Checking UML and OCL Model Consistency: An Experience Report on a Middle-Sized Case Study

Martin Gogolla, Lars Hamann, Frank Hilken, Matthias Sedlmeier
University of Bremen, Database Systems Group

Motivation and context

- consider 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)

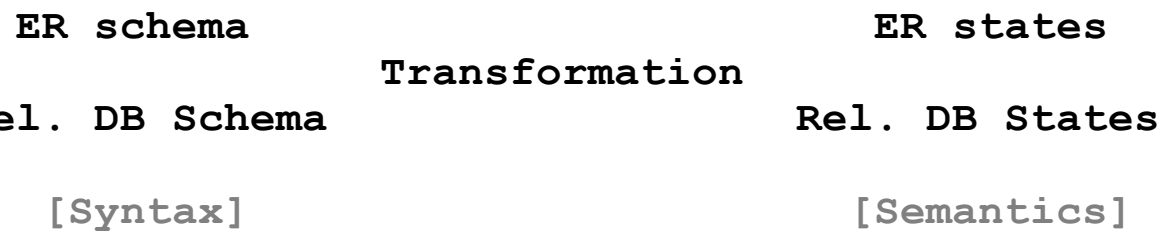
- USE gives support for object, statechart, sequence and communication diagrams and imperative operation implementation

- model validator on basis of Kodkod automatically constructs object diagrams for UML and OCL models

- prove model properties

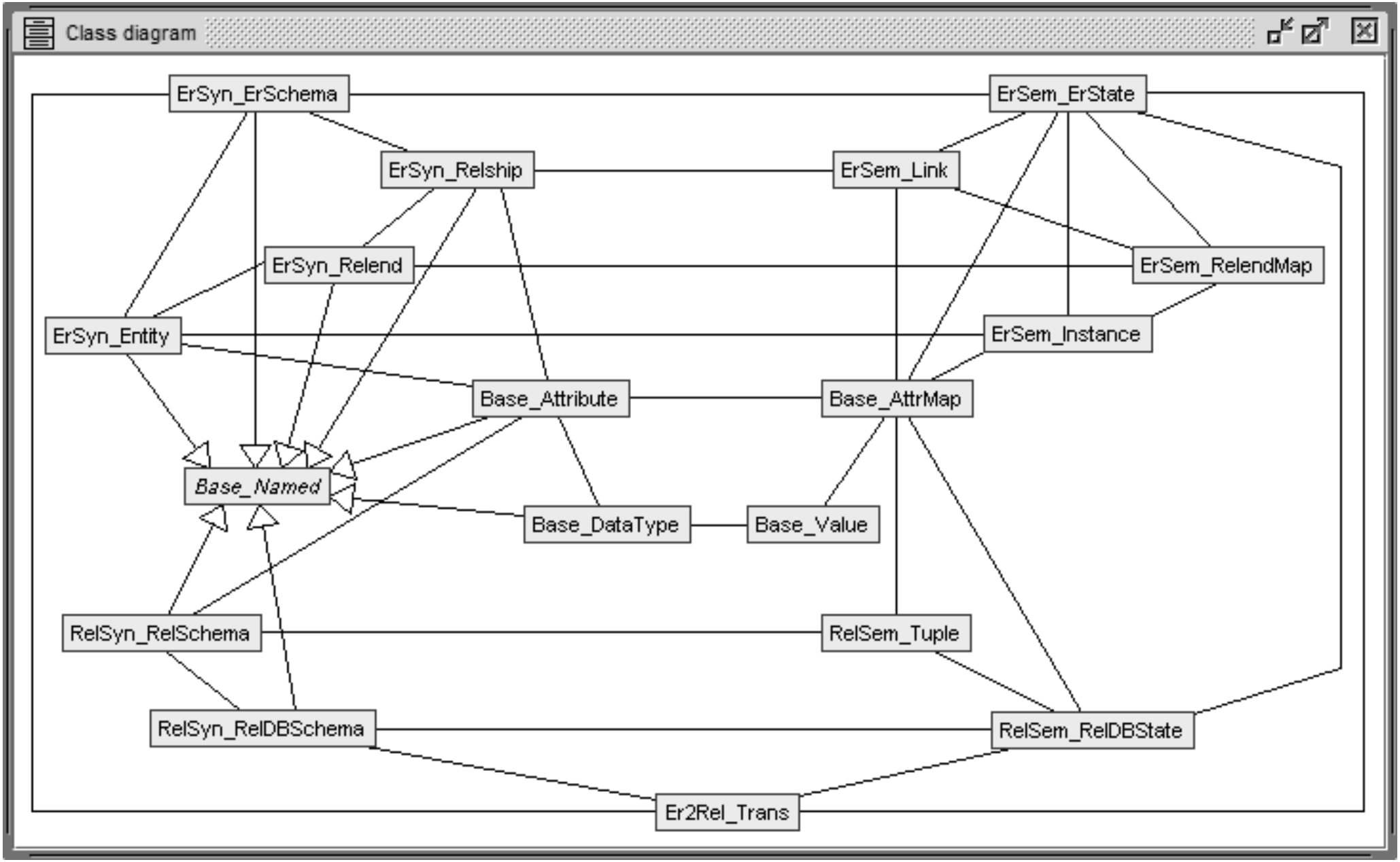
  - model consistency, i.e., automatically construct a valid model instance

- consider example model representing the transformation between the Entity-Relationship (ER) and the relational data model; consider schemata and states for both data models



ER Schema

ER State



Rel. DB Schema

Transformation

Rel. DB State

Case study class diagram and invariants

- Class diagram

  - 18 classes
  - 34 associations
  - 10 OCL helper operations

- 59 Invariants

  - within one table, two distinct attributes have distinct names
  - every table must have at least one key attribute

  - all tuples in relational DB state have unique key attribute values

  - every entity is transformed into a table
  - every relationship is transformed into a table
  - every tuple (in a relational DB state) is transformed either into
    - an instance (typed by an entity) or
    - a link (typed by a relationship)

  ...

Example invariant

```
context self:Er2Rel_Trans inv forTupleExistsOneInstanceXorLink:

  self.relDBState->forAll(relSt | self.erState->one(erSt |
    relSt.tuple->forAll(t |

      erSt.instance->one(i |
        t.attrMap->forAll(amRel | i.attrMap->one(amEr |
          amEr.attribute.name=amRel.attribute.name and
          amEr.value=amRel.value)))

      xor

      erSt.link->one(l | t.attrMap->forAll(amRel |
        ( amRel.attribute.isKey=false implies
          l.attrMap->one(amEr |
            amEr.attribute.name=amRel.attribute.name and
            amEr.value=amRel.value) )
        and
        ( amRel.attribute.isKey=true implies
          l.relendMap->one(rm |
            rm.instance.attrMap->
              select(amEr | amEr.attribute.isKey)->one(amEr |
                amRel.attribute.name =
                plus(times10(rm.relend.name),amEr.attribute.name) and
                amRel.value=amEr.value)))))))
```

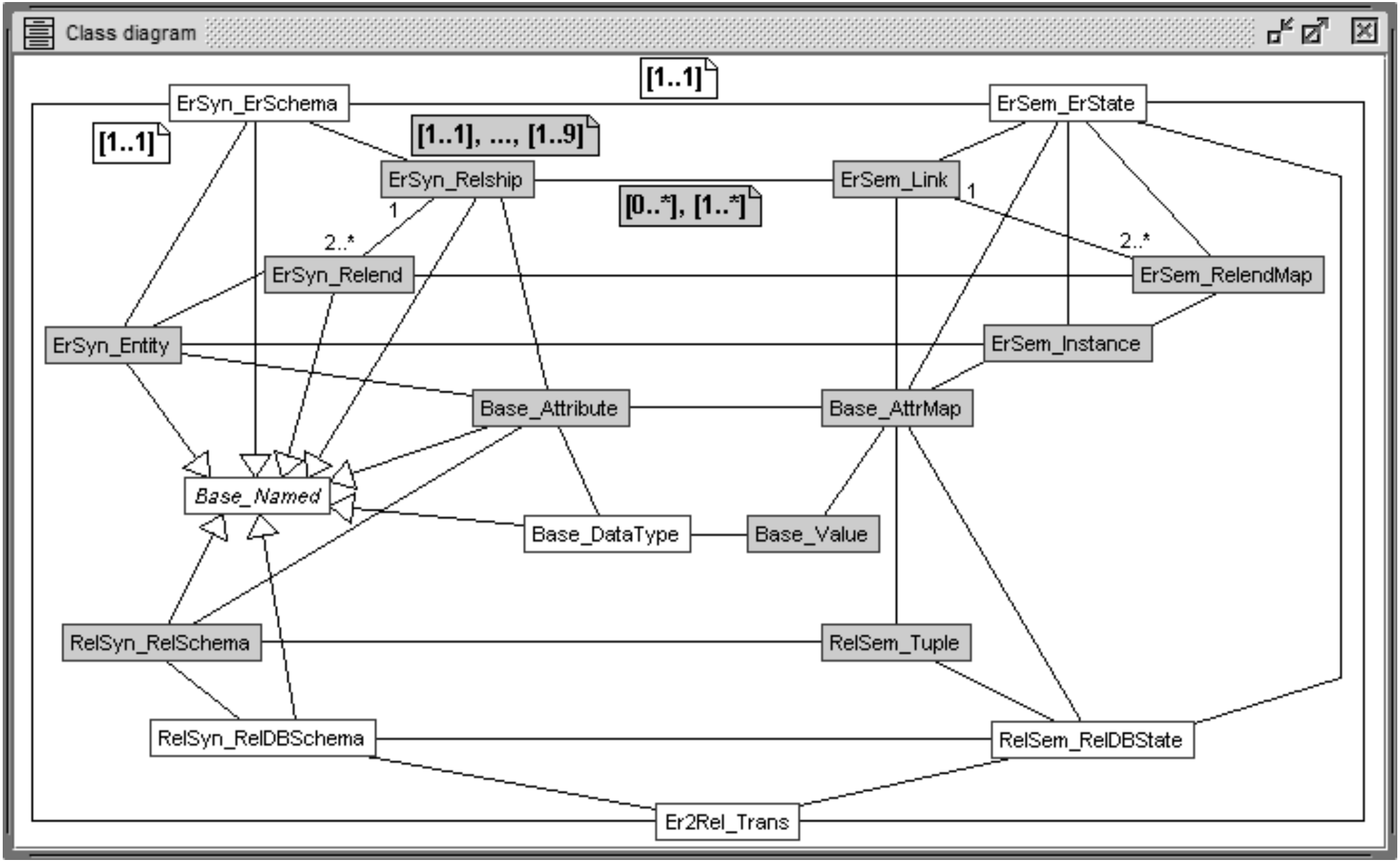
USE model validator and configuration

- USE Model validator
  - automatically construct object diagram
  - based on translation of UML and OCL into relational logic and implemented in form of Alloy and Kodkod
  - model validator uses Kodkod
  - translate found results back into UML

- Configuration
  - guarantees: models elements (classes, attributes, associations, datatypes) are populated with finite sets

- Building an object diagram shows consistency of invariants

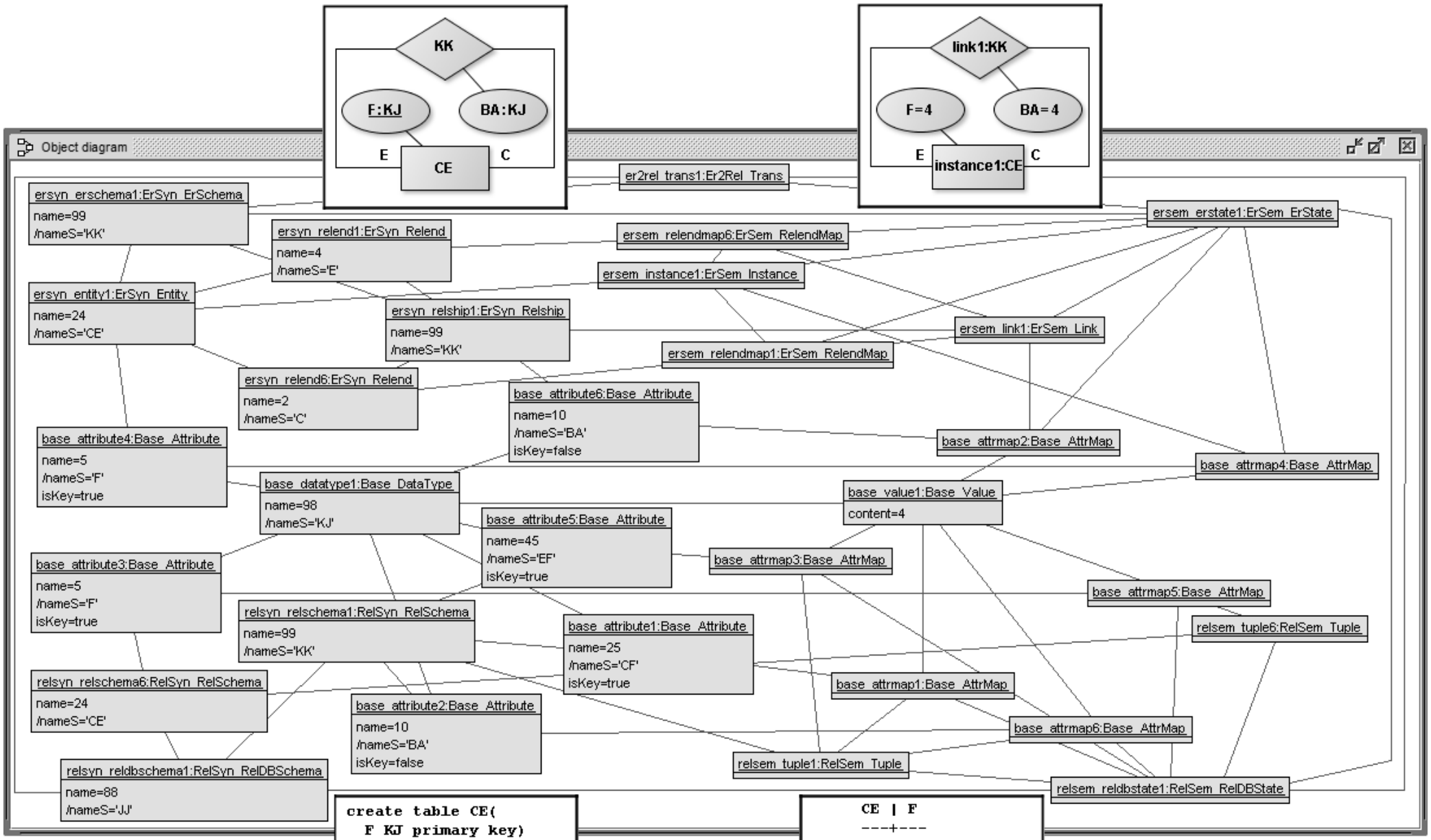
18 configurations: grey classes 1..9 objects, assocs 0..* or 1..*

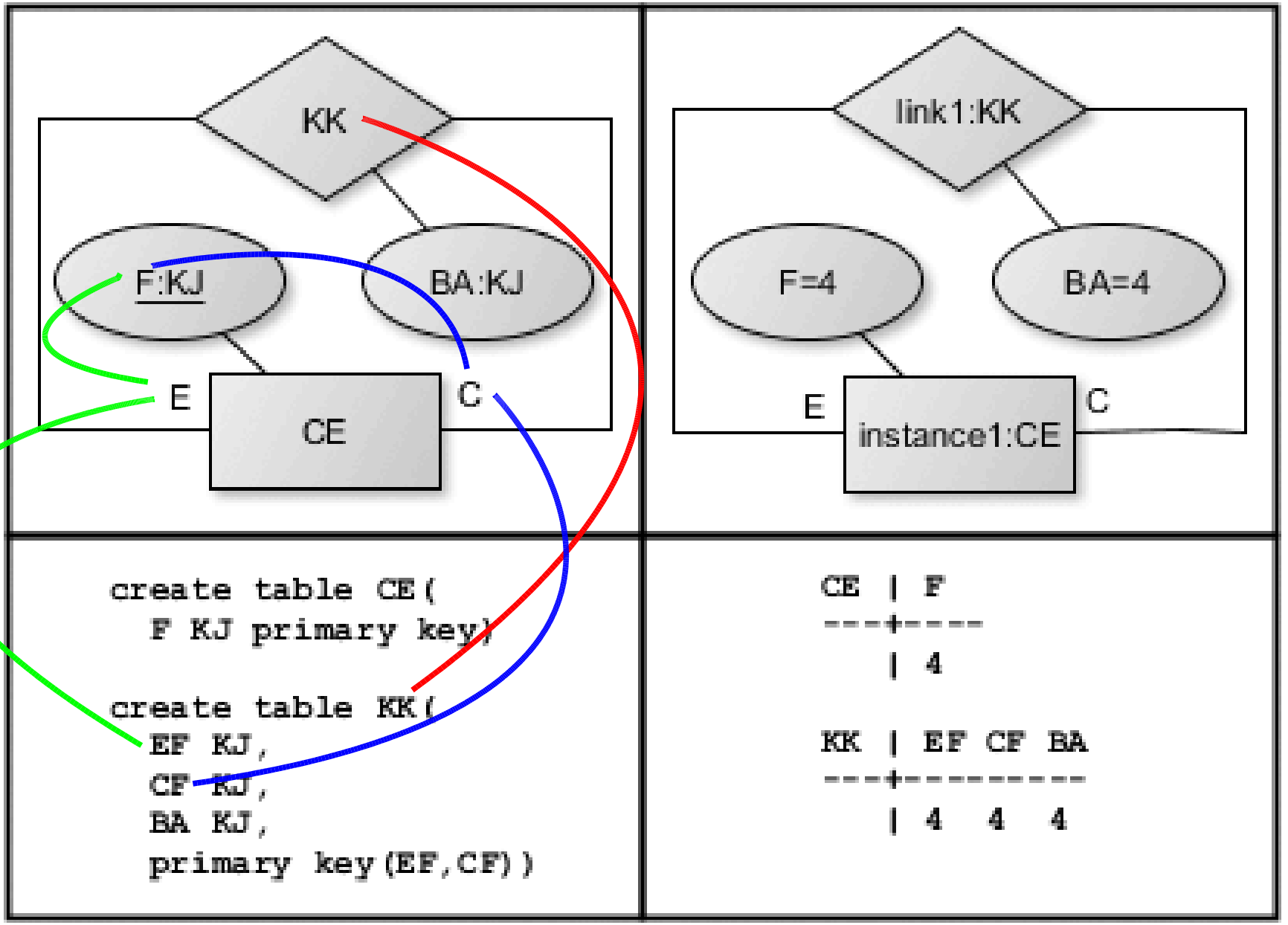


Num Objects	Num Links	USE Response	Times [in milliseconds]		
			Translation	Translation	Solving
1..1	0..*	trivially unsat	358 ms	202 ms	0 ms
1..2	0..*	unsat	328 ms	811 ms	31 ms
1..3	0..*	unsat	359 ms	3292 ms	827 ms
1..4	0..*	sat	359 ms	11092 ms	8205 ms
1..5	0..*	sat	327 ms	31231 ms	45022 ms
1..6	0..*	sat	328 ms	73445 ms	8533 ms
1..7	0..*	sat	327 ms	158839 ms	231053 ms
1..8	0..*	sat	343 ms	301907 ms	149480 ms
1..9	0..*	sat	343 ms	557427 ms	459233 ms
1..1	1..*	trivially unsat	312 ms	203 ms	0 ms
1..2	1..*	unsat	328 ms	827 ms	16 ms
1..3	1..*	unsat	343 ms	3338 ms	78 ms
1..4	1..*	unsat	340 ms	10951 ms	219 ms
1..5	1..*	unsat	343 ms	30857 ms	3572 ms
1..6	1..*	sat	375 ms	74412 ms	134878 ms
1..7	1..*	sat	343 ms	157264 ms	17628 ms
1..8	1..*	sat	394 ms	301315 ms	120432 ms
1..9	1..*	sat	375 ms	551758 ms	607059 ms

Fig. 4. Applied 18 model validator configurations and USE results.

All classes and assocs instantiated; construction time: ca. 200.000 ms ≈ 3.5 mins






```
create table CE(
  F KJ primary key)

create table KK(
  EF KJ,
  CF KJ,
  BA KJ,
  primary key(EF,CF))
```

```
CE  | F
----+---
    | 4

KK  | EF CF BA
----+---------
    | 4  4  4
```

Conclusion

- presented a case study for automatically checking model properties
  - instantiated a class diagram with 18 classes, 34 assocs, 59 invs by an object diagram with 32 objects and 67 links in 3.5 mins
  - consistency, i.e., class instantiability, class and association instantiability
  - approach can also check for implied model properties
- model validator based on relational model finder Kodkod
- relationship to Tests And Proofs (TAP): build a test case (object diagram) and by this prove that a property (consistency) holds

Future work

- handling of strings has to be improved
- incorporation of model behavior: filmstripping
- `observation terms´ in the case that not only one solution, but all solutions should be considered; achieve substantially different solutions, i.e. object diagrams
- show invariant independence for the example transformation model
- further larger case studies must check the practicability

Thanks for your attention!