahm and Bernstein, 2001). These can be distinguished by their use of information about the database structure (schema-level information), information about the database content (instance-level information), external information (e.g. thesauruses, domain ontologies, etc.) or some combination of these three.

Schema level information comprises structural data, i.e. the relationships between tables as expressed through foreign key constraints, and the relationships between tables and their columns. Additionally, column data types and column constraints, such as uniqueness and optionality constraints, are useful information for schema matching. Finally, some matching algorithms examine the names of database elements via some form of string edit metric.

Information about database instances can be used in addition to, or instead of, schema-level information. Typically, identification of similar table columns is based on aggregate instance information, such as value distributions, term frequencies or averages, which is computed for all table columns and then used in similarity measures. For example, when two table columns contain the same distribution of values, then the columns may be argued to be similar. Machine learning techniques, such as neural networks or Bayesian learners can be used to establish characteristic features of an attribute.

External information is usually externally supplied through lexical databases such as WordNet or ontologies such as Cyc. These databases and ontologies provide lists of homonyms, synonyms and other semantic relationships between words or concepts. Such external information is most applicable to problems where schema element names are descriptive, rather than acronyms, or cryptic abbreviations.

THE IMPORTANCE OF THE COGNITIVE APPROACH

Schema matching methods and tools are evaluated by comparing their results to a set of reference matches (Evermann, 2008b). Their performance is typically assessed in terms of recall (how many of the reference matches are actually identified) and precision (how many non-matches have been incorrectly identified) (Evermann, 2008b). The construction of reference matches is done by human users and these reference matches therefore represent the output of a human cognitive process (Evermann, 2008b). Thus, schema matching methods and tools are judged as successful if they conform to this human cognitive process, i.e. produce the same output (Evermann, 2008). Hence, the identification of this cognitive process is important to guide the development and improvement of schema matching methods (Evermann, 2010; Lukyanenko and Evermann, 2011). However, many existing schema matching methods have been developed without a theoretical foundation and are based on the ad-hoc assumptions of their creators and an underlying theory, rooted in human cognition is lacking. They are rooted in human cognition only to the extent that they are the product of the researcher's intuition, but they neglect the existing knowledge of similarity judgment that systematic theorizing and empirical work in cognition has provided. Without this, there is little to guide the development of new and improvement of existing approaches.

The importance of identifying the human cognitive processes that underlie similarity judgments has only been recently recognized. A review of the schema matching literature approaches the similarity issue from the perspective of the similarity of the meaning of two database elements and consequently applied theories of meaning found in philosophy and cognitive psychology (Evermann, 2008b). That review pointed out the need for systematic empirical work in this area. Subsequently, experimental studies were unable to clearly identify a single theory of meaning held by database integrators (Evermann, 2008a, Evermann, 2009). A subsequent Delphi study explored the information that is used in schema matching (Evermann, 2010) and confirmed that the information used by existing schema matching methods and tools is indeed relevant to the human process of making matching decisions. However, overall, the approach based on theories of meaning as applied to the meaning of database elements, does not appear to yield conclusive results about the matching process. Accordingly, more recent work suggests that instead of focusing on the meaning of database elements, research might be based directly on cognitive theories and processes of similarity (Lukyanenko and Evermann, 2011). The paper by Lukyanenko and Evermann (2011) provided a review of major developments in cognition and presented avenues for further empirical work in this area. The present study can be seen as building on that foundation. In the next section, we describe a specific theory of cognitive similarity and explain its adaptation and application to the database context.

SIMILARITY AS INTERACTIVE ACTIVATION AND MAPPING

In this section, we provide an overview of the Similarity as Interactive Activation and Mapping (SIAM) theory of similarity (Goldstone, 1994; Goldstone and Medin, 1994) and apply it to the problem of schema matching. SIAM was developed as a

process model of human similarity judgment and has been shown to be superior to competing similarity theories in explaining the process and outcome of human similarity judgment (Larkey and Markman, 2005). Structure-mapping theory, of which SIAM is an instance, proposes three universal principles of human similarity judgment (Gentner, 1983; Gentner and Toupin, 1986; Markman and Gentner; 1993). First, SIAM is based on the principles of structural consistency. Structural consistency suggests that 1:1 mappings are preferred over 1:n mappings and that parallel mapping should be respected. Parallel mapping requires that if two relationships are mapped to each other, their participating elements must also be mapped. Second, SIAM is based on the principle of systematicity, which suggests that mappings of relationships and objects within relationships should be preferred over mappings of features. Third, the principle of tiered identicality suggests that relationships must ideally be matched exactly, but the exactness requirement can be relaxed if the relaxation allows larger sets of elements to be matched. These principles are based on systematic experimental evidence of human similarity judgments and a schema matching method based on cognitive principles ought to follow these principles.

SIAM and other cognitive theories of similarity, such as the Structure Mapping Engine SME (Falkenhainer, Forbus, Gentner, 1989) have originally been developed to evaluate simple *scenes* in which *objects* that possess *features* play a part in *roles* that relate the objects to each other. The features of an object belong to *dimensions* of the object. We have chosen to apply this to relational database as follows. The two database schemata to be matched are the two scenes, the tables within each schema are the objects within the scenes, the foreign keys between tables are the relationships between objects, the columns of the tables are the dimensions of the objects and the names and data types of the columns are the features of the objects. It is easy to see that this is not the only possible way of applying SIAM to relational databases, a point to which we return in the discussion.

We illustrate the principles of SIAM by examining the trivial match problem of the two databases in Figures 1 and 2. Each database schema comprises two tables, related by a foreign key relationship. A detailed description of SIAM is provided in the original work by Goldstone (1994) and Goldstone and Medin (1994) along with experimental evaluations.

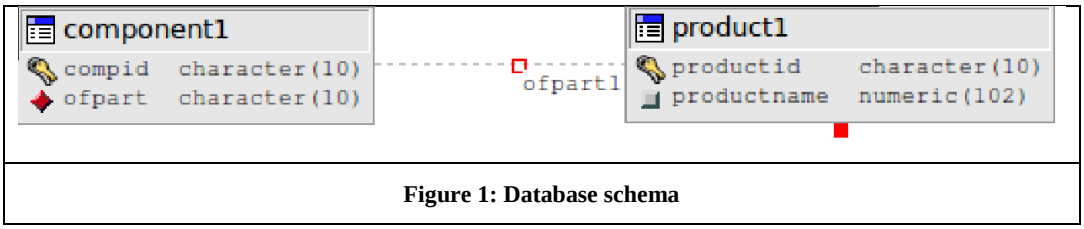

Figure 1: Database schema

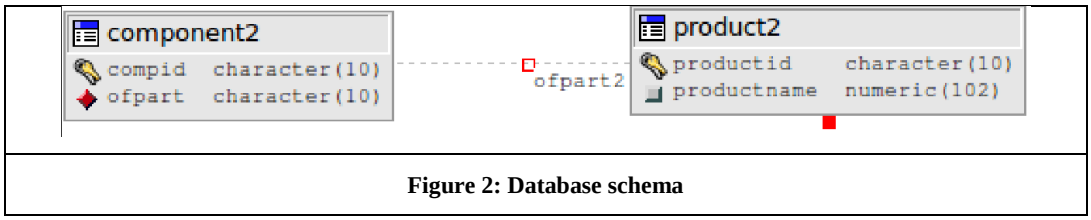

Figure 2: Database schema

The architecture of SIAM consists of a set of nodes. Each node relates either two objects (i.e. tables), two features (i.e. attributes) or two roles (i.e. foreign keys) and represents a *match hypothesis*. Thus, there is a set of object-object nodes, a set of role-role nodes, and a set of feature-feature nodes. Every node has an *activation value*, between 0 and 1, that expresses the similarity of the related objects, attributes, or roles.

For the initialization of the highest and lowest layers of nodes, surface similarity measures are used, whereas all nodes on the middle layer(s) are initialized to 0.5. In our case, the activation values at the lowest layer (i.e. those of feature-feature nodes relating database columns) are initialized by computing surface similarity metrics. We have chosen to compute attribute name similarity based on the Monge-Elkan string similarity metric (Monge and Elkan, 1996), which is frequently used in string matching problems. Additionally, we examine the similarity of data types. The initial activation values at the highest layer (i.e. those of role-role nodes) are also based on the Monge-Elkan name similarity, as well as on the similarity of the update and delete propagation rules for each foreign key constraint. The activation values of object-object nodes (relating database tables) are set to 0.5. Figure 3 below shows a matrix-like representation of the three sets of nodes with their initial activation

values, first for the object-object nodes, followed by the feature-feature nodes, and finally the single role-role nodes for this example.

Nodes are connected to other nodes. Specifically, each object-object node is connected to all other object-object nodes, to all role-role nodes, and to all feature-feature nodes. Each role-role node is connected to other role-role nodes and to all object-object nodes, but not to feature-feature nodes. Each feature-feature node is connected to other feature-feature nodes, and all object-object nodes, but not to role-role nodes. Thus, the SIAM architecture sets up a system of layers of nodes with connection within each layer and to the next higher and next lower layer, but not across multiple layers. It is easy to see that this system of layers can be generalized to more than three layers, a point to which we return in the discussion section.

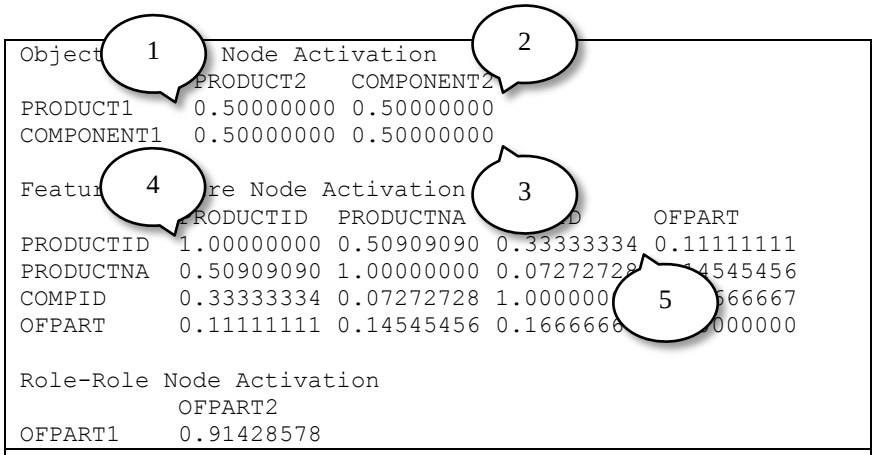

Figure 3: Initial node activations

Connected nodes are either consistent or inconsistent with a given node. For example, consider the object-object node that matches Product1 with Product2 (top left node in the object-object matrix in Figure 3, note 1). Now consider the object-object node that matches Product1 with Component2 (top right node in the object-object matrix in Figure 3, note 2). These two nodes are inconsistent, as the Product1 table cannot be matched to two different tables at the same time. Consider instead the object-object node that matches Component1 with Component2 (lower right node in the object-object matrix in Figure 3, note 3). This node is consistent with the Product1-Product1 node as the two match hypotheses do not conflict or contradict each other. Similarly, the Product1-Product2 node is consistent with the feature-feature node that matches ProductID to ProductID (top left node in the feature-feature matrix of Figure 3, note 4) as the matched attributes belong to the respective tables matched by the Product1-Product2 node. However, it is inconsistent with the feature-feature node that matches ProductID to CompID (pink highlight in the feature-feature matrix in Figure 3, note 5) as the matched attributes do not belong to the respective tables matched by the Product1-Product2 node.

The SIAM method proceeds iteratively from the initial activation values of nodes. Each node's activation value is updated by examining all the nodes it is connected to. In the simplest way, the new activation value of a node is calculated as

$$A_{i,t+1} = A_{i,t}(1 - L) + M_i L$$

Where L is a lag factor (between 0 and 1) and M_i is defined as

$$M_i = \frac{\sum R_{ji} W_{ji} S_j}{\sum W_{ji} S_j}$$

We can interpret M_i as the weighted average of recommended activations R_{ji} of all connected nodes. The weights W_{ji} can be used to influence the relative impact of object-object nodes on feature-feature nodes, etc. The salience values S_j can be used to make certain dimensions (and their features) more salient than others in the similarity computation. In our initial adaptation, all weights and saliences, as well as the lag factor L have been set to 1. The central part of the SIAM activation update calculation is the computation of the R_{ji} values that represent the activation of node i as recommended by connected node j. These recommendations differ for connected nodes j that are consistent with node i and connected nodes j that are inconsistent with node i and are computed as follows.

For consistent node connections:

$$R_{ji} = A_i + (1 - A_i)\big(A_j - 0.5\big) \text{ if } A_j > 0.5$$

$$R_{ji} = A_i - A_i\big(0.5 - A_j\big) \text{ if } A_j < 0.5$$

For inconsistent node connections:

$$R_{ji} = A_i + (1 - A_i)\big([\,1 - A_j] - 0.5\big) \text{ if } (1 - A_j) > 0.5$$

$$R_{ji} = A_i - A_i\big(0.5 - [1 - A_j]\big) \text{ if } (1 - A_j) < 0.5$$

It is important to note that the activation values of inconsistent nodes are not set to 0 immediately, as that would presuppose that the currently updated node i is in fact the correct match. Instead, inconsistent nodes recommend a lower activation whereas consistent nodes recommend a higher activation for the node whose update is currently considered.

The second iteration, updated nodes for our illustrative example are shown in Figure 4. The activation of the role-role node has not changed, as the initial activations of all connected nodes were identical at 0.5. The activation values of the object-object nodes have changed due to the differential influence of the feature-feature nodes. In turn, the activation values of the feature-feature nodes have changed due to the influence of other feature-feature nodes, as the initial activation of all connected object-object nodes were identical at 0.5. However, in the following iterations, the differential activations of consistent and inconsistent object-object nodes will have influence on the activation values of role-role and feature-feature nodes.

```
Object-Object Node Activation
            PRODUCT2   COMPONENT2
PRODUCT1    0.61101365 0.38898626
COMPONENT1  0.38898626 0.61101365

Feature-Feature Node Activation
           PRODUCTID  PRODUCTNA  COMPID     OFPART
PRODUCTID  0.88165867 0.48922619 0.35583913 0.18012877
PRODUCTNA  0.48922619 0.90547580 0.15193367 0.22074817
COMPID     0.35583913 0.15193368 0.92174369 0.24655323
OFPART     0.18012877 0.22074817 0.24655320 0.93748015

Role-Role Similarity
           OFPART2
OFPART1    0.91428578
```

Figure 4: Node activations in second iteration

The algorithm continues until the activation values of all nodes converge, i.e. the absolute changes in activation value between successive iterations for all nodes is less than a threshold (we have adopted 0.001 as the threshold change amount). Figure 5 shows the final activation values for all nodes.

```
Object-Object Node Activation
            PRODUCT2   COMPONENT 2
PRODUCT1    0.99973452 0.00026543
COMPONENT2  0.00026543 0.99973452

Feature-Feature Node Activation
           PRODUCTID  PRODUCTNA  COMPID     OFPART
PRODUCTID  0.74273050 0.73981845 0.25848016 0.25712132
PRODUCTNA  0.73981839 0.74357074 0.25697675 0.25778526
COMPID     0.25848019 0.25697675 0.74400651 0.73869342
OFPART     0.25712129 0.25778523 0.73869342 0.74455899

Role-Role Similarity
           OFPART2
OFPART1    0.99999988
```

Figure 5: Converged node activations (after 26 iterations)

After node activations have converged, the similarity of features, objects, and roles are determined as a combination of their final activation values and their surface similarity. The similarity of feature-feature and role-role nodes is computed as the product of their initial activation value (i.e. their surface feature) and their node activation value (i.e. the strength of their structural correspondence). The similarity of objects is determined as follows:

$$sim = \frac{\sum A_1 A_t S}{A_t S}$$

The sum is taken over all features of the two objects whose similarity is computed. The term A_1represents the surface similarity (i.e. the initial activation value) and, as above, A_t represents the activation of the object node at convergence time, and S represents the salience (in our case, all salience values are set to 1). Figure 6 below shows the final similarity values for our illustrative example.

```
Object-Object Similarity
            PRODUCT2   COMPONENT
PRODUCT1    0.75523901 0.16593461
COMPONENT   0.16593462 0.58530802

Feature-Feature Similarity
           PRODUCTID  PRODUCTNA  COMPID     OFPART
PRODUCTID  0.74273050 0.37663484 0.08616006 0.02856904
PRODUCTNA  0.37663481 0.74357074 0.01868922 0.03749604
COMPID     0.08616006 0.01868922 0.74400651 0.12311557
OFPART     0.02856903 0.03749604 0.12311557 0.74455899

Role-Role Similarity
           OFPART2
OFPART1    0.91428566
```

Figure 6: Final similarity values

Finally, to determine the best object-object mappings, the probability of each scene-to-scene mapping (i.e. a mapping of all objects in the scenes) is determined as follows:

$$P(M) = \frac{\sum_{i \in C(M)} A_i}{\sum_{d=1}^{n} A_d}$$

Here, the sum in the numerator is over the activation values A_i of all elements i of the consistent set of object-object nodes in the mapping M, denoted by $C(M)$. This is normalized by dividing by the total sum of activation values over all object-object nodes. In our example, there are only two consistent mappings, shown in Figure 7 with their respective P values (non-normalized).

```
Mapping with p = 1.999469
product1 --- product2
component1 --- component2

Mapping with p = 5.3086155E-4
product1 --- component2
component1 --- product2
```

Figure 7: Mapping probabilities (non-normalized)

A REALISTIC PROBLEM

The previous section has illustrated the principles of SIAM using a very simple example. To evaluate its performance on a more realistic problem, we have used the two database schemata shown below in Figures 9 and 10. The smaller schema in Figure 9 must be mapped to the larger schema in Figure 10 with the understanding that some elements of the larger schema

will be unmapped. The application of SIAM to this problem yields an object to object mapping that is shown in Figure 8 below. While the mapping is not perfect, it is an overall consistent mapping. We note that this specific matching problem does not have a single, unambiguously correct mapping solution.

```
source --- supplier
supplies --- supplies
buyer --- storage
shipment --- customer
part --- component
warehouse --- location
article --- product
```

Figure 8: SIAM best mapping

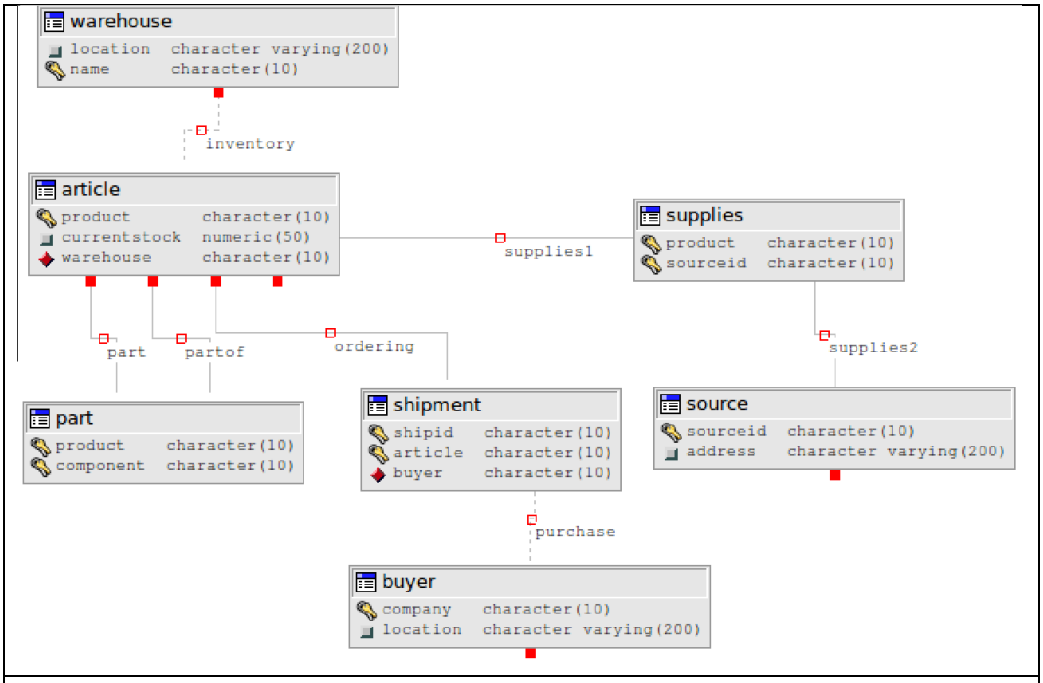

Figure 9:Database schema 1

DISCUSSION

This paper has made two contributions. First, we have argued for the importance and relevance of cognitive theories of similarity as a theoretical foundation for a research field in business intelligence that to date has been dominated by ad-hoc methods and heuristics. This theoretical foundation can form the basis for improvement and further development of schema matching methods. Second, the paper has demonstrated how one particular, well researched and accepted theory of similarity, Similarity as Interactive Activation and Matching (SIAM) can be applied to the database integration context.

While we selected the SIAM model due to its successful evaluation and application in the reference discipline of cognitive psychology, the SIAM model is also a good fit in terms of its architecture of objects, roles, and features to the database context with tables, foreign keys, and columns. One main difference is however that SIAM (and other theories of similarity) are not primarily intended to generate mappings but to calculate the overall similarity of two scenes. In database terminology, the main goal of the SIAM algorithm is to compute the overall similarity of two database schemata.

Another difference to the database context is that the SIAM model assumes that objects possess identical dimensions for which 1:1 mappings already exist. For example, SIAM was originally employed to visually compare scenes comprising butterflies. Here, the dimensions were the head, the wings, the tail, etc., each of which possessed certain visual patterns, the

features. In this context, it makes no sense to map the head of one butterfly to the tail of another. In the data integration context, columns take the place of dimensions and their characteristics (name, data types, etc.) take the place of features. We cannot assume that all tables have the same columns or even the same number of columns. We can also not assume an existing 1:1 mapping of column mappings but must instead identify and evaluate possible mappings.

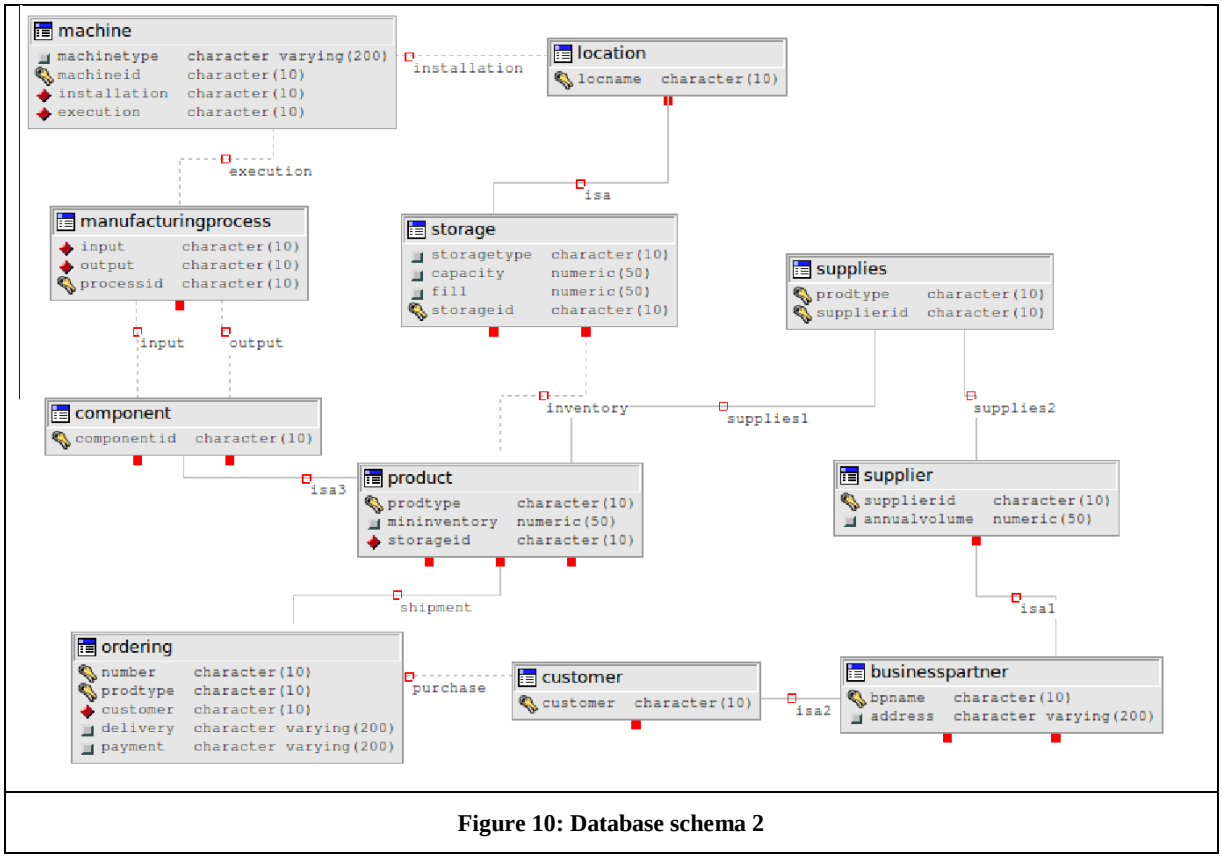

Figure 10: Database schema 2

We note a number of limitations to this paper, which point the way to future research. First, the research of similarity is an active area of investigation in the reference discipline of cognitive psychology (Gentner and Forbus, 2011; Lovett et al., 2009a, 2009b). Thus, the work presented here is based on a single snapshot in time. While this may mean that the specific theories that explain similarity judgment process may change in the future, cognitive theories remain valid reference theories and our argument for their importance remains valid. Schema matching algorithms will be considered successful and useful if they provide the same results as those of a human matching expert. Thus, the goal to understand human matching processes remains important. As we noted earlier, previous work in this area has identified a number of competing theories of similarity (e.g. Holyoak and Thagard, 1989; Falkenhainer, et al., 1989), and while SIAM has been demonstrated to be superior in experimental evaluations in cognitive psychology (Larkey and Markman, 2005), that superiority may not transfer to applications in the business intelligence context. Hence, future research needs to apply and evaluate other candidate reference theories in the data integration context. Further, theories of similarity are closely related to theories that explain metaphorical and analogical reasoning (Forbus et al., 2005; Gentner and Markman, 1997; Gentner and Toupin, 1986; Lovett et al, 2009a, 2009b; Spellman and Holyoak, 1992), enlarging the set of candidate theories further.

Second, the theory applied in this paper, SIAM, is extensively parameterized. For this initial application, we have achieved reasonable results with the default parameters. For example, we have assumed that all features are equally salient, we have assumed that every node to node connection has equal weight, and we have assumed a lag factor of $L = 1$. Every one of these parameters can be changed to improve our matching results. Given the fact that similarity matches are highly variable, it may even be possible that, in order to yield acceptable results with respect to user expectations, distinct sets of parameter values may be required for each user/data integrator.

Third, the chosen application of the SIAM architecture to the database context, i.e. mapping foreign keys to relationships, tables to objects, etc. was a conscious design decision. Other applications are possible. For example, on could map table ownership of columns to relations between objects, the columns to objects, the different types of constraints (uniqueness, optionality, check, and primary key constraints) to dimensions and the presence/absence of each constraint to the features. Other applications to the database context might also extend the SIAM model to allow for multiple types of relationships or different types of objects. Different applications and adaptations of the SIAM model will yield different results for the process and process outcome. The effects of this design decision on matching performance need to be systematically explored in future research.

Fourth, we have indicated that the SIAM architecture of connected layers of nodes is generalizable or extendible to more than three layers. This opens up the possibility of explicitly modeling, e.g. different types of roles other than foreign key constraints. One example could be modelling the ownership of columns by tables as explicit roles. This would allow the easier inclusion of primary keys and other column characteristics. Another example might be to split the roles between 1:n foreign keys and 1:1 foreign keys, i.e. foreign keys that relate the primary keys of two tables, so that these could be assigned different weights.

A final area for future research is the ongoing evaluation of the SIAM application to new problems and an experimental assessment of the results. Specifically, this implementation of SIAM needs to be evaluated using standard problems in schema matching. Moreover, while SIAM itself has been extensively evaluated in the cognitive psychology reference discipline, the type of problems considered there is substantially different than the schema matching problem. This points to the need to conduct our own experimental investigations and to compare the SIAM results against observed human schema matching processes and matching judgments.

In summary, we believe that this initial application of SIAM to the business intelligence and data integration field shows promise. We have demonstrated the suitability and feasibility of application and shown that the SIAM architecture is a good fit to the problem area of schema matching.

REFERENCES

1. Doan, A., and Halevy, A.Y. (2005) Semantic-integration research in the database community - A brief survey, *AI Magazine 26*, 1, 83-94.
2. Evermann, J. (2008a) An Exploratory Study of Database Integration Processes, *IEEE Transactions on Knowledge and Data Engineering*, 20, 1, 99-115.
3. Evermann, J. (2008b) Theories of meaning in schema matching: A review, *Journal of Database Management 19*, 3, 55-82.
4. Evermann, J. (2009) Theories of meaning in schema matching: An exploratory study, *Information Systems 34*, 1, 28-44.
5. Evermann, J. (2010) Contextual Factors in Database Integration — A Delphi Study, in J. Parsons, M. Saeki, P. Shoval, C. Woo and Y. Wand (Eds.) *Conceptual Modeling – ER 2010*, Springer Berlin / Heidelberg, 2010, 274-287.
6. Falkenhainer, B., Forbus, K.D., and Gentner, D. (1989) The structure-mapping engine: Algorithm and examples, *Artificial Intelligence 41*, 1, 1-63.
7. Forbus, K.D., Usher, J., and Tomai, E. "Analogical learning of visual/conceptual relationships in sketches," in: *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, AAAI Press, Pittsburgh, Pennsylvania, 2005, pp. 202-208.
8. Gentner, D. (1983) Structure-Mapping: A Theoretical Framework for Analogy, *Cognitive Science 7*, 2, 155-170.
9. Gentner, D. and Forbus, K.D. (2011) Computational models of analogy. *WIREs Cognitive Science 2*, 266-276.
10. Gentner, D., and Markman, A.B. (1997) Structure Mapping in Analogy and Similarity, *American Psychologist 52*, 1, 45-56.
11. Gentner, D., and Toupin, C. (1986) Systematicity and Surface Similarity in the Development of Analogy, *Cognitive Science 10*, 3, 277-300.
12. Goldstone, R.L. (1994) Similarity, Interactive activation, and mapping. *Journal of Experimental Psychology: Learning, Memory, and Cognition 20*, 1, 3-28.
13. Goldstone, R.L., and Medin, D.L. (1994) Time Course of Comparison, *Journal of Experimental Psychology: Learning, Memory, and Cognition 20*, 1, 29-50.

14. Holyoak, K.J., and Thagard, P. (1989) Analogical Mapping by Constraint Satisfaction, *Cognitive Science 13*, 3, 295-355.
15. Larkey, L.B., and Markman, A.B. (2005) Processes of Similarity Judgment, *Cognitive Science 29*, 6, 1061-1076.
16. Lovett, A., Gentner, D., Forbus, K., and Sagi, E. (2009a) Using analogical mapping to simulate time-course phenomena in perceptual similarity, *Cognitive Systems Research 10*, 3, 216-228.
17. Lovett, A., Tomai, E., Forbus, K., and Usher, J. (2009b) Solving Geometric Analogy Problems Through Two-Stage Analogical Mapping, *Cognitive Science 33*, 7, 1192-1231.
18. Lukyanenko, R. and Evermann, J. (2011) A Survey of Cognitive Theories to Support Data Integration. *Proceedings of the 17th Americas Conference on Information Systems, Detroit, MI.*
19. Markman, A.B., and Gentner, D. (1993) Splitting the Differences: A Structural Alignment View of Similarity, *Journal of Memory and Language 32*, 4, 517-535.
20. Monge, A. and Elkan, C. (1996) The field matching problem: Algorithms and applications. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 267-270.
21. Rahm, E., and Bernstein, P.A. (2001) A survey of approaches to automatic schema matching, *The VLDB Journal 10*, 4, 334-350.
22. Spellman, B.A., and Holyoak, K.J. (1992) If Saddam Is Hitler Then Who Is George Bush? Analogical Mapping Between Systems of Social Roles, *Journal of Personality and Social Psychology 62*, 6, 913-933.

APPENDIX: IMPLEMENTATION

The prototype for this research was implemented by the researcher in approximately 800 lines of Java code, of which only approximately 400 are used to implement the core SIAM algorithm, while the remainder performs input, output, and user interface functions. The prototype imports SQL databases using JDBC connections. It accesses an ANSI compliant information schema to identify tables, columns, and constraints. The present research was conducted using PostgreSQL databases, but MySQL, Oracle or any JDBC and ANSI compliant database is suitable. The source code is available from the author on request.