

Un aporte gráfico para las herramientas de transformación de modelos

Gabriela Pérez

Directora: Dra. Claudia Pons

Co-Directora: Dra. Roxana Giandini

Facultad de Informática
Universidad Nacional de La Plata



Agenda

- **Introducción**
- **MDD**
- **Objetivos de la tesis**
- **Ejemplo de transformación**
- **Arquitectura 4 capas de modelado**
- **OCL - Reglas de buena formación**
- **Tools**
- **Tools - Atl**
- **Tools - Mofscript**
- **Conclusiones de las herramientas**
- **Aporte de la tesis**
- **Conclusiones**
- **Trabajos futuros**

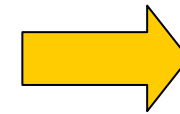
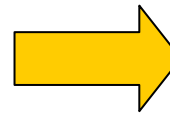
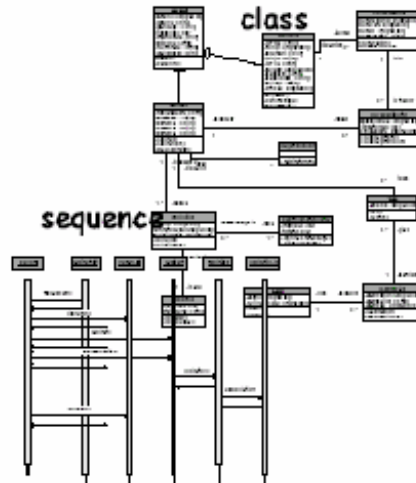
Introducción

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



```
static void main(String  
  
    FileOutputStream out;  
    PrintStream p; // dec  
  
    try  
    {  
  
        // Create a ne  
        // connected :  
        out = new File  
  
        // Connect pr:  
        p = new Print
```

```
C# .println ("Tl  
  
    p.close();  
    }  
    catch (Exception e)
```

Cambios

El problema de la productividad
documentos y diagramas pierden el valor
no hay tiempo disponible para actualizar modelos

**Nuevas
tecnologías**

problema de la portabilidad
aparecen nuevas tecnologías que rápidamente llegan a ser
populares

Evolución

problema del mantenimiento y la documentación

MDD - una propuesta de solución

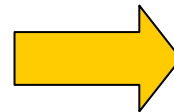
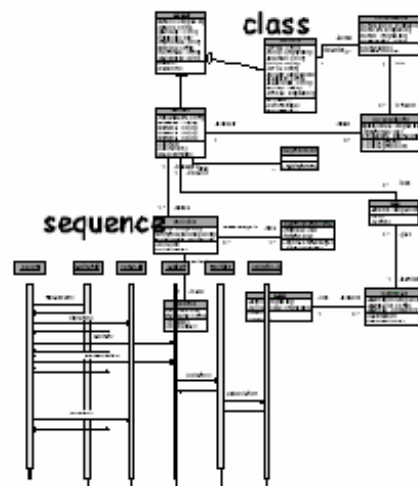
Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

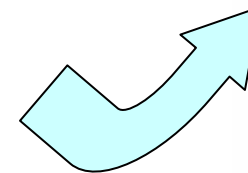
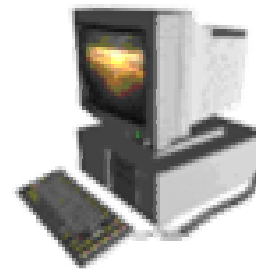
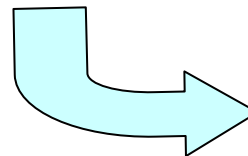
Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

- MDD = Model Driven Development o desarrollo conducido por modelos



```
static void main(String  
  
    FileOutputStream out;  
    PrintStream p; // dec  
  
try  
{  
    // Create a ne  
    // connected  
    out = new File  
  
    // Connect pr  
    p = new Print  
  
    p.println ("T  
  
    p.close();  
  
}  
catch (Exception e)  
Código  
Java
```



"from human-readable to computer-understandable"
J. Bézin

MDD - Elementos principales

*Introducción
MDD
Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

*Conclusiones
Trabajos
futuros*

En MDD los modelos son considerados los conductores primarios en todos los aspectos del desarrollo de software

- **Platform Independent Model (PIM)**

“Un modelo de un sistema que **no** contiene información acerca de la plataforma o la tecnología que es usada para implementarlo”

- **Platform Specific Model (PSM)**

“Un modelo de un sistema que incluye información acerca de la tecnología específica que se usará para su implementación sobre una plataforma específica”

- **Transformación de modelos**

especifica el proceso de conversión de un modelo en otro modelo del mismo sistema.

Cada transformación incluye (al menos):

- un PIM,
- un Modelo de la Plataforma,
- una Transformación, y
- un PSM

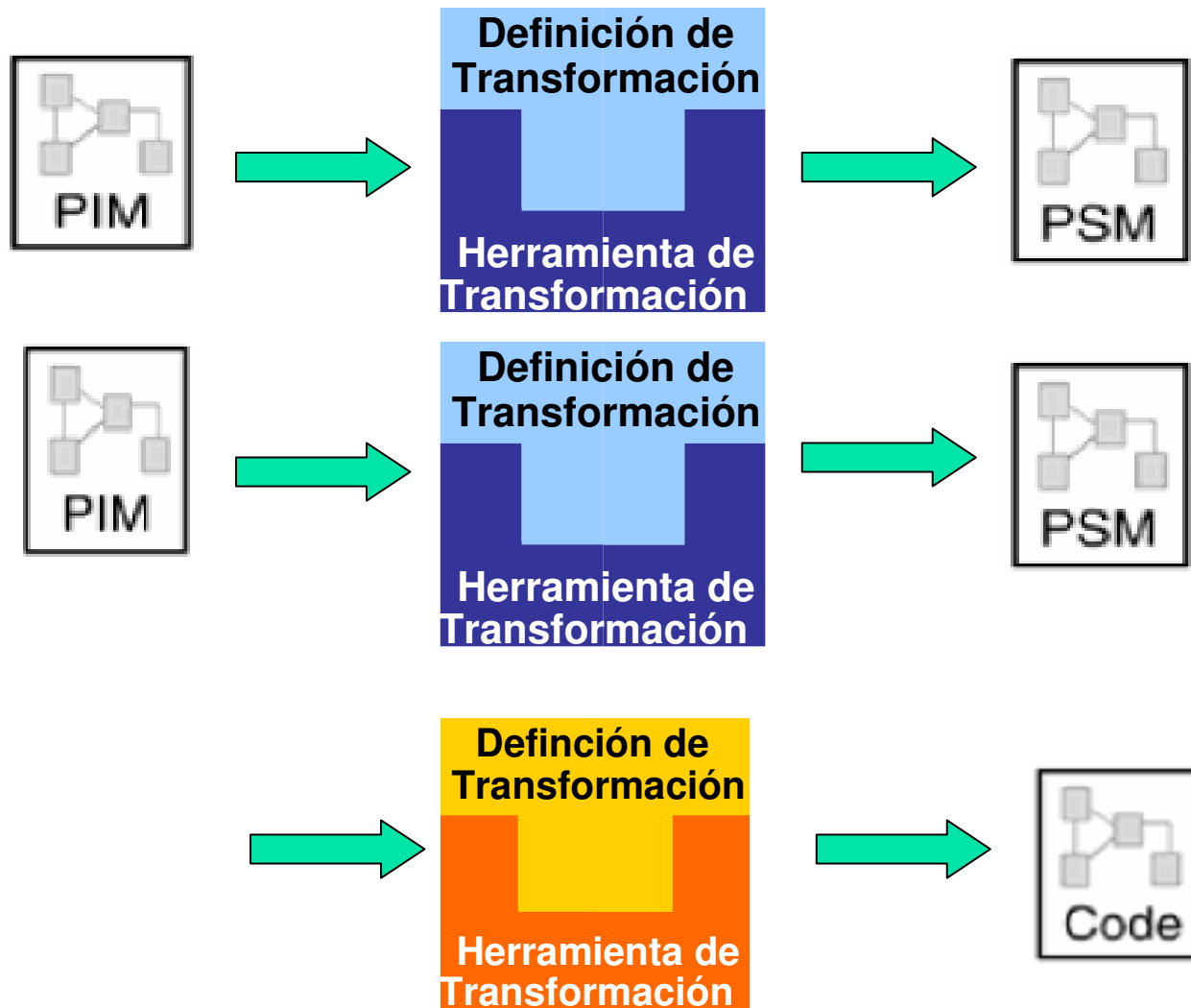
MDD – Relación entre modelos

Introducción
MDD
Objetivos


Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



MDD - Beneficios


Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

■ Productividad

- menos trabajo que hacer
- el sistema desarrollado se acerca más a las necesidades del usuario final
- mejora en la implementación de modificaciones

■ Portabilidad

- puede ser automáticamente transformado en muchos PSMs para diferentes plataformas

■ Mantenimiento y documentación

- el modelo será una exacta representación del código

Objetivo de la Tesis

*Introducción
MDD*

■ *Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

*Conclusiones
Trabajos
futuros*

- analizar el concepto de transformaciones de modelos;
- analizar cómo se da soporte a MDD en las herramientas de ingeniería de software actuales
- hacer un aporte a las herramientas estudiadas para facilitar su utilización mediante la implementación de un plugin que permite la definición de metamodelos, modelos y chequeo de reglas de buena formación en los metamodelos.

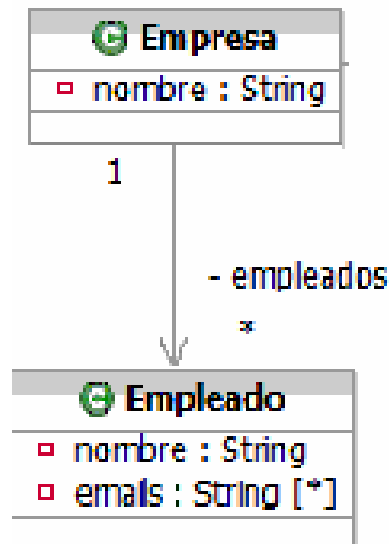
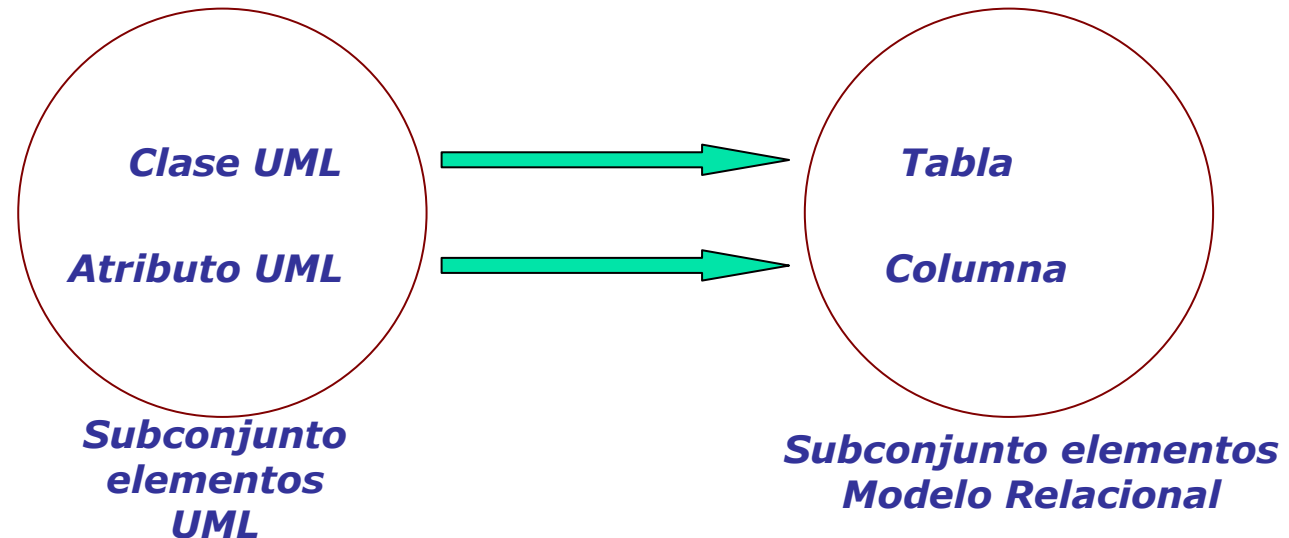
Ejemplo de transformación

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



Modelo Relacional



Ejemplo de transformación

*Introducción
MDD
Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

*Conclusiones
Trabajos
futuros*

Transformation UML2Relational (Uml:UML2.0, Rel: Relational) {

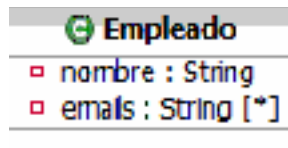
Relation UMLClass2Table {

checkonly domain Uml c: Class

enforced domain Rel t: Table

when { }

where { c.name = t.name and c. ownedAttribute -> forAll (
a : Attribute | (t.ownedColumns -> exists(col:Column |
col.name = a.name)) }



Empresa_Empleados

empresaID: Integer
empleadoID: Integer

MDA

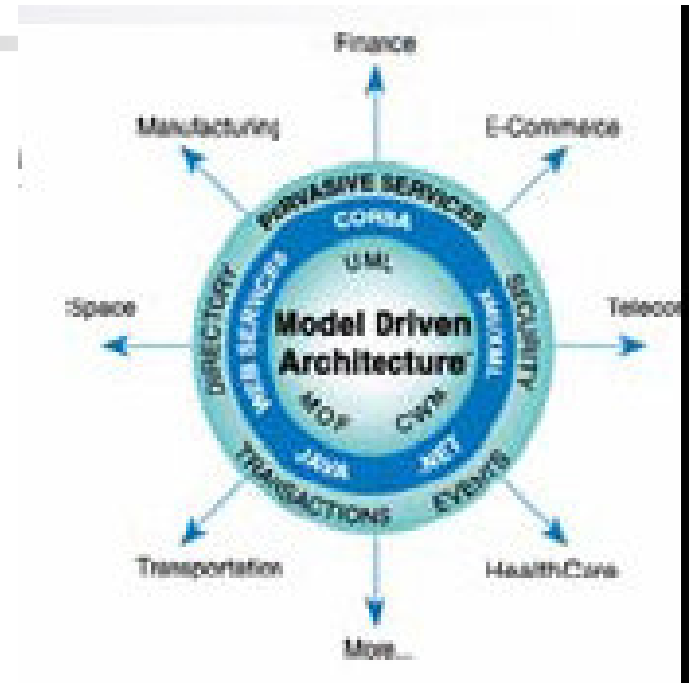
*Introducción
MDD
Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

*Conclusiones
Trabajos
futuros*

- MDA = Model Driven Architecture o arquitectura dirigida por modelos
- MDA = MDD + lenguajes estandares (MOF, XMI, QVT)



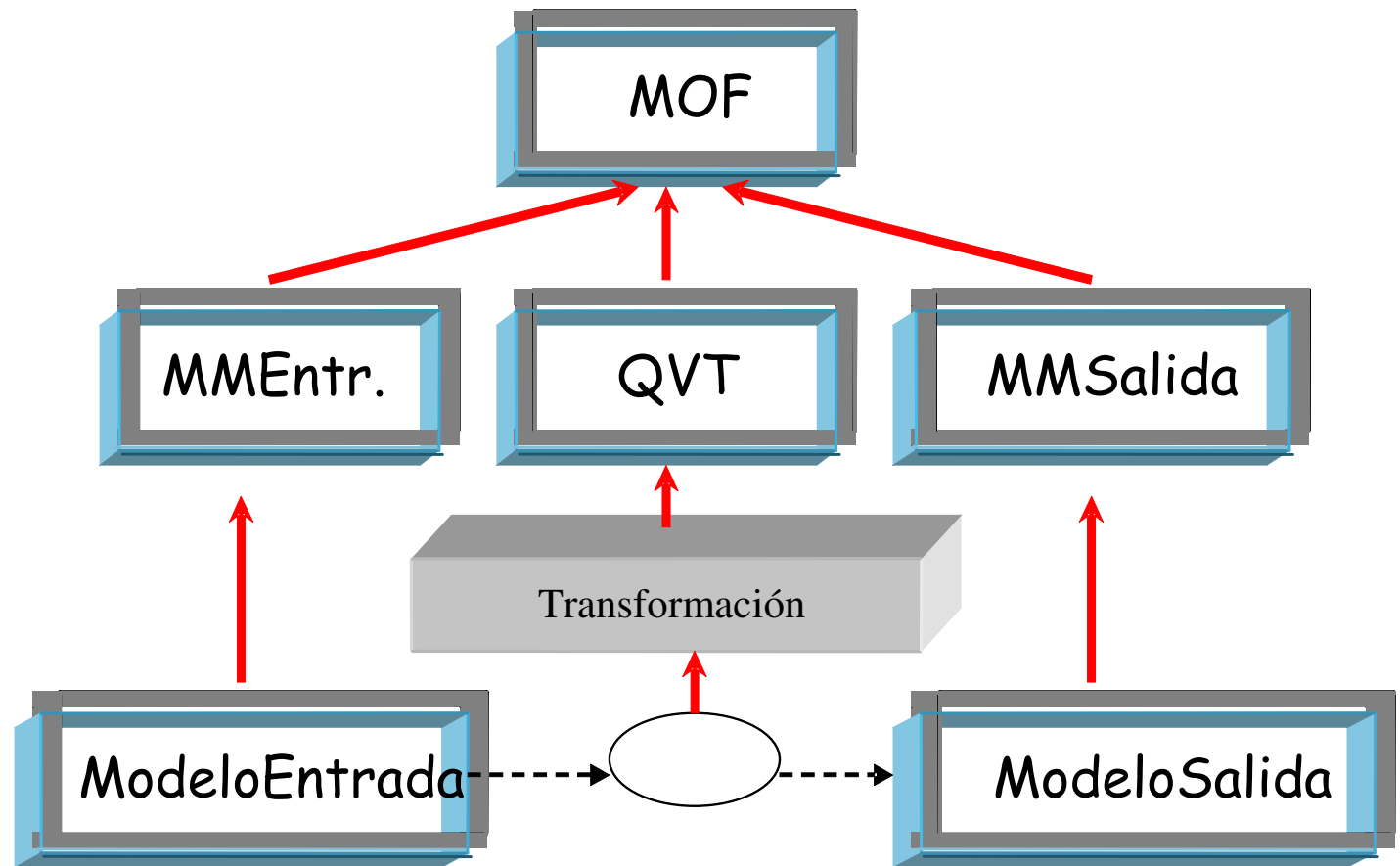
Esquema de transformación

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



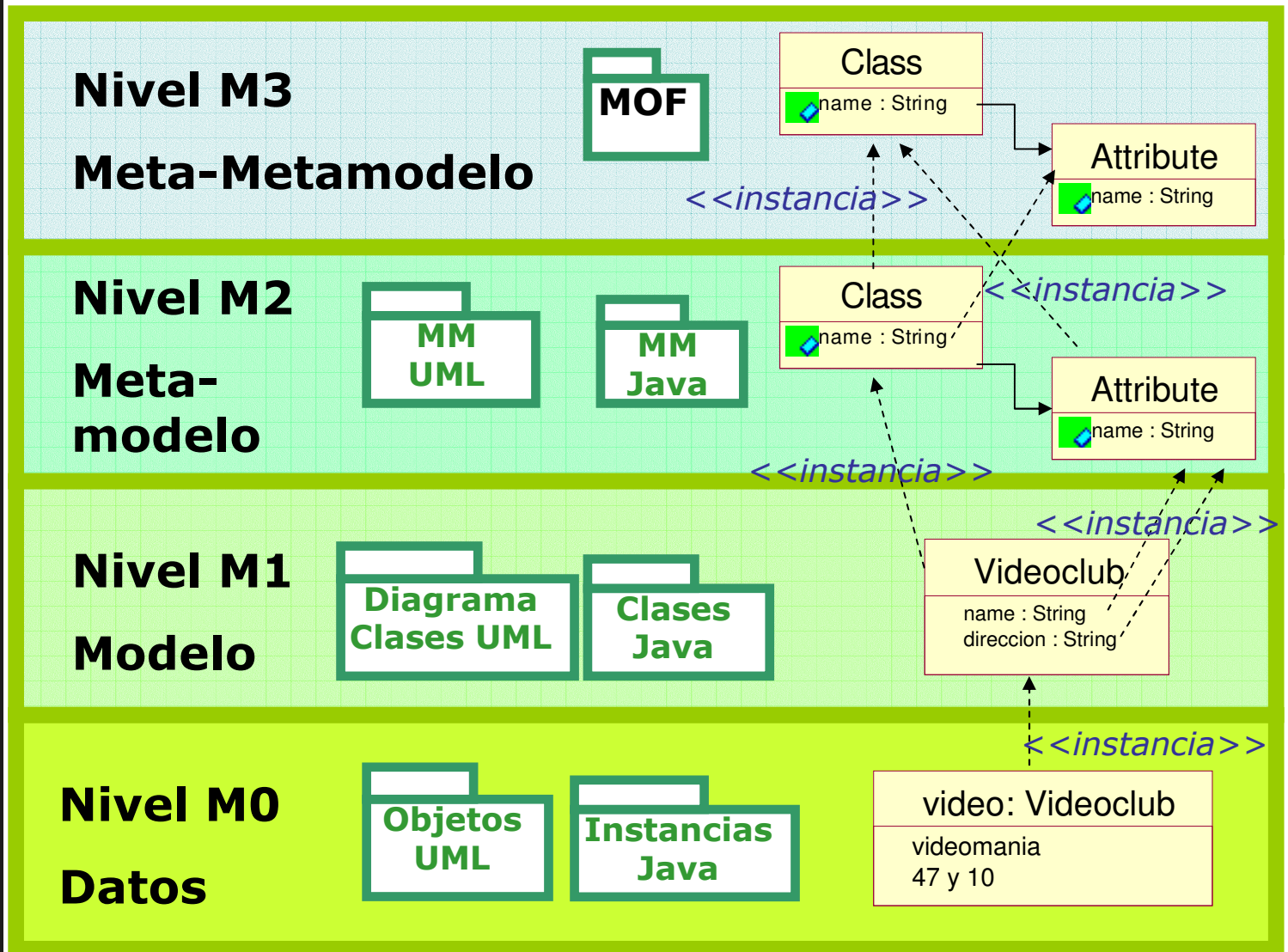
Arquitectura 4 capas de modelado

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



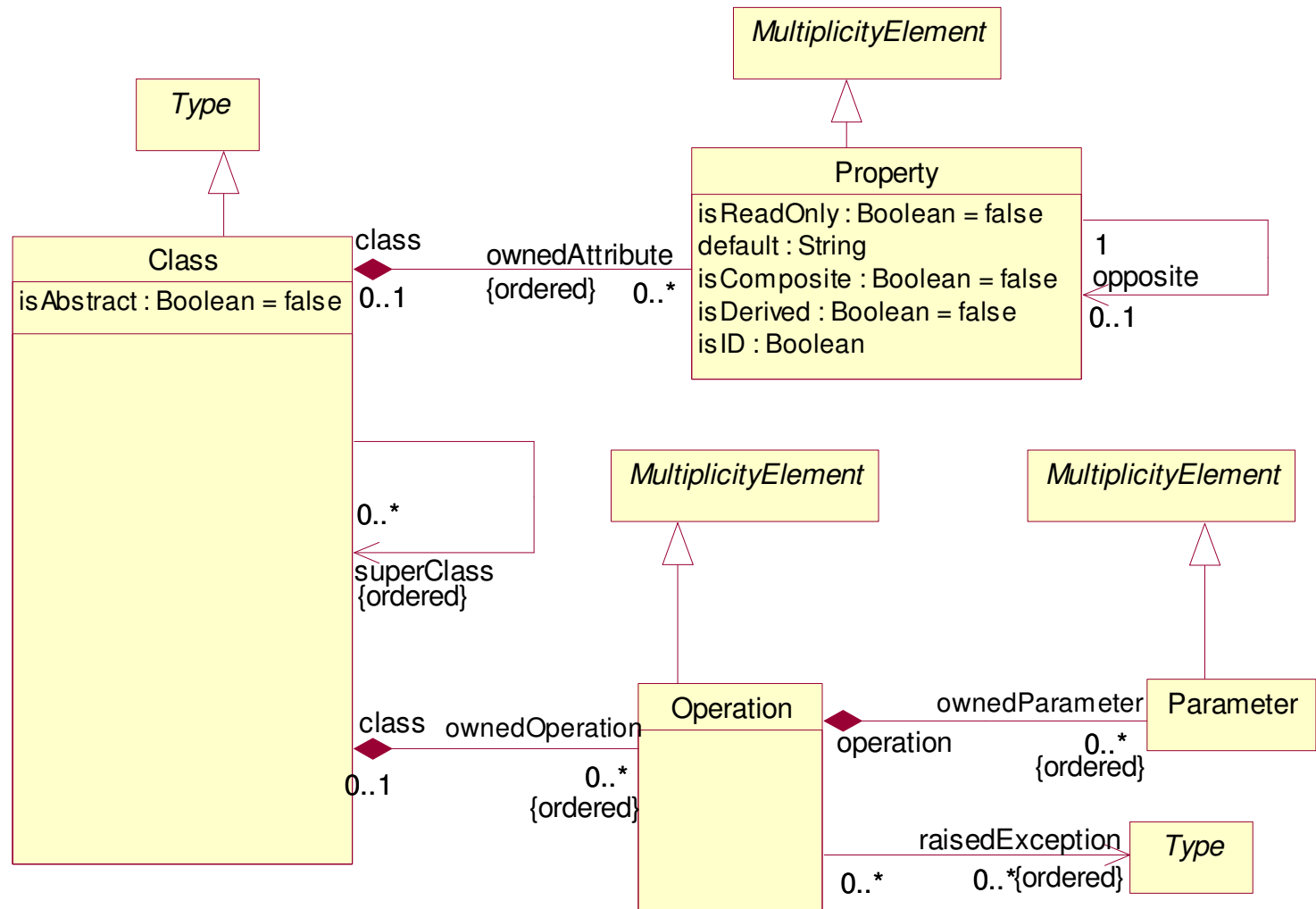
MOF - Clases

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



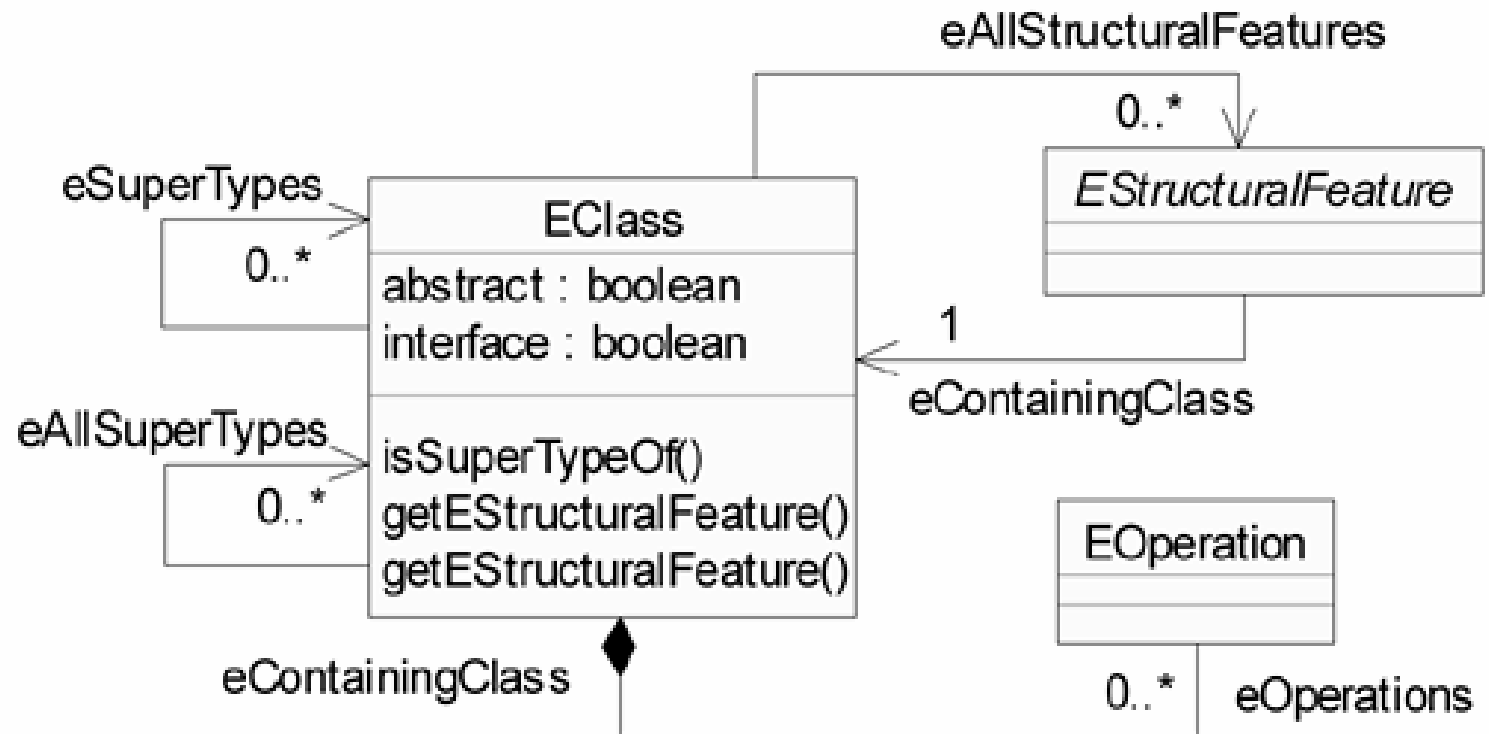
Implementación de MOF - Ecore

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros





OCL

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

OCL permite especificar diferentes tipos de restricciones:

- Invariantes
- Pre y Postcondiciones

Con estas restricciones se pueden definir reglas de buena formación (Well – Formedness Rules) en los distintos niveles.

Arquitectura 4 capas de modelado

OCL - Reglas de buena formación

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

Nivel M3
Meta-
Metamodelo

MOF

Nivel M2
Meta-modelo

WFR nivel meta-metamodelo

Las clases de un paquete MOF deben tener distintos nombres

Meta
Modelo

Nivel M1
Modelo

WFR nivel metamodelo

Las asociaciones de UML deben tener distinto nombre de rol

Diagrama
Clases

Nivel M0
Datos

WFR nivel modelo

El mail de los clientes deben contener un caracter @

Objetos

Arquitectura 4 capas de modelado

OCL - Reglas de buena formación

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

Nivel M3
Meta-
Metamodelo

MOF

Nivel M2
Meta-modelo

MM
UML

context EPackage

inv WFR-EPackage:

--The EClass must have different names.

self.eClassifiers -> select(c|c.ocllsKindOf (EClass)) -> forAll (c, c2 |
c.name = c2.name implies c = c2)

WFR nivel meta-metamodelo

Nivel M1
Modelo

Diagrama
Clases UML

context Association

inv WFR_Association:

--[1] Los roles deben tener distinto nombre

self.memberEnd -> forAll(p,q| p.name = q.name implies p = q)

WFR nivel metamodelo

Nivel M0
Datos

Objetos
UML

context Cliente

inv:

-- Los mails validos contienen @
self.email.contains('@')

WFR nivel modelo



Agenda

- **Introducción**
- **MDD**
- **Objetivos de la tesis**
- **Ejemplo de transformación**
- **Arquitectura 4 capas de modelado**
- **OCL - Reglas de buena formación**
- **Tools**
- **Tools - Atl**
- **Tools - Mofscript**
- **Conclusiones de las herramientas**
- **Aporte de la tesis**
- **Conclusiones**
- **Trabajos futuros**

Herramientas de transformación

*Introducción
MDD
Objetivos*

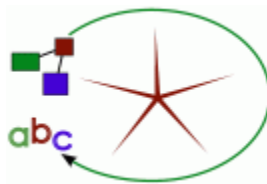
*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

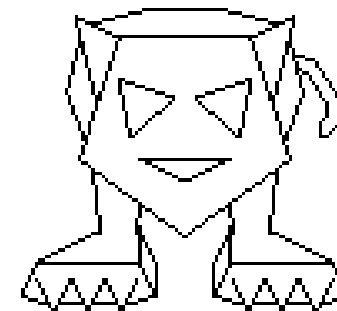
*Conclusiones
Trabajos
futuros*



VIA TRA



MOFScript



Tefkat

Herramientas de transformación

*Introducción
MDD
Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

*Conclusiones
Trabajos
futuros*

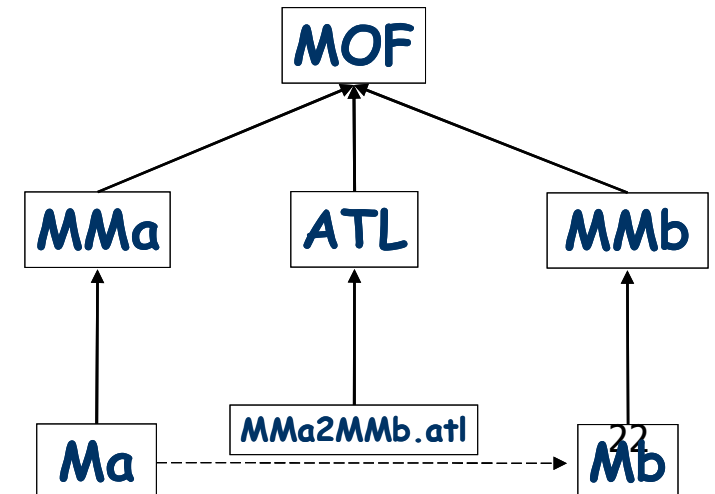
Nombre de la herramienta	Definición de los metamodelos	Lenguaje de transformación
VIATRA		
TefKat	Compatible con MOF	
ATL	Compatible con MOF	Basado en QVT
Epsilon		
AToM3		
MOLA	Compatible con MOF	
MofScript	Compatible con MOF	Basado en QVT
Kermeta	Compatible con MOF	
Kent	Compatible con MOF	Basado en QVT

Herramienta de transformación - ATL



ATLAS Transformation Language

Transforma modelo fuente en un modelo destino.



Introducción
MDD
Objetivos

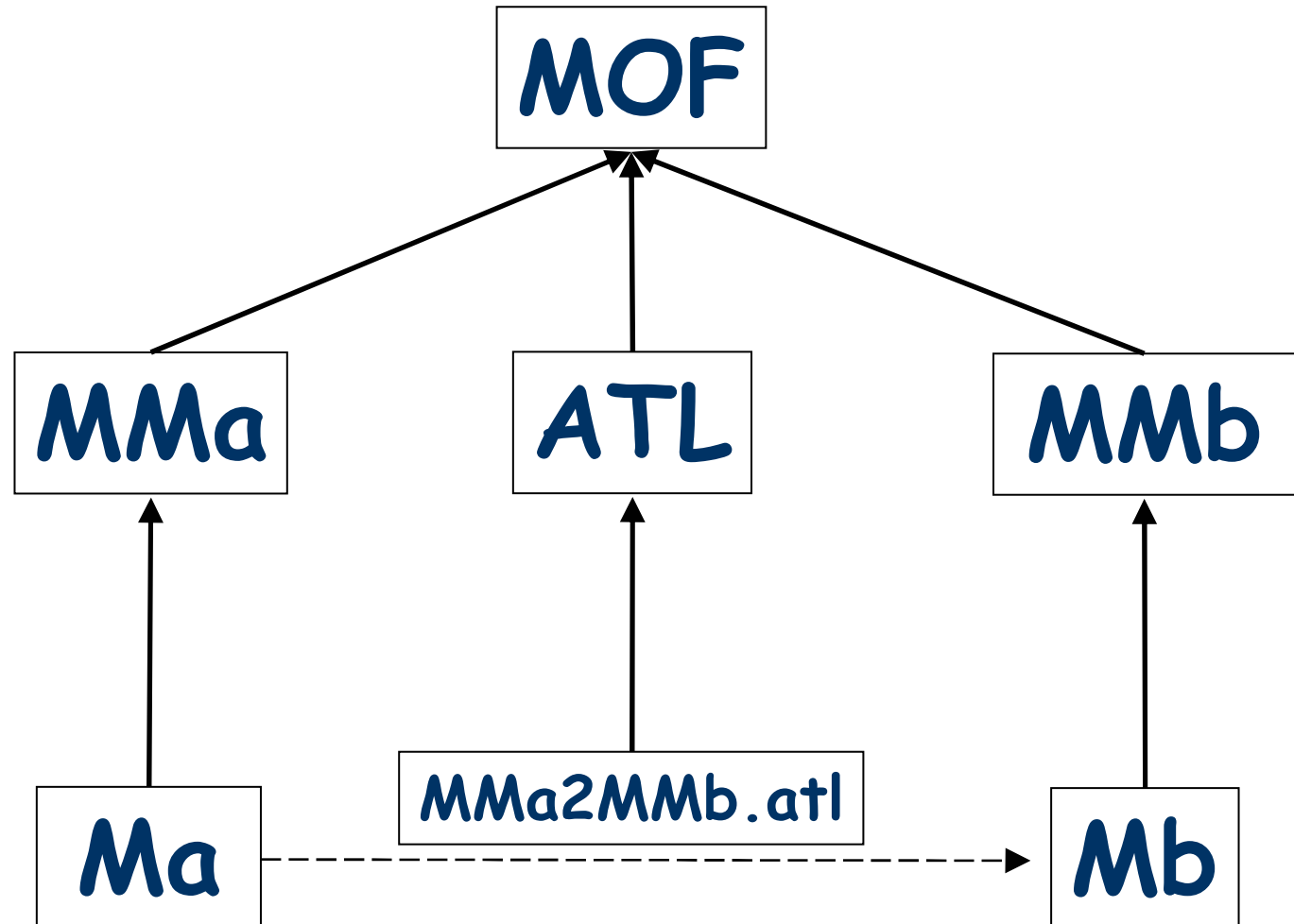
Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

ATL

- Introducción
- MDD
- Objetivos
- Transf. Ej.
- Arq. 4 capas
- OCL
- Tools
 - Atl
 - Mofscript
 - Aporte tesis
- Conclusiones
- Trabajos futuros



ATL – Especificar metamodelos

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

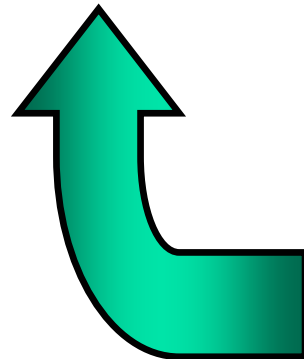
Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

Ecore - XMI

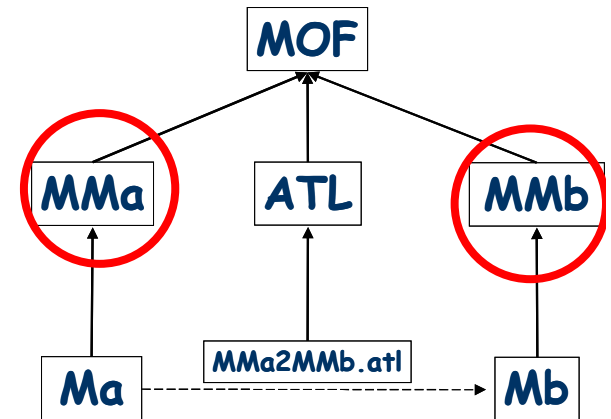
```
<eClassifiers
  xsi:type="ecore:EClass"
  name="NamedElement"
  abstract="true">

<eStructuralFeatures
  xsi:type="ecore:EAttribute"
  name="name"
  lowerBound="1"
  eType="/0/String"/>
</eClassifiers>
```



Kernel MetaMetaModel (KM3)

```
class Attribute extends NamedElement {
  attribute name : String;
}
```



ATL – Especificar metamodelos KM3

Introducción
MDD
Objetivos

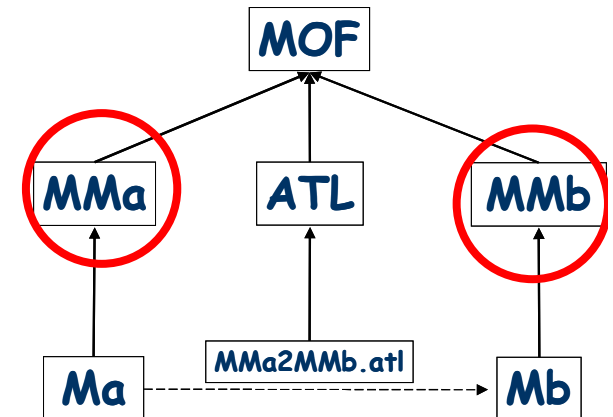
Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

Kernel MetaMetaModel (KM3)

```
class Class extends Classifier {  
    reference super[*] : Class;  
    reference attr[*] ordered container :  
        Attribute oppositeOf owner;  
    attribute isAbstract : Boolean;  
}  
  
class Attribute extends NamedElement {  
    attribute multiValued : Boolean;  
    reference type : Classifier;  
    reference owner : Class oppositeOf attr;  
}
```



ATL – Especificar transformación

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

rule Class2Table {

from

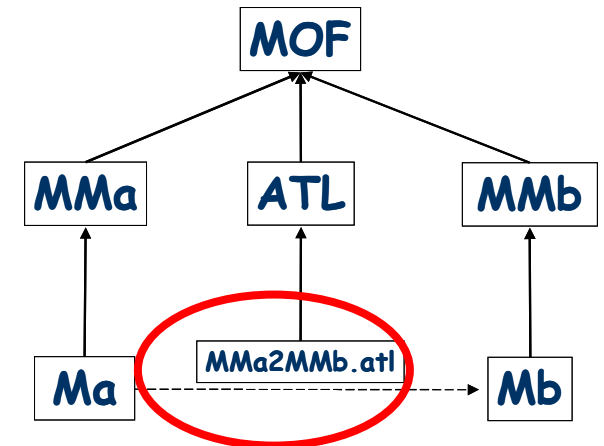
c : Class!Class

to

out : Relational!Table (
 name <- c.name,
 col <- Sequence {key} ->
 union(c.attr->select(e | not
e.multiValued)),
 key <- Set {key}
),

key : Relational!Column (
 name <- 'objectId',
 type <- thisModule.objectIdType
)

}



ATL – Modelo de entrada

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

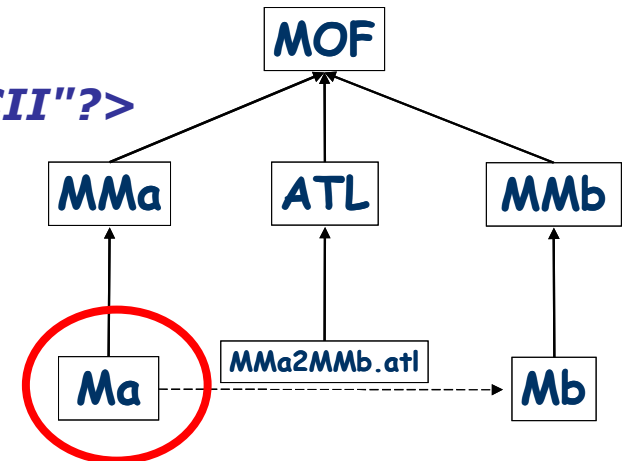
```
<?xml version="1.0" encoding="ASCII"?>
```

```
<XMI:XMI XMI:version="2.0"
xmlns:XMI="http://www.omg.org/
XMI" xmlns="Class">
```

```
<Class name="Empresa">
  <attr name="nombre" multiValued="false" type="/2"/>
  <attr name="empleados" multiValued="true"
type="/1"/>
</Class>
```

```
<Class name="Empleado">
  <attr name="nombre" multiValued="false" type="/2"/>
  <attr name="emailAddresses" multiValued="true"
type="/2"/>
</Class>
```

```
<DataType name="String"/>
</XMI:XMI>
```



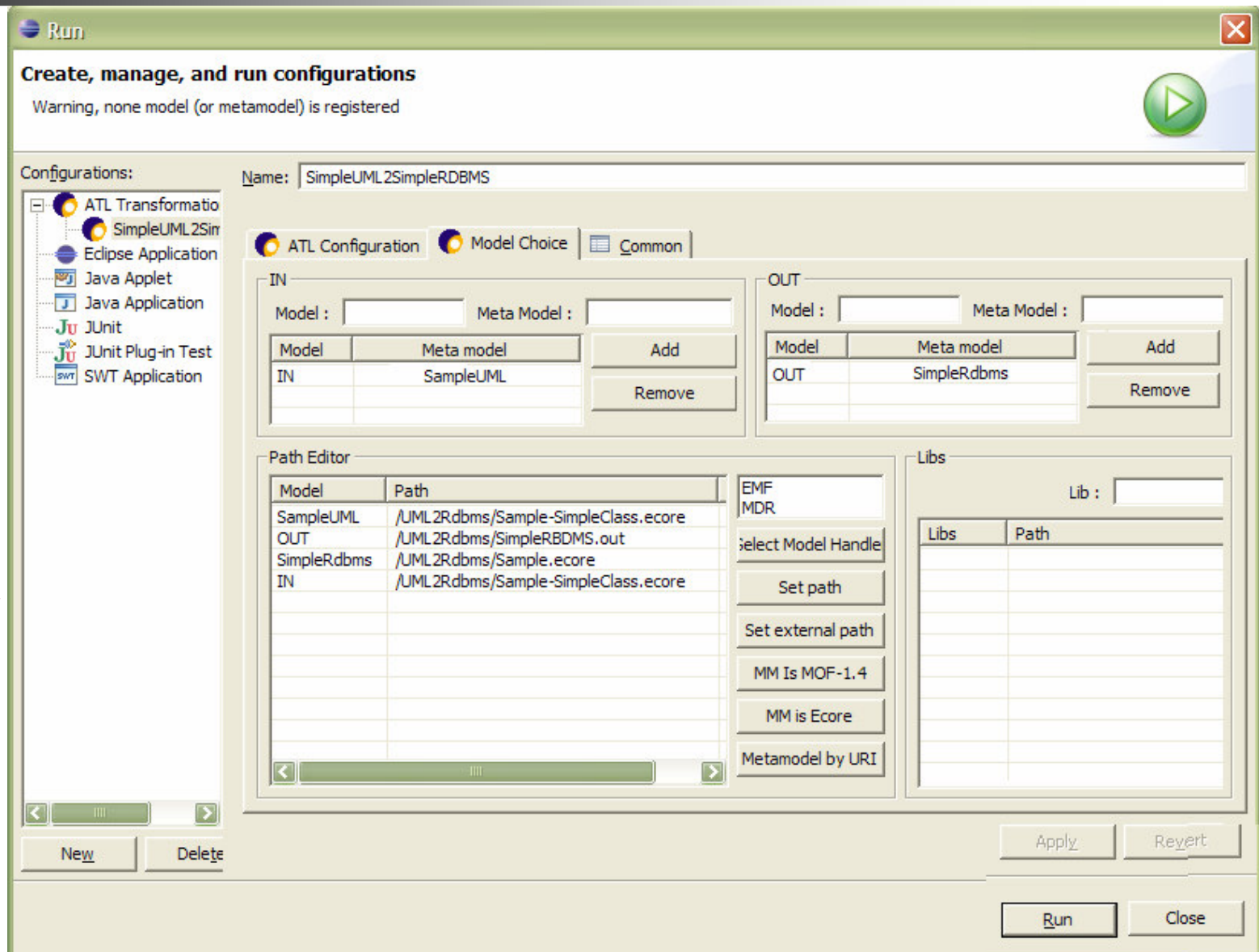
ATL – Ejecutar la transformación

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



ATL – Modelo de Salida

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

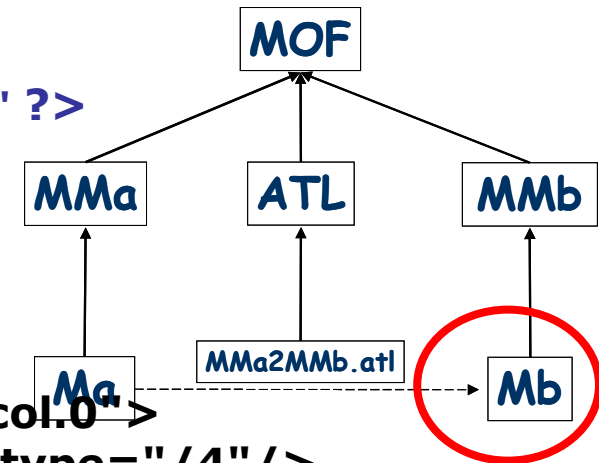
Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

```
<?xml version="1.0" encoding="ISO " ?>
<XMI:XMI XMI:version="2.0"
xmlns:XMI="http://www.omg.org
/XMI" xmlns="Relational">
```

```
<Table name="Empresa" key="/0/@col.0">
  <col name="objectId" keyOf="/0" type="/4"/>
  <col name="nombre" type="/3"/>
</Table>
<Table name="Empleado" key="/1/@col.0">
  <col name="objectId" keyOf="/1" type="/4"/>
  <col name="nombre" type="/3"/>
</Table>
<Table name="Empresa_Empleado" key="/1/@col.0">
  <col name="empresaID" type="/4"/>
  <col name="empleadoID" type="/4"/>
</Table>
<Type name="String"/>
<Type name="Integer"/>
```

```
</XMI:XMI>
```



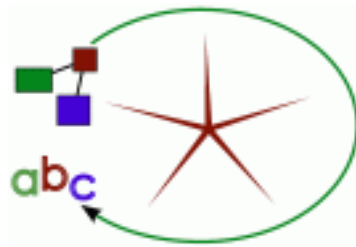
Herramienta de transformación - MOFScript

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

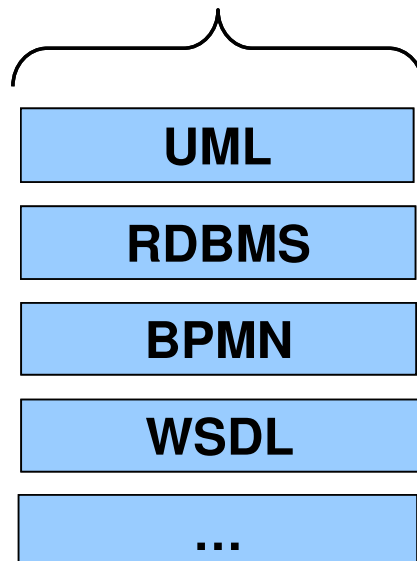
Conclusiones
Trabajos
futuros



MOFScript

• **Transforma Modelos a texto**

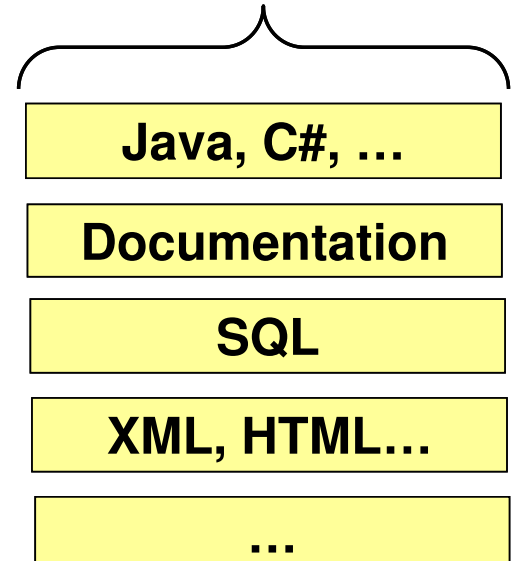
Metamodelos MOF



MOFScript



Textual output



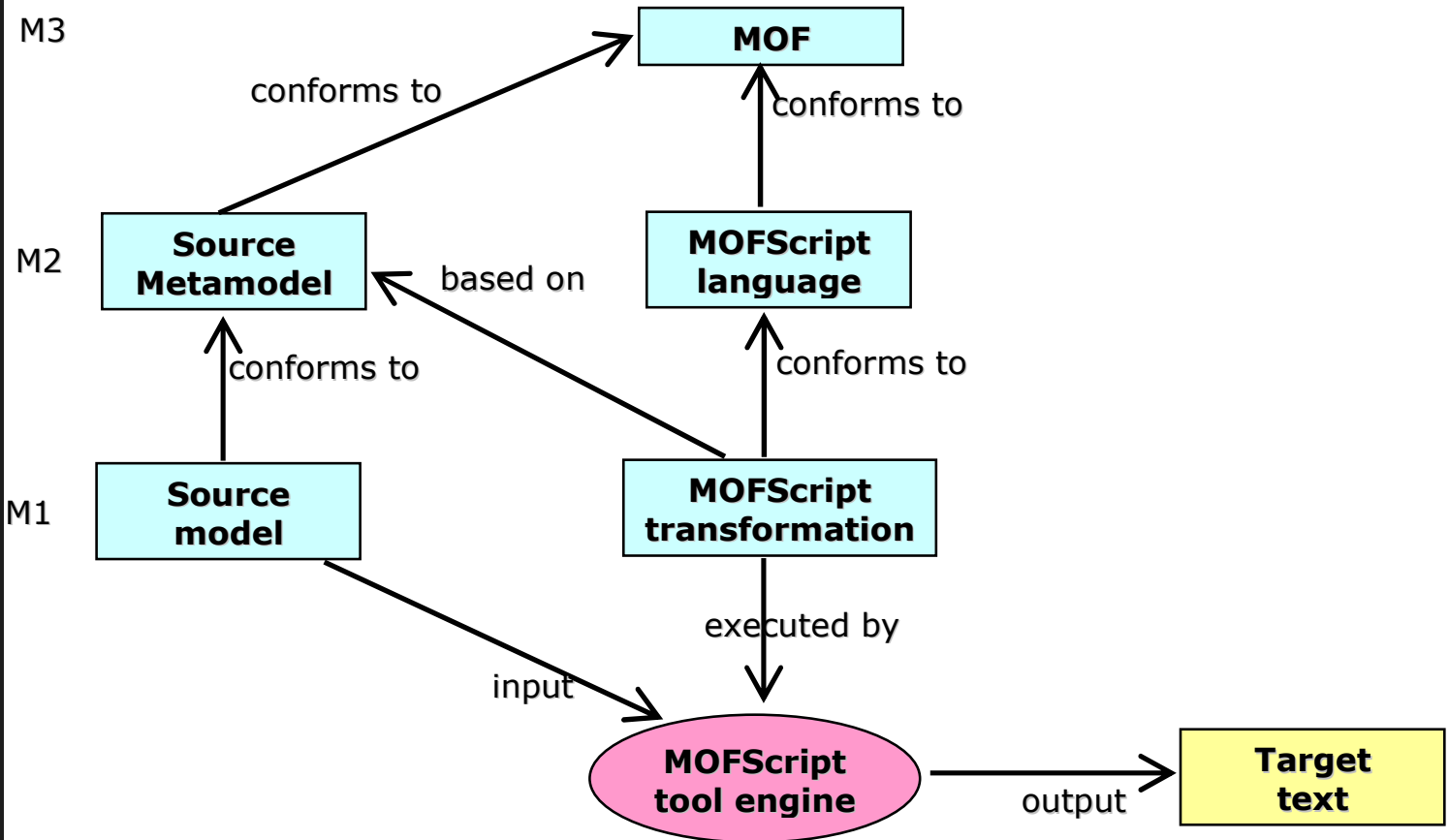
Esquema de transformación - MOFScript

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



MOFScript – Especificar metamodelo

Introducción
MDD
Objetivos

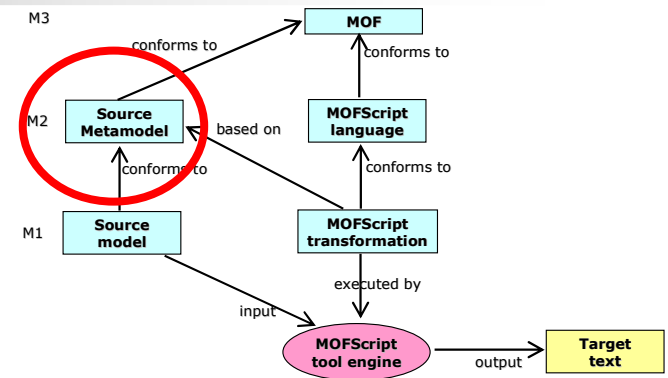
Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl

Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

- Define una librería de metamodelos on-line
- Permite importar un diagrama de clases UML dibujado en algún editor y convertirlo a un modelo ecore.
- Para poder utilizar un metamodelo nuevo hay que “instalarlo”



MOFScript – Especificar transformación

Introducción
MDD
Objetivos

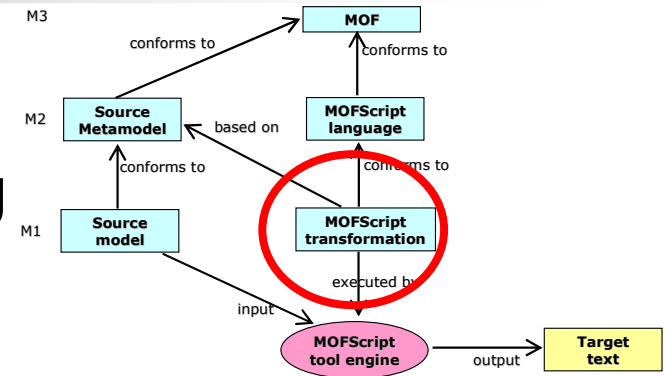
Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

```
texttransformation Ex (in  
    uml:"http://www.eclipse.org  
    /uml2/2.0.0/UML") {
```

```
    uml.Class::main () {  
        file (self.name + ".java")  
  
        'public class ' self.name ' { '  
        self.ownedAttribute->forEach(p:uml.Property) {  
            p.privateProperty()  
        }  
        '  
    } // end of class ' self.name '
```



MOFScript – Modelo de entrada

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

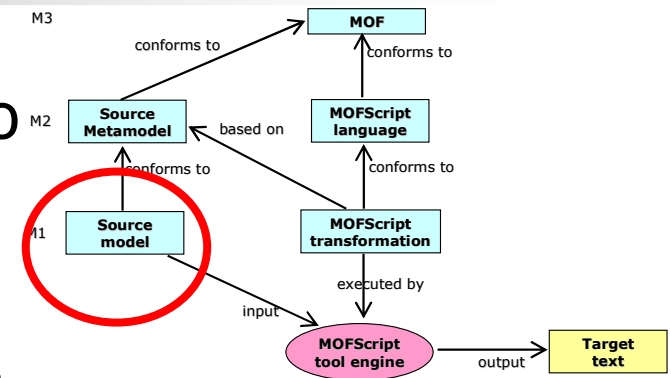
Tools
Atl

Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

- Admite modelos xmi pero no da ayudas para definirlos.

Si se usa como metamodelo UML, se puede importar un modelo realizado con alguna herramienta UML (x ej Rational).



MOFScript – Texto de salida

Introducción
MDD
Objetivos

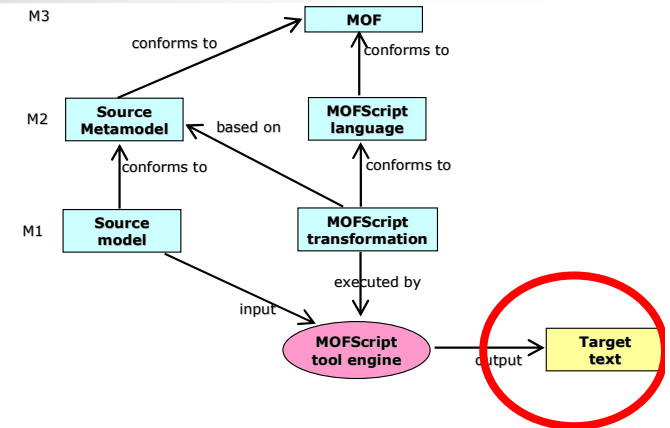
Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

```
public class Persona {  
    private String nombre;  
    private int edad;  
    private String direccion;  
    private Empresa trabajaen;  
} // end of class Persona
```

```
public class Empresa {  
    private String nombre;  
    private String direccion;  
} // end of class Empresa
```



Conclusiones de las herramientas

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl

 *Mofscript*
Aporte tesis

Conclusiones
Trabajos
futuros

- No proveen ayuda gráfica para la edición de metamodelos
- No garantiza que los metamodelos estén bien formados.
- No proveen ayuda para la instanciación de modelos a partir de metamodelos definidos

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl

Mofscript
 *Aporte tesis*

Conclusiones
Trabajos
futuros

- Creación y edición gráfica de modelos UML
- Edición y validación de restricciones OCL en modelos UML
- Otras funciones no relacionadas...



Aporte de la tesis

Los objetivos agregados en la presente tesis son:

- Creación y edición gráfica de modelos Ecore (para la definición de metamodelos)
- Instanciación del metamodelo definido (para la definición de modelos)
- Edición y validación de restricciones OCL en el nivel del meta-metamodelo, para verificar si un metamodelo esta bien definido.
- La posibilidad de interoperar con otras herramientas, que soporten MDA, mediante el uso del estándar xmi.

Introducción

MDD

Objetivos

Transf. Ej.

Arq. 4 capas

OCL

Tools

Atl

Mofscript

■ *Aporte tesis*

Conclusiones

Trabajos

futuros

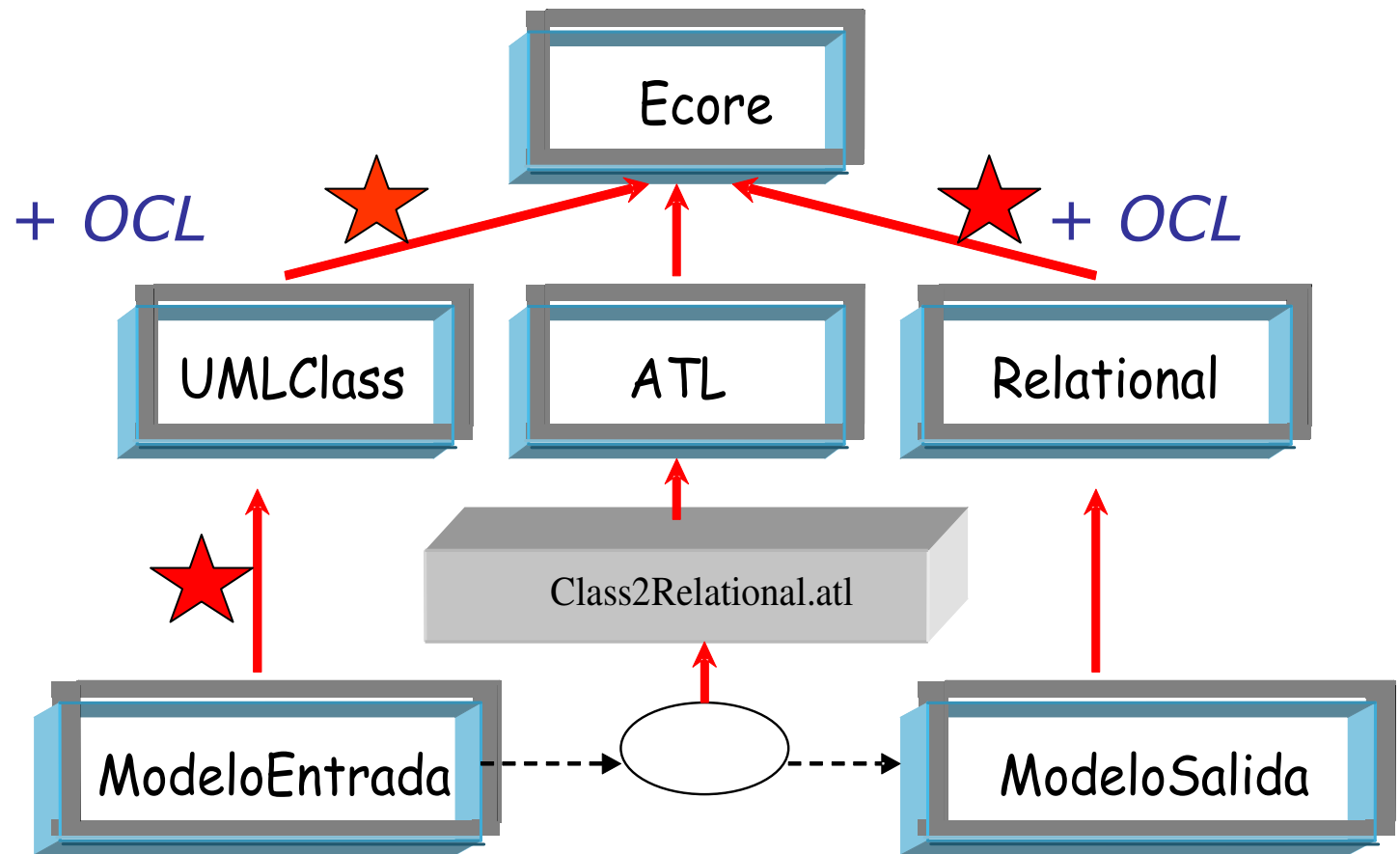
Aporte de la tesis

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



Editor de metamodelos

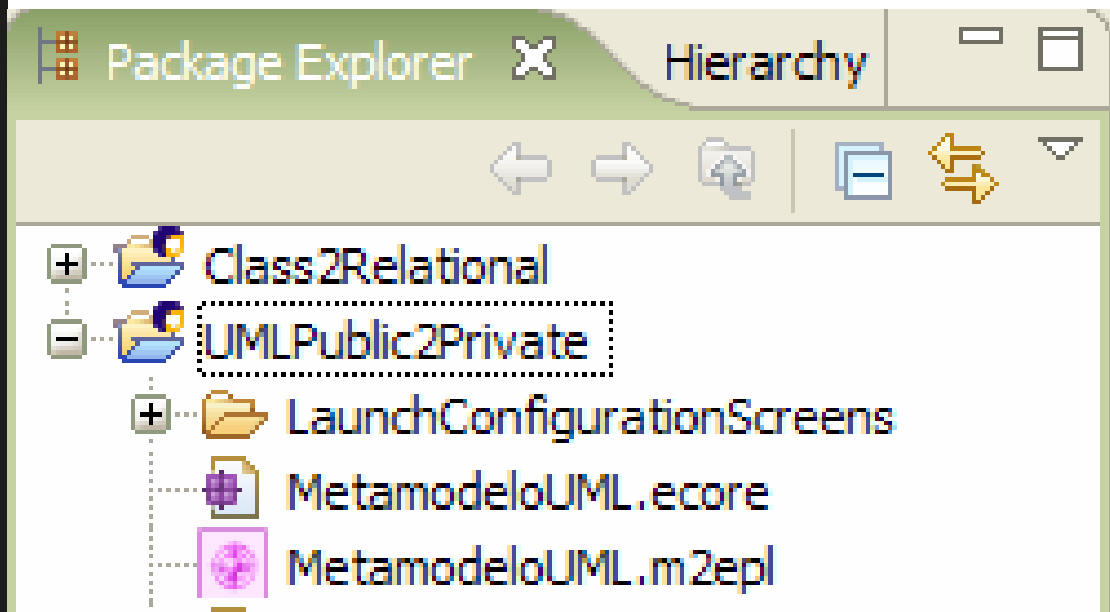
*Introducción
MDD
Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl*

*Mofscript
Aporte tesis*

*Conclusiones
Trabajos
futuros*



.m2epl -> guarda información perteneciente al diagrama

.ecore -> las instancias de los elementos del metamodelo Ecore.

Editor de metamodelos

Introducción
MDD
Objetivos

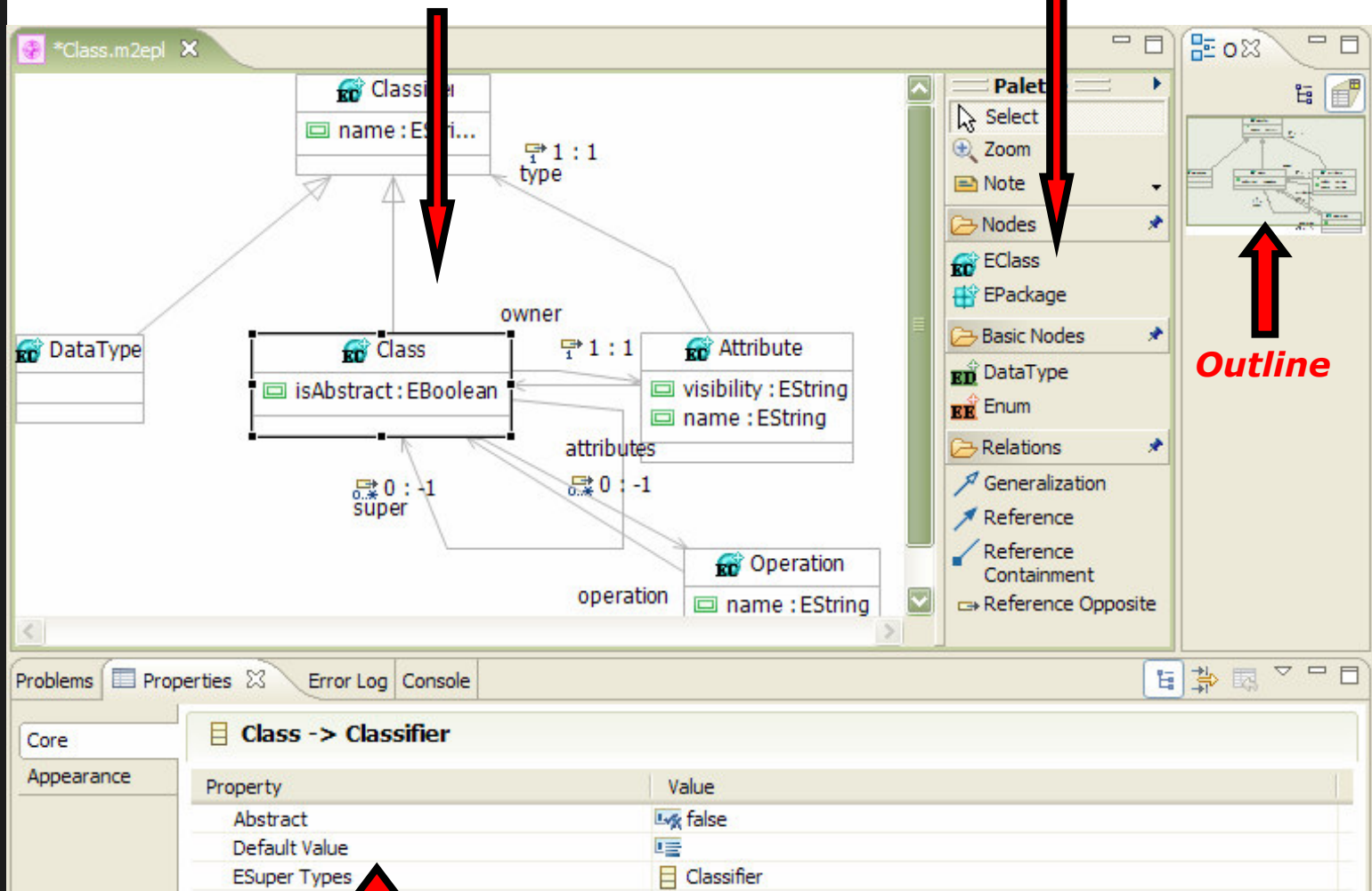
Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

Editor de diagrama de clases

Elementos Ecore



Outline

Editor de propiedades de los elementos ecore

Editor de modelos

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros

The screenshot displays the ATL Transformation Editor interface. The main workspace shows a class diagram with the following elements:

- Class Cliente
 - Attribute name
- Class Banco
 - Attribute nombre
 - Attribute direccion
- Class CuentaBancaria
 - Data Type String

The Properties tab at the bottom shows the following details for the selected element:

Property	Value
Name	nombre
Owner	Class Banco
Type	Data Type String
Visibility	public

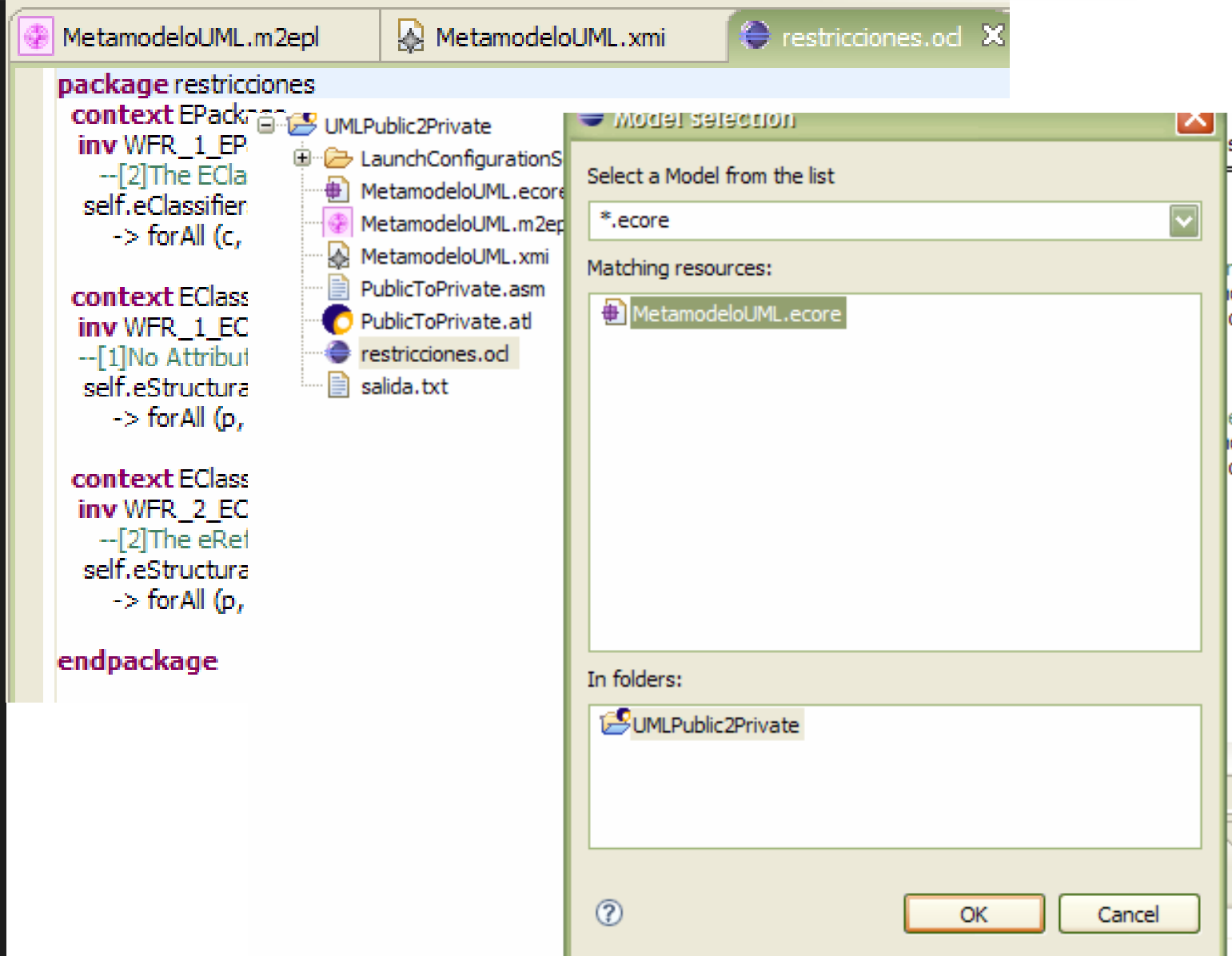
Evaluación de restricciones OCL

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



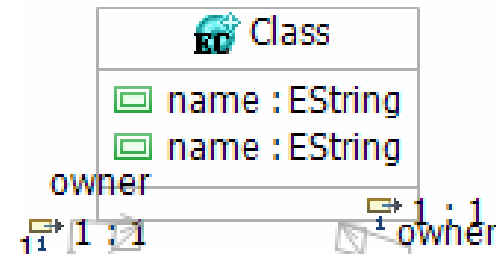
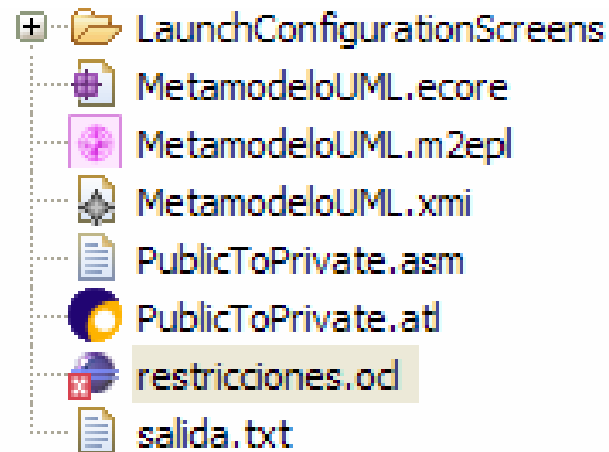
Evaluación de restricciones OCL

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

Conclusiones
Trabajos
futuros



Description	Resource	Path	Location
Errors (1 item)			
[1]No Attributes may have same name within an EClass	restricciones.odl	UMLPublic2Private	MetamodeloUML.ecore :: Class



Conclusiones

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

 *Conclusiones*
Trabajos
futuros

En este trabajo se estudiaron las falencias en el soporte automático para el desarrollo de software guiado por modelos.

- No proveen ayuda gráfica para la edición de metamodelos
- No garantiza que los metamodelos estén bien formados.
- No proveen ayuda para la instanciación de modelos a partir de metamodelos definidos



Conclusiones

*Introducción
MDD
Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

 *Conclusiones
Trabajos
futuros*

En este trabajo se ha desarrollado un plugin para el entorno Eclipse que:

- Permite especificar gráficamente metamodelos, brindando una forma más intuitiva para estas construcciones.
- Posee una base formal, ya que permite la edición y evaluación de restricciones definidas en OCL.
- Permite la instanciación de un modelo a partir del metamodelo definido. Esto es de gran ayuda, ya que da lugar a un marco de trabajo mucho más amigable para los desarrolladores.



Conclusiones

Introducción
MDD
Objetivos

Transf. Ej.
Arq. 4 capas
OCL

Tools
Atl
Mofscript
Aporte tesis

 *Conclusiones*
Trabajos
futuros

- El plugin puede incorporarse a distintas herramientas de transformación (en particular, fue probada con ATL y MofScript)

Trabajo futuro

*Introducción
MDD
Objetivos*

*Transf. Ej.
Arq. 4 capas
OCL*

*Tools
Atl
Mofscript
Aporte tesis*

*Conclusiones
Trabajos
futuros*

- Completar el prototipo presentado para permitir la transformación de modelos con el lenguaje estándar para transformaciones de modelos: QVT. Para esto será necesario implementar un nuevo plugin que permita definir la transformación en QVT. Al utilizar el lenguaje estándar, será posible la reutilización de la transformación y de los modelos.
- Permitir la evaluación de propiedades a nivel modelo. Para ello es necesario enriquecer al evaluador para que tenga la capacidad de evaluar propiedades en el nivel de metamodelo o M1, definidas junto con los metamodelos.



Preguntas

