Reduction Transformations in ORM

Terry Halpin, Andy Carver and Kevin M. Owen

Neumont University, Utah, USA.
e-mail: {terry, andy, kevin}@neumont.edu

Abstract: This paper proposes extensions to the Object-Role Modeling approach to support schema transformations that eliminate unneeded columns that may arise from standard relational mapping procedures. A "unique where true" variant of the external uniqueness constraint is introduced to allow roles spanned by such constraints to occur in unary fact types. This constraint is exploited to enable graphic portrayal of a new corollary to a schema transformation pattern that occurs in many business domains. An alternative transformation is introduced to optimize the same pattern, and then generalized to cater for more complex cases. The relational mapping algorithm is extended to cater for the new results, with the option of retaining the original patterns for conceptual discussion, with the transforms being applied internally in a preprocessing phase. The procedures are being implemented in NORMA, an open-source tool supporting the ORM 2 version of fact-oriented modeling.

1 Introduction

Object-Role Modeling (ORM) is a fact-oriented approach for modeling, transforming, and querying information in terms of the underlying facts of interest, where facts and rules may be verbalized in language easily understood by non-technical users of the business domain. Unlike Entity-Relationship (ER) models [3] and Unified Modeling Language (UML) class diagrams [17], ORM models are attribute-free, treating all facts as relationships (unary, binary, ternary etc.). ORM includes techniques for mapping to attribute-based structures, such as those of ER or UML. Fact-oriented modeling includes a number of closely related dialects, such as Natural language Information Analysis Method (NIAM) [18] and Fully-Communication Oriented Information Modeling (FCO-IM) [1]. First generation ORM is covered in detail in [8], and compared with UML in [11]. This paper uses the notation of second generation ORM (ORM 2) [12]. For a recent overview of fact-oriented modeling, see [14].

To ensure correctness and completeness, informational models are best specified first at a conceptual level where they can be validated by the business domain expert, before being transformed to implementation structures such as relational database schemas, class models or XML schemas. Regardless of the modeling notation used, there are always many correct ways to model the same business domain. The problem of determining whether different schemas model the same domain is known as the schema equivalence problem. Schema equivalence at the conceptual level has long been investigated in ER (e.g. [2]), ORM (e.g. [6, 5, 7]) and other approaches.

Conceptual schema transformations enable one conceptual schema to be transformed to another conceptual schema that is either equivalent to it or implied by it. When passed to a standard logical mapping procedure, different but equivalent conceptual schemas may map to logical data structures (e.g. relational schemas or XML schemas) that differ in efficiency. Various heuristics have been developed to perform conceptual schema optimization, wherein conceptual schemas are pre-transformed to acceptably equivalent conceptual schemas that result in more efficient logical schemas when logical mapping procedures are applied.

This paper extends earlier work on conceptual schema transformation and optimization in ORM [6, 7, 8] by considering *reduction transformations*, a class of transforms that eliminate unneeded or redundant columns that may arise when applying standard relational mapping procedures (e.g. Rmap [8]) to ORM schemas. The Rmap algorithm is extended to cater for the new results, with the option of retaining the original patterns for conceptual discussion, with the transforms being applied internally in a preprocessing phase. The procedures are being implemented in NORMA [4], an open-source tool supporting the ORM 2 version of fact-oriented modeling.

The rest of this paper is structured as follows. Section 2 introduces a "unique where true" variant of the external uniqueness constraint to allow roles spanned by such constraints to occur in unary fact types. Section 3 exploits this constraint to enable graphic portrayal of a new corollary to a schema transformation pattern that occurs in many business domains. An alternative transformation is also introduced to optimize the same pattern. Sections 4 and 5 generalize the alternative transformation to cater for more complex cases. Section 6 summarizes the main results, suggests topics for further research, and lists references.

2 "Unique Where True" Constraints

As an extension to ORM, we now add a *"unique where true" constraint*. This is essentially a restricted form of external uniqueness constraint, where at least one role spanned by the constraint must belong to a unary fact type, and the constraint applies only where it's true that instances of the join type (here Politician) instantiate the unary. The constraint is symbolized by a circled uniqueness bar "⊖" overlaid with "T" (for "True"), attached to the roles it spans, as shown in Fig. 1.

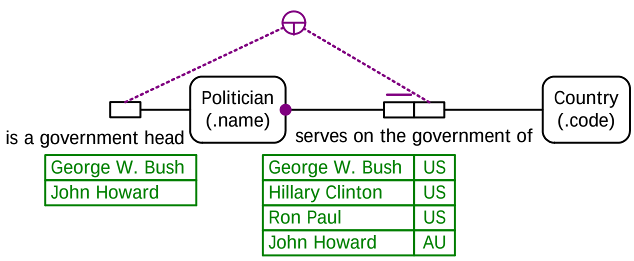
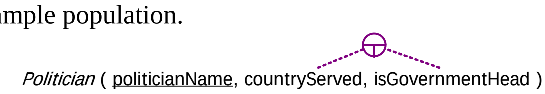


Fig. 1. Example of a "Unique where True" constraint

In this example, the constraint requires at most one head of government (President, Prime Minister etc.) for each country. For example, the United States has many politicians but only one head (illustrated by the sample population). In positive form [9], the constraint verbalizes as For each Country, at most one Politician serves on the government of that Country and is a government head, and in negative form as For each Country, it is impossible that more than one Politician serves on the government of the same Country and is a government head. If we were to add an entry for Hillary Clinton to the population of the unary fact type, this would provide a counterexample to the constraint.

The relational mapping algorithm is extended to enforce the constraint by applying the constraint to the columns mapped from the constrained roles, as shown in Fig. 2 along with the sample population.



George W. Bush	US	True
Hillary Clinton	US	False
Ron Paul	US	False
John Howard	AU	True

Fig. 2. Relational schema mapped from the ORM schema in Fig. 1

In the SQL standard, the constraint may be trivially enforced by the following check clause. In most industrial SQLs, which do not support rich check clauses, such constraints may be easily recoded as triggers or stored procedures.

```
check not exists (select countryServed, isGovernmentHead from Politican
                  where isGovernmentHead is true
                  group by countryServed, isGovernmentHead having count(*) > 1)
```

To cater for the "unique where true" constraint, we have added yet another graphic symbol to the already rich constraint notation of ORM 2. However, we feel this is justified, because the constraint is common in business domains whose models include unary fact types (UML and ER typically do not support unaries directly), and the constraint is easy to enforce in code. We have met dozens of such examples in real applications (department heads, country capitals etc.) and have often been asked by others "How can you state this kind of constraint in ORM?". An alternative is to remodel the unary as the 1:1 binary Politician heads Country with a subset constraint between the associations, leading to the relation scheme Politician (politicianName, countryServed, [countryHeaded]) with an equal-where-not-null constraint between the country attributes.

3 Reduction Transforms for a Common Data Model Pattern

We now consider extensions to a class of schema transformation that one of us introduced in [6, pp.6-51, 7-13, 7-14]. We call these *reduction transformations*, since they eliminate unneeded or redundant columns that may arise when applying standard relational mapping procedures. Two basic schema equivalence patterns discussed in [8, pp. 611, 640] are shown in Fig. 3. We now refer to transforms to the right-hand alternative as *role elimination* (since they eliminate an unneeded role).

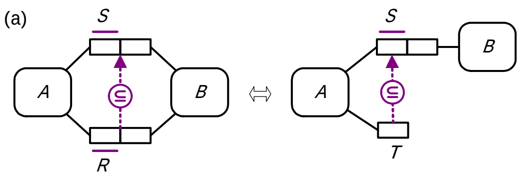


Fig. 3. Two basic reduction transformations

For both cases shown in Fig. 3, the subset constraint combined with the UC (uniqueness constraint) on the upper predicate implies that player instances of the second role in the lower predicate must match the player instances of the second role in the upper predicate (for the same player of the first role). Hence without information loss, the lower predicate may be reduced by removing its second role.

Suppose now that our politician example is modeled using just binary fact types as shown in Fig. 4(a). This matches the pattern of Fig. 3(a), but adds a mandatory role constraint, and a uniqueness constraint on the lower right role. The equivalence in Fig. 3(a) licenses the reduction transform to our earlier schema, repeated in Fig. 4(b), except for the unique-when-true constraint, which clearly derives from the additional uniqueness constraint on the lower right role of Fig. 4(a). The mandatory constraint transfers without change, and implies a subset constraint from the unary to the left upper role in Fig. 4(b), so the subset constraint in Fig. 3(b) is satisfied.

This example illustrates the following corollary to the equivalence theorem in Fig. 3(a). *Adding a uniqueness constraint to the righthand role of R adds a unique-where-true constraint spanning T and the righthand role of S.*

If the schema in Fig. 4(a) is passed to the Rmap algorithm, we obtain the relation scheme Politician(politicianName, countryServed, [countryGoverned]) along with the constraint check(countryGoverned is null or countryGoverned = countryServed). For most cases however, the relational schema in Fig. 2 mapped from the reduced ORM schema is preferable.

The equivalent capital city models in Fig. 5 illustrate an extended version of this kind of reduction transform, where the new constraint has been strengthened to a *"unique and mandatory where true" constraint*. Here the addition of the mandatory dot on the constraint symbol indicates that it is mandatory for a country to have a capital city. In the rare case where more than one unary is involved in the constraint, both individual and disjunctive mandatory constraints may be added as needed. Another difference here is the composite identification scheme for MainCity (both the country code and the city name are required to identify a main city).

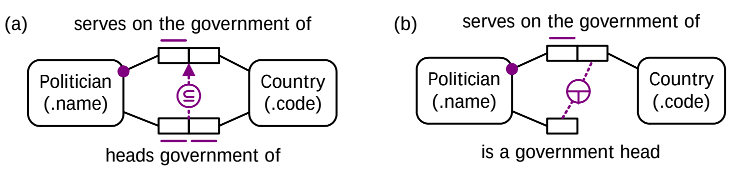


Fig. 4. Adding the right-hand UC in (a) adds the "unique where true" constraint in (b)

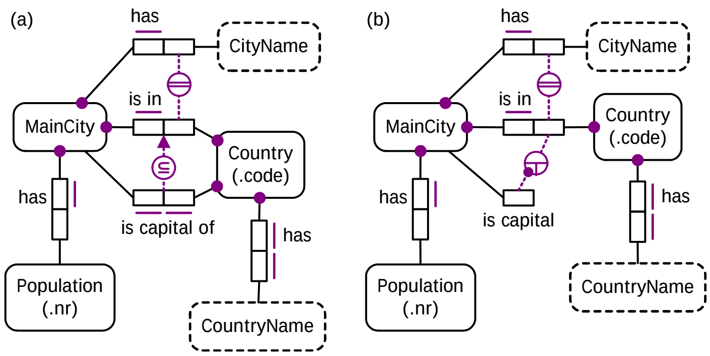


Fig. 5. Schema (a) remodeled in (b) with a "unique and mandatory where true" constraint

The relational schemas obtained by mapping the ORM schemas in Fig. 5 are displayed in Fig. 6. Keys are underlined, doubly-underlining primary keys if other candidate keys exist. Arrows denote subset constraints, and the other symbols should be self-explanatory. In Fig. 5(a), the 1:1 fact type has only one mandatory role, played by Country, so this fact type maps to the Country table. Notice again the "redundant" second country code in the relational schema of Fig. 6(a).

The addition of the mandatory constraint and the composite reference scheme in Fig. 5(a) allows an alternative reduction transform (that we call *role redirection*) to be performed, that was not possible with the previous pattern in Fig. 4. To avoid repeating the country information when declaring a country's capital, we connect the capital city predicate directly to CityName instead of MainCity, as shown in Fig. 7. The pair subset constraint involves a conceptual join [10] on MainCity, and ensures that the capital city name is one of the country's main city names.

The relational schema obtained by mapping this alternative ORM schema is shown in Fig. 7. Compared with the relational schema in Fig. 6(a), the new relational schema avoids the "redundant" country code column and is clearly more efficient. Yet another way to avoid the "redundant" column is to introduce a simple, artificial identifier for MainCity (e.g. MainCityNr)—this option is less attractive unless we wish to track cities that change their name over time. In implementing these new features in NORMA [4], we plan to detect such patterns and offer options of performing either reduction transform (either explicitly or internally only) or introducing artificial keys.

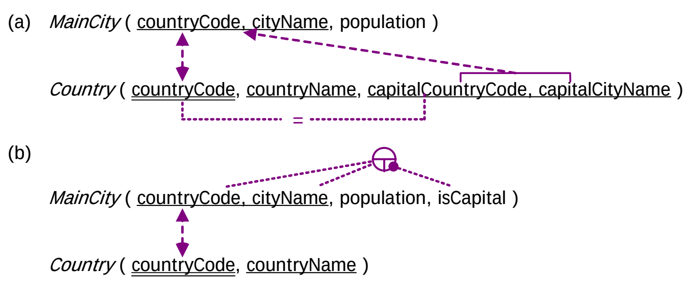


Fig. 6. Relational schemas mapped from the ORM schemas in Fig. 5

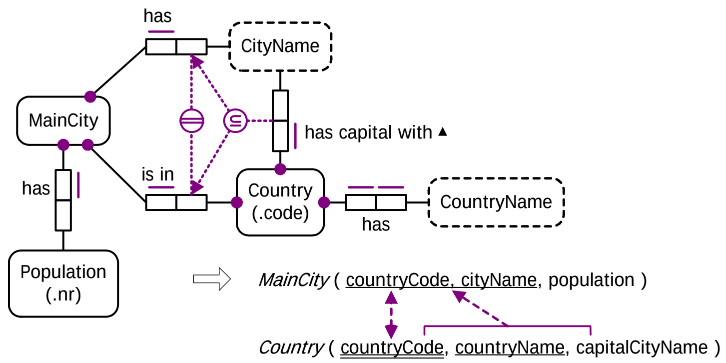


Fig. 7. A second alternative ORM schema for the city example and its relational map

Fig. 8 shows a new schema equivalence pattern that underlies the role redirection transform from Fig. 5(a) to Fig. 7. The following two corollaries are also employed in this transform: adding a UC to *A*'s role in *S* transforms to a UC on *A*'s role in *S'*; making *A*'s roles mandatory in the left schema makes them mandatory in the other. This result can be extended further to *n*-part (*n* > 2) composite reference schemes.

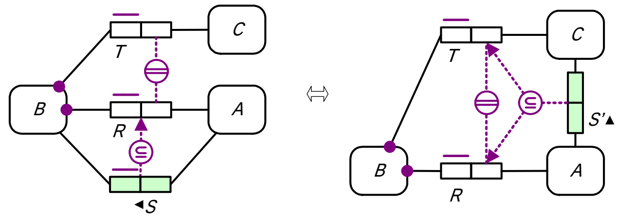


Fig. 8. A schema equivalence pattern underlying the basic role direction transform

4 A More Complex Case

We now analyze some more-complex schemas in order to illustrate the more general pattern to which role redirection can be applied when role elimination cannot. First, we perform a simple transform on the schema in Fig. 5(a).

In Fig. 9(a) Country's simple reference scheme is made explicit, and MainCity is in Country is transformed to MainCity has CountryCode. Consequently, the subset constraint of Fig. 5(a) becomes a join subset constraint. Fig. 9(b) is the same as 9(a) except that both the constraints in Fig. 9(a) that are implied by other constraints are left implicit, viz., the uniqueness constraint on the left-hand role of MainCity is capital of Country and the mandatory-role constraint on the right-hand role of MainCity has CountryCode.

Though not shown here, a derived binary fact type may be defined as an abbreviation of the path from MainCity through CountryCode to Country, the first role of which is functional (transitively implied FD). Treating this derived fact type as *S* in Fig. 3(a) implies a uniqueness constraint on the left-hand role of MainCity is capital of Country (the uniqueness constraint on predicate *R* in this pattern is derivable).

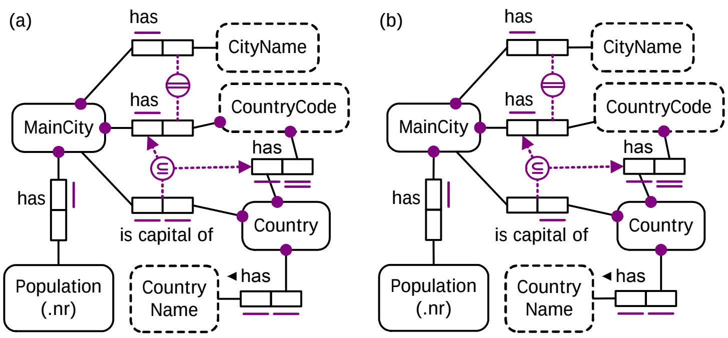


Fig. 9. Two displays of a schema equivalent to that of Fig. 5(a)

But compare the schema in Fig. 9(b) with that of Fig. 10(a). Note that in the USA, the same postal code might be used in more than one State. The lack of a uniqueness constraint on the lower role of StateOrProvince prevents the first role of the implicit, derived binary fact type defined as the path from LegislativeDistrict through StateOrProvince to Address from being functional. Without an implied UC on the district role in Address is in LegislativeDistrict, we cannot replace that role with a unary fact type without loss of information. Thus here, the role elimination transform is unavailable.

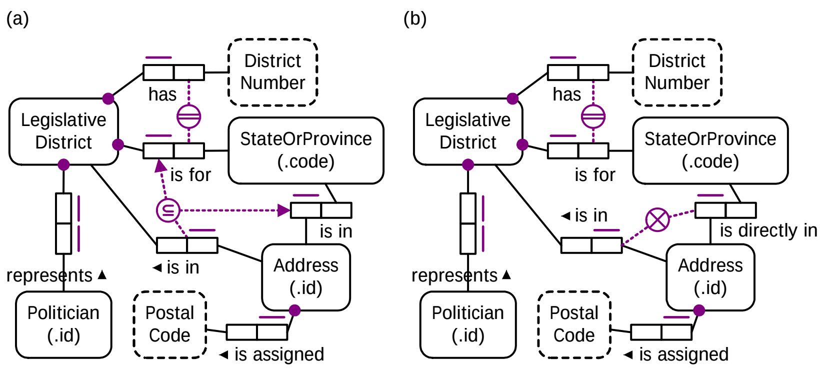


Fig. 10. Two schemas like Fig. 8's, but which cannot use the role elimination transform

Yet there is still scope for a reduction transform; for the relational schema to which the ORM schemas of Figs. 9(a) and 10(a) would map has a "redundant" column in the Address table, similar to the one in the Country table in Fig. 6(a). One could include the exclusion constraint which, in Fig. 10(b), replaces the subset constraint. However, the relational mapping for this alternative ORM schema does not eliminate the "redundant" column; it only forbids the "redundant" data that would fill it, and replaces this data with nulls. Thus, deploying the exclusion constraint gains us very little.

As in the earlier example, the source of the "redundant" column is the way the subset constraint is situated in relation to the composite reference scheme. Thus we may use a transform similar to the one we used in Fig. 7, but more complex, as in Fig. 11.

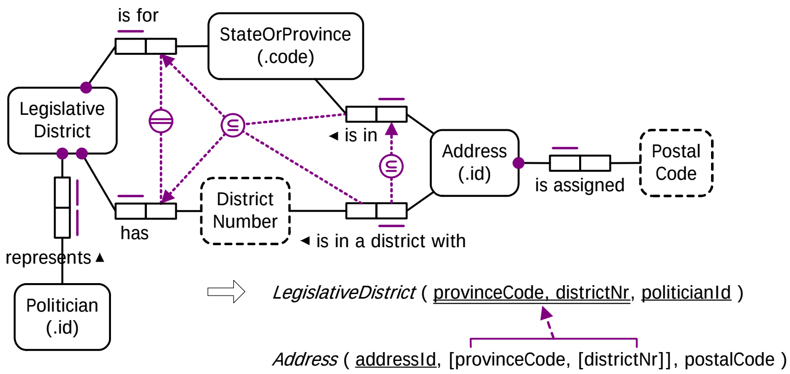


Fig. 11. An alternative ORM schema for the legislative-district example and its relational map

The equivalence pattern underlying this more complex transform is shown in Fig. 12 (predicate *S* is redirected from *B* to become *S'* connected directly to *C*). In Fig. 11, Address corresponds to *A* and DistrictNumber to *C*. A simpler form, as well as corollaries to this result are omitted here because of space limitations.

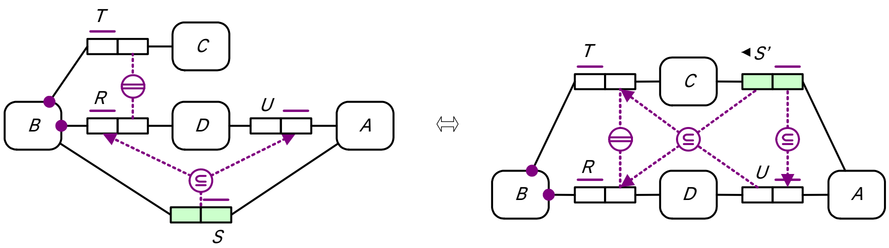


Fig. 12. A schema equivalence pattern underlying the more complex role redirection transform

5 An Even More Complex Case

The role redirection transform in Fig. 12 may sometimes be reapplied to multiple portions of an ORM schema. Fig. 13(a) is a fragment[1] from a real-world schema, in which a comment can be about at most one of a song, an album, or a band. As in Fig. 10, an alternative is provided in Fig. 13(b), with the exclusion constraint replaced by join subset constraints; here, if a comment is about a song it must also be about the album that that song is part of. Likewise, if a comment is about an album, it must also be about the band for that album. Although not shown here, this alternative could be represented by removing the join subset constraints and making the Comment is about Album and Comment is about Band fact types semi-derived [13] (with obvious derivation rules). The ORM schemas are essentially equivalent when relationally mapped, each having a comment id and at most one of a song name, album name, or band name.

[1] The fragment includes only the relevant portions of the schema. The comment, song, album, and band object types play additional roles not pertinent to this reduction transform.

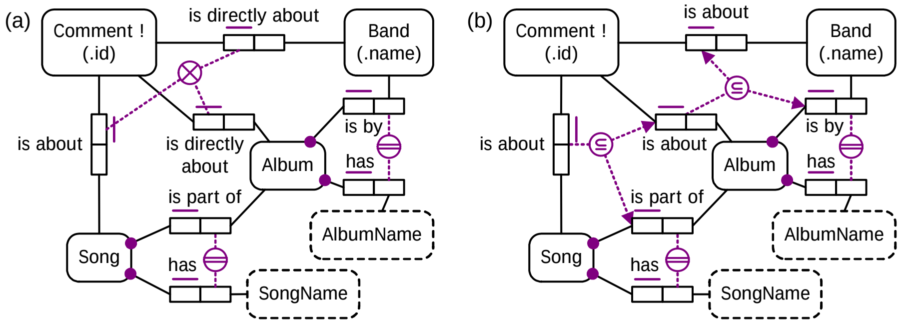


Fig. 13. Two schemas to which role redirection can be applied in multiple locations

Applying the reduction transform to Comment is about Song and Comment is about Album, eliminates the "redundant" columns, resulting in the schemas depicted in Fig. 14.

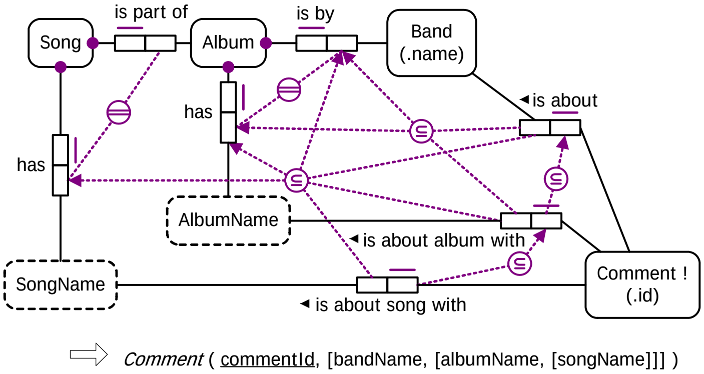


Fig. 14. The fully-reduced form of the ORM and relational schemas from Fig. 13

6 Conclusion

This paper examined two classes of reduction transformations in ORM called role elimination and role redirection that eliminate unneeded columns that may arise from standard relational mapping procedures. Two extensions to external uniqueness constraints ("unique where true", and "unique and mandatory where true") were introduced to graphically portray various role elimination transforms. Complex cases of reduction transforms were considered, along with their impact on relational mapping, and new schema equivalence theorems were identified.

The procedures are currently in the early stages of being implemented in NORMA [4], an open-source tool supporting the ORM 2 version of fact-oriented modeling. Different options (e.g. role elimination, role redirection, or surrogate identifier introduction) are often available to reduce the kind of column duplication addressed in this paper. Moreover, applications of these transform options may have positive or negative impact of the readability of the conceptual schema. For this reason, our current implementation strategy is to first automatically detect opportunities for optimization and then provide modelers with the option of retaining the original patterns for conceptual discussion; if they choose that option, the transforms will be applied only internally in a preprocessing phase to the relational mapping.

NORMA currently maps to other targets besides relational schemas (e.g. class models and XML schemas). One research question still to be investigated is the impact of applying reduction transformations when mapping to such different targets.

References

1. Bakema, G., Zwart, J. & van der Lek, H. 2000, *Fully Communication Oriented Information Modelling*, Ten Hagen Stam, The Netherlands.
2. Batini, C., Ceri, S. & Navathe, S. 1992, *Conceptual Database Design: an entity-relationship approach*, Benjamin/Cummings, Redwood City.
3. Chen, P. P. 1976, 'The entity-relationship model—towards a unified view of data'. ACM Transactions on Database Systems, 1(1), pp. 9–36.
4. Curland, M. & Halpin, T. 2007, 'Model Driven Development with NORMA', *Proc. 40th Int. Conf. on System Sciences (HICSS-40)*, 10 pages, CD-ROM, IEEE Computer Society.
5. De Troyer, O. 1993, 'On Data Schema Transformations', PhD thesis, University of Tilburg (KUB), Tilburg, The Netherlands.
6. Halpin, T. 1989, 'A Logical Analysis of Information Systems: static aspects of the data-oriented perspective', doctoral dissertation, University of Queensland.
7. Halpin, T. & Proper, H. 1995, 'Database schema transformation and optimization', *Proc. OOER'95*, Springer LNCS, vol. 1021, pp. 191-203.
8. Halpin, T. 2001, *Information Modeling and Relational Databases*, Morgan Kaufmann, San Francisco.
9. Halpin, T. 2004, 'Business Rule Verbalization', *Information Systems Technology and its Applications*, Proc. ISTA-2004, (eds Doroshenko, A., Halpin, T., Liddle, S. & Mayr, H.), Salt Lake City, Lec. Notes in Informatics, vol. P-48, pp. 39-52.
10. Halpin, T. 2005, 'Constraints on Conceptual Join Paths', *Information Modeling Methods and Methodologies*, eds J. Krogstie, T. Halpin, & K. Siau, Idea Pub., Hershey, pp. 258-77.
11. Halpin, T. 2005, 'Information Modeling in UML and ORM: a Comparison', *Encyclopedia of Inf. Science and Tech.*, vol. 3, ed. M. Khosrow-Pour, Idea Pub., Hershey, pp. 1471-5.
12. Halpin, T. 2005, 'ORM 2', *On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops*, eds R. Meersman, Z. Tari, et al., Cyprus. Springer LNCS 3762, pp 676-87.
13. Halpin, T. 2007, 'Subtyping Revisited', *Proc. CAiSE'07 Workshops vol. 1*, eds. B. Pernici & J. Gulla, Tapir Academic Press, pp. 131-141.
14. Halpin, T. 2007, 'Fact-Oriented Modeling: Past, Present and Future', *Conceptual Modelling in Inf. Sys. Eng.*, eds. J. Krogstie, A. Opdahl & S. Brinkkemper, Springer, pp. 19-38.
15. ter Hofstede, A. H. M. 1993, 'Information Modelling in Data Intensive Domains', PhD thesis, University of Nijmegen.
16. ter Hofstede, A. H. M., Proper, H. A. & Weide, th. P. van der 1993, 'Formal definition of a conceptual language for the description and manipulation of information models', *Information Systems*, vol. 18, no. 7, pp. 489-523.
17. Object Management Group 2003, *UML 2.0 Superstructure Specification*. Online at: www.omg.org/uml.
18. Wintraecken J. 1990, *The NIAM Information Analysis Method: Theory and Practice*, Kluwer, Deventer, The Netherlands.