Induction of Global Rules Considering Schema Conflicts in Multi-database Systems*

Yufu Song†, Zengyou He†, Xiaofei Xu†, Shengchun Deng†

†*Department of Computer Science and Engineering Harbin Institute of Technology, Harbin, 150001 China*

Summary
Semantic query optimization is comparatively a recent approach for the transformation of given query into equivalent alternative queries using matching rules in order to select an optimal query based on the costs of executing alternative queries. There is the potential to reduce query execution cost by applying this process in a multi-database system. In such an environment, rules can be classified into three types: global, inter-schema and local. In this paper, a systematic method is proposed for inducing global rules from holding rules in local databases with the consideration of schema conflicts.

Key words:
Rules, Semantic Query Optimization, Multi-database Systems

1. Introduction

With the development of database technology and new applications of information systems, such as multidatabase system and electronic commerce, we need to handle various database objects in a heterogeneous environment where things become very different. In such situation, traditional query optimization technology is not very suitable. Therefore new query optimization technologies such as semantic query optimization are proposed. Semantic query optimization (SQO) use semantic integrity constraints in the database to transform the original query into a more efficient one to reduce the execution cost.

A multi-database system allows its users to simultaneously access heterogeneous and autonomous databases using an integrated schema and a single global query language. The global schema of a multi-database system results from the integration of the schemas exported from the underlying local databases which maybe relational or object-oriented DBMSs. A global query language is used to issue queries against the global schema.

While SQO has been successfully applied[1] in centralized databases, its potential for distributed and heterogeneous systems is enormous, as there is the potential to eliminate inter-site joins which are the single biggest cost factor in query processing. Further justification for its use is provided by the fact that users of multidatabase typically issue queries through high-level languages, which may result in very inefficient queries if mapped directly, without consideration of the semantics of the system. Even if this is not the case, users cannot be expected to be familiar with the semantics of component databases, and may consequently issue queries that are unnecessarily complicated.

In a multidatabase environment, rules used for SQO can be classified into three types: global, inter-schema and local. Local rules are constraints held on local component databases, which are used for the optimization of global sub-queries that can be executed in local databases. Automatic acquisition of these kinds of rules has been studied extensively [1,2,3,4]. Inter-schema rules identify the relationships between local databases, which can reduce the cost of data transformation by reducing unnecessary data retrieval. Global rules are defined for global schema virtually, inducted from local rules. These kinds of rules must satisfy all their corresponding source constraints in local databases and are used for global semantic query transformation. The purpose of global semantic query transformation is to identify redundancies and inconsistencies in the specification of the global query. Global rules can be expected to be small in number and simple in their form, and so are easy and cost effective to apply.

The global schema of a multi-database system results from the integration of the schemas exported from the underlying local databases. The schemas of these databases may be different in various ways, while the same information is represented. To make global rules satisfy all their corresponding source constraints in local databases, induction of global rules in global schema must consider the effect of conflicts between these schemas.

To induce global rules, a three-step approach is proposed which takes schema conflicts between global schema and local schemas into consideration. In the first step local rules are mapped into global rules according to schema conflicts and meta data in dictionary. In the second step, consistency is checked and contradictory rules are eliminated. Final global rule set is inferred by deleting redundant rules in the last step.

Manuscript received April 5, 2006.
Manuscript revised April 30 , 2006.

2. Conflicts between Schemas

First, a university multi-database example is given to illustrate our description. It is composed of local databases of different departments, registrar's office, student union, etc. The schemas of these databases are shown in Figure 1. The data stored in those local databases contain information about students and teachers.

Registrar's Office Database (ROD)
Student(s_no, name, Sex, Birthday, Degree, Department, Enrolled_Time, T_no, GPA, Allowance, street, city, state, association) Teacher(t_no, name, Sex, Birthday, Degree, Department, Phone, Start_Time, Salary, street, city, state)
Computer Science Department Database (CSD)
Student(s_no, name, sex, Age, Education, address, t_no) Teacher(t_no, name, sex, Age, Phone, Education, Work_Period) Address(t_no, street, city, state)
Association Database (ASD)
Student(s_no, name, Age, hobby, Dept, t_no) Address(s_no, street, city, state) Teacher(t_no, name, Age, male, female, phone, Dept, Start_Time)
Foreign Language Department Database (FLD)
S_Female(s_no, name, Birthday, Country, Salary, t_no) S_Male(s_no, name, Birthday, Country, Salary, t_no) T_Female(t_no, name, Birthday, Country, Wage, Work_Period) T_Male(t_no, name, Birthday, Country, Wage, Work_Period)
Student Union Database (SUD)
Student(s_no, name, sex, Age, Dept, t_no, association, Country) Address(s_no, street, city, state) Teacher(t_no, name, sex, Age, Country, Dept, Start_Time)
An Integrated Schema
Student(s_no, name, Sex, Birthday, Degree, Department, Enrolled_Time, T_no, GPA, Allowance, street, city, state, association, Hobby, Country) Teacher(t_no, name, Sex, Birthday, Degree, Department, Phone, Start_Time, Salary, street, city, State, Country)

Figure 1. An integrated schema and its local schemas for databases in a university

Various types of conflicts could exist between any two schemas. The classification used here is just the same as presented in [5]. We briefly introduce them as following:

- Value-to-value conflicts

This type of conflicts occurs when databases use different representation for the same data value. The difference may appear in three aspects: expressions, units and precision. Examples of these conflicts are U.S. dollars versus RMB (different units), a score of 1 to 100 versus A to E (different precision), etc.

- Value-to-attribute conflicts

This type of conflicts occurs when the same information is expressed as values in one database and as an attribute(s) in another database. For example, the values of attributes sex of CSD.Student are represented as attributes (female and male) in ASD.Student. This type of conflicts is called a value-to-attribute conflict.

- Value-to-table conflicts

This type of conflicts occurs when the attribute values in one database are expressed as tables in another database. For example, Figure 1 shows that S_Female and S_Male of ROD are relations for female and male students, respectively. The sex information, however, is represented as values in CSD. We refer to this kind of conflicts as the value-to-table conflict

- Attribute-to-attribute conflicts

Using different definitions for the semantically equivalent attributes in different databases causes these conflicts. For instance, the address is one attribute in CSD.Student. It is however represented by three attributes street, city and

state in IS.Student. This type of conflicts is referred to as the attribute-to-attribute conflict.

- Attribute-to-table conflicts

This type of conflicts occurs if an attribute of a database is represented as a table in another database. For example, address is attribute in CSD.Student. It is however represented as a relation Address in SUD. This type of conflicts is termed the attribute-to-table conflict.

- Table-to-table conflicts

These conflicts are caused by representing the information of a set of semantically equivalent tables in a different set of tables in another databases. For example, the IS.Student has a table-to-table conflict with the ASD.Student and ASD.Address.

3. Induction of Global Rules

In Section 2, it is stated that schema conflicts can be roughly classified into six types: value-to-value, value-to-attribute, value-to-table, attribute-to-attribute, attribute-to-table and table-to-table. Basing on this classification, we present our approach for inducing global rules.

3.1 Map Local Rules into Global Rules Considering Schema Conflict

Before proceeding any further, we will provide some definitions to be used in inducing global rules.

Definition 1. Suppose GS is the global schema and LS is the local schema. If V is a local value, we define Mapping(V, LS, GS) as the corresponding global value according to global data dictionary.

Definition 2. Suppose GS is the global schema with attribute set GAttr={A | A is the attribute of a special relation in GS} and LS_i is the local schema with attribute set $LAttr_i$={A | A is the attribute of a special relation in LS_i}, where i =1,2…n. For any $A_L \in LAttr_i$, we define the mapping of local attribute A_L to global schema GAttr as:

Mapping (A_L, LS_i, GS) = {A_G | $A_G \in$ GAttr and A_G is the corresponding attribute in GS for A_L}

Information about the correspondence between local attributes and global attributes can be found in the global data dictionary, which has been formed in the process of schema integration. The details of mapping are not our concern. It also should be noted the obvious fact that any mappings from local schema to global schema will not be empty sets.

For a local rule in form of $r=X \rightarrow Y$, we can regard the problem of mapping it into a global rule as mapping the local predicts, X and Y, into global predicts separately. Next, we will present a method for mapping local rules into global rules based on schema conflict.

Assume predict P in form of $\alpha\ op\ \beta$, where *op* represent operator such as >, =, <, etc, α is a local attribute and β is its value.

- Value-to-value conflicts

In this case, Mapping (α, LS, GS) = α, what we have to do is to map local value into global value. Therefore, the transformed predict will be $\alpha\ op$ Mapping (β, LS, GS). For example, if there is a predict salary>=3000 in local schema using RMB as its currency unit, after mapping, the global predict will be salary>=375 with U.S. dollars as its currency unit.

- Value-to-attribute conflicts

In this case, it is obvious that *op* equals '='. If the same information is expressed as values in LS and as an attribute(s) in GS, and Mapping (α, LS, GS) = $\{a_1 \ldots \beta \ldots a_n\} \supseteq \beta$, the transformed predicts will be a_1=FALSE $\wedge \ldots \wedge \beta$ =TRUE… $\wedge$ a_n=FALSE. For example, sex='male' can be mapped into predicts male=TRUE $\wedge$ female=FALSE. If the same information is expressed as values in GS and as an attribute(s) in LS, the transformed predict will be Mapping (α, LS, GS) = α.

- Value-to-table conflicts

If the global attribute a, whose values set $\{a_1 \ldots a_n\}$ in GS are expressed as tables in LS, and this predict is defined on the table corresponding to the value a_i, the mapped predicts should be expressed as $\alpha\ op\ \beta \wedge a=a_i$. For instance, the predict salary>=3000 defined for FLD.S_Male will be mapped into global predicts as salary>=3000 $\wedge$ sex=male.

- Attribute-to-attribute conflicts or Attribute-to-table conflicts

Suppose Mapping (α, LS, GS)= $\{a_1 \ldots a_n\}$ and Mapping (β, LS, GS)= $\{b_1 \ldots b_n\}$, the transformed predicts will be $a_1\ op\ b_1 \wedge \ldots \wedge a_n\ op\ b_n$. The predict address='Jiaotong, Harbin, China' will be expressed as street='Jiaotong' $\wedge$ city='Harbin' $\wedge$ state='China'.

- Table-to-table conflicts or No schema conflicts exist

In this case, predict can be mapped directly, transformed predict will be just the same as the original one.

For a special rule in a local schema, after all the predicts in it have been mapped into global predicts, we can easily get the transformed rule by combining them in their original orders.

3.2 Checking for Contradiction and Redundancy

After the transformation process from local rules to global rules has completed, rule reduction must be performed to insure the consistency and non-redundancy of the global rule set. This task can be done by existing works [6,7] efficiently, which is not our main concern.

4. Conclusion

In this paper we discuss the problem of inducing global rules from local rules in multidatabase systems. By considering schema conflicts between database schemas, we present a systematic methodology to deal with this problem. Checking consistency and non-redundancy of the transformed global rule set will be addressed in our future work.

Acknowledgments

We would like to thank Dr. Min Ni for his valuable comments to improve this paper.

References

[1] C. T. Yu and W. Sun. Automatic Knowledge Acquisition and Maintenance for Semantic Query Optimization. IEEE Trans. Knowledge and Data Engineering, vol. I, no.3, pages 362-375, 1989.

[2] C. N. Hsu. Learning Effective and Robust Knowledge for Semantic Query Optimization. PhD thesis, Department of Computer Science, University of Southern California, 1996.

[3] M. D. Siegel, E. Sciore and S. Salveter. A Method for Automatic Rule Deviation to Support Semantic Query Optimization. ACM Transactions on Database Systems, vol.17, no.4, pages 563-600, 1992 (12).

[4] S. Shekhar, B. Hamidzadeh and A. Kohli. Learning Transformation Rules for Semantic Query Optimization: A Data-driven Approach. IEEE Trans. Knowledge and Data Engineering, pages 949-964, 1993.

[5] Chiang Lee and Chia-jung Chen. Query Optimization in Multidatabase Systems Considering Schema Conflicts. IEEE transactions on knowledge and data engineering, pages 941-955, 1997, 9(6).

[6] X. Zhang, Z. M. Ozsoyoglu. Implication and Referential Constraints: A New Formal Reasoning. IEEE Transactions on Knowledge and Data Engineering, pages 894-910, 1997, 9(6).

[7] N.Ishakbeyoglu, Z. M.Özsoyoglu. Maintenance of Implication Integrity Constraints under Updates to Constraints. VLDB Journal, pages 67-78, 1998, 7(2).

YuFu Song, male, born in 1964, Ph. D. candidate. His major research interests include database, workflow management, advanced transaction processing and data mining.

ZengYou **HE,** male, born in 1976. He is a PhD candidate at the Department of Computer Science and Engineering, Harbin Institute of Technology. His researches areas are data mining.

Xiaofei Xu, male, born in 1962, professor at the Department of Computer Science and Engineering, Harbin Institute of Technology. His current research interests include database and CIMS.

ShengChun Deng, male, born in 1974, assistant professor at the Department of Computer Science and Engineering, Harbin Institute of Technology. His current research interests include database and CIMS.

---

* This work was supported by the High Technology Research and Development Program of China (No. 2003AA4Z2170, 2003AA413021)