Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2013 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

6-18-2013

DYMS (Dynamic Matcher Selector) – Scenario-based Schema Matcher Selector

Youngseok Choi
Seoul National University, aquinas9@snu.ac.kr

DYMS (DYNAMIC MATCHER SELECTOR) – SCENARIO-BASED SCHEMA MATCHER SELECTOR

Youngseok Choi, College of Business Administration, Seoul National University, Seoul, Republic of Korea, aquinas9@snu.ac.kr

Abstract

Schema matching is one of the main challenges in different information system integration contexts. Over the past 20 years, different schema matching methods have been proposed and shown to be successful in various situations. Although numerous advanced matching algorithms have emerged, schema matching research remains a critical issue. Different algorithms are implemented to resolve different types of schema heterogeneities, including differences in design methodologies, naming conventions, and the level of specificity of schemas, amongst others. The algorithms are usually too generic regardless of the schema matching scenario. This situation indicates that a single matcher cannot be optimized for all matching scenarios. In this research, I proposed a dynamic matcher selector (DYMS) as a probable solution to the aforementioned problem. The proposed DYMS analyzes the schema matching scenario and selects the most appropriate matchers for a given scenario. Selecting matchers are weighted based on the parameter optimization process, which adopts the heuristic learning approach. The DYMS returns the alignment result of input schemas.

Keywords: schema matching, schema integration, matcher selection, schema matching strategy

1. INTRODUCTION WITH MOTIVATION

Schema matching involves matching among concepts which describe the meaning of data in various heterogeneous, distributed data (Gal 2006). Schema matching is recognized as one of the basic operations required in the process of data integration (Bernstein & Melnik 2004). This problem necessitates finding correspondence among elements of given schemas. A schema is a formal structure of an engineered artifact, such as SQL schema, XML schema, entity–relationship diagram, ontology description, interface definition, and form definition (Bernstein et al. 2011). Manually specifying schema matches is obviously a tedious, time-consuming, error-prone, and consequently, expensive process. This problem is becoming worse given the rapidly increasing number of Web data sources and e-businesses to integrate (Rahm & Bernstein 2001). Therefore, numerous researchers have tried to find more effective and efficient means of matching schemas automatically.

Schema matching has been a very active research area, particularly in the last decade, during which hundreds of techniques and prototypes for automatic matching have been developed (Rahm & Bernstein 2001). Although numerous advanced matching algorithms have emerged, schema matching research remains a critical issue. Various algorithms are implemented to resolve different types of schema heterogeneities, including differences in design methodologies, naming conventions, and the level of specificity of schemas, among others (Batini et al. 1986). The algorithms for matchers are usually too generic regardless of the schema matching scenario. This situation indicates that a single matcher cannot be optimized for all matching scenarios. A matching algorithm cannot be effective in all scenarios.

In this dissertation, I propose a dynamic matcher selector (DYMS) that provides effective matching results based on the optimal combination of existing matchers by reflecting the features of the matching scenario (such as the features of the input schemas). In Section 2, the theoretical foundations of schema matching research are presented. Classifications of existing schema matching research are introduced in Section 3. In Section 4, the details of the DYMS architecture are described using a figure. The progress in implementing the DYMS and future plans for completion are summarized in Section 5. The conclusion and expected contributions of this research are presented in Section 6.

2. THEORETICAL FOUNDATIONS OF SCHEMA MATCHING RESEARCH

Several attempts at setting theoretical foundations for schema matching exist in literature. The theoretical aspect of schema matching research can be divided into model management and operation matching. Most theoretical foundations are related to symbolic algebra.

2.1. Model Management

Model management is a framework for supporting applications related to metadata, wherein models and mappings are manipulated as first-class objects using various operations (Bernstein & Rahm 2000). A model is a structure representing a designed artifact, such as an XML DTD and Web site schema, among others. A number of mathematical foundations make it easier to manage such models. Many examples of high-level algebraic operations are currently being used for specific metadata applications (Jannink et al. 2009; Miller et al. 1994; Mitra et al. 2000).

Bernstein et al. (2000) suggested key algebraic operations for model management. Fundamental operations include:

- Matching – taking two models as input and returning a mapping between them as output;
- Composing – taking two mappings as input and returning their composition as output;

- Merging –taking two models and a mapping between them as input and returning a model merging the two models (using the mapping to guide the merging) as output;
- Setting operations on models – involves unions, intersections, and differences; and
- Projecting and selecting models – which are comparable to relational algebra.

These algebraic operations manipulate models and their mappings, each of which connects the elements of two models. A matching operation, which is one of the most important operations in model management, can be defined in a more formal manner. The details of matching operations are discussed in the next subsection.

2.2. Matching Operation and Matching Process

The foundations of a matching operation in a schema matching process can be found in numerous related studies. Based on the research of Shvaiko et al. (2005) and Euzenat (2004), a matching operation can be defined as a quintuple: <*id, e, e', n, R*>, where

- *id* is a unique identifier of a given mapping element;
- *e* and *e'* are the entities (tables, XML elements, properties, classes) of the first and second schema, respectively;
- *n* is a confidential measure in several mathematical structure holdings for the correspondence between entities *e* and *e'*; and
- R is a relation [equivalence (=), more general (⊒), disjointness (⊥), overlapping (∩)] holding between entities *e* and *e'*.

Based on this matching operation, we can define the matching process which determines the alignment (*A*) for a pair of schemas. Alignment is a set of mapping elements (Shvaiko et al. 2005). The general matching process is depicted in Figure 1.

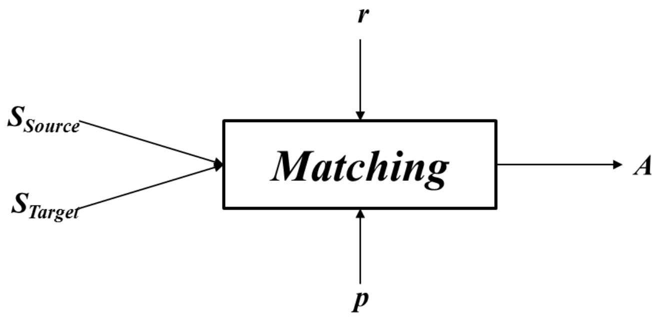

Figure 1. The matching process.[1]

The input of this matching process is a pair of schemas [source (S_{Source}) and target (S_{Target})]. External resources or auxiliary information (*r*) (such as thesauri, ontology) can be used by the matching process. To determine the result of the matching process (alignment, *A*), the relation among mapping elements is defined based on the matching parameters (*p*) in advance.

[1] This figure was modified based on Figure 2 in Shvaiko et al. 2005. The descriptions for this figure were also refined and modified.

3. RELATED STUDIES

Existing schema matching techniques are shown in Rahm & Bernstein (2001). An implementation of a schema-matching process may use multiple matching algorithms, thus allowing us to select individual matchers depending on the matching scenario (Rahm & Bernstein 2001). Rahm and Bernstein (2001) suggested the following classification for schema matching techniques (Figure 2).

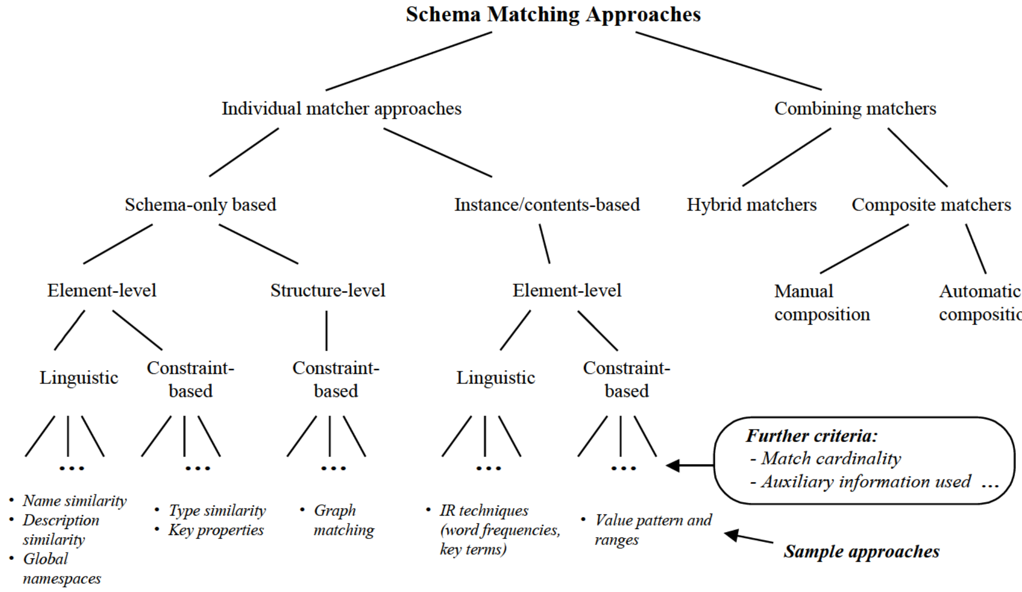

Figure 2. Classification of schema matching approaches (Rahm & Bernstein 2001).

For individual matchers, Rahm & Bernstein (2001) considered the following largely orthogonal classification criteria:

- **Instance vs. schema**: Matching approaches can consider instance data or only the information of the schema element.
- **Element vs. structure matching**: Matches can be performed for individual schema elements or for combinations of elements, such as complex schema structures. Structural matching mainly uses the topological approach.
- **Language vs. constraint**: A matcher can use a linguistic-based approach (such as those based on names and textual descriptions of schema elements) or a constraint-based approach (such as those based on keys and relationships).
- **Matching cardinality**: The overall match result may relate one or more elements of one schema to one or more elements another, thus yielding four cases: 1:1, 1:n, n:1, and n:m. In addition, each mapping element may interrelate one or more elements of the two schemas. Furthermore, different match cardinalities may be found at the instance level.
- **Auxiliary information**: Most matchers do not only rely on input schemas S_1 and S_2, but also on auxiliary information, such as dictionaries, global schemas, previous matching decisions, and user input.

4. THE DYMS ARCHITECTURE

The DYMS is the implementation of a schema matching process based on the analysis of a schema matching scenario. The DYMS selects schema matchers that are most appropriate for a schema matching scenario and combines the selected matchers using the optimizing parameter. The DYMS is expected to offer a noble approach to schema matching techniques that use multiple matchers. Although previous studies combining matchers do not consider the features of input schemas (such as the schema matching scenario), the DYMS suggests the means of optimally selecting a matcher based on the given schema matching scenario. The DYMS has three sub-modules (**scenario analyser, matcher selector,** and **parameter optimizer**) and one external resource. The details of the DYMS architecture are shown in Figure 3.

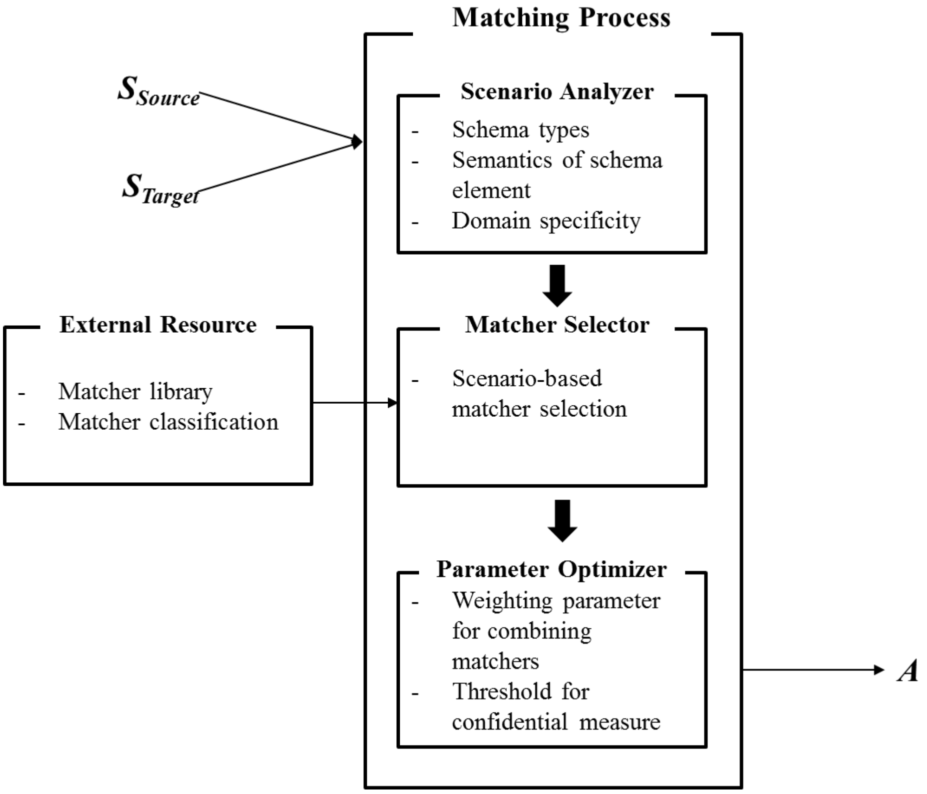

Figure 3. The architecture of the DYMS.

4.1. Analyzing the Matching Scenario–Scenario Analyser Module

The first step in the matching process is analyzing the features of input schemas. In this research, the schema matching scenario indicates the features of given input schemas. The **scenario analyser** module analyzes input features to select the most appropriate matcher for the features of the input schemas. This module determines what type of schema (XML schema, ER schema, ontology, Web schema, and so on) the input is. Then, this module judges whether schema elements have semantic (meaning). Based on this judgement, the **matcher selector** decides whether linguistic matchers will be selected. In addition, the possibility of using instance data of given schemas can be another important feature for selecting matchers. All analysis results of the **scenario analyser** are used by the **matcher selector** module.

4.2. Scenario-based Matcher Selection–Matcher Selector Module

The **matcher selector** module selects the existing matcher from the matcher library and matcher classification based on the analysis results of the **scenario analyser** module. The matchers in the library will be classified based on their suitability for the matching scenario.[2] The matchers will be selected according to the result of the scenario analysis.

4.3. Parameter Optimization–Parameter Optimizer Module

Based on the selected matchers from the **matcher selector**, the **parameter optimizer** module determines the weighting factor for each matcher and the threshold value for confidential measurement. This module adopts the heuristic learning approach for calculating the optimal parameter for a given schema matching problem.

5. CURRENT STAGE OF THE RESEARCH AND PLANS FOR COMPLETION

The current implementation level of the DYMS architecture and plans for completion are presented in this section. The details are shown in Table 1.

Modules	Description	Plan and Progress
Scenario Analyser	**Schema type checker:** Confirming which type of schema (XML schema, ER schema, ontology, Web schema, and so on) the input is.	Done
	Semantic checker: Confirming whether schema elements have semantic/meaning or do not use auxiliary information (such as dictionary, WordNet)	February 2013 to March 2013; Progress : 80%
	Checking the other features of input schemas (domain specificity, existence of instance data, and so on)	March 2013 to July 2013
External Source	**Matcher library:** Selecting candidate matchers from more than 200 schema matching papers	December 2012 to May 2013; Progress: 50 %
	Matcher classification: Rebuilding the classification of Rahm and Bernstein (2001). Matching scenario-driven criteria will be used for this classification. The result of this classification will be the literature review section of my doctoral dissertation.	
Matcher Selector	This module selects the matcher using the result from the **scenario analyser** and **external source**. After finishing the implementation of the **scenario analyser** and **external source**, the **matcher selector** can be implanted.	May 2013
Parameter Optimizer Module	**Heuristic learner for parameter optimization:** Implementing the	May 2013 to June 2013

[2] Matcher library and classification will be rebuilt using existing literature. The classification from Rahm & Bernstein (2001) is not appropriate for the proposed approach. The input features will be the criteria for rebuilding the classification of the existing matcher.

	calculation algorithm based on heuristic learning. Heuristic learner uses the results of the pre-experiment which combines selected matchers. Based on the learning data and the results, the optimal weighting parameter for each matcher will be determined.	
	Determining the threshold for confidential measure: Determining the threshold based on a number of experiment results. The threshold value for confidential measurement should fully reflect the statistical nature of the confidential measure. Therefore, numerous experiment results are required for determining the threshold.	June 2013 to September 2013

Table 1. Current stage of the research and plans for completion.

6. CONCLUSION

Despite two decades of research in schema matching fields, most schema matching studies still seem to involve ad-hoc solutions (Gal 2006). More than 200 schema matching articles have been published, and each suggests a limited solution to their own matching scenario. Researchers have already made sufficient matchers. Therefore, reusing and combining the existing hundreds of matchers should be the next focus of studies.

In this research, I proposed a DYMS architecture. The proposed DYMS can provide a generic solution for any kinds and scenarios of input schema by selecting an optimal matcher. Matchers are selected based on the features of the input schema (such as the scenario of schema matching), and the parameters for the selected matcher are calculated. The result of the DYMS architecture is the alignment of input schemas, *A*.

This research is expected to offer a novel and effective means of combining existing schema matching techniques. We cannot develop a schema matching algorithm applicable to all possible situations. However, sufficient algorithms for specific situations exist. In this manner, the DYMS is developed based on the design science paradigm (Hevner et al. 2004). To find the optimal solution for a schema matching problem, the DYMS selects existing alternatives in a reasonable manner. I believe that the DYMS can be milestone that can change the direction of schema matching research.

References

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

Bernstein, Philip A., and Rahm, E. (2000). Data Warehouse Scenarios for Model Management. In *Proceedings of Conceptual Modeling-ER 2000: 19th International Conference on Conceptual Modeling*, Salt Lake City, Utah, USA, October 9-12, 2000, Vol. 1920. Springer.

Bernstein, P. A., Halevy, A., and R. A. Pottinger. (2000) A Vision for Management of Complex Models. *ACM SIGMOD Record*, 29(4), 55-63.

Bernstein, P. A., and S. Melnik. (2004). Meta data management. In *Proceedings of the IEEE CS International Conference on Data Engineering*. IEEE Computer Society.

Bernstein, P. A., Madhavan, J. and Rahm, E. (2011). Generic schema matching, ten years later. In *Proceedings of the VLDB Endowment 4.11*, 695-701.

Euzenat, Jérôme. (2004) An API for ontology alignment. In *Proceedings of the Semantic Web–ISWC 2004*, 698-712.

Gal, Avidgor. (2006). Why is Schema Matching Tough and What Can We Do About It? *SIGMOD Record*, 35(4), 1-11.

Jannink, J., Mitra, P., Neuhold, E., Pichai, S., Studer, R., Wiederhold, G.(1999) An Algebra for Semantic Interoperation of Semistructured Data. In *Proceedings of 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, 77-84.

Miller, R., Ioannidis, Y.E., Ramakrishnan, R. (1994) Schema Equivalence in Hetereogeneous Systems: Bridging Theory and Practice. *Information Systems* 19(1), 3-31.

Mitra, P., Wiederhold, G., Kersten, M. (2000). A Graph-Oriented Model for Articulation of Ontology Interdependencies. *In Proceedings of Extending DataBase Technologies, EDBT 2000*, LNCS, Springer Verlag

Rahm, E. and Bernstein, P. A. (2001). A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal*, 10(4), 334-350.

Shvaiko, P. and Jérôme Euzenat. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics IV* , 146-171.