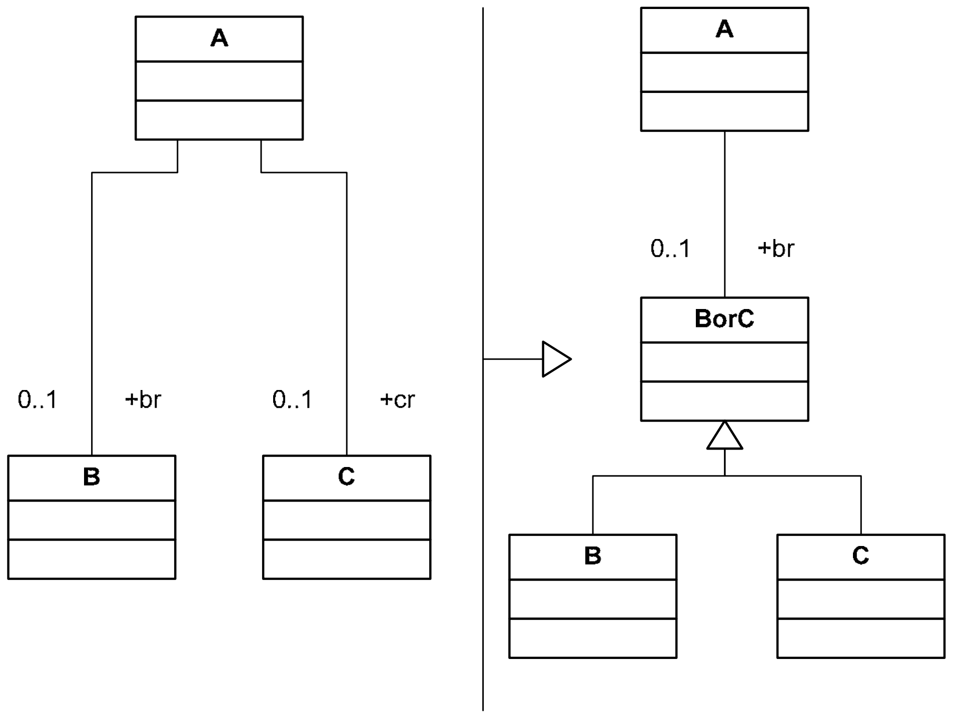
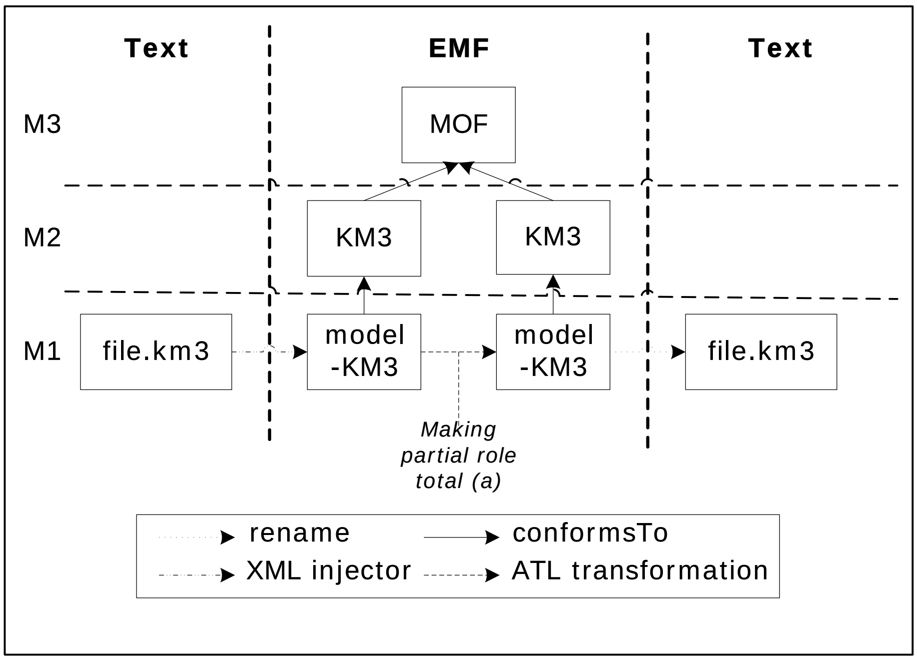
INRIA	ATL Transformation Catalogue of Model Transformations	Author Baudry Julien Jul.baudry <*at*> gmail.com
	Documentation	Aug 8th 2006

1. ATL Transformation Example: making partial role total (a)

This example is extract from Catalogue of Model Transformations by K. Lano.
Section 2.14: making partial role total (a), page 23.





2. ATL Transformation overview

2.1. Description

A 0..1 multiplicity role of a class A may be turned into a 1 multiplicity role by either moving the role to a superclass of its current target, or by moving the other end to a subclass of A on which the association is total.

2.2. Purpose

Total associations are generally easier to implement and manage than partial associations. The previous figure shows the ‘generalise target’ version of this transformation.

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, we must distinguish two cases :
    - If the class has a reference with a 0..1 cardinality :
        - We create another class with the same name,
        - Abstract if the source class is abstract,
        - Linked to the same supertypes and the same structuralFeatures,
        - Another Class element named '<name of the current class> + total order'
            - The value of isAbstract is true,
            - Is linked to the same package as the current class,
            - Supertypes is linked to the structural features of the current class.

    - If the class has not references with a 0..1 cardinality :
        - We create another class with the same name,
        - Abstract if the source class is abstract,
        - Linked to the same supertypes and the same structuralFeatures.

- For a Reference element with a 0..1 cardinality, two reference are created :
    - with the same properties of the current reference,
    - named '<name of the current class> + TotalOrder' and '<name of the current class> + OppositeTotalOrder',
    - with upper and upper value equal to 1,
    - Their types are the abstract class created by the previous rule.

2.4. ATL Code

```
-- @name    Making partial role total (a)
-- @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 making a patial role total
-- @see http://www.dcs.kcl.ac.uk/staff/kcl/tcat.pdf
-- @see section 2.14, page 23
-- @see author of article : K. Lano

module Replace; -- 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 ClassWithPartialOrder
rule ClassWithPartialOrder {
    from
        inputClass:KM3!Class (
            not(inputClass.structuralFeatures->select(r|(r.upper=1)and(r.lower=0))->isEmpty())
        )

    to
        outputClass:KM3target!Class (
            name <- inputClass.name,
            isAbstract <- inputClass.isAbstract,
            structuralFeatures <- inputClass.structuralFeatures,
            supertypes <- inputClass.supertypes
        ),
        totalOrderClass : KM3target!Class (
            name <- inputClass.name+'TotalOrder',
            isAbstract <- true,
            package <- inputClass.package,
            supertypes <- inputClass.structuralFeatures->iterate(a;acc:Sequence(KM3!Class)=Sequence{}|
                                                          acc->including(a.opposite.owner))
        )
}
--@end ClassWithPartialOrder

--@begin ClassWithoutPartialOrder
rule ClassWithoutPartialOrder {
    from
        inputClass:KM3!Class (
            inputClass.structuralFeatures->select(r|(r.upper=1)and(r.lower=0))->isEmpty()
        )

    to
        outputClass:KM3target!Class (
            name <- inputClass.name,
            isAbstract <- inputClass.isAbstract,
            structuralFeatures <- inputClass.structuralFeatures,
            supertypes <- inputClass.supertypes
        )
}
--@end ClassWithoutPartialOrder

--@begin reference partial order
rule referencePartialOrder {
    from
        inputRef : KM3!Reference (
            inputRef.upper = 1 and inputRef.lower = 0
        )
    to
        outputRef : KM3target!Reference (
            name <- inputRef.opposite.owner.name+'TotalOrder',
            isOrdered <- inputRef.isOrdered,
            isUnique <- inputRef.isUnique,
            location <- inputRef.location,
            lower <- 1,
            upper <- 1,
            type <- KM3target!Class.allInstances()->select(a|a.name=inputRef.owner.name+'TotalOrder')->asSequence()-
>first(),
            owner <- inputRef.owner,
            opposite <- inputRef.opposite
        ),
        outputOppositeRef : KM3target!Reference (
            name <- inputRef.opposite.owner.name+'OppositeTotalOrder',
            isOrdered <- inputRef.opposite.isOrdered,
            isUnique <- inputRef.opposite.isUnique,
            location <- inputRef.opposite.location,
            lower <- 1,
            upper <- 1,
            type <- inputRef.opposite.type,
            owner <- KM3target!Class.allInstances()->select(a|a.name=inputRef.owner.name+'TotalOrder')->asSequence()-
>first(),
            opposite <- outputRef
        )
}

--@end reference partial order
```

3. References

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