RESOLVING FRAGMENTATION CONFLICTS IN SCHEMA INTEGRATION

Yann DUPONT

Laboratoire de Bases de Données
Département d'Informatique
Ecole Polytechnique Fédérale
IN - Ecublens
CH-1015 Lausanne (Switzerland)

Tel. (+41) 21-693-5255
Fax. (+41) 21-693-5195
email: yann.dupont@di.epfl.ch

Abstract. Research on schema integration leads to the identification of many different conflict types. Some of them received much attention and many papers proposed solutions for their resolution. However, literature usually focuses on traditional problems, whilst new kinds of schema discrepancies, due to the object orientation or the generalization concept, are not really treated. Moreover, most of the proposed methodologies and strategies only allow binary comparisons between items to be integrated.

This paper discusses n-ary (also called one-many) conflicts, and particularly the three fragmentation conflict types. These conflict types need specific operators for schema comparison. We propose a simple unified language for easy specification of these conflicts, and give many different techniques to solve them. We emphasize the benefit of separating the declaration of the correspondences from the choice of resolution technique.

Our discourse is illustrated with the entity-relationship model, but our solutions can be applied in any data model. We propose several conflict resolution techniques that can be applied on a concept which has the generalization (the entity type) and different techniques that must be applied on a concept which has not (the relationship type).
1 Introduction

A very important step in database design methodologies is the schema integration process which merges external schemas into a single global schema, the conceptual schema of the future database. Schema integration is also of the utmost importance for the creation of distributed or federated database systems based on existing databases [22]. This domain of research, very active for two decades, is still open. Moreover, the increasing importance of object orientation and of numerous powerful data models, such as extended entity-relationship (ECR [7], ERC+ [17] for instance), binary models (NIAM [28]), semantic networks, brings new kinds of problems and explains the interest of numerous researchers.

Two main distinct ways of research have been followed. In the first methodologies [1], the database administrator (DBA) modifies input schemas by using a set of restructuring operators, for each found discrepancy between input schemas. When all conflicts have been solved, the conformed schemas can be merged by superimposition of common concepts. Later, several methods and tools have been proposed to help the user with an automatic search of correspondences and conflicts between schemas, and to simplify his/her task with a more powerful and user-friendly language for performing the integration. Nevertheless, whatever the approach, only the simplest conflict types have been mastered. 1:n conflicts (in which an item in one schema corresponds to a set of items in a second schema) are rarely addressed ([4] and [12] seem to carry the earliest reference to this problem), and the conflict resolution techniques are almost non-existent, except for [19], in the particular context of data/meta-data conflict, and [11] which proposes an SQL-based multidatabase language which could be used to define a virtual class based on 1:n conflicting classes.

However, relational and semantic models, and many object oriented data models or database systems (COCOON [20], Dual [18], among others) support multiple instanciation which allows the same object to be present in different classes. This concept, which includes generalization/specialization as a particular case, gives a great freedom for the conceptual design, but at the same time is at the origin of 1:n conflicts called fragmentation conflicts (by analogy with fragmentation in the distributed database domain). The first problem we tackle is the classification conflict, in which the same set of objects is classified differently from one schema to the other. The second one is the decomposition conflict in which the same real world object is perceived and implemented as a whole in one schema and as several points of view in the other schema. The last one is the aggregation conflict in which a complex real world object is perceived and implemented as an aggregated occurrence in one schema and as a set of occurrences in the other schema.

The next section is concerned with the methodological context in which we study 1:n conflicts. Section 3 talks about fragmentation conflicts and proposes a dedicated declarative language. Our discourse is illustrated with schemas which we will use in all sections throughout this paper. In Section 4, we give many conflict resolution techniques and discuss conflict resolution scenarios, while section 5 defines some schema integration strategies. These two parts show the interest of separating the conflict declaration and the conflict resolution. Finally, the conclusion analyzes possible extensions to our work.