Mapping Publishing and Mapping Adaptation in the Middleware of Railway Information Grid System

Ganmei You, Huaming Liao, Yuzhong Sun

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080
`ganmeiu@ict.ac.cn`

Abstract. When adopting the mediator architecture to integrate distributed, autonomous, relational model based database sources, mappings from the source schema to the global schema may become inconsistent when the relational source schema or the global schema evolves. Without mapping adaptation, users may access no data or wrong data. In the paper, we propose a novel approach the global attribute as view with constraints (GAAVC) to publish mappings, which is adaptive for the schema evolution. Also published mappings satisfy both source schema constraints and global schema constraints, which enable users to get valid data. We also put forward the GAAVC based mapping publishing algorithm and mapping adaptation algorithms. When we compare our approach with others in functionality, it outperforms. Finally the mapping adaptation tool GMPMA is introduced, which has been implemented in the middleware of railway information grid system.

1 Introduction

A serious issue in integrating distributed, autonomous, relational model based databases is the evolution of schemas. As for mediator architecture, mappings from source schemas to the global schema are used to create source views that query data from sources. When schemas change, users will get invalid data without modifying mappings. However it is time-waste and heavy work to revise mappings manually. In this paper we propose a GAAVC based mapping publishing algorithm and corresponding mapping adaptation algorithms to automatically adjust mappings.

Our main contributions are as follows. (1) We put forward the nested schema decomposition model according to which we propose the approach the global attribute as view to construct mappings. (2) We develop the GAAVC based mapping publishing approach and the corresponding algorithm. The approach is adaptive to the evolution of the source schema and the global schema. Also source views created by the mappings guarantees that data from sources is valid. (3) We develop algorithms to adjust invalid mappings automatically as schemas evolve. (4) We implement the GMPMA tool that implements the algorithms.

Section 2 introduces related work. Section 3 defines valid mappings. Section 4 gives the GAAVC approach and the mapping publishing algorithm. Section 5 introduces mapping adaptation algorithms. Section 6 compares the GAAVC approach with other approaches. Section 7 describes the architecture of GMPMA tool before section 8 concludes.

2 Related Works

The approach adapts to the schema evolution if only local attributes that have no invalid mappings can still be accessed as schemas evolve.

The local as view (LAV) method [1] defines the local schema as the view over the global schema. When schemas evolve, all the local attributes that are in the same schema cannot be accessed. So LAV is not adaptive to schema changes. Because source schemas match the global schema, source schema constraints are satisfied when global constraints are satisfied by views. The global as view (GAV) approach [2] defines the global schema as the view over the source schema. When schemas change, all the local attributes that are in the same view cannot be accessed. So GAV has problems when schemas change. Because the global schema matches source schemas, global schema constraints are satisfied when source constraints are satisfied by views. The global-local as view (GLAV) [3] approach is a variation of LAV. The approach has the same problem with the LAV approach. The both as view (BAV) [4] approach defines the global attribute with the view of the local schema and the local attribute with the view of the global schema. The approach is adaptive for the schema evolution. However, the view is defined manually. The approach has considered constraints of the global schema and source schemas. The correspondence view (CV) [5] approach sets up the mapping of view, other attributes of the same view cannot be accessed when schemas change. So the approach is not adaptive to the schema evolution. What is more, the approach considers foreign key constraints but not key constraints of the global schema.

3 Valid Mappings

We study relational based XML and relational schemas with key constraints and foreign key constraints in this paper. We suppose that there is at most one view for a set of elements in XML schema or tables in relational schemas.

Definition 3.1 An **attribute** is of the form <ID, label, type>.

We use Greek alphabets, α, β, ⋯, to represent a set of attributes.

Definition 3.2 A **schema** is of the form (< ID, label>, {attribute}, {schema}).

Definition3.3 A **mapping of global attributes** is an expression of source views:

$Mapping\,(\alpha_g):\ \alpha_g = G(V(\alpha_{S_1}),V(\alpha_{S_2}),...,V(\alpha_{S_n})) = F(\alpha_{S_1},\alpha_{S_2},...,\alpha_{S_n})$. α_g are global attributes. V is a function of attributes of one source and the $V(\alpha_{S_i})$ is the view of the source S_i (i = 1, 2, … , n). G is a function of views of different sources. F is a composite function of G and V and F is a function of attributes of different sources. The expression is made up of union, intersection and set difference operators.

Definition 3.4 The **mapping of the global schema R** is the set of the mappings of all the attributes of the schema: $Mapping_R = \{\alpha_g = F(\alpha_{S_1},\alpha_{S_2},...,\alpha_{S_n}) \mid \alpha_g \in R\}$.

The expression $\alpha \rightarrow_{KEY} \beta$ denotes that α is the key of the schema of β and $\alpha \rightarrow_{FK} \beta$ denotes that β is the foreign key of α. $Mapping_{S_1}(\alpha)$ denotes the mapping of α in source S_1.

Lemma 3.1 (Null value constraints for foreign key, briefly $\alpha \to_{FK\_NULL} \beta$ **)** Suppose $\alpha \to_{FK} \beta$,if α =null then β =null.

Definition 3.5 Valid data are data that satisfy key constraints and null value constraints for foreign key of the global schema.

Definition 3.6 A **valid source view** is a source view that satisfies key constraints and null value constraints for foreign key of the global schema.

Definition 3.7 Valid mappings of a global schema are mappings that produce valid source views.

   By valid mappings of a global schema, users can get valid data from databases.

4 The Global Attribute as View with Constraints (GAAVC) Based Mapping Publishing Approach

4.1 The Nested Schema Decomposition Model

$R (\kappa, A_1, R'(\kappa',A_1',\cdots),\cdots,A_m) = R_1(\kappa) \cup R_2(\kappa,A_1) \cup R_3(\kappa, \kappa') \cup R_4(\kappa, \kappa',A_1') \cup \cdots \cup R_n(\kappa, A_m)$, R is a nest relational schema or XML schema. R' is the sub-schema of R. κ is the key of R and κ' is the key of R'. $\kappa = (k_1, k_2, \cdots, k_n)$. $\kappa' = (k_1', k_2', \cdots, k_n')$. Atom schemas, $R_1, \cdots, R_n$, inherit foreign key constraints of R.

Lemma 4.1 In the nested schema decomposition model, the decomposition is a lossless join decomposition.

   The lemma 4.1 indicates that no information will be lost after the decomposition.

4.2 The Global Attribute as View with Constraints (GAAVC) Approach

Definition 4.1 A **view key set** is the keys' union of all the schemas of the view:
$key(R_1,R_2,\ldots,R_n) = key(R_1) \cup key(R_2) \cup \cdots \cup key(R_n)$

Lemma 4.2 The view key set is the key of the view.

Definition 4.2 Key constraints on mapping are the global schema key constraints on mapping: If $\beta_g \to_{KEY} \alpha_g$, $\beta_g = F(\beta_s)$, $\alpha_g = F(\alpha_s)$, $\beta_s = (\beta_{s_1}, \beta_{s_2}, ..., \beta_{s_n})$, $\alpha_s = (\alpha_{s'_1}, \alpha_{s'_2}, ..., \alpha_{s'_n})$, then for any $\alpha_{s'_i} \in \alpha_s$ (i =1,2,$\cdots$, m), there is a view key set β_{s_j} (1$\leqslant$j$\leqslant$n), $\beta_{s_j} \to_{KEY} \alpha_{s'_i}$, $\beta_{s_j} \in \beta_s$, $S'_i = S_j$.

Lemma 4.3 (key constraints on null mapping) If $\beta_g \to_{KEY} \alpha_g$, Mapping$_s$($\beta_g$)= null, and the mappings of β_g and α_g satisfy key constraints on mapping, then Mapping$_s$(α_g)=null.

Lemma 4.4 (foreign key constraints on null mapping, $\beta_g \to_{FK\_NULL} \alpha_g$**)** if $\beta_g \to_{FK} \alpha_g$, Mapping($\beta_g$)=null, then Mapping($\alpha_g$)=null.

Definition 4.3 (the global attribute as view approach, GAAV) the GAAV approach is using source views to express each atom schema of the global schema.

Definition 4.4 (the global attribute as view with constraints (GAAVC) approach)
Given the nested schema $R(\kappa, A_1, R^1(\kappa^1, A^1_1,...),..., A_m)$, the mapping expression of R: $R_{mapping}=\{ \kappa = F_1(\alpha_{s_1}, \alpha_{s_2},..., \alpha_{s_n})$, $A_1 = F_2(A_{1,s_1}, A_{1,s_2},..., A_{1,s_n})$, $\kappa^1 = F_3(\alpha^1_{s_1}, \alpha^1_{s_2},..., \alpha^1_{s_n})$, $A^1_1 = F_4(A^1_{s_1}, A^1_{s_2},..., A^1_{s_n})$,..., $A_m = F_k(A_{m,s_1}, A_{m,s_2},..., A_{m,s_n})$ \}, mappings are constructed according to the following rules:

1) According to the nested schema decomposition model, the approach constructs mappings between atom schemas and source schemas, which are equal to mappings between the original global schema and source schemas
2) Mappings of atom schemas satisfy key constraints on mapping.
3) Mappings of atom schemas satisfy foreign key constraints on null mapping.
4) The mapping expression of an atom schema is the expression of the attribute in deepest schema, while operands are all the attributes of the atom schema.

Theorem 4.1 Mappings that created by the GAAVC approach are valid mappings.

4.3 Mapping Expressions

Mappings of atom schemas of the global schema are expressed as follows:

The key mapping expression: <addKey, <global key>, expression (<<source attributes>, source view constraints>)>

The non-primary attribute mapping expression: <addAttribute, <global key, global non-primary attribute>, expression (<<source attributes>, source view constraints>)>.

The GAAVC approach adapts to schema evolutions because attributes that have no invalid mapping can still be accessed when schemas change.

Lemma 4.5 Given $\kappa \rightarrow_{KEY} A$, source view constraints of $R(\kappa)$ and $R(\kappa, A)$ are the same.

4.4 The GAAVC Based Mapping Publishing Algorithm

Given a global schema, source schemas and a set of mappings between the global schema and source schemas, we set up mappings according to GAAVC approach.

Step 1: verify if mappings satisfy key constraints on mapping. If not, exit.

Step 2: get M the minimal cover of global functional dependencies.

Step 3: get ordered partitions of M so that β and α are in the same partition and β is before α if $\beta \rightarrow_{KEY} \alpha$ or $\beta \rightarrow_{FK} \alpha$.

Step 4: build mapping expressions of attributes of each ordered partition in sequence. The mapping expression of α is built only if for any attributes β, $\beta \rightarrow_{KEY} \alpha$ or $\beta \rightarrow_{FK} \alpha$, Mapping($\beta$)$\neq$null.

Step 2 gets rid of redundant functional dependencies of global schema and step 3 makes partitions to decrease searching times.

5 Mapping Adaptations

When a schema changes, we should adjust mappings to ensure them valid. Adding attributes has no influence on mappings. When renaming a schema or an attribute, we only review all the mapping expressions to change the name.

5.1 Deleting an Attribute or Deleting the Mapping of an Attribute

The step 2, 3, 7 and 8 in the following algorithm ensure key constraints on mapping. The step 4, 5,11, and 12 ensure foreign key constraints on null mapping.

```
deleteAttribute(α_k){

if α_k is a global attribute, that is α_g              1

  for each mapping expression E that contains α_g      2

    delete E                                           3

    for each attribute set λ, Mapping(λ)≠null,

    κ →_FK λ, κ is the key of E                         4

      deleteAttribute(λ)                               5

if α_k is an attribute of source S, that is α_s        6

  for each mapping expression E that contains α_s      7

    delete E the mapping of source S                   8

    if the global attributes β of E is key             9

      if no mapping expression contains β              10

        for each attribute set λ, Mapping(λ)≠null

        and β →_FK λ)                                   11

          deleteAttribute(λ)   }                       12
```

5.2 Deleting a Foreign Key Constraint

Step 2 and step 3 in the following algorithm ensure key constraints on mapping, while step 5 and step 6 ensure foreign key constraints on null mapping.

```
deleteForeignKey (α →_FK β){

if α and β are attributes of source S                  1

  for each mapping expression E that contains α →_FK β  2

    delete the mapping of E in source S                3
```

```
    if the global attributes β in E are the primary
    key and no mapping expression contains β            4
      for each global attribute set λ,
      Mapping(λ)≠null and β →_FK λ                       5
        deleteAttribute(λ)  }                            6
```

5.3 Adding a Foreign Key Constraint

When adding a source foreign key constraint, the existing mappings are still valid. When adding a global foreign key constraint, step 3 and step 4 in the following algorithm ensure foreign key constraints on null mapping.

```
addForeignKey (α →_FK β){                               1
if α and β are global attributes                        2
  if Mapping(β)≠null and Mapping(α)=null                3
    deleteAttribute (β) }                               4
```

5.4 Adding a Source Mapping

The foreign key constraints on null mapping are ensured by step 1, step 2 and step 3.

```
AddSourceMapping(Mapping_s(α_m)= F(α_s)){
for each global attributes λ, λ →_FK α_m                1
  if Mapping(λ) = null                                  2
    exit                                                3
if the mapping of α_m satisfy key constraints
on mapping when adding Mapping_s(α_m)= F(α_s)           4
  add the mapping Mapping_s(α_m)= F(α_s) }              5
```

5.5 Deleting a Source Mapping

Step 2 ensures key constraints on mapping. Step 4, 5 and 6 ensure foreign key constraints on null mapping.

```
deleteSourceMapping(Mapping_s(α_m)= F(α_s)){
for each mapping expression E that contains α_m         1
  delete the mapping of E in source S                   2
```

```
if α  is a set of attributes of key κ                    3
      m

  if Mapping(κ) = null                                    4

    for each attributes λ,Mapping(λ)≠null and κ →  λ     5
                                                  FK
      deleteAttribute(λ)   }                              6
```

6 Comparisons

Table 1. Mapping publishing approaches comparisons

Publishing methods / Comparisons	LAV	GAV	GLAV	BAV	CV	GAAVC
Be adaptive to schema evolution	No	No	No	Yes	No	Yes
Consider key and foreign key constraints	Yes	Yes	Yes	Yes	Only foreign key constraints	Yes
Adjust mappings manually	No	No	No	Yes	No	No

The detailed discussion is in related works. From the table 1, it can be seen that our GAAVC approach outperforms.

7 Architecture of the GAAVC Based Mapping Publishing and Mapping Adaptation (GMPMA) Tool

We implement the GAAVC based mapping publishing algorithm and mapping adaptation algorithms using the GMPMA tool. Figure 1 shows the architecture of the GMPMA tool. With the graphical user interface or the monitor, schema evolutions are detected. Then the mapping publishing and mapping adaptation engine constructs mappings or adjusts mappings. The architecture has been implemented in the middleware of the railway information grid system.
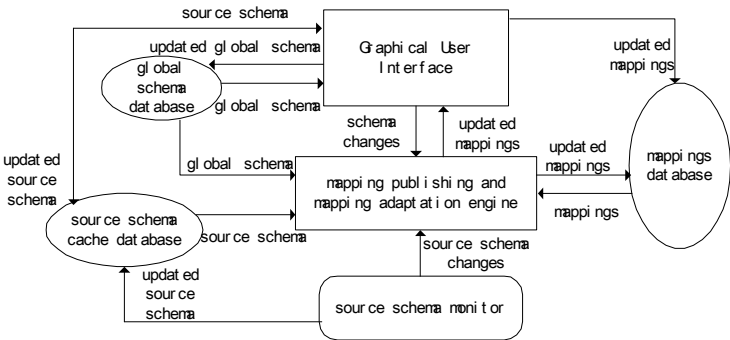

Fig. 1. The GMPMA Architecture

8 Conclusions

To ensure users to get valid data when schemas evolve, we propose the GAAVC based mapping publishing algorithm and mapping adaptation algorithms and implement them in the GMPMA tool in the middleware of railway information grid system. Our approach is unique in many ways: 1) The GAAVC approach constructs the mappings that adapt to the schema evolution. 2) Our mappings enable users to get valid data. 3) We consider mappings from one computation expression of some source attributes to one global attribute. How to make adaptation to schema evolutions of object-oriented databases is our future work.

References

1. T. Kirk, A.Y. Levy, and Y. Sagiv. The information manifold. In: Proc of the AAAI Spring Symposium on Information Gathering From Heterogeneous, Distributed Environments. AAAI press, 1995, 85~91.
2. H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajararnan, Y. Sagiv, J. Ullman, V. Vassalos, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. Journal of Intelligent Information Systems, 1997,8(2): 117~132.
3. M. Friedman, A. Y. Levy, and T. D. Millstein. Navigational Plans For Data Integration. In: Proc of the Sixteenth National Conference on Artificial Intelligence and Eleventh conference on Innovative Applications of Artificial Intelligence. AAAI Press / The MIT Press, 1999, 67~73.
4. P. McBrien , A. Poulovassilis. Data integration by Bi-directional Schema Transformation Rules. In: Proc of the 19th International Conference on Data Engineering. Morgan Kaufmann, 2003, 227~238.
5. Y. Velegrakis, R. J. Miller, L. Popa. Mapping Adaptation under Evolving Schemas. In International Conference of Very Large Databases (VLDB), September 2003, pp. 584-595.