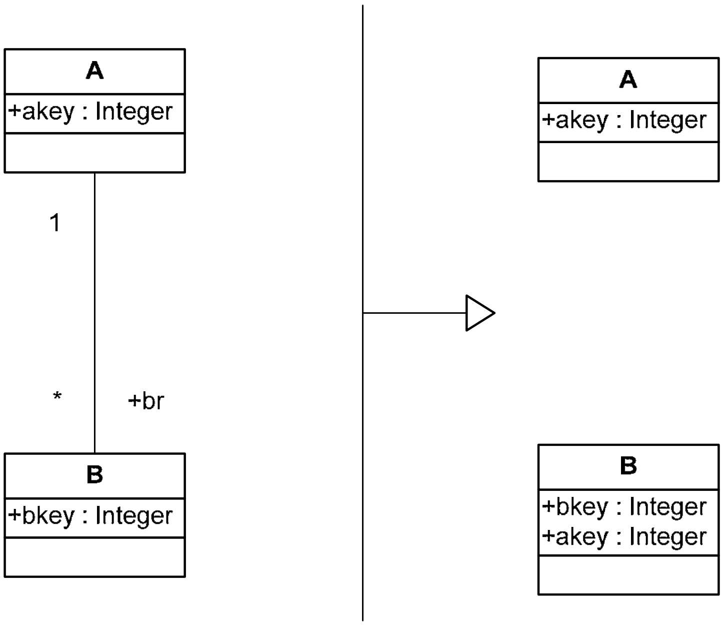
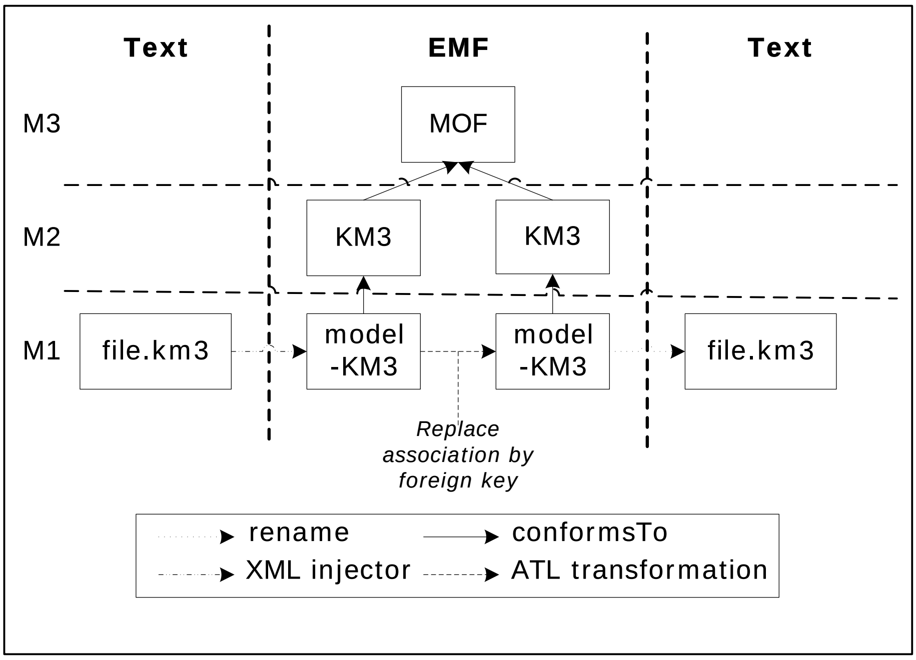
INRIA	ATL Transformation Catalogue of Model Transformations	Author Baudry Julien Jul.baudry <*at*> gmail.com
	Documentation	Aug 7th 2006

1. ATL Transformation Example: replace association by foreign key

This example is extract from Catalogue of Model Transformations by K. Lano.
Section 1.6: replace association by foreign key, page 7.





2. ATL Transformation overview

2.1. Description

This transformation applies to any explicit many-one association between persistent classes. It assumes that primary keys already exist for the classes linked by the association. It replaces the association by embedding values of the key of the entity at the one end of the association into the entity at the many ends.

2.2. Purpose

This is an essential step for implementation of a data model in relational database.

2.3. Rules specification

Our transformation has the same source and the target metamodel, KM3. We use 2 different names (KM3 and KM3target), but they refer to the same metamodel.

- For a Metamodel element, another Metamodel element is created :
  - with the same name and location,
  - Linked to the same contents.

- For a Package element, another Package element is created :
  - with the same name,
  - Linked to the same contents.

- For a Class element, another Class is created :
  - with the same name,
  - Abstract if the source class is abstract,
  - Linked the same structural feature.

- For a DataType element, another DataType is created :
  - With the same name and location.

- For a Reference element, a Attribute element is created :
  - with the name '<name of the class> + ForeignKey', isOrdered <- false,
  - isUnique <- false,
  - location is empty,
  - lower <- 1,
  - upper <- 1,
  - type <- Integer (defined in the package Primitive Type).

- For a Attribute element, another Attribute is created
  - with the same name, location, and type,
  - With the same properties isUnique,
  - With the same values upper and lower.

2.4. ATL Code

```
-- @name    Replacing association by foreign key
-- @version    1.0
-- @domains    Catalogue of Model Transformations
-- @authors    Baudry Julien (jul.baudry<at>gmail.com)
-- @date    2006/08/02
-- @description   The purpose of this transformation is to replace an association by a foreign
key
-- @see http://www.dcs.kcl.ac.uk/staff/kcl/tcat.pdf
-- @see section 1.6, page 7
-- @see author of article : K. Lano


module ForeignKey; -- Module Template
create OUT : KM3target from IN : KM3;

--@begin rule Metamodel
rule Metamodel {
   from
      inputMm:KM3!Metamodel
   to
      outputMm:KM3target!Metamodel (
         location <- inputMm.location,
         contents <- inputMm.contents
      )
}
--@end rule Metamodel

--@begin rule Package
rule Package {
   from
      inputPkg:KM3!Package
   to
      outputPkg:KM3target!Package (
         name <- inputPkg.name,
         contents <- inputPkg.contents
      )
}
--@end rule Package

--@begin rule Class
rule Class {
   from
      inputA:KM3!Class
   to
      outputA:KM3target!Class (
         name <- inputA.name,
         isAbstract <- inputA.isAbstract,
         structuralFeatures <- inputA.structuralFeatures
      )
}
--@end rule Class

--@begin rule datatype
rule DataType {
from
      inputData:KM3!DataType
   to
      outputData:KM3target!DataType(
         name <- inputData.name,
         location <- inputData.location
      )
}
--@end rule datatype

--@begin foreignkey
rule reference2Key {
   from
      r:KM3!Reference (
         r.upper = 1
      )
   to
      key:KM3target!Attribute (
         name <- r.type.name.toLower()+'ForeignKey',
         isOrdered <- false,
         isUnique <- false,
         location <- '',
         lower <- 1,
         upper <- 1,
         type <- KM3!DataType.allInstances()->select(a|a.name = 'Integer')->first(),
         owner <- r.owner
      )
}
--@end foreignkey

--@begin rule attribute
rule attribute {
   from
      inputAtt:KM3!Attribute
   to
      outputAtt:KM3target!Attribute(
         name <- inputAtt.name,
         isOrdered <- inputAtt.isOrdered,
         isUnique <- inputAtt.isUnique,
			location <- inputAtt.location,
			lower <- inputAtt.lower,
			upper <- inputAtt.upper,
			type <- inputAtt.type,
			owner <- inputAtt.owner

		)
}
```

3. References

[1] Catalogue of Model Transformations
http://www.dcs.kcl.ac.uk/staff/kcl/tcat.pdf