A Formal Framework for ER Schema Transformation

Peter MCBrien and Alexandra Poulovassilis

Department of Computer Science, King's College London,
Strand, London WC2R 2LS.
alex,pjm@dcs.kcl.ac.uk

Abstract. Several methodologies for semantic schema integration have been proposed in the literature, often using some variant of the ER model as the common data model. As part of these methodologies, various transformations have been defined that map between ER schemas which are in some sense equivalent. This paper gives a unifying formalisation of the ER schema transformation process and shows how some common schema transformations can be expressed within this single framework. Our formalism clearly identifies which transformations apply for any instance of the schema and which only for certain instances.

1 Introduction

When data is to be shared or exchanged between heterogeneous databases, it is necessary to build a single integrated schema expressed using a common data model (CDM) [13]. Conflicts may exist between the export schemas of the component databases, which must be removed by performing transformations on the schemas to produce equivalent schemas.

In this paper we examine the behaviour of the schema transformations commonly found in the literature within a new formal framework, and distinguish in a precise manner between schema transformations which are dependent on knowledge about the instances associated with the schema, and those which are not. This distinction has the benefit of precisely defining what assumptions are made when a database object is transformed or is considered to have the same "real world state" [8] as some other object.

Previous work on schema transformation has either been to some extent informal [2, 7, 6], has formalised only transformations that are independent of database content [10, 11], or is limited to certain types of transformation only [3, 8, 14, 5]. The latter cases assume that specific types of dependency constraints are employed to limit the instances of schemas in order that the schemas can be regarded as equivalent. In contrast, our approach allows arbitrary constraints on instances to be specified as part of the transformation rules. A similar approach has recently been adopted in [6] where the notion of a database "context" constrains instances so that schemas can be considered equivalent.

In common with much previous work on semantic schema integration [1, 2, 4], we use a variant of the ER model as the CDM, namely a binary ER model with
subtypes. This model supports entity types with attributes, subtype relationships between entity types, and binary relationships (without attributes) between entity types. Subtype relationships give sufficient modelling expressiveness for representing object-oriented schemas. By omitting generalisation hierarchies for $n > 2$, some alternate representations of the universe of discourse are avoided. There is no loss of modelling expressiveness since there are obvious transformations that can be applied from these more complex structures into our CDM (see, for example, [9]).

Similarly to [1], in our CDM each relationship, $r$, between two entity types, $e_1$ and $e_2$, has associated with it a pair of cardinality constraints, $l_1 : u_1$ and $l_2 : u_2$, where $l_1 : u_1$ ($l_2 : u_2$) indicates the lower and upper cardinalities of the participation in $r$ by each instance of $e_1$ ($e_2$). Each association between an entity type and an attribute similarly has a pair of cardinality constraints associated with it. If not explicitly stated, a cardinality constraint of $1 : 1$ should be assumed.

The main tasks of database schema integration are pre-integration, schema conforming, schema merging and schema restructuring [2]. The last three of these tasks involve a process of schema transformation, and Figures 1-3 illustrate some of the common transformations used. In practice, schema conforming transformations are applied bi-directionally and schema merging and restructuring ones uni-directionally. For each of these transformations, the original and resulting schema obey one or more alternative notions of schema equivalence [12, 1, 8]. This paper presents a unifying formalism for the schema transformation process, including all of the transformations of Figures 1-3.

In Section 2 we define the notions of ER schemas, instances and models. In Section 3 we define the notion of schema equivalence which provides the semantic foundation of schema transformation in our approach. In Section 4 we define a set of primitive transformations and explore their properties with respect to schema equivalence. We then extend these transformations into "knowledge-based" versions, which allow conditions on instances to be expressed. Section 5 demonstrates the expressiveness of our primitive transformations by showing how they can be used to define all the transformations of Figures 1-3. In Section 6 we give our concluding remarks.

2 ER Schemas, Instances and Models

Our definitions of schemas, instances and models (Definitions 1-3 below), assume the availability of two disjoint sets: $Vals$ (values) and $Names$ (names of entity types, attributes and relationships). The distinguished constant $Null$ is a member of $Names$. $Seq(Vals)$ is the set of sequences of values of finite length. $P$ is the powerset operator. $Cards$ is the set of cardinality constraints, a cardinality constraint being a pair $l:u$, where $l$ is a natural number and $u$ is either a natural number or $N$ (denoting no upper limit). For any function $f$, $Range(f)$ denotes the range of $f$.

Definition 1. An ER schema, $S$, is a quadruple $(Ents, Incs, Atts, Assocs)$ where: