Model Transformations? Transformation models!

Jean Bézivin, Fabian Buttner, Martin Gogolla, Frederic Joualt, Ivan Kurtev,

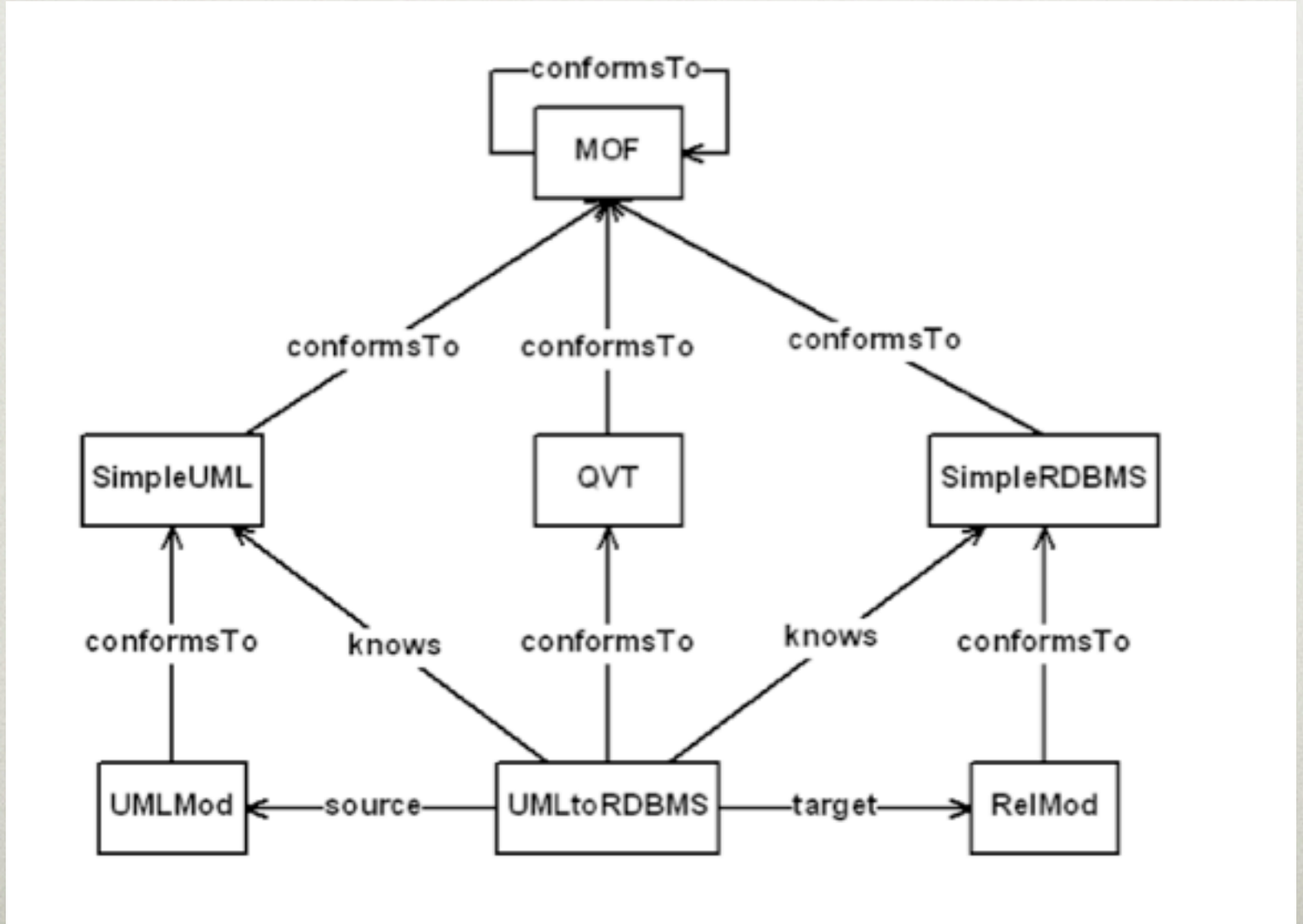
Presented by: Rafael Olaechea

Why use Transformation Models?

- Focus on the properties of transformations instead of an operational description of them.
  - Undirected.
  - Higher Order Transformations.
  - Validations and Completions.

Operational view of Model Transformations

Query View Transformation (QVT) Standard



Operational View of Model Transformations

```
map packageToSchema in umlRdbms {

        uml () {
            p:Package

        }

        rdbms () {
            s:Schema
        }
        where () {

            p2s:PackageToSchema|
            p2s.umlPackage = p;
            p2s.schema = s;
        }
        map {
            where () {
                    p2s.name := p.name;
                    p2s.name := s.name;
                    p.name := p2s.name;
                    s.name := p2s.name;
            }

        }

}
```

Operational View of Model Transformations

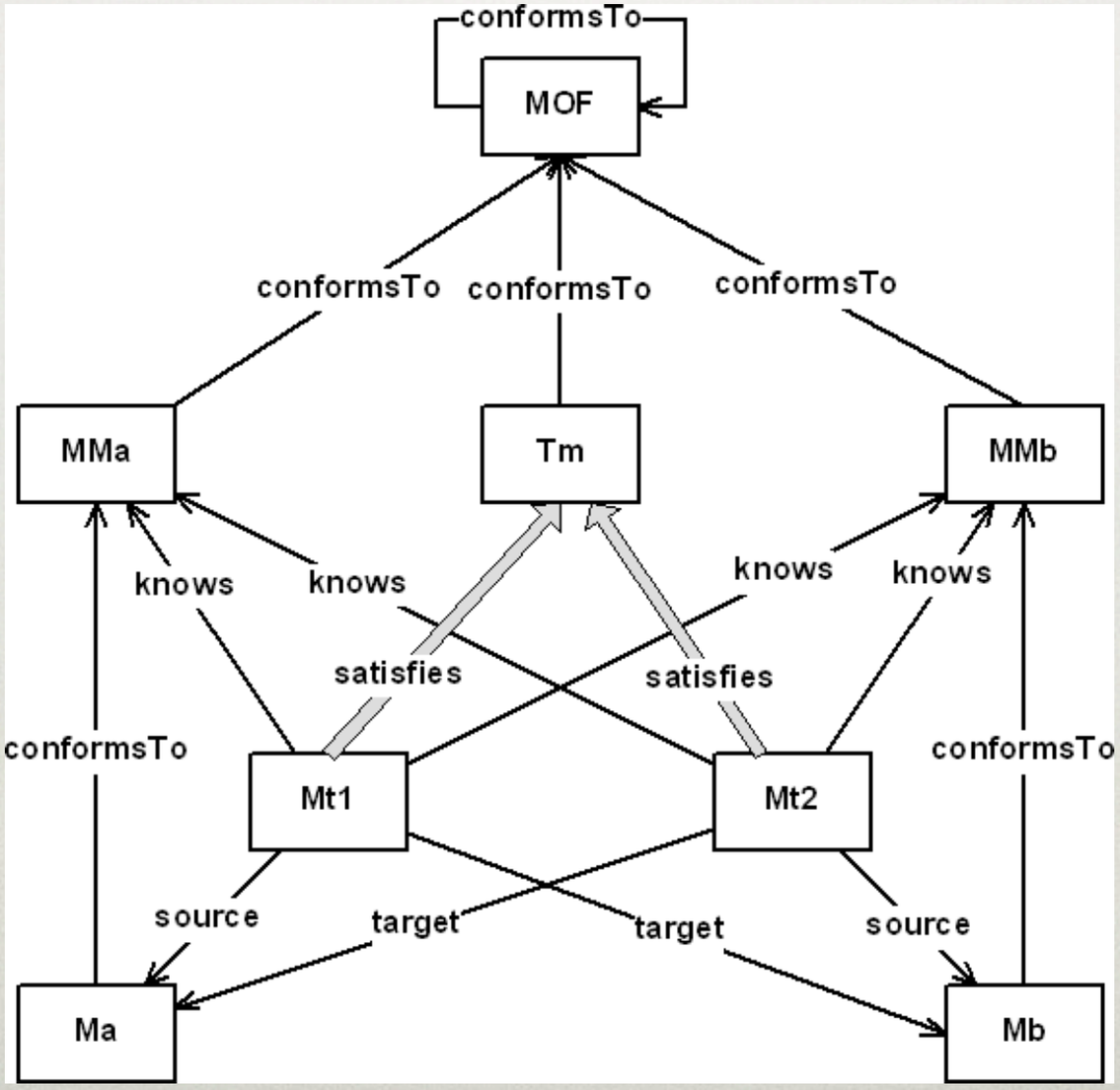
```
 class PrimitiveToName {
      owner : PackageToSchema opposites primitivesToNames;
      name : String;
      -- uml
      primitive : PrimitiveDataType;
      --rdbms
      typename: String;
}

map primitiveToName in umlRdbms {
    uml (p:Package) {
        prim:PrimitiveDataType|
           prim.owner = p;
    }
      check enforce rdbms () {
           sqlType:String
      }
      where (p2s:PackageToSchema| p2s.umlPackage=p) {
           realize p2n:PrimitiveToName|
           p2n.owner := p2s;
           p2n.primitive := prim;
           p2n.typeName := sqlType;
     }
     map {
          where () {
            p2name := prim.name + '2' + sqlType
          }
     }
```

```
map integerToNumber in umlRdbms
refines primitiveToName {
    uml () {
        prim.name = 'Integer';
    }
    check enforce rdbms () {
        sqlType := 'NUMBER';
    }

}
```

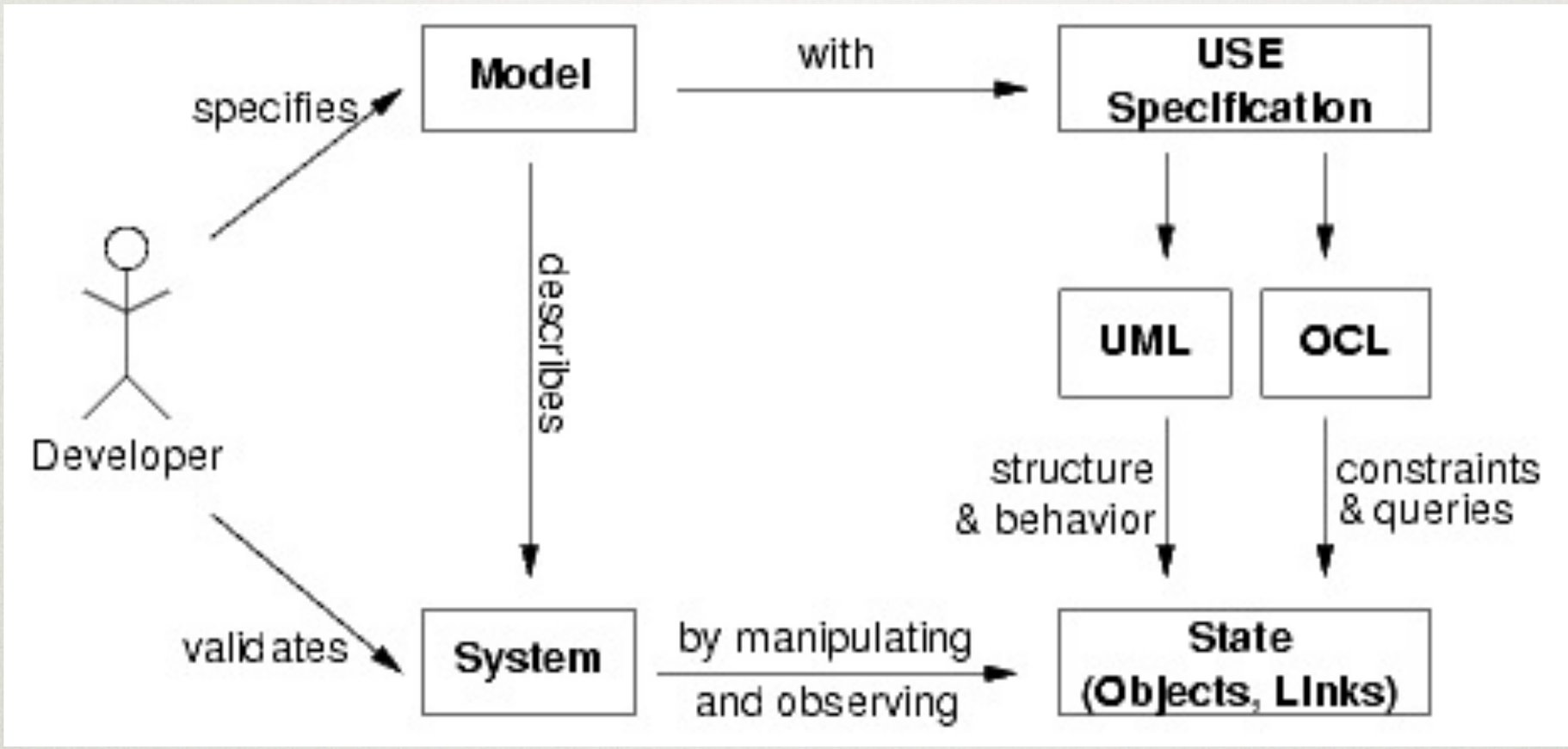
Modelling View of Model Transformations



What are transformation models?

- Transformation models express transformations as a MOF model relating source and target elements , including OCL constraints.

UML-based Specification Environment (USE)



Transformation model of Entity Relationship into a Relational model , is specified and validated using USE.

Example: ER to Rel Transformation Model

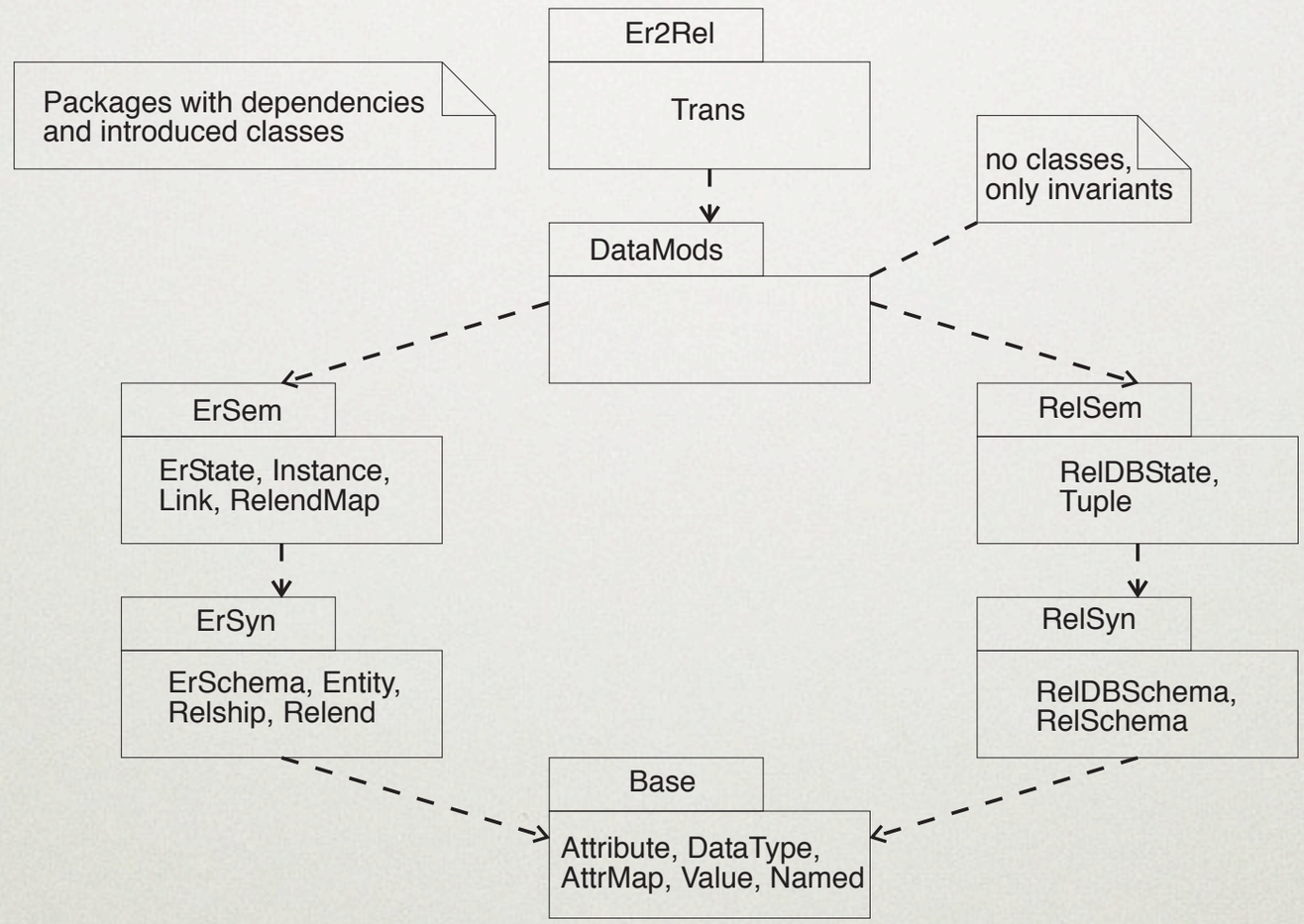
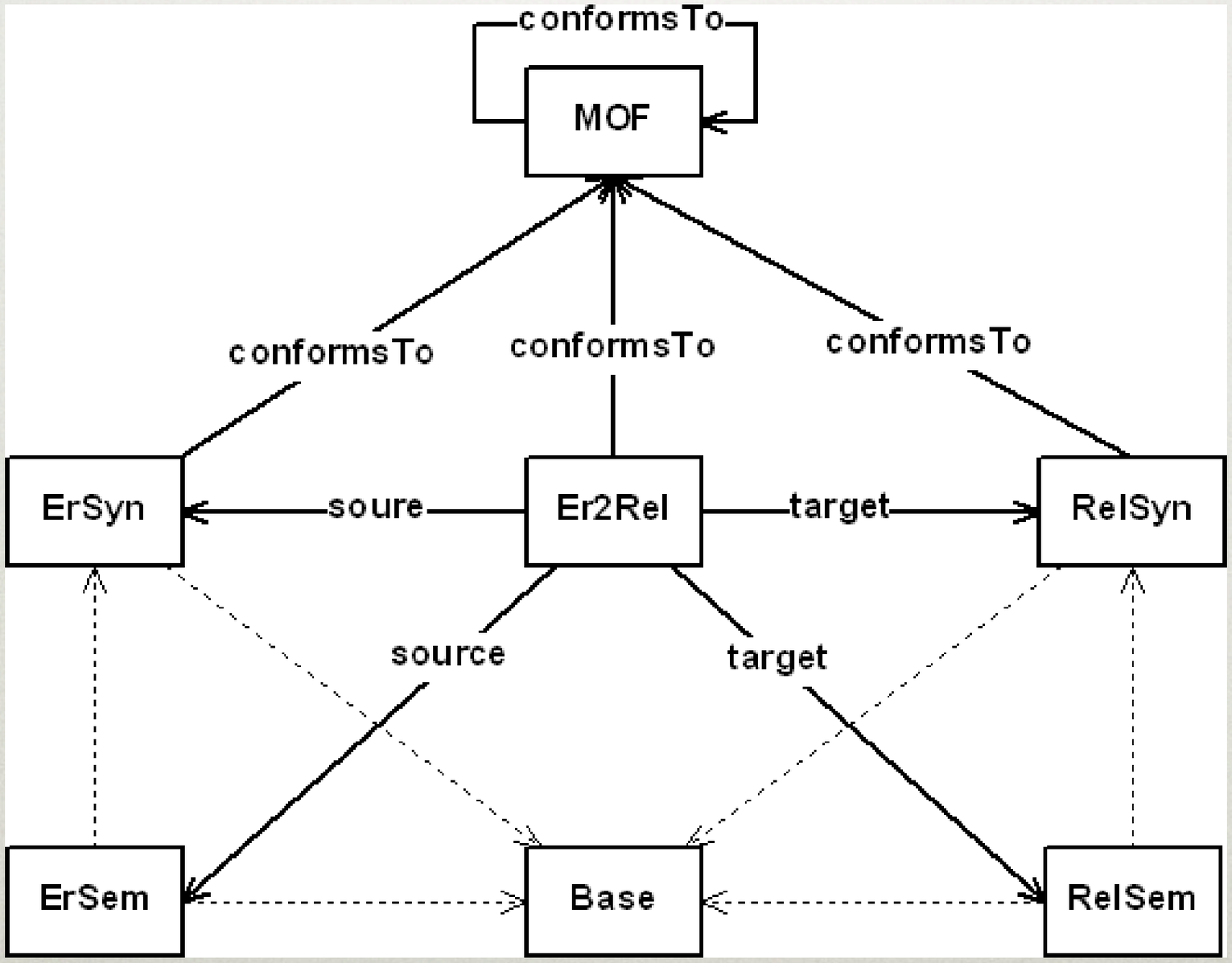
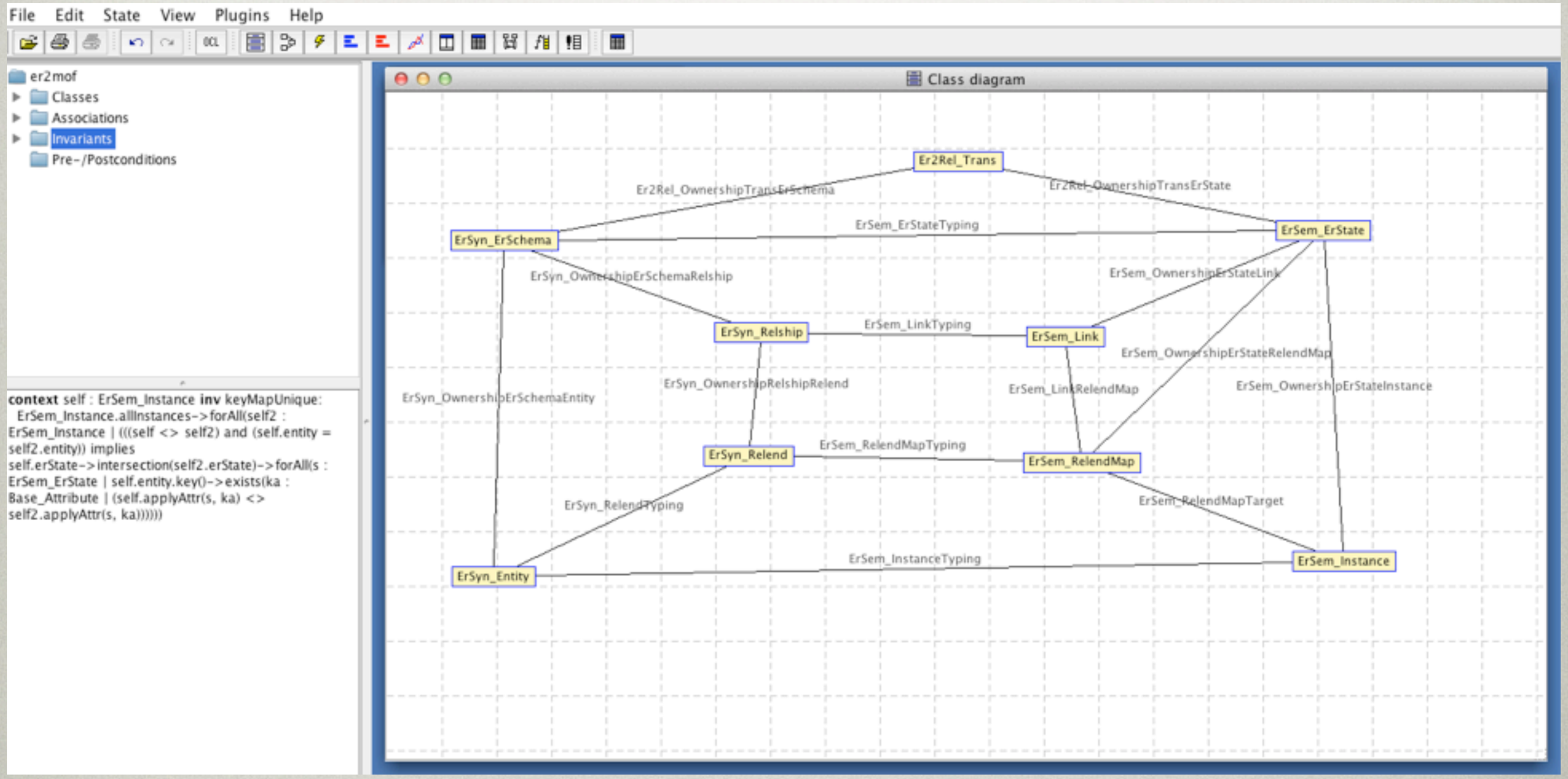


Figure 2.1: Package Diagram with Class Names

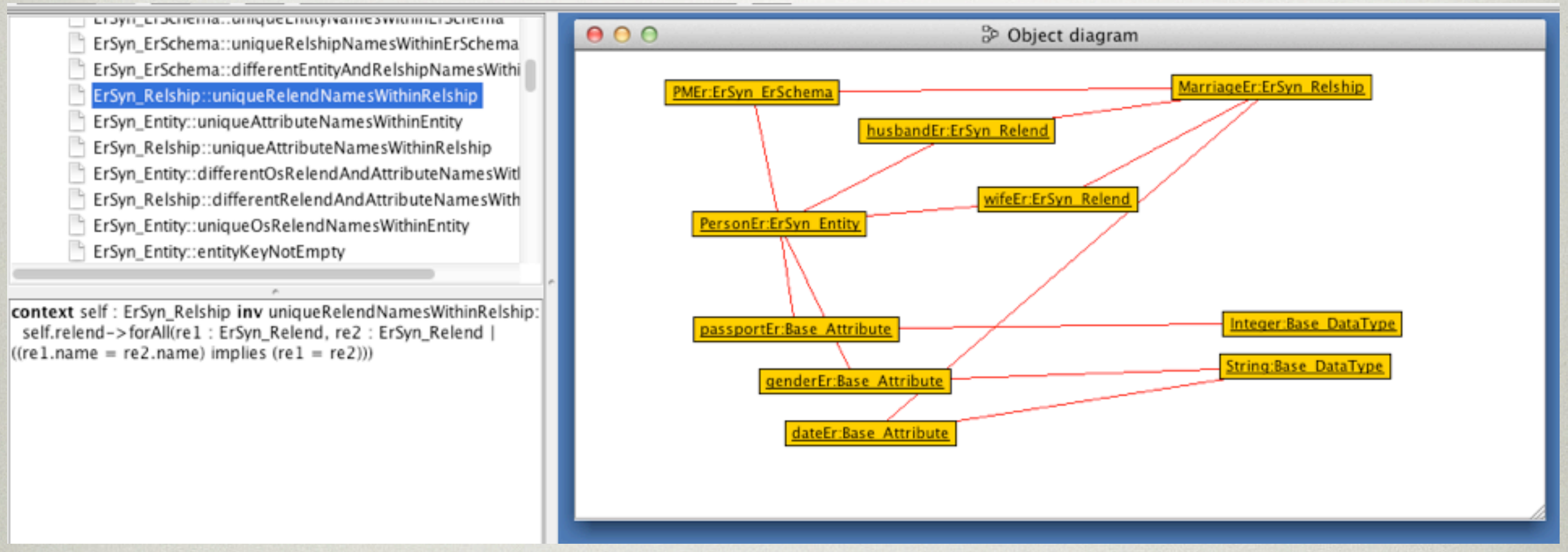
Example: ER to Rel Transformation Model



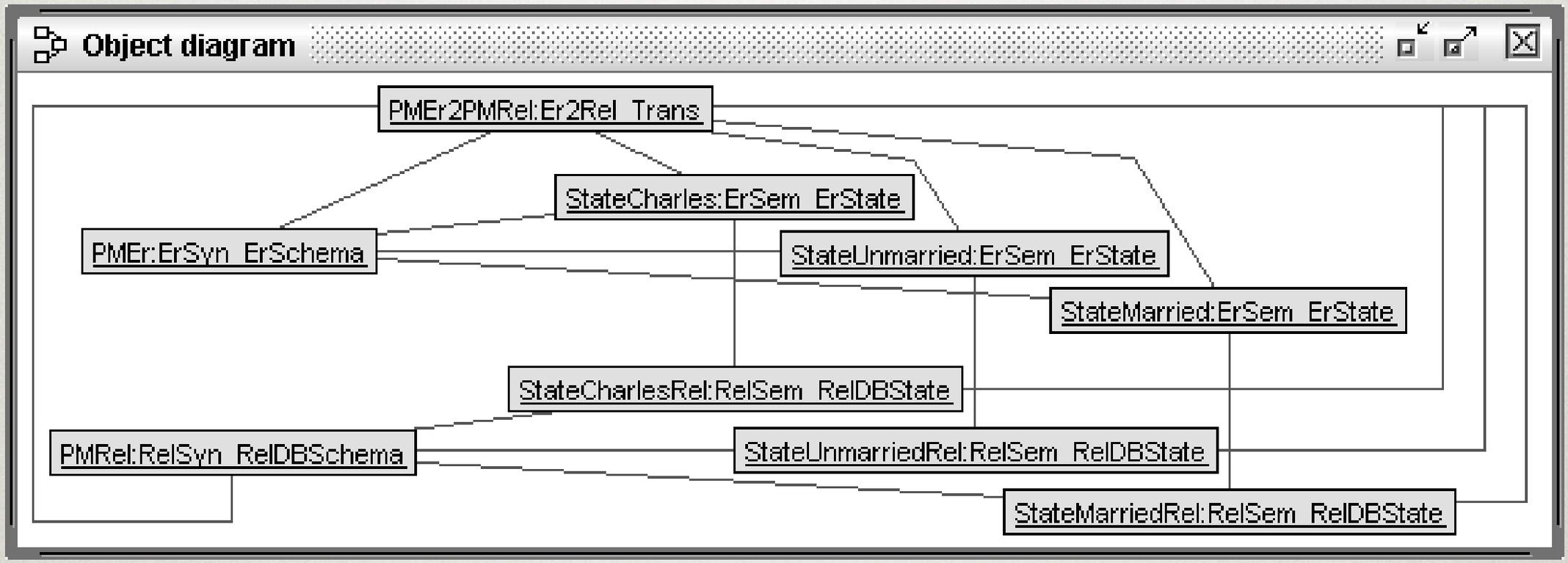
Syntax and Semantics Metamodels for the Er Model



Syntax for the Er model

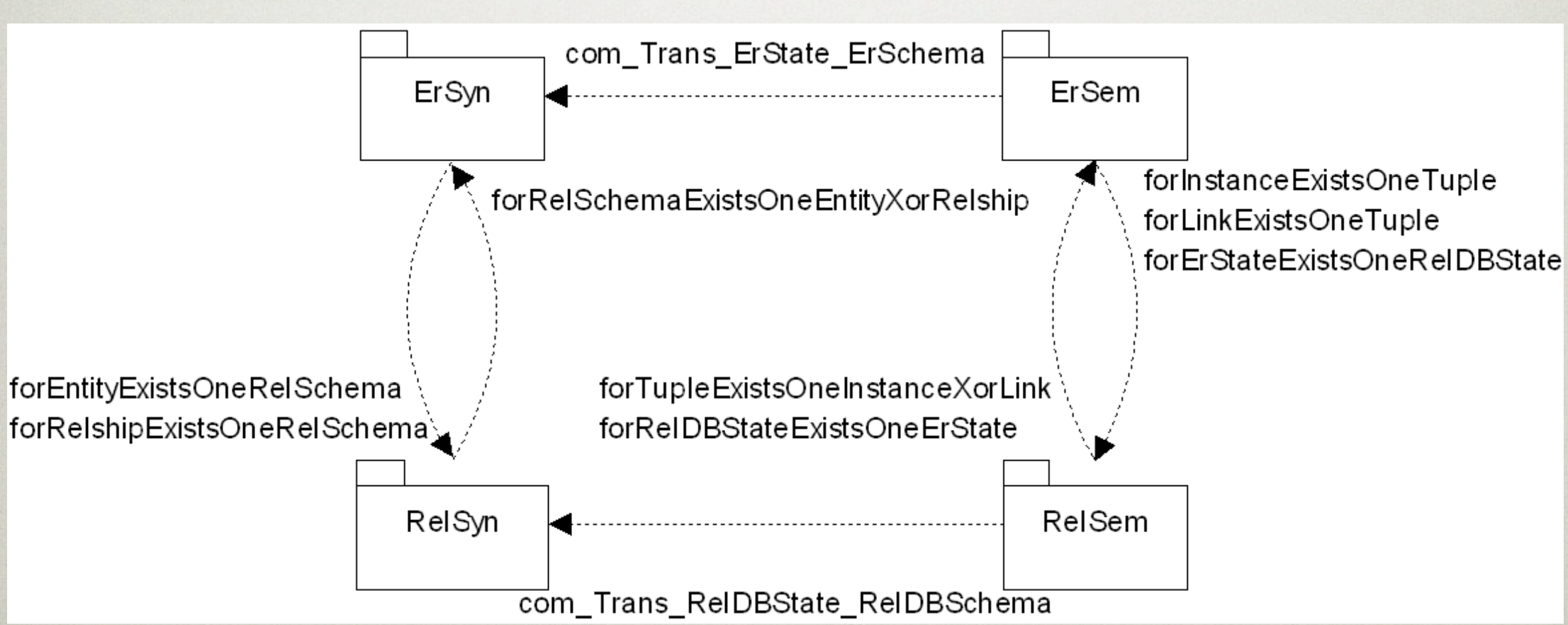


Transformation Model Object Diagram



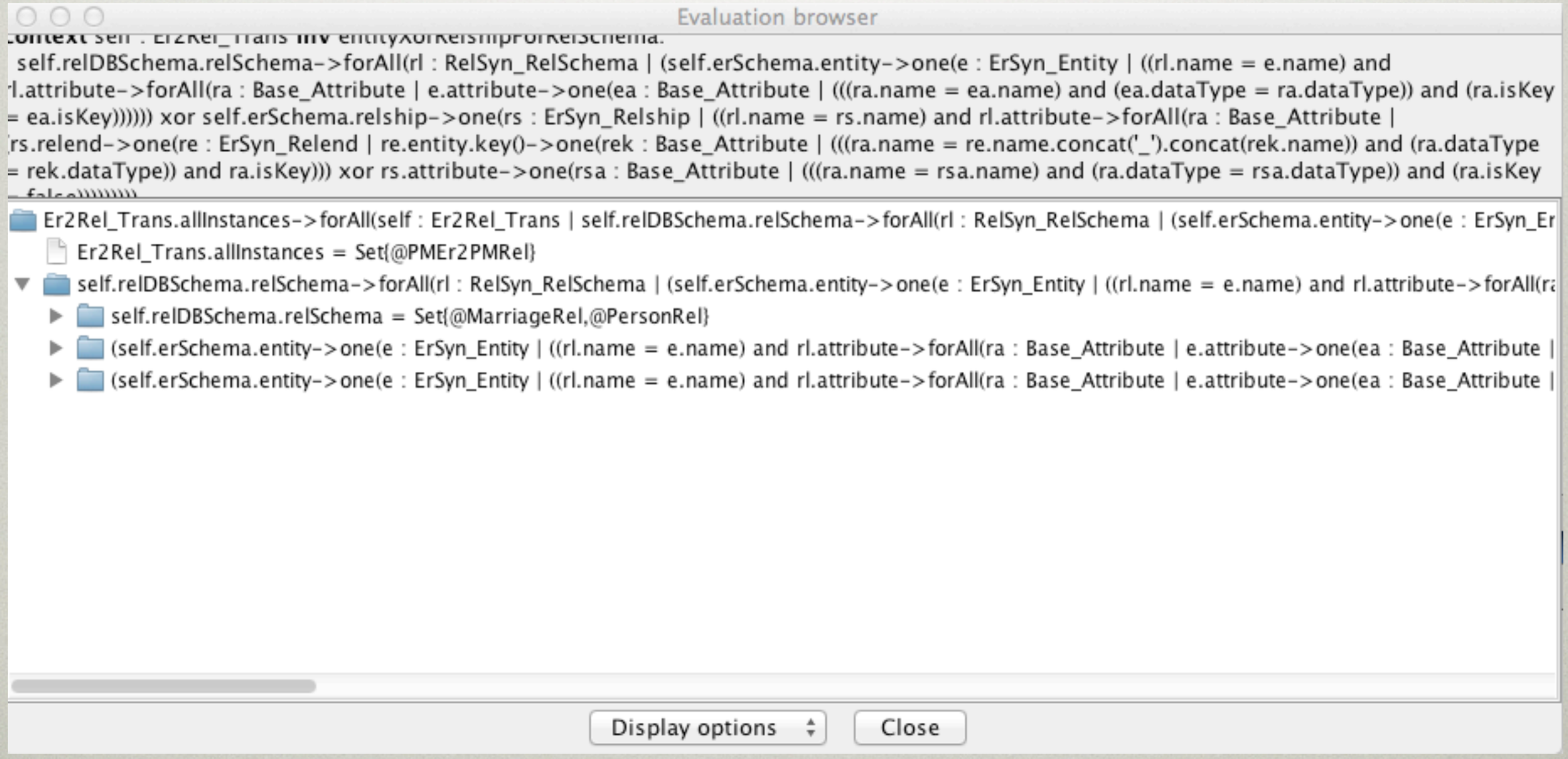
Constraints Captured in the ER2REL Transformation Model

- Er2Rel captures 10 constraints:



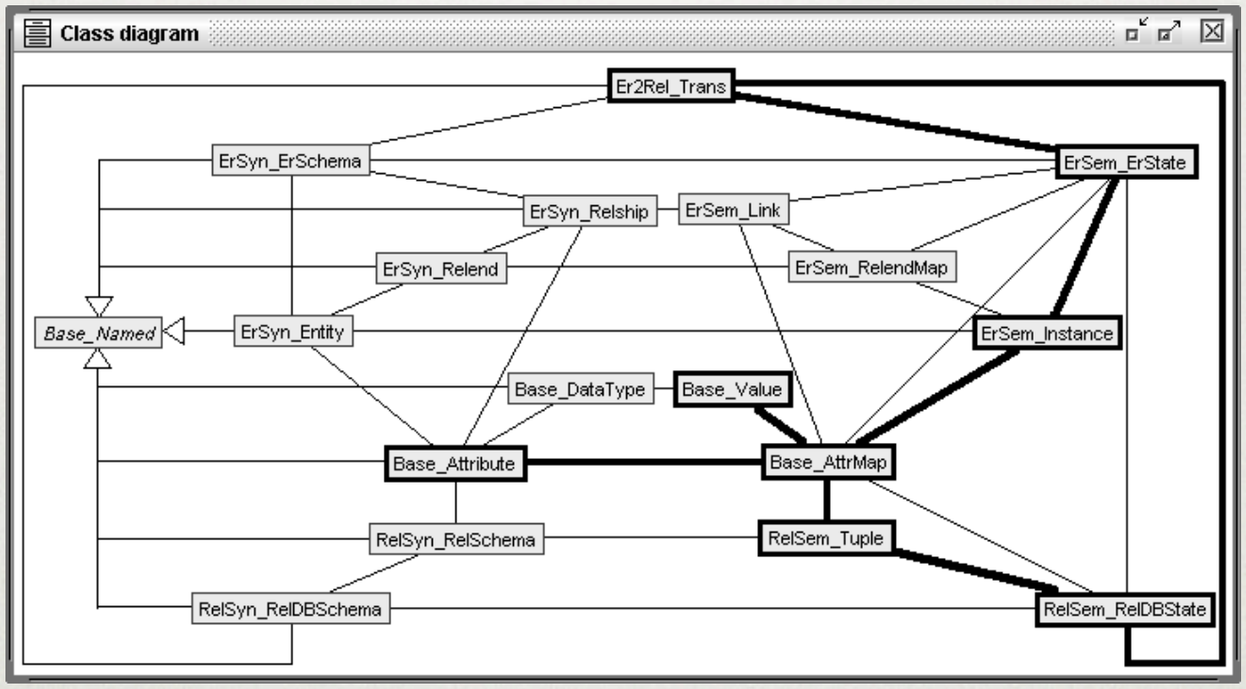
Constraint I

- forRelSchemaExistsOneEntityXorRelship



Constraint II

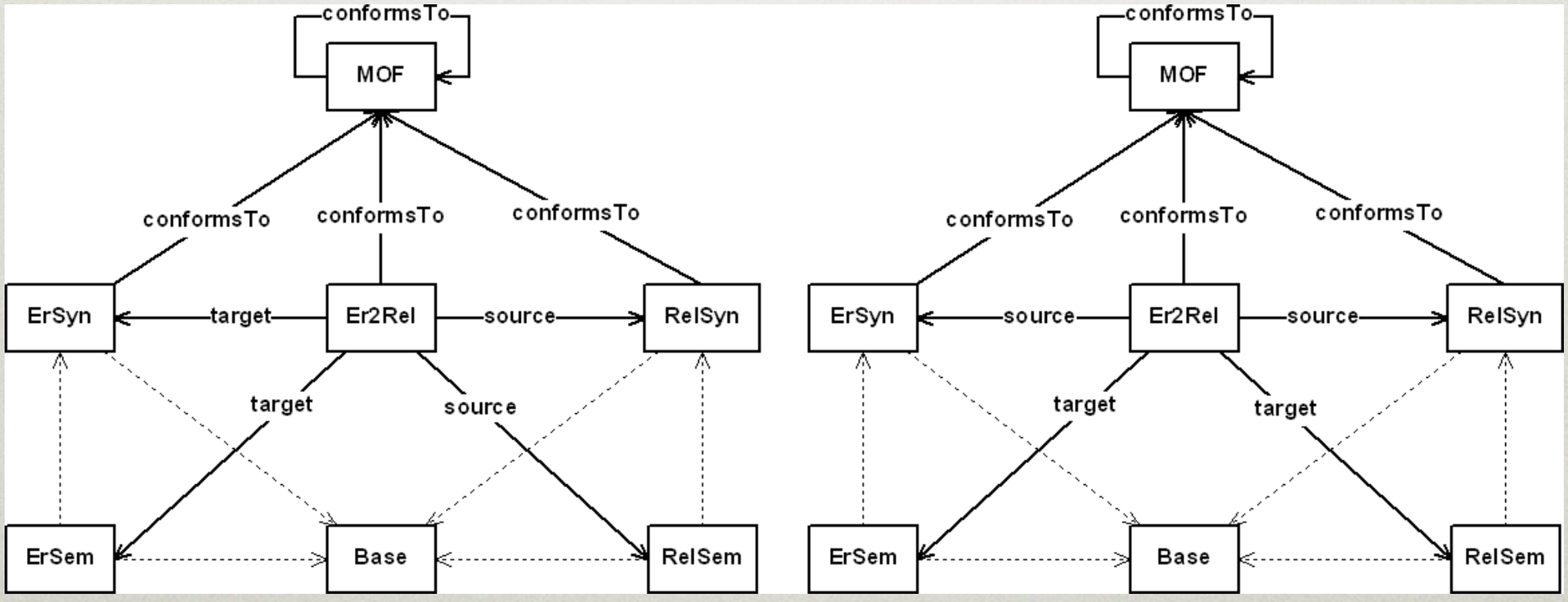
- forInstanceExistsOneTuple




```
context self:Er2Rel_Trans inv forInstanceExistsOneTuple:
  self.erState->forAll(erSt | self.relDBState->one(relSt |
    erSt.instance->forAll(i | relSt.tuple->one(t |
      i.attrMap->forAll(amEr |
        t.attrMap->one(amRel |
          amEr.attribute.name=amRel.attribute.name and
          amEr.value=amRel.value))))))
```

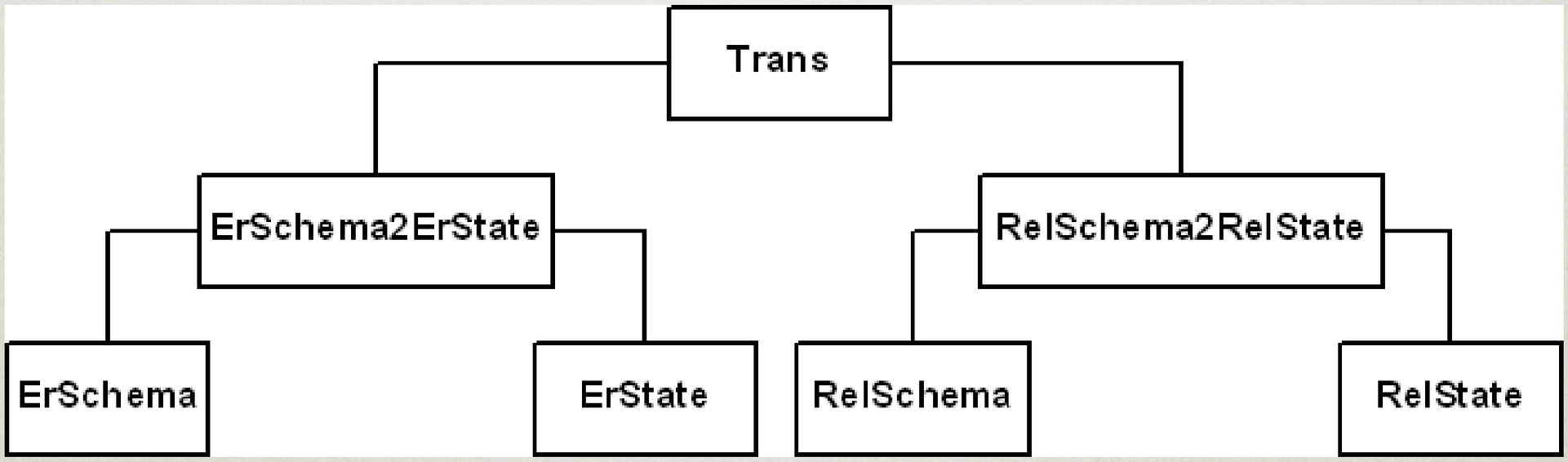
Advantages of Transformation Models

- Direction Neutral



Advantages of Transformation Models

- Higher Order Transformations



Advantages of Transformation Models

- Uniformity of Development
  - Avoiding QVT and using only UML and OCL.
- Transformations of Transformations
  - E.g Refactorings

Advantages of Transformation Models

- Validations and Completions.
  - "Analysis of Model Transformations via Alloy".

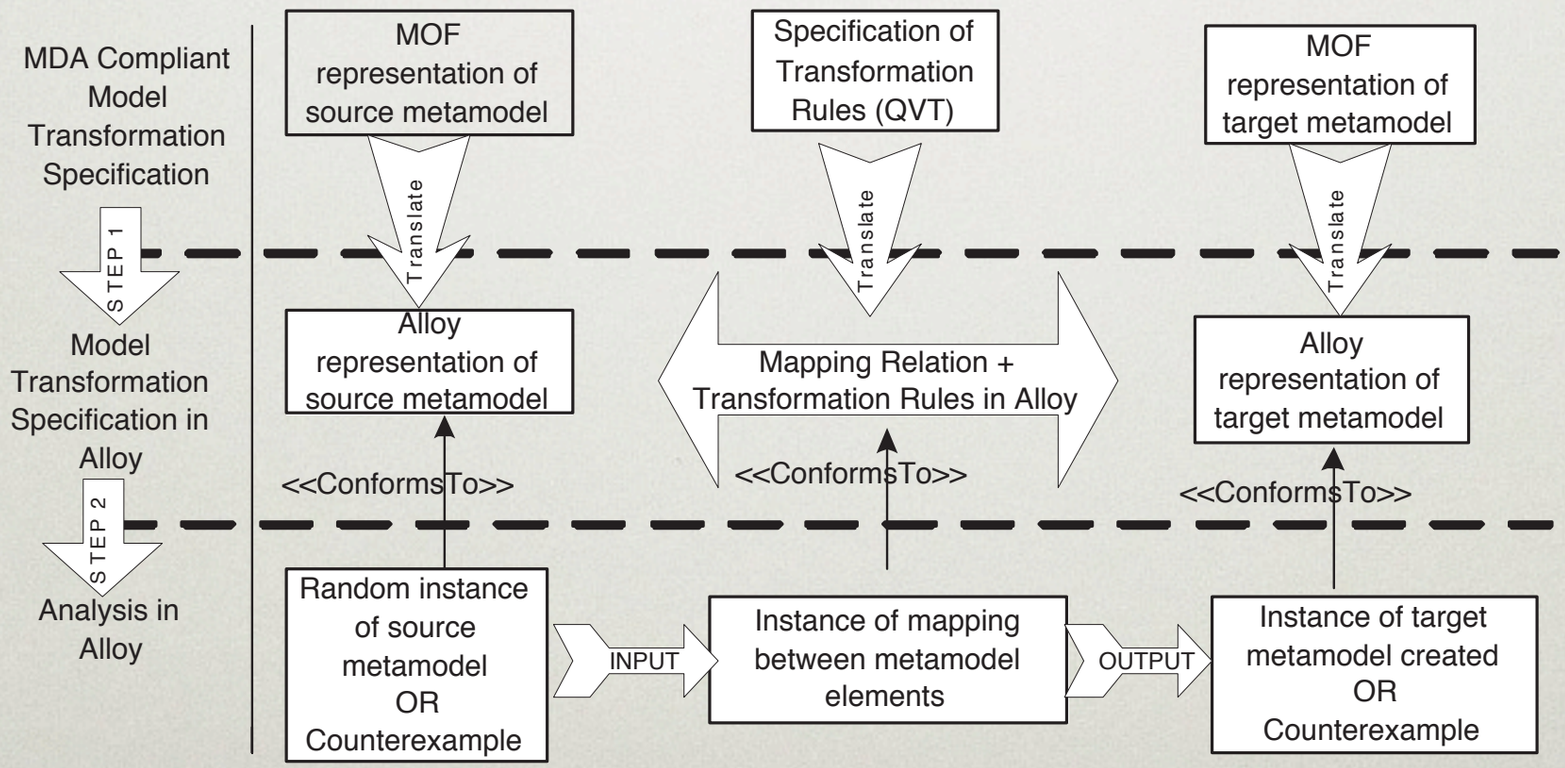


Fig. 1. An outline of our approach

Advantages of Transformation Models

- Complete Language Description.
  - Syntax and Semantics part.

Discussion Questions

- Why would you prefer a transformation model instead of standard QVT to describe transformation?
- Would it be possible to automatically complete transformations defined by the transformation model?
- What are the advantage of partitioning the models into semantic and syntax part?