

**UML and OCL Transformation Model Analysis:
Checking Invariant Independence**

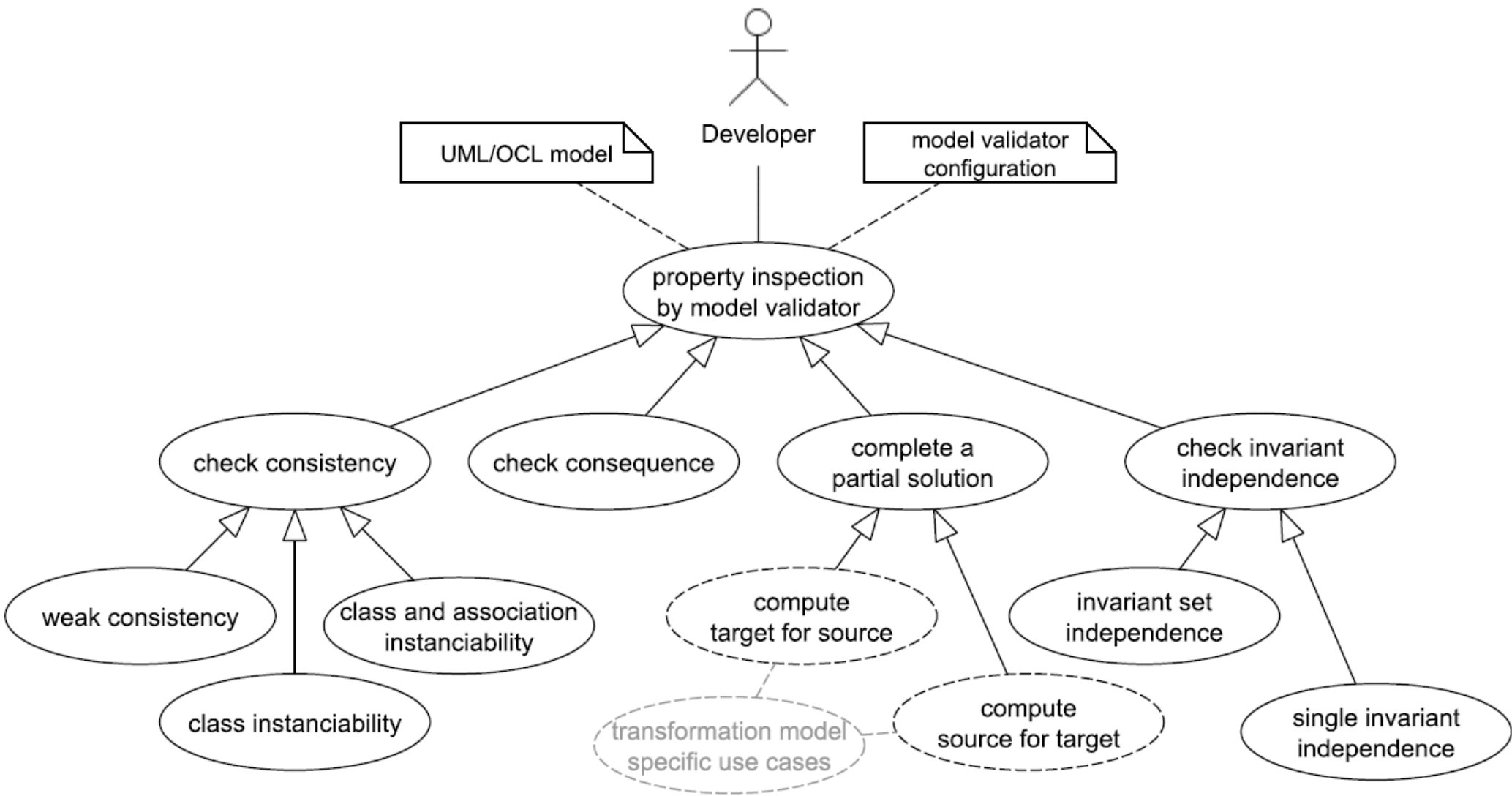
**Martin Gogolla, Frank Hilken
University of Bremen, Database Systems Group**

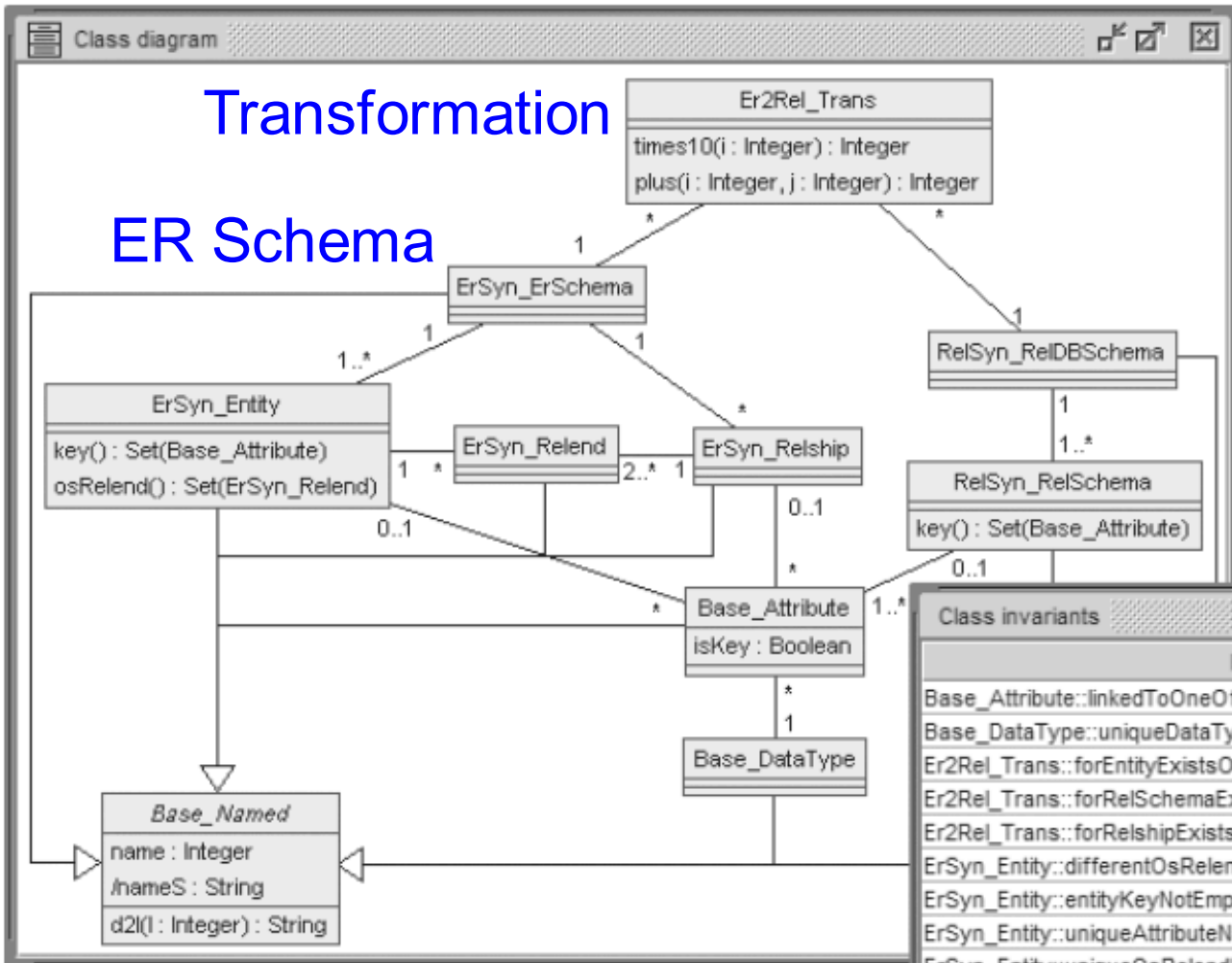
Motivation and context

- consider transformation models in form of UML **class** diagrams and enriched by OCL **invariants**
- support development of such models with the tool **USE** (Uml-based Specification Environment)
- USE gives support for **object**, statechart, sequence and communication **diagrams** and imperative operation implementation
- model validator on basis of Kodkod automatically **constructs object diagrams** for UML and OCL models
- prove transformation model properties
 - **invariant independence**, i.e., automatically check that each single invariant is not a consequence of other invariants
- consider example model representing the transformation between the Entity-Relationship (ER) and the relational data model

ER schema <-----> Rel. DB Schema
Transformation

aim: independence can effectively be proved for a non-trivial transformation model





Class invariants

Invariant	Result
Base_Attribute::linkedToOneOfEntityRelshipRelSchema	true
Base_DataType::uniqueDataTypeNames	true
Er2Rel_Trans::forEntityExistsOneRelSchema	true
Er2Rel_Trans::forRelSchemaExistsOneEntityXorRelship	true
Er2Rel_Trans::forRelshipExistsOneRelSchema	true
ErSyn_Entity::differentOsRelendAndAttributeNamesWithinEntity	true
ErSyn_Entity::entityKeyNotEmpty	true
ErSyn_Entity::uniqueAttributeNamesWithinEntity	true
ErSyn_Entity::uniqueOsRelendNamesWithinEntity	true
ErSyn_ErSchema::differentEntityAndRelshipNamesWithinErSchema	true
ErSyn_ErSchema::uniqueEntityNamesWithinErSchema	true
ErSyn_ErSchema::uniqueErSchemaNames	true
ErSyn_ErSchema::uniqueRelshipNamesWithinErSchema	true
ErSyn_Relend::c_Relend_Entity_Relship_ErSchema	true
ErSyn_Relship::differentRelendAndAttributeNamesWithinRelship	true
ErSyn_Relship::relshipKeyEmpty	true
ErSyn_Relship::uniqueAttributeNamesWithinRelship	true
ErSyn_Relship::uniqueRelendNamesWithinRelship	true
RelSyn_ReIDBSchema::uniqueReIDBSchemaNames	true
RelSyn_ReIDBSchema::uniqueRelSchemaNamesWithinReIDBSchema	true
RelSyn_RelSchema::relSchemaKeyNotEmpty	true
RelSyn_RelSchema::uniqueAttributeNamesWithinRelSchema	true

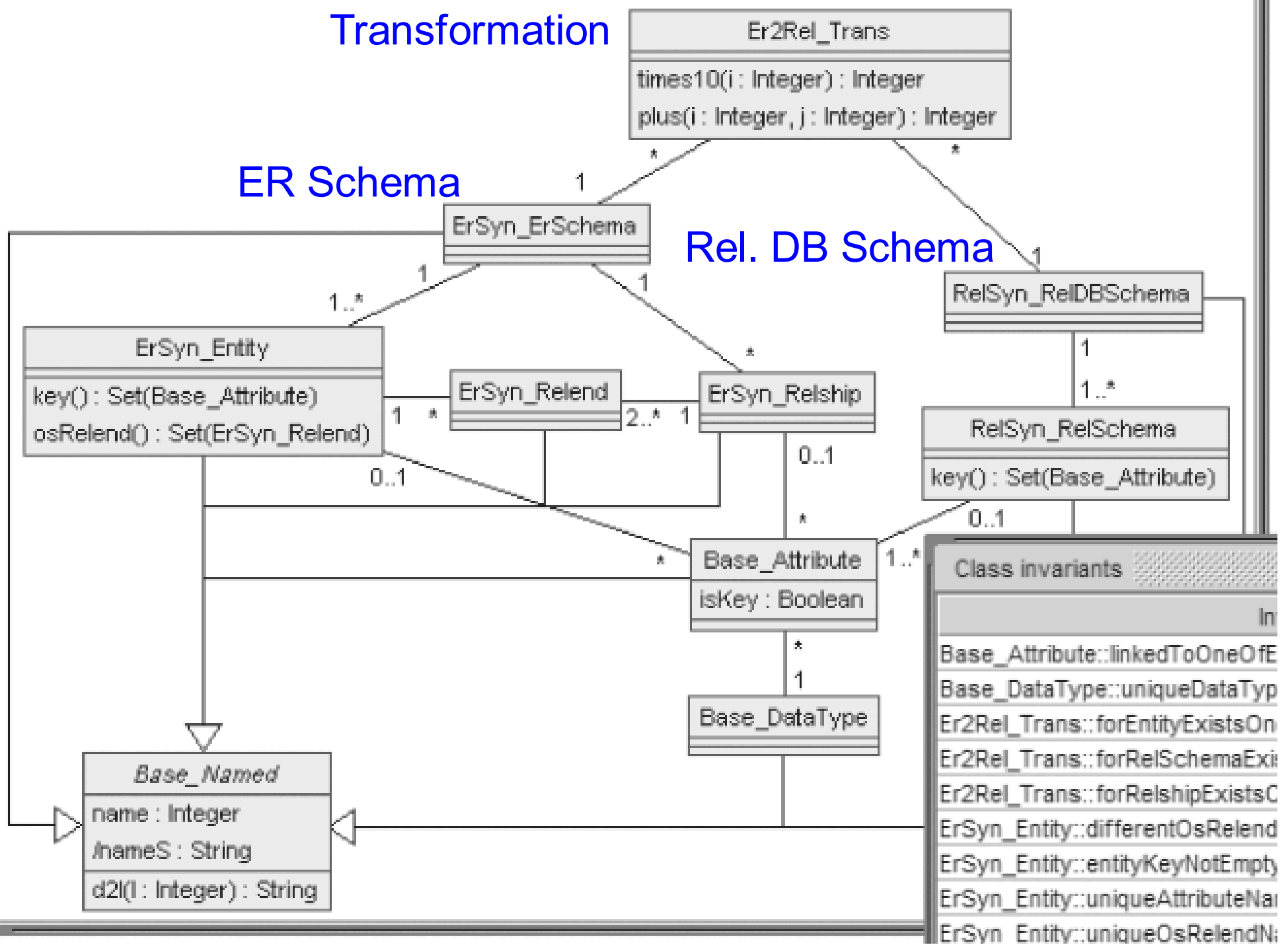
Constraints ok. (0ms) 100%

Invariants

Transformation

ER Schema

Rel. DB Schema



Class invariants	
Class	Invariant
Base_Attribute	linkedToOneOfE
Base_DataType	uniqueDataTyp
Er2Rel_Trans	forEntityExistsOn
Er2Rel_Trans	forRelSchemaExi
Er2Rel_Trans	forRelshipExistsC
ErSyn_Entity	differentOsRelend
ErSyn_Entity	entityKeyNotEmpty
ErSyn_Entity	uniqueAttributeNa
ErSyn_Entity	uniqueOsRelendN

Class invariants



Invariant	Result
Base_Attribute::linkedToOneOfEntityRelshipRelSchema	true
Base_DataType::uniqueDataTypeNames	true
Er2Rel_Trans::forEntityExistsOneRelSchema	true
Er2Rel_Trans::forRelSchemaExistsOneEntityXorRelship	true
Er2Rel_Trans::forRelshipExistsOneRelSchema	true
ErSyn_Entity::differentOsRelendAndAttributeNamesWithinEntity	true
ErSyn_Entity::entityKeyNotEmpty	true
ErSyn_Entity::uniqueAttributeNamesWithinEntity	true
ErSyn_Entity::uniqueOsRelendNamesWithinEntity	true
ErSyn_ErSchema::differentEntityAndRelshipNamesWithinErSchema	true
ErSyn_ErSchema::uniqueEntityNamesWithinErSchema	true
ErSyn_ErSchema::uniqueErSchemaNames	true
ErSyn_ErSchema::uniqueRelshipNamesWithinErSchema	true
ErSyn_Relend::c_Relend_Entity_Relship_ErSchema	true
ErSyn_Relship::differentRelendAndAttributeNamesWithinRelship	true
ErSyn_Relship::relshipKeyEmpty	true
ErSyn_Relship::uniqueAttributeNamesWithinRelship	true
ErSyn_Relship::uniqueRelendNamesWithinRelship	true
RelSyn_ReIDBSchema::uniqueReIDBSchemaNames	true
RelSyn_ReIDBSchema::uniqueRelSchemaNamesWithinReIDBSchema	true
RelSyn_RelSchema::relSchemaKeyNotEmpty	true
RelSyn_RelSchema::uniqueAttributeNamesWithinRelSchema	true

Constraints ok. (0ms)

100%

Configuration for proving invariant independence

Er2Rel_Trans : 1..1		
Er2Rel_OwnershipTransErschema : 1..1		
Er2Rel_OwnershipTransRelDBSchema : 1..1		
ErSyn_Erschema : 1..2	RelSyn_RelDBSchema : 1..2	
ErSyn_Entity : 0..3	RelSyn_RelSchema : 0..3	
ErSyn_Relship : 0..3		
ErSyn_Relend : 0..4		
ErSyn_OwnershipErschemaEntity : *	RelSyn_OwnershipRelDBSchemaRelSchema : *	
ErSyn_OwnershipErschemaRelship : *	RelSyn_OwnershipRelSchemaAttribute : *	
ErSyn_OwnershipEntityAttribute : *		
ErSyn_OwnershipRelshipAttribute : *		
ErSyn_OwnershipRelshipRelend : *		
ErSyn_RelendTyping : *		
Base_Attribute : 0..8		
Base_DataType : 1..2		
Base_Named_name : Set{0,1,2,3,4,5,6,7,8,9,10, ..., 89,90,91,92,93,94,95,96,97,98,99}		
Base_Attribute_isKey : Set{false,true}		
Base_AttributeTyping : *		
Integer : 0..99		
Real : Set{}		
String : Set{}		

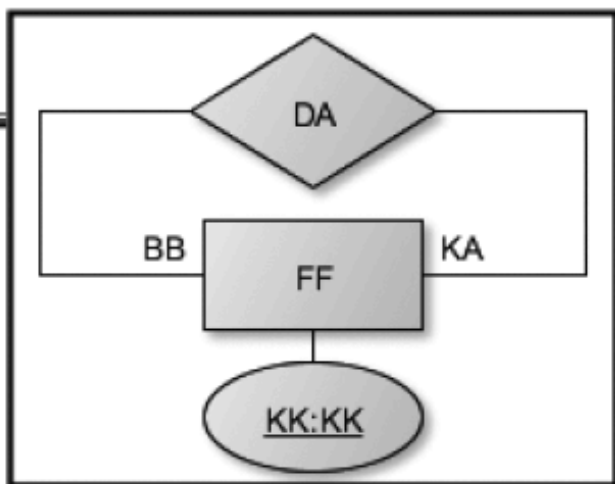
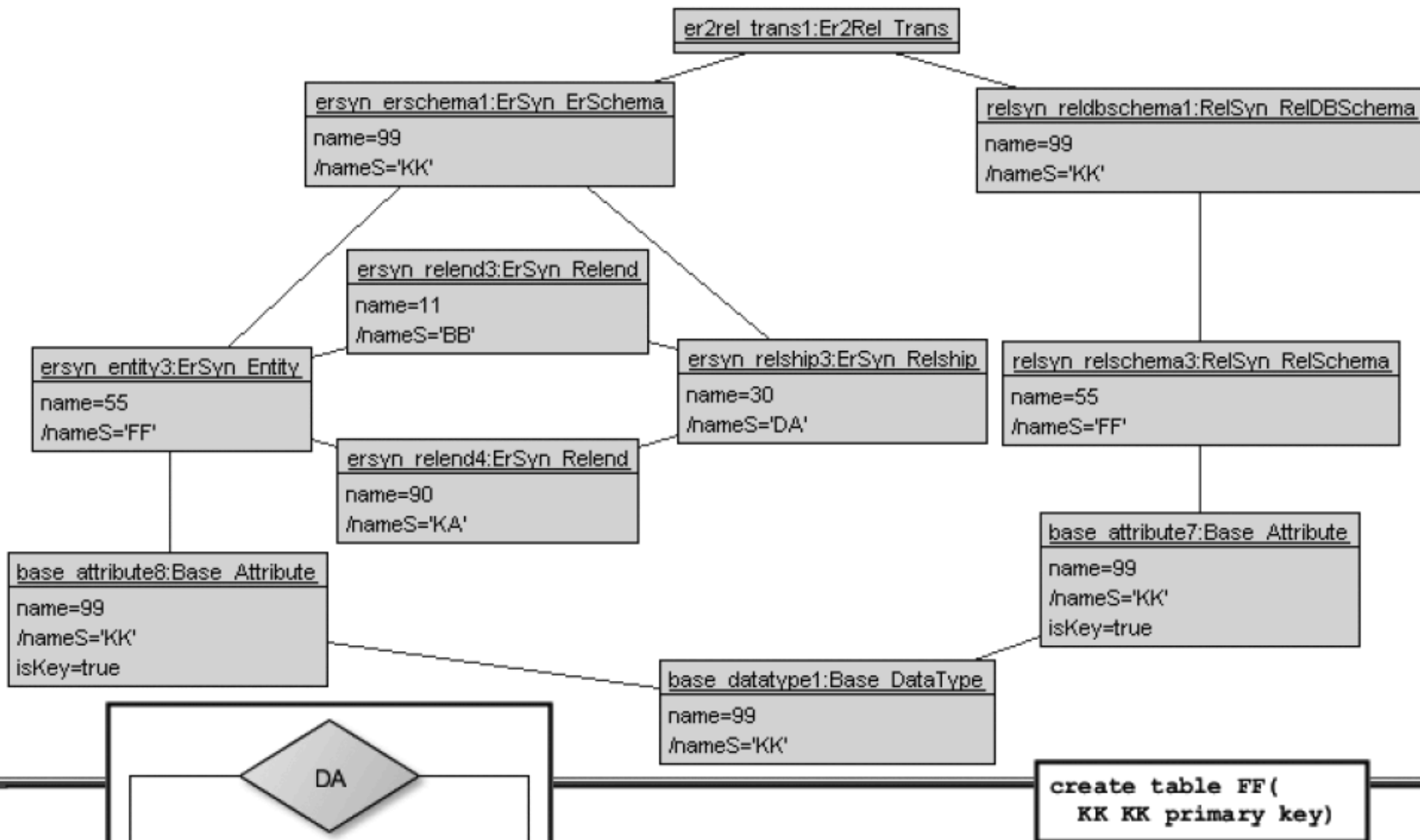
Class black-on-white
Association black-on-light-grey

Inv_B is independent of $\{Inv_A, Inv_C\}$

$$\begin{aligned} & \exists od: ObjectDiagram (Inv_A \wedge \neg Inv_B \wedge Inv_C) \\ \Leftrightarrow & \neg \forall od: ObjectDiagram (Inv_A \wedge Inv_C \Rightarrow Inv_B) \end{aligned}$$

modelvalidator -invIndep invariantIndependenceConfig.properties

```
15:18:24 Base_Attribute::linkedToOneOfEntityRelshpRelSchema: Independent
15:18:37 Base_DataType::uniqueDataTypeNames: Independent
15:18:48 Er2Rel_Trans::forEntityExistsOneRelSchema: Independent
15:19:17 Er2Rel_Trans::forRelSchemaExistsOneEntityXorRelshp: Independent
15:19:29 Er2Rel_Trans::forRelshpExistsOneRelSchema: Independent
15:19:40 ErSyn_Entity::differentOsRelendAndAttributeNamesWithinEntity: Independent
15:19:50 ErSyn_Entity::entityKeyNotEmpty: Independent
15:20:00 ErSyn_Entity::uniqueAttributeNamesWithinEntity: Independent
15:20:11 ErSyn_Entity::uniqueOsRelendNamesWithinEntity: Independent
15:20:21 ErSyn_ErSchema::differentEntityAndRelshpNamesWithinErSchema: Independent
15:20:34 ErSyn_ErSchema::uniqueEntityNamesWithinErSchema: Independent
15:20:45 ErSyn_ErSchema::uniqueErSchemaNames: Independent
15:20:55 ErSyn_ErSchema::uniqueRelshpNamesWithinErSchema: Independent
15:21:06 ErSyn_Relend::c_Relend_Entity_Relshp_ErSchema: Independent
15:21:19 ErSyn_Relshp::differentRelendAndAttributeNamesWithinRelshp: Independent
15:21:30 ErSyn_Relshp::relshpKeyEmpty: Independent
15:21:41 ErSyn_Relshp::uniqueAttributeNamesWithinRelshp: Independent
15:21:52 ErSyn_Relshp::uniqueRelendNamesWithinRelshp: Independent
15:22:05 RelSyn_RelDBSchema::uniqueRelDBSchemaNames: Independent
15:22:16 RelSyn_RelDBSchema::uniqueRelSchemaNamesWithinRelDBSchema: Independent
15:22:28 RelSyn_RelSchema::relSchemaKeyNotEmpty: Independent
15:22:39 RelSyn_RelSchema::uniqueAttributeNamesWithinRelSchema: Independent
```

Er2Rel_Trans::forRelSchemaExistsOneEntityXorRelship	true
Er2Rel_Trans::forRelshipExistsOneRelSchema	false
ErSyn_Entity::differentOsRelendAndAttributeNameWithinEntity	true

Conclusion

- presented a case study for automatically checking **invariant independence**
- model validator based on **relational model finder** Kodkod
- **relationship to Verification of model Transformation (VOLT)**:
verify a central transformation model property, i.e.
invariant independence, by constructing object diagrams

independence could also be viewed as **invariant set minimality**

Future work

- **handling of strings** has to be improved
- **'observation terms'** in the case that not only one solution,
but all solutions could be considered; achieve substantially
different solutions, i.e. object diagrams
- apply observation terms for **source** and **target** metamodels in
transformation models
- further **larger case studies** must check the practicability

Thanks for your attention!