Resolving Schematic Discrepancy in the Integration of Entity-Relationship Schemas

Qi He and Tok Wang Ling

School of Computing, National University of Singapore
{heqi,lingtw}@comp.nus.edu.sg

Abstract. In schema integration, schematic discrepancies occur when data in one database correspond to metadata in another. We define this kind of semantic heterogeneity in general using the paradigm of context that is the meta-information relating to the source, classification, property etc of entities, relationships or attribute values in entity-relationship (ER) schemas. We present algorithms to resolve schematic discrepancies by transforming metadata into entities, keeping the information and constraints of original schemas. Although focusing on the resolution of schematic discrepancies, our technique works seamlessly with existing techniques resolving other semantic heterogeneities in schema integration.

1 Introduction

Schema integration involves merging several schemas into an integrated schema. More precisely, [4] defines schema integration as “the activity of integrating the schemas of existing or proposed databases into a global, unified schema”. It is regarded as an important work to build a heterogeneous database system [6, 22] (also called multidatabase system or federated database system), to integrate data in a data warehouse, or to integrate user views in database design. In schema integration, people have identified different kinds of semantic heterogeneities among component schemas: naming conflict (homonyms and synonyms), key conflict, structural conflict [3, 15], and constraint conflict [14, 21].

A less touched problem is schematic discrepancy, i.e., the same information is modeled as data in one database, but metadata in another. This conflict arises frequently in practice [11, 19]. We adopt a semantic approach to solve this issue. One of the outstanding features of our proposal is that we preserve the cardinality constraints in the transformation/integration of ER schemas. Cardinality constraints, in particular, functional dependencies (FDs) and multivalued dependencies (MVDs), are useful in verifying lossless schema transformation [10], schema normalization and semantic query optimization [9, 21] in multidatabase systems. The following example illustrates schematic discrepancy in ER schemas. To focus our contribution and simplify the presentation, in the example below, schematic discrepancy is the only kind of conflicts among schemas.

Example 1. Suppose we want to integrate supply information of products from several databases (Fig. 1). These databases record the same information, i.e., product numbers, product names, suppliers and supplying prices in each month, but have discrepant schemas. In DB1, suppliers and months are modeled as entity types. In DB2, months are modeled as meta-data of entity types, i.e., each entity type models...
the products supplied in one month, and suppliers are modeled as meta-data of attributes, e.g., the attribute S1_PRICE records the supplying prices by supplier s1. In DB3, months are modeled as meta-data of relationship types, i.e., each relationship type models the supply relation in one month. We propose (in Section 4) to resolve the discrepancies by transforming the metadata into entities, i.e., transforming DB2 and DB3 into a form of DB1. The statements on the right side of Fig. 1 provide the semantics of the constructs of these schemas using ontology, which will be explained in Section 3.

Fig. 1. Schematic discrepancy: months and suppliers modeled differently in DB1, DB2 and DB3

**Paper organization.** The rest of the paper is organized as follows. Section 2 is an introduction to the ER approach. Section 3 and 4 are the main contributions of this paper. In Section 3, we first introduce the concepts of ontology and context, and the mappings from schema constructs of ER schemas onto types of ontology. Then we define schematic discrepancy in general using the paradigm of context. In Section 4, we present algorithms to resolve schematic discrepancies in schema integration, without any loss of information and cardinality constraints. In Section 5, we compare our work with related work. Section 6 concludes the whole paper.

---

1 Without causing confusion, we blur the difference on entities and identifiers of entities. E.g., we use supplier number s1 to refer to a supplier with identifier S# = s1, i.e., s1 plays both the roles of an attribute value of S# and an entity of supplier.