Model Transformation Composition

Mark Asztalos, Marouane Kessentini, **Eugene Syriani**, and Manuel Wimmer

The HOT group

MOTIVATION

- Transitive transformation

$$\text{UML} \rightarrow \text{ER} \rightarrow \text{RDBMS}$$

- Language evolution

$$\text{UML} \rightarrow \text{RDBMS} \rightarrow \text{RDBMS'}$$

COMPOSITION OF TRANSFORMATIONS

- **Definition**

  Given: $MM_1 \xrightarrow{T1} MM_2 \xrightarrow{T2} MM_3$

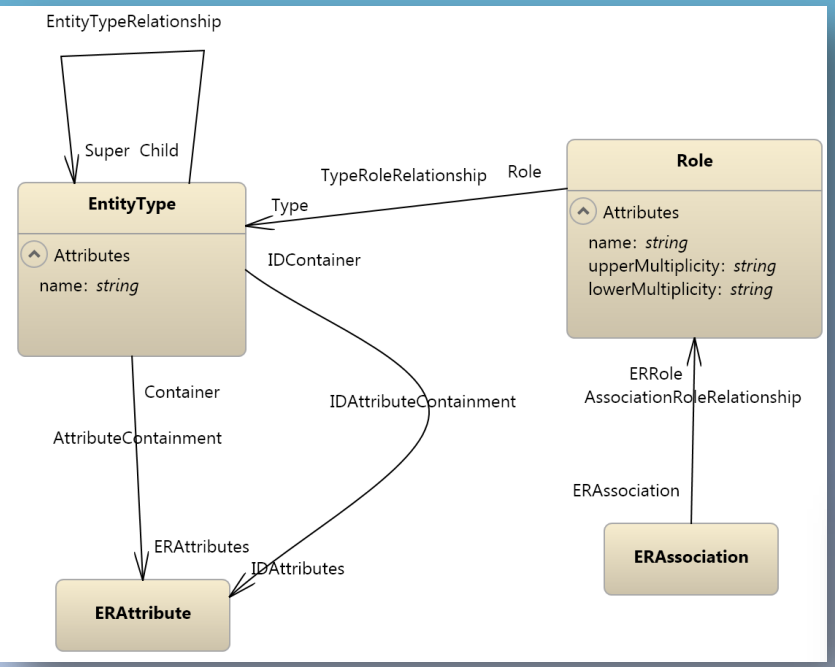
  1. $T_3 \equiv T_2 \circ T_1$
  2. No occurrences of an element of MM_2 shall be found in T_3

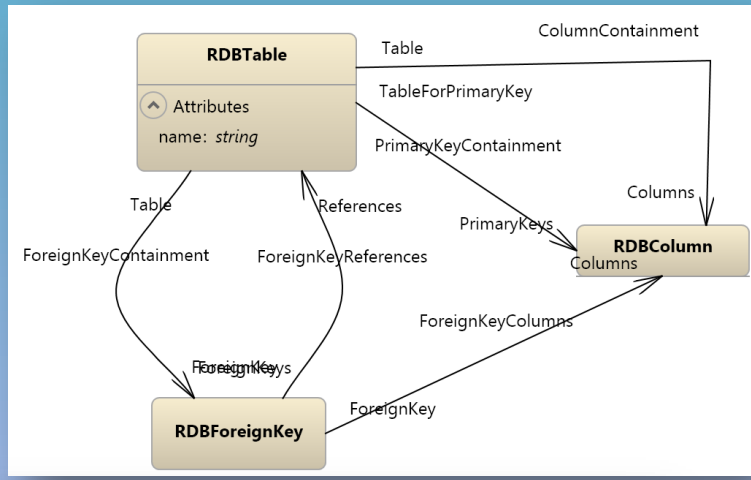
- **Theoretical foundation**
  - Graph transformation rule composition
  - Formalized in Category Theory

EXAMPLE

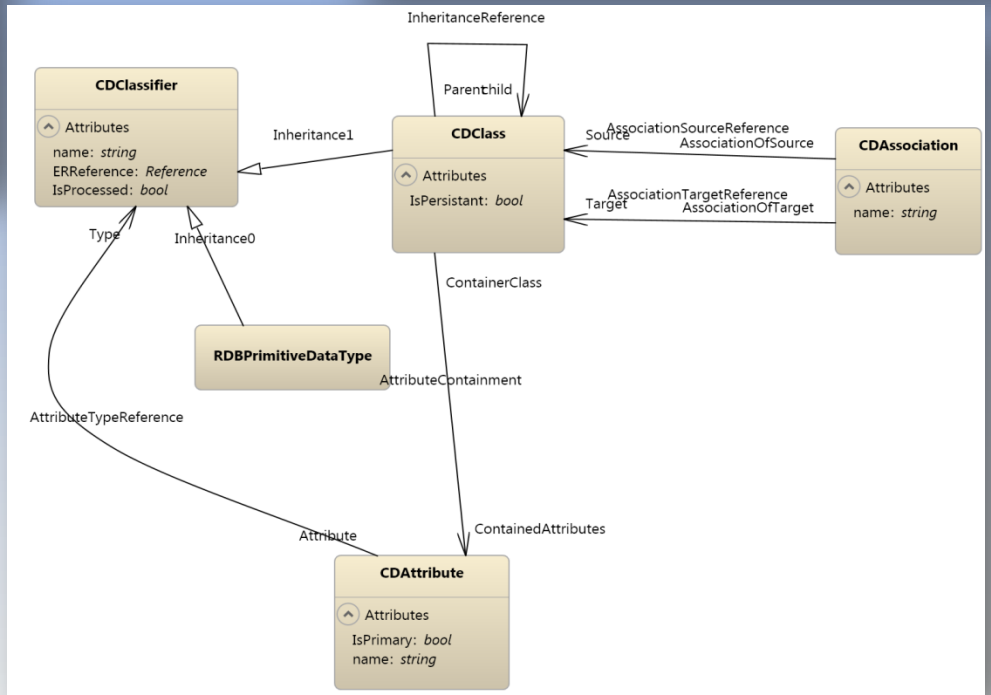
UML→ER →RDBMS

RDBMS MM





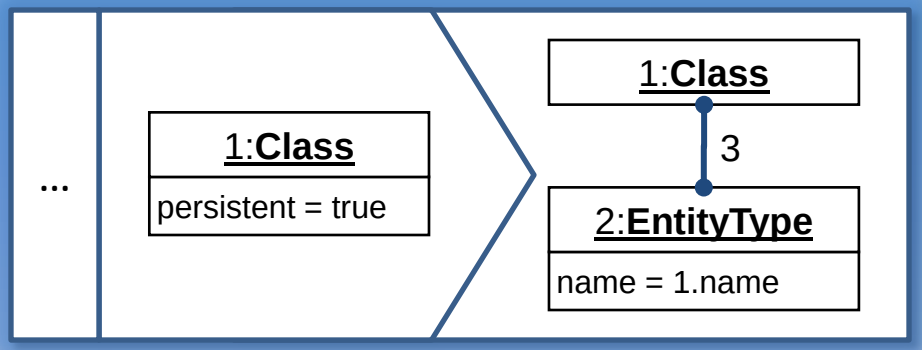
Entity-Relationship MM

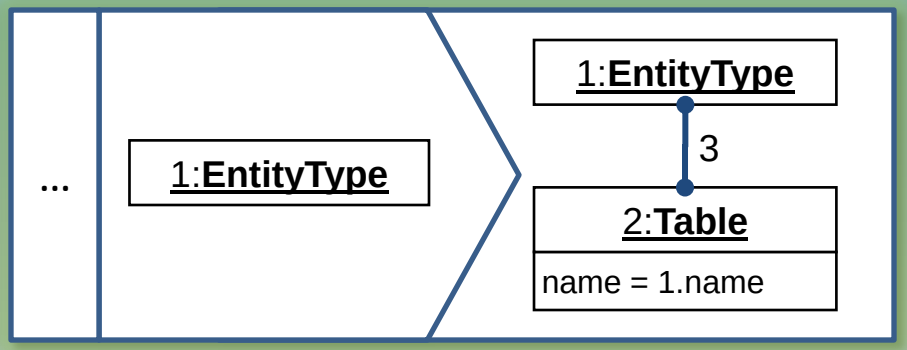


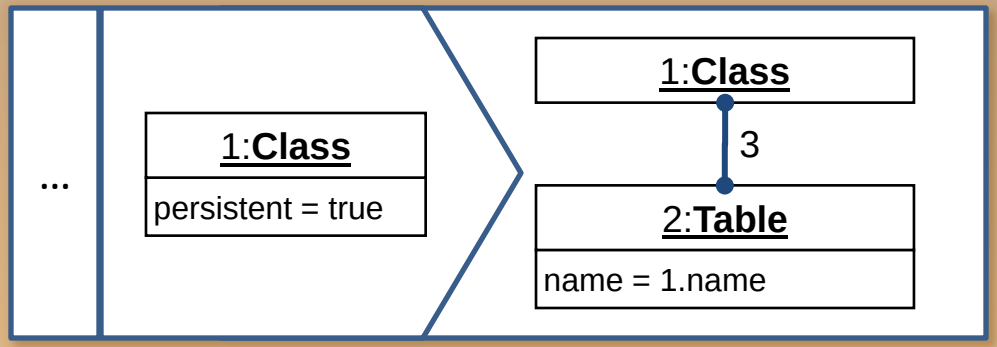
UML-CD MM

EXAMPLE RESULT

- 2 simple rules (CD-ER & ER-RDBMS)
- Result (CD-RDBMS)





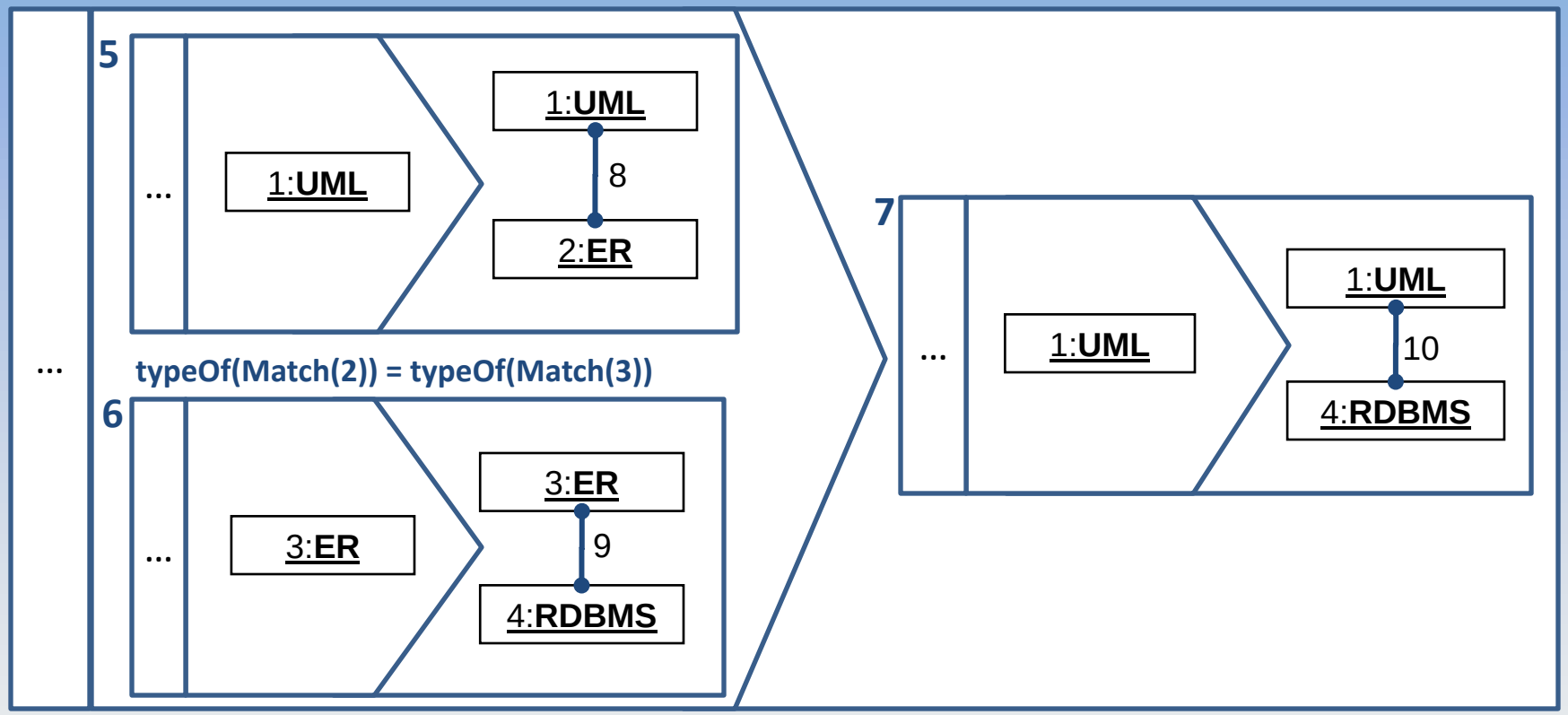


SOLUTION

$$MM_1 \xrightarrow{T1} MM_2 \xrightarrow{T2} MM_3$$

$$T_3 = T_C(T_1, T_2)$$

- **HOT rule**



CHALLENGES

- Rule complexity
  - Attribute assignment
  - n-m correspondences
  - Constraints + NACs
- Assumption
  - There exists a MM of the MTL (cf. CAMPaM'09)
  - Rule creates traceability links
  - When to delete intermediate MM?
- Validation
  - Test of exactly same model outputs for same input model
    - If fails: analyze generated transformation to give incite on the design fault in the HOT
  - Given an optimal transformation, compare generated transitive transformation
- Control flow (out of scope)

IMPLEMENTATION

- VMTS

RELATED WORK

In DB community

- Context: **Model management**
- Models are defined as schemas
- Mappings are defined as correspondences between schemas
- Operators that can modify a schema or a mapping
  - Diff
  - Merge
  - Inverse
  - Compose