Australian Journal of Basic and Applied Sciences, 7(7): 515-519, 2013
ISSN 1991-8178

Semi-Automated Schema Integration (Icase): A Tool To Identify And Resolve Naming Conflicts

Said Tahat, Dr. Kamsuriah Ahmad

Faculty of Information Science and Technology, University Kebangsaan Malaysia

Abstract: Creating global schema over a set of heterogeneous and distributed database entails dealing with a different data model or different structure. The most important issue in schema integration is how to identify correspondences among schema element names. Element names may be given a different meaning at different schemas. For example, the noun Area based on the WorldNet dictionary has five word-senses (location, square feet, country, region and sphere), so how is it ensured that the concept name Area in different schema is used in the same meaning. In this paper, a new approach called (ICASE) Identify Correspondences among Schema Elements, ICASE takes the challenge to solve the problems of naming conflict which includes homonym conflict and synonym conflict. Homonym conflict arises as a result of using the same name to represent different elements while the synonym conflict means the presence of two elements in different schemas, represented by different names. WorldNet dictionary helps ICASE to represent the intended meaning of the element name since fully automatic schema integration is still impossible due to the huge amount of semantic diversity in designing databases. A friendly user interface is built to allow the user to interact with the system and to automate the system.

Key words: Naming conflicts**,** schema integration**,** schema redundancy, semantic, global view.

INTRODUCTION

The increase in the number of databases has entailed the management of related data in different formats across these databases. Many designers preferred to use relational model even though there are many database models. The relational model is the most dominant model for many database applications; applications for distributed environment in particular, are still adopting this model due to its simplicity compared to other models. Meanwhile, the available databases have been developed for distributed environment which may lead to heterogeneity conflicts. According to (Batini, Lenzerini, & Navathe, 1986), heterogeneity conflicts on schema can be categorized into two types: naming conflicts and structural conflicts. The naming conflicts include synonym and homonym conflicts.

Hence, it is vital for an organization to use other organizational data for better decision-making and success, and consequently, they need to understand the semantics and retrieve data from these other distributed and heterogeneous data sources (Ozgul Unal & Afsarmanesh, 2010). Therefore, it is very significant to integrate or interconnect these databases or enable interoperability between them. It is noteworthy that the demand of collaboration among organizations is apparently escalated. In order to create a correct, complete, minimal and understandable unified global schema (Peter Bellström, 2010), the problem of naming conflicts needs to be resolved and this is the main task of this study. Therefore, the aim of this paper is to propose a method for recognizing the homonyms and synonym conflicts in schema integration.

2. Schema Integration:

Combining multiple database schemas into a global schema is a big challenge in database sharing (Ahmad, Chiew, & Samad, 2011) due the diversities of these source schemas, such as different formats, with different meanings, and references using different names. Subsequently, schema integration must handle the structural conflicts, naming conflicts, naming conflicts that are difficult to recognize and resolve, and it is required to be aware of the intended meaning of the name for each element (O. Unal & Afsarmanesh, 2009; Ozgul Unal & Afsarmanesh, 2010). The naming conflict on schema integration scenario is divided into types: homonym conflict and synonym conflict, Most of the previously developed models to solve schema integration issues (Chiticariu, Kolaitis, & Popa, 2008; Melnik, Rahm, & Bernstein, 2003; R. Pottinger & Bernstein, 2008) (Madhavan, Bernstein, & Rahm, 2001; R.A. Pottinger & Bernstein, 2003) (Saleem, Bellahsene, & Hunt, 2008; Ozgul Unal & Afsarmanesh, 2010) aim to handle schema integration of relational databases. However, these approaches are not generic enough and still have limitations. Instead of concentrating on the integration result and providing a complete solution, these approaches concentrate on the schema integration process by reducing the amount of manual work in order to reduce the time and cost. For instance, (Gardarin, Gannouni, Finance, & Fankhauser, 1995) only considers structural conflict, and assumed that correspondences among source schemas

Corresponding Author: Said Tahat, Faculty of Information Science and Technology, University Kebangsaan Malaysia
E-mail: Saed_tahat@yahoo.com

are already given.(Dou & LePendu, 2006) considered the ontology as a possible solution to represent the content of heterogeneous data sources, using ontology as the mediated schema for integration. Using the ontology is not generic enough because firstly, we need to analyze the similarities and differences among relational model, and then discuss how to map relational schema to ontology. Most of the existing approaches ignore the problems of naming conflict during the integration, they assume that inter-schema name correspondences are established during pre-integration phase, and then no naming conflicts can arise.

The proposed approach in this paper aims to resolve the naming conflict which is deduced from the schemas to enhance the integration result. In this study, the problem of recognizing homonym and synonym conflict that exists in the element name of two different schemas is explored. Homonym conflict arises as a result of the representation of two different elements on different schemata by using the same name. Synonym conflict arises when the same element are represented by different names in different schemata (P. Bellström, 2005). For example in schema S1 there is an element name Area and in schema S2 there is also the same element name Area however Area in both schemata represents different concepts. For instance, Area in S1 means Location while on the other hand, in S2, Area is used to represent square-feet. Another example for synonym conflict is Customer and Client. In schema S1 there is an element name, Customer and in schema S2 there is also the same element name Client. In fact Customer and client in both schemata represent the same concept; Fig. 1 shows two database schemata that have homonym and synonym conflict in relational representation.

Getting awareness of the meaning of the element name will depend on how the user interprets the meaning. If we are going to integrate these schemas together, then a special method is required to identify the occurrence of naming conflict before the schemas are integrated, otherwise redundancy or consistency of schema will be appear on the global schema. In order to integrate these two schemas into a global view, an integration method is needed to solve this diversity of schema.

3. Proposed Method:

An integration tool called (ICASE) is proposed to identify correct correspondences among schema elements to create a correct, complete, minimal and understandable unified global schema in consideration of the problems of naming conflicts (homonym and synonym). According to (Batini, et al., 1986), there are two types of integration strategies; binary and n-binary strategies. The proposed method follows the binary strategy where two schemas are analyzed and integrated at the same time.

Figure 1 presents the proposed method which consists of four steps, namely: extraction and translation, comparison, renaming and integration. As a case study, two heterogeneous relational database schemas related to each other (from the same domain), both having conflicts - homonyms and synonym are involved. Note that only a part of the schema is shown due the space limitation.

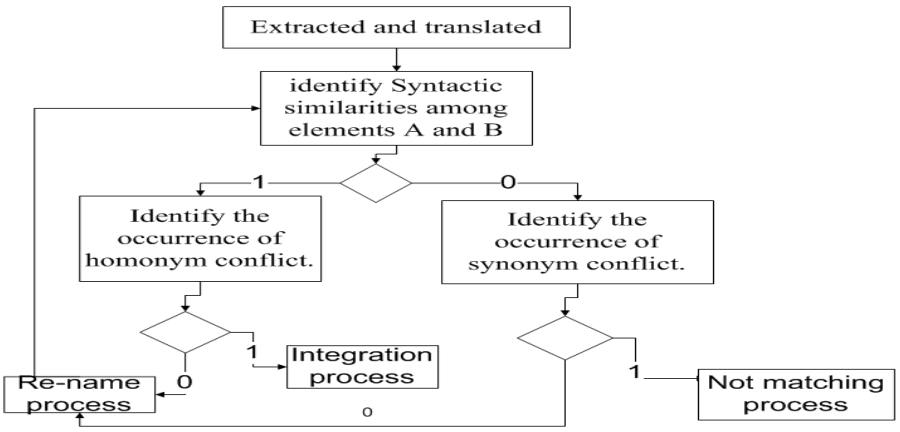


Fig. 1: The flows of integrating the elements of relation schemas.

Table from schema B: home_information (home_ID, Area, Square-feet, Customer,Ad_target)
Table from schema B: Home_information (home_ID, Area, Location, Client, Ad_target)

Step1: Extraction and translation: the responsibility of this process is to extract two schemas L and G, and translate them into a Directed Acyclic Graph (DAG) format as shown in figure 2, by using JGraphT, a free Java graph library (Ozgul Unal & Afsarmanesh, 2010).Then both schemas are loaded to the next step where the schema G is taken as the base schema in the integration process, which will be the global schema.

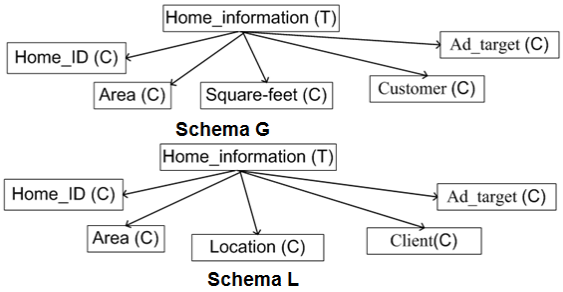


Fig. 2: G and L schemas in graph format.

Step 2: involves the comparison process which is the most important step in ICASE. Figure 1 presents the flows of comparison process based on our approach. There are two levels: Syntactic and Semantic level. The semantic level is further divided into two processes, homonym and synonym identifier.

(i) *Syntactic level:* this takes two string names A and B as input, and it checks whether or not the two strings are similar to each other. If they are identical, the syntactic similarity is set to 1; otherwise, syntactic similarity is set to 0. The syntactic similarity does not decide whether both elements' names are identical or not, it just decides which conflicts (Homonym or synonym) will be considered on the semantic level. Table 1 presents the syntactic similarity between schema G and schema L.

Table 1: Syntactic similarity.

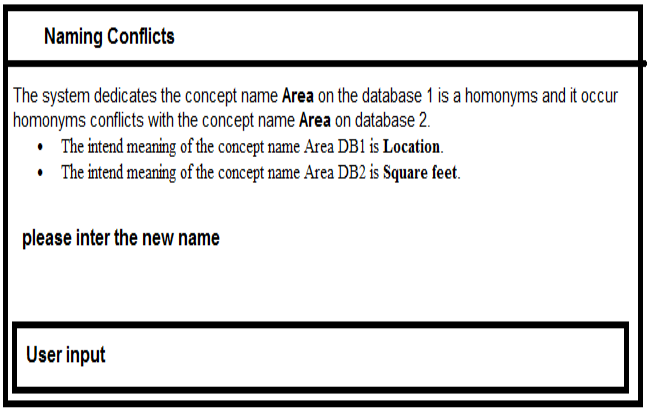
G/L	*Home_ID*	*Area*	*Location*	*Client*	*Ad_target*
Home_ID	*1*	*0*	*0*	*0*	*0*
Area	*0*	*1*	*0*	*0*	*0*
Square-feet	*0*	*0*	*0*	*0*	*0*
customer	*0*	*0*	*0*	*0*	*0*
Ad_target	*0*	*0*	*0*	*0*	*0*

(ii) *Semantic level:* the main task here is identifying the occurrence naming conflict, by using the WorldNet dictionary and if the result obtained from the syntactic level is 1, the semantic level will identify the occurrence of homonym conflict, by the homonym process. On the other hand, if the result is 0, the semantic will identify the occurrence synonym conflict by the synonym process, based on the result of the homonym or synonym process. The semantic similarity is set to 0 if the manning conflicts occur; otherwise, semantic similarity is set to 1.

Homonym process: by using WorldNet dictionary, this process retrieves the word-sense related to element name A and compares it with all elements in the same schema. If any related word-sense matches with any element name of the schema, that means the homonym conflict is detected and needs to be solved; otherwise, no conflicts exist and the final result of homonym process is set to 1. For instance, we take the name Area, because the syntactic similarity is 1. The homonym process begins, retrieving Area (location, square feet, country, region and sphere), comparing it with other elements in schema A. We will find that the related word-senses square feet will match with square feet element in schema G, and hence, the semantic is set to 0.

Synonym process: by using WorldNet dictionary, this process retrieves the synonym word related to element name A and compares it with all the elements in the other schema B. If any related synonym word matches with any element name of the schema B, it means that the synonym conflict is detected and needs to be solved; otherwise, there is no conflict and the final result of homonym process is set to 1. For example, we take the name customer from schema G with client from schema L with the syntactic similarity at 0. The synonym process begins and retrieves the related word sense customer (client, purchaser, buyer, province, shopper and consumer), and compares it with client. If it is clear then the semantic similarity will be set to 1.

Re-naming: the responsibility of this process is to solve detected conflicts (homonym or synonym), through the re-name strategy using to solve naming conflicts. The system will ask the user about the new name for both conflicts. For instance, in Fig. 2, the Area element occurs as homonym conflicts and needs to be re-named. Figure 3 shows the interaction between the user and the system by solving conflict screen.



After the user enters the new name, the system will return the new name to the syntactic process to compare it again with other elements.

Integration process: this process maps schema L into schema G (global schema) after the correspondences among schema elements are identified. Any element in schema L corresponding to any element in schema G will be removed from schema L. If the elements in schema L do not exist in schema G then it will map directly to G. At the end of schema L, the system can load other schema to integrate it.

4. An Example for Evaluation:

As an evaluation, we run the developed approach manually by using the schema in Fig 2 in the previous section. The global schema in Fig 4 shows that (ICASE) is able to produce the output correctly. (ICASE) is able to resolve the existence of naming conflict that appears in the locale schema.

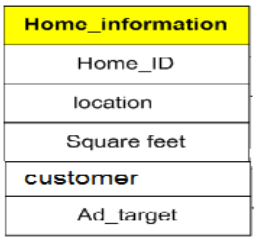


Fig. 4: The (ICASE) output.

In the evaluation, two quality attributes have been considered to evaluate the effectiveness of the proposed approach namely, completeness and correctness. Completeness means the global schema should cover all concepts of local schemas [20] and shows how complete the alignment was found in terms of finding all correct matches; it can be defined by the following formula:

$$Completeness\ (Recall) = \frac{TP}{FN + TP}$$

Where TP is a number of correctly found mappings (True Positive) and FN is a number of not found mappings which should be found (False Negative). At the same time, Correctness (or Precision) shows how much the matching is correct, according to. Correctness can be defined by the following formula:

$$Correctness\ (Precision) = \frac{TP}{FP + TP}$$

Where FP is the number of found mappings which should not be found (False Positive), while TP is a number of correctly found mappings (True Positive). The results of both quality attributes are analyzed, as in Table 2.

Table 2: Quality attributes results.

quality attributes	formula	result
Completeness	$\frac{7}{0+7}$	1
Correctness	$\frac{7}{0+7}$	1

As can be seen from Table 2, the (ICASE) method is able to produce the output correctly. ISI is able to detect the existence of homonym and synonym conflict that appears in the source schema. Furthermore, it is able to create a correct, complete, minimal and understandable unified global schema.

5. Conclusion and Future Work:

In this paper, a new approach for identifying and resolving homonym and synonym conflict in schema integration is proposed. The proposed approach identifies the occurrence of homonym and synonym conflict by using WorldNet dictionary during the comparison, and asking the designer which elements he wants to rename. The significant contribution of this research work is a new technique that has the ability to detect and resolve naming conflict effectively. Furthermore, compared with the existing schema integration approach, the proposed approach has sufficient power to create a correct, complete, minimal and understandable unified global schema. In the future, we are looking to resolve other conflicts such as key constraints, cardinality, structural and type constraints.

REFERENCES

Ahmad, K., H.K. Chiew, R. Samad, 2011. *Intelligent Schema Integrator (ISI): A tool to solve the problem of naming conflict for schema integration*.

Batini, C., M. Lenzerini, S.B. Navathe, 1986. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv., 18*(4): 323-364.

Bellström, P., 2005. Using Enterprise Modeling for identification and resolution of homonym conflicts in view integration. *Information Systems Development*, 265-276.

Bellström, P., 2010. *Schema Integration.* Unpublished DISSERTATION, Karlstad University Studies.

Chiticariu, L., P.G. Kolaitis, L. Popa, 2008. *Interactive generation of integrated schemas*. Paper presented at the Proceedings of the 2008 ACM SIGMOD international conference on Management of data.

Dou, D., P. LePendu, 2006. *Ontology-based integration for relational databases*.

Gardarin, G., S. Gannouni, B. Finance, P. Fankhauser, 1995. *IRO-DB-A Distributed System Federating Object and Relational Databases*.

Madhavan, J., P.A. Bernstein, E. Rahm, 2001. *Generic Schema Matching with Cupid*. Paper presented at the Proceedings of the 27th International Conference on Very Large Data Bases.

Melnik, S., E. Rahm, P.A. Bernstein, 2003. *Rondo: a programming platform for generic model management*. Paper presented at the Proceedings of the 2003 ACM SIGMOD international conference on Management of data.

Pottinger, R., P.A. Bernstein, 2008. *Schema merging and mapping creation for relational sources*. Paper presented at the Proceedings of the 11th international conference on Extending database technology: Advances in database technology.

Pottinger, R.A., P.A. Bernstein, 2003. *Merging models based on given correspondences*. Paper presented at the Proceedings of the 29th international conference on Very large data bases - Volume 29.

Saleem, K., Z. Bellahsene, E. Hunt, 2008. PORSCHE: Performance ORiented SCHEma mediation. *Inf. Syst., 33*(7-8): 637-657.

Unal, O., H. Afsarmanesh, 2009. Schema matching and integration for data sharing among collaborating organizations. *Journal of software, 4*(3): 248-261.

Unal, O., H. Afsarmanesh, 2010. Semi-automated schema integration with SASMINT. Knowledge and Information Systems, 23(1): 99-128.