2011 International Conference on Electrical Engineering and Informatics
17-19 July 2011, Bandung, Indonesia

Intelligent Schema Integrator (ISI): A Tool to Solve the Problem of Naming Conflict for Schema Integration

Kamsuriah Ahmad[#1], Hea Khim Chiew[#2], Reduan Samad[*3]

Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
[1]kam@ftsm.ukm.my
[2]chiewmail@yahoo.com
[*]*School of ICT*
Asia e-University, Malaysia
[3]reduan.samad@aeu.edu.my

Abstract - **The data stored in the data warehouse are mostly coming from different sources. It may be developed using different model or structure for the schema. In order to improve the usability of these data, the process of combining or integrating is needed so that it can provide users with a unified view or a global view of these data. The most important issue in data integration is the schema integration: that is to solve the problem of "how can equivalent real-world entities from multiple data sources be matched up?" This is referred to as entity identification process. Terms may be given a different interpretation at different sources by different people. For example, how can data analyst be sure that *customer_id* in one database and *cust_number* in another refer to the same entity? In this paper, a tool which is called an Intelligent Schema Integrator (ISI) is built to increase the uses of data from the data warehouse and to make the process more simple, systematic and impressive. ISI is an intelligent tool which can be used to integrate two different schemas from different sources into a unified schema (global schema). ISI is developed to solve the problems of naming conflict which are homonym conflict and synonym conflict. Homonym conflict means the same element name is used to represent different concept. Synonym conflict means different element name is used to represent the same concept. Thesaurus is used to get the meaning of each element concept and compares it with the other concept. An interface is built to allow the user to choose which elements are going to be renamed or removed, if there are occurrences of homonym and synonym conflicts in the schemas. These are the intelligence features built for ISI. The methodology used in this study consists of 4 phases: Design the Input and Output, Extraction, Comparison, and Integration. The development of this tool is an important direction for more efficient and effective implementation of data integration in data warehousing.**

Keywords: **schema integration, homonym conflict, synonym conflict, naming conflict**

I. INTRODUCTION

Data warehouse is a repository to store and to process important data from all aspects of organization. For instance if the organization is from educational institution, the stored data might be the student information, academician, and daily operation of the institution. The data stored in data warehouse environment will be processed to produce valuable information to increase the knowledge of particular organization. The introduction of the World Wide Web (WWW) in 30th April 1993 enables us to achieve and to share information without boundaries. This capability has increased the storage of organization data in the data warehouse [8].

Extensible Markup Language (XML) is widely used as an intermediate language in a web environment. Hence, most data sources stored in data warehouses are based on XML. The advantage of this is that data can be shared among the organizations. But the problem is the diversity of the stored data consists of heterogeneous data sources. The data are derived from different sources and may use different schema structures and terminology. If the sharing of data between organizations is needed, these data should be combined and integrated into a unified one. This unified schema will enable users to access a single global view to a set of distributed, but related data. This requires the development of reliable and scalable schema integration to enhance the exchanging of data between homogeneous and heterogeneous databases, enabling communication between applications or organization. This paper will be organized as follows: section II reviews previous work in schema integration. In section III states the pending issues in schema integration and the objectives of this paper. Section IV discusses the development of ISI. While in section V evaluate the performance of ISI and section VI concludes the paper and states the future works.

II. SCHEMA INTEGRATION

Schema integration is a big challenge in database industry. It faces two major problems which are the structural and semantics diversities of source schemas that to be merged.

978-1-4577-0752-0/11/$26.00 ©2011 IEEE

Semantic similarities and differences are difficult to recognize and resolve, it needs to understand the intended meaning of a concept for each element ([5], [6]). The semantic conflict of schema during the schema integration can be in terms of naming conflict (homonym and synonym), type conflict, key and cardinality conflict, and etc. Most of the previously developed integration systems, such as Tukwila [9], DIXSE [8], LoPix [7], LSD [1] and DIKE [3], aim to handle integration of XML databases. Even though these systems have achieved certain results, however they still have some limitations. For instance LSD [1] used neural network technique to integrate the schemas from different sources. DIKE [3] only focused on the structural different of the source schemas. Most of the existing approaches are not focused on solving the problems of naming conflict during the integration. The work done in this paper is aimed to remove the naming conflict which may occur in the schemas for better integration result.

In this study the problems of identifying homonym and synonym conflict that exists in the element of two different schemas is explored. Homonym conflict means that two or more elements of the same name, but each has a different concept of representation. Synonyms conflict means that two or more elements that use a different name, but has the same concept of representation [7]. For instance in schema A there is an element name E1 and in schema B there exist an element E2 with the same name however E1 and E2 represent different concept. If this scenario occurs then these schema are said to suffer from homonym conflict. As an example, Fig. 1 shows part of student schema for Reed College (schema A) and Washington State University, WSU (schema B) in tree representation. The course element has two simple elements which are registration number (reg_num) and days. Element name *reg_num* appears both in schema A and schema B, however the concept used is different. The concept of element *reg_num* in schema A is used to represent the course registration number, while on the other hand, in schema B it is used to represent the course time table. Another example of homonym conflict is the concept of element *area* where it might represent dimension or location; this will depend on how the designer interprets the meaning of this concept. If these two schemas are going to be merged then a special method is needed to identify the occurrence of homonym conflict before the integration process take place, otherwise redundancy of schema will be produced.

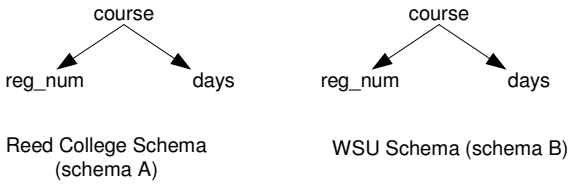

Fig. 1 Example of homonym conflict

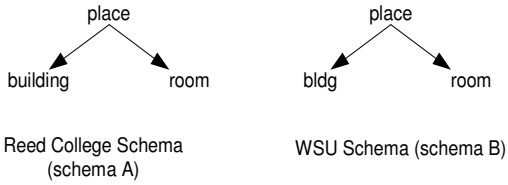

Fig. 2 Example of synonym conflict

Fig. 2 shows an example of synonym conflict where this conflict means different element name is used to represent the same concept. Element *place* has two simple elements, where in schema A the two simple elements are *building* and *room*. In schema B the two simple elements for schema B are *bldg* and *room*. The simple elements *building* and *bldg* are actually has the same concept where they are used to represent the name of the building however they have different name. In order to integrate these two schemas into a global view, an integration method is needed to solve this diversity of schema.

III. PROBLEM STATEMENT

One of the advantages of XML is it allows the designers to create their own tag names when constructing the data structures in order to meet their needs. However this will lead the XML documents stored in the data warehouse environment to have a different XML structures. Consequently, this will lead to a conflict when the sharing of data between two different XML is required. There are several issues of conflict that may arise in this situation, one of which is the problem of naming conflicts which are the homonym and synonym conflicts. Homonym and synonym conflicts can be seen in student data at Reed College and Washington State University (WSU) as discussed in previous section. These student data are based on XML documents and are used as a case in this study. In order to integrate these data the problem of naming conflicts need to be developed and this is the intention of this study. Therefore the objectives of this study are:

- To propose a method to solve the problems of homonym and synonym conflicts
- To integrate two different DTD schema from source schema into global schema
- To develop Intelligent Schema Integrator (ISI) to solve the homonym and synonym conflicts.

IV. INTELLIGENT SCHEMA INTEGRATOR (ISI)

An integration tool called Intelligent Schema Integrator (ISI) is proposed to help in integrating the data from local schema to the global schema and focused to solve the problems of homonym and synonym conflict. To solve this conflict the thesaurus approach is used where the concept of representative elements is compared with respect to the meaning of the element in data dictionary or thesaurus. As a case study two DTD schemas of XML documents at Reed College and Washington State University (WSU) are used as illustrate in Fig. 3.

<!ELEMENT root (course*)> <!ELEMENT course (reg_num,subj,crse,sect,title,units,instructor,days,time,place)> <!ELEMENT reg_num (#PCDATA)> <!ELEMENT subj (#PCDATA)> <!ELEMENT crse (#PCDATA)> <!ELEMENT sect (#PCDATA)> <!ELEMENT title (#PCDATA)> <!ELEMENT units (#PCDATA)> <!ELEMENT instructor (#PCDATA)> <!ELEMENT days (#PCDATA)> <!ELEMENT time (start_time,end_time)> <!ELEMENT start_time (#PCDATA)> <!ELEMENT end_time (#PCDATA)> <!ELEMENT place (building,room)> <!ELEMENT building (#PCDATA)> <!ELEMENT room (#PCDATA)>	<!ELEMENT root (course*)> <!ELEMENT course (footnote,reg_num,prefix,crs,lab,sec,title,credits,days,times,place,instructor,limit,enrolled)> <!ELEMENT footnote(#PCDATA)> <!ELEMENT reg_num(#PCDATA)> <!ELEMENT prefix(#PCDATA)> <!ELEMENT crs(#PCDATA)> <!ELEMENT lab(#PCDATA)> <!ELEMENT sec(#PCDATA)> <!ELEMENT title(#PCDATA)> <!ELEMENT credit(#PCDATA)> <!ELEMENT days(#PCDATA)> <!ELEMENT times (start,end)> <!ELEMENT place (bldg,room)> <!ELEMENT instructor(#PCDATA)> <!ELEMENT limit(#PCDATA)> <!ELEMENT enrolled(#PCDATA)>
Schema A(Reed College)	Schema B (WSU)

Fig. 3 Schema for Reed College and WSU

The integration process of ISI is as follows:

i. Read the input schema both from schema A and schema B.
ii. Extract element from both schemas and store into two different tree structures to represent each schema.
iii. The integration process between schema A and B begins.
iv. The representation concept and the name of each element are compared between schema A and B. Thesaurus will be used during this comparison.
v. Identify the occurrence of homonym and synonym conflict.
vi. If homonym conflict detected, identify which element needs to be re-named.
vii. If synonym conflict detected, identify which element needs to be removed from the list.
viii. Global schema which is the integrated schema between schema A and B is constructed.

A.Reading and Extraction Process

Schema from both file are read. Tree structure is used to extract the data structure of both schemas. Tree structure is used since XML documents normally modeled as a tree representation [10]. This process will prepare the data for the integration process.

B.Comparison Process

This process is the most important process in ISI. It is used to identify the occurrence of homonym and synonym conflict. During this process every element of schema will be compared with schema B. The comparison is based on the name and the concept of each element. 1:N approach is used during the comparison, meaning one element from schema A will be compared with all the element in schema B to find a matching element. The comparisons of the next elements of schema A will start after the previous element have finished the comparison process on all elements of schema B. This process continues after all the elements in schema A are exhausted. This process is illustrated in Fig. 4.

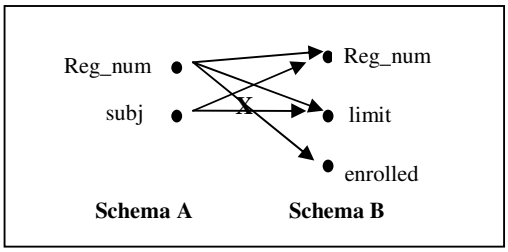

Fig. 4 The comparison process

For instance, the *reg_num* of schema A will be compared with *reg_num*, *limit* and *enrolled* element of schema B for the first round. In the second round the *subj* element of schema A will be compared against *reg_num*, limit and enrolled element of schema B. The description of each element is captured and thesaurus is used to get the meaning of these elements and compares them. Cases will be constructed during the comparison process between schema A and schema B. These cases are used to identify the occurrence of homonym and synonym conflict. Case is a set of algorithms developed in ISI environment that are able to identify the problems automatically. Table 1 state the category of cases that may occur during the comparison process between these schemas and identifies the occurrence of homonym and synonym conflicts. No conflict indicates that either both schemas have similar element names and concept or both schemas have different element names and concept.

TABLE 1: CATEGORY OF CASES CONSTRUCTED

	Element name between Schema A and schema B	**Concept element between Schema A and schema B**	**Naming conflict**
Case 1	Similar	Similar	No conflict
Case 2	Similar	No similar	Homonym conflict
Case 3	Not similar	Similar	Synonym conflict
Case 4	Not similar	not similar	No conflict

The rule representations for the four cases are stated as follows;

Case 1: IF A.name = B.name) ^ (A.concept = B.concept) THEN No conflict

Case 2: IF A.name = B.name) ^ (A.concept ≠ B.concept) THEN Homonym conflict

Case 3: IF A.name ≠ = B.name) ^ (A.concept = B.concept) THEN Synonym conflict

Case 4: IF A.name ≠ B.name) ^ (A.concept ≠ B.concept) THEN No conflict

ISI uses two approaches to solve the naming conflict which are the re-naming strategy and remove strategy. Re-naming strategy is to solve the homonym conflict, where remove strategy is to solve the synonym conflict.

C. Re-naming Strategy

If element name in schema A is the same with one of the element in schema B but used different concept then

homonym conflict has occurred. To solve this problem, the re-naming strategy is used [2]. This strategy is needed to avoid similar name to be used to represent different concept in the schema global. One of the elements needs to be changed to other name. The user will receive a message from the system through the dialog box asking which of the element need name change. As an example in Fig. 4 *reg_num* element needs to be changed to other name in order to distinguish them. Figure 5 is the screen shot of ISI that shows the interaction between user and the system.

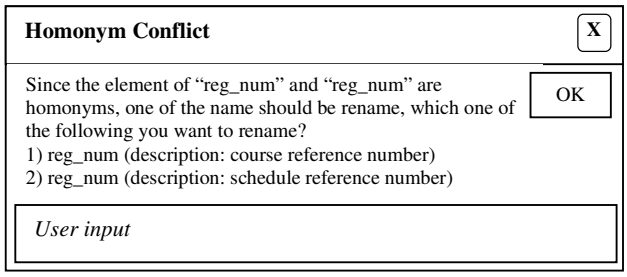

Fig. 5 The interface for Re-naming Strategy

D. Removing Strategy

If there is an element name in schema A has the same concept with one of the element in schema B but have different element name then synonym conflict has occurred. To solve this problem, the removing strategy is used [2] where one of the elements needed to be removed and replaced with the one that is going to remain. This strategy is needed to avoid different name used to represent the same concept in the schema global. The user will receive a message from the system through the dialog box that asking which of the element need to be removed and replace with the same name of other element. As an example in Fig. 4 either *building* or *bldg* need to be removed. The user is given a choice to choose. If they prefer to keep the name *building* then the name *bldg* in schema B will have to be removed and replaced with the name *building*.

E. Integration Process

This process means to integrate both schema A and schema B into global schema after the naming conflict is solved. Every element in schema A will be compared against elements in schema B. If the elements in schema A do not exist in schema B then it will map directly to global schema and vice versa. At the end of this process, global schema is generated and will represent both schemas.

V. EVALUATION

As an evaluation, ISI is tested to verify its effectiveness. Two mode of testing is used: expected result testing and actual result testing. The first test will use ISI as a tool and get the output. The second one will use the algorithm developed for ISI and runs it manually. Both tests will use Reed College and WSU schemas as an input data. The results of both testing are then compared and analyzed, as in Table 2.

TABLE 2 Results Generated from ISI and Manual Process

	Homonym Conflict	**Synonym Conflict**	**Total Occurrence**
ISI (actual result)	1	7	8
Manual (expected result)	1	7	8

As can be seen from Table 2, the actual result (using ISI) is the same with the expected result (runs manually). The result shows that ISI 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 of Reed College and WSU

VI.CONCLUSION

This paper proposed an integration tool called SIS to identify the existence of homonym and synonym conflict in the source schemas and removed it during the construction of the global schema. This tool provides a simple solution using thesaurus during the comparison and allows user to identify which elements that they want to maintain or remove. Providing these features in the integration tool is very important so that during the construction of global schema there will be no redundant or repeated elements used to represent the same concept. These features provide an extra advantage to ISI when compared with the existing integration tools. However there are other semantic constraints that are not counted in the development of ISI, such as key constraints, cardinality, structural and type constraints. Adding these constraints into the development of ISI will make it more comprehensive and impressive. The development of this tool is an important direction for more efficient implementation of data integration in data warehouse.

REFERENCES

[1] A. Doan and A. Halevy, "Semantic integration research in the database community: A brief survey," AI magazine, vol. 26, p. 83, 2005.

[2] C. Batini, M. Lenzerini and S.B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*. Vol.18. 1986.

[3] I. Palopoli, G. Terracina, and D. Ursino. A System Supporting The Semi-Automatic Construction of Cooperative Information Systems From Heterogeneous Databases. *Softw. Pract. Exper*: 847-884. 2003

[4] J. Madhavan, P. A. Bernstein, A. H. Doan and A. Halevy, "Corpus-based schema matching," IEEE,pp. 57-68, 2005.

[5] O. Unal and H. Afsarmanesh, "Schema matching and integration for data sharing among collaborating organizations," Journal of Software, vol. 4, p. 248, 2009.

[6] O. Unal and H. Afsarmanesh, "Semi-automated schema integration with SASMINT," Knowledge and Information Systems, vol. 23, pp. 99-128, 2010.

[7] P. Bellstrom. 2005. Using Enterprise Modeling for Identification and Resolution of Homonym Conflicts in View Integration. *Information Systems Development: Advances in Theory, Practice and Education*: 265-276.

[8] P. Gianolli, J. Mylopoulos. A semantic approach to XML based data integration. Proc. Of the 20th. International Conference on Conceptual Modelling (ER), Yokohama, Japan. 2001

[9] R. Pottinger and A. Levy. "A scalable algorithm for answering queries using views". Proc. of 26th VLDB Conference, Cairo, Egypt,484–495 2000.

[10] S. Abiteboul, P. Buneman, and D. Suciu. *Data On The Web*. California. Morgan Kaufman Publishers. 2000.

[11] W. May. A Framework for Generic Integration of XML Data Sources. International Workshop on Knowledge Representation meets Databases (KRDB 2001), Roma, Italy. 2001.