	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

1. Bridging UML profiles and Domain Specific Languages

1.1. Abstract


With UML, it is possible to describe domain specific concepts using Profiles. In the DSL approach, a new metamodel is created for each domain-specific language. This paper identifies more precisely the relation between UML profiles and metamodels and presents a tool to bridge these two approaches. This tool is implemented using ATL (Atlas Transformation Language) and AMW (ATLAS Model Weaver).

1.2. Introduction

Model engineering is in continuous development. Nowadays basically two approaches appear and evolve. The first is the UML approach. It uses profiles to model domain specific concepts. A UML profile is an extension mechanism of UML. The goal is to express the semantics of systems or applications which are not supported by UML elements. It allows designers to customize UML to their particular domain or purpose. The second approach is based on DSL (Domain Specific Languages). It aims at representing each domain with a specific metamodel. We need interoperability between these approaches: to produce an UML profiled model from a model conforming to a metamodel and vice versa. In this document we will present first the specification of a tool, which performs automatic transformations between UML models that were defined using a profile and models conforming to a metamodel (the profile model and the metamodel describe the same system). Then, we will describe our solution for a prototype of this tool that we have implemented.

1.3. Metamodels

A metamodel is a language allowing describing a field of an activity. It contains a set of concepts, relations between these concepts and constraints. Metamodels describe the structure of models. A model conforms to a metamodel. The MOF (MetaObject Facility) describes a generic and universal language. It is used for designing metamodels; it is thus a metamodel. It should be noticed that the MOF auto-defines itself. Figure 1 shows the hierarchy of models with MOF on top of the hierarchy.

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

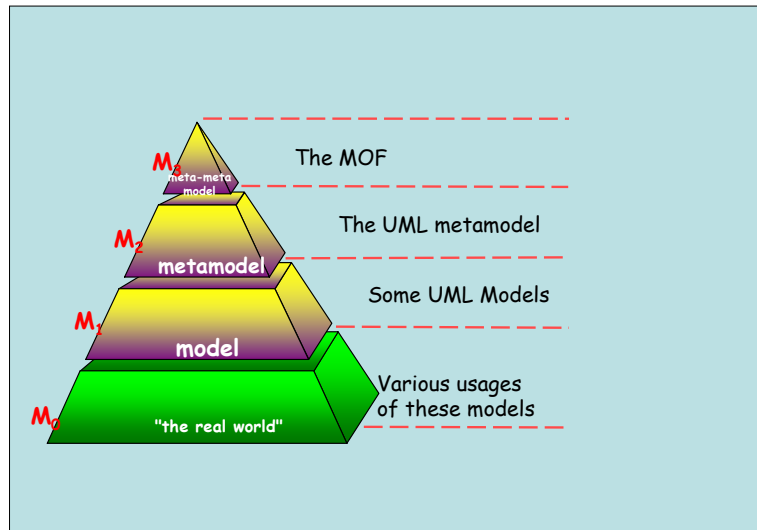


Figure 1

1.4. UML Profiles


UML profiles are dedicated to the strategic intention to formalize and support the development of applications with UML. When UML is unable to represent a system or an application in a convenient way, an UML profile may give additional mechanisms to do it. It defines virtual UML subclasses by associating stereotypes, tag definitions and constraints to provide an additional meaning to UML classes.

A stereotype provides a way of defining virtual subclasses of UML metaclasses with additional semantics. It may define tags with additional properties as well as additional constraints over its base metamodel class. The actual properties of individual model elements are specified using tagged values.

A constraint is a semantic restriction represented as a text expression. They can be specified using OCL (Object Constraint Language).

Table 1 presents a part of the profile for MOF, an UML profile that is used as an example throughout the paper.

MOF Element	UML Element
Package	Model or Package, both with <<metamodel>> stereotype
Association	Association
Exception	Exception or Class with <<exception>> stereotype
Attribute	Attribute

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

Class	Class
Constant	DataValue
Constraint	Constraint
Import	Dependency with <<import>> or <<clustering>> stereotype
PrimitiveType	DataType
Reference	Attribute with <<reference>> stereotype or AssociationEnd (if implicitReferences is set to true)

Table 1

Each profile element maps to a specific MOF element as shown in the table above. This mapping table is completed by mapping details for each element. They define more precisely each map between UML and MOF elements. Each mapping detail contains subsections covering these topics: tags, mapping properties, constraints, and limitations.

Tags are used for MOF properties not directly supported by UML. However, some MOF details cannot be rendered in UML using this profile. Those details are described as limitations. The tables below show an example of mapping details of the UML Model to the MOF Package.

Tags:


Tag	Value
org.omg.uml2mof.hasImplicitReferences	true (default) or false

Table 2

Feature Map:

MOF Feature	UML
container	namespace or null if the namespace is either null or not mapped to a MOF Package
contents	ownedElement, taggedValue
isRoot	isRoot
isLeaf	isLeaf
isAbstract	isAbstract
visibility	always set to public_vis
supertypes	other packages on supplier end of UML dependencies stereotyped as <<subtyping>> that binds them to this package.

Table 3

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

Limitations:

The order of elements is not fully preserved when rendered using the profile as UML since it has separate associations for ownedElement and taggedValue.

Table 4

UML Model Constraints:

UML Model/Package representing a nested MOF Package must not have a tag org.omg.uml2mof.hasImplicitReferences.


Table 5

1.5. Specification

1.5.1. Requirements

The tool takes as input an UML profile as defined above and the metamodel of the system that the profile describes. It allows transforming between models conforming to those inputs. On one side, transforming UML models designed with this profile to models conforming to the profiled metamodel. On the other side, transforming models conforming to the metamodel in input to UML models conforming to the profile in question. Thus, the tool aims at automatically transformations between models and not transforming a metamodel to a profile or the opposite. So, it already requires the load of the metamodel and the UML profile which can be created manually or obtained using another means.

Figure 2 below shows a use case of the tool. MMA is a metamodel for which a profile already exists. Ma.uml is an UML model designed using the profile for MMA. The tool will create Ma. Ma is the model conforming to MMA and corresponds to Ma.uml. It also will be able to do the opposite transformation; creates Ma.uml from Ma.

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

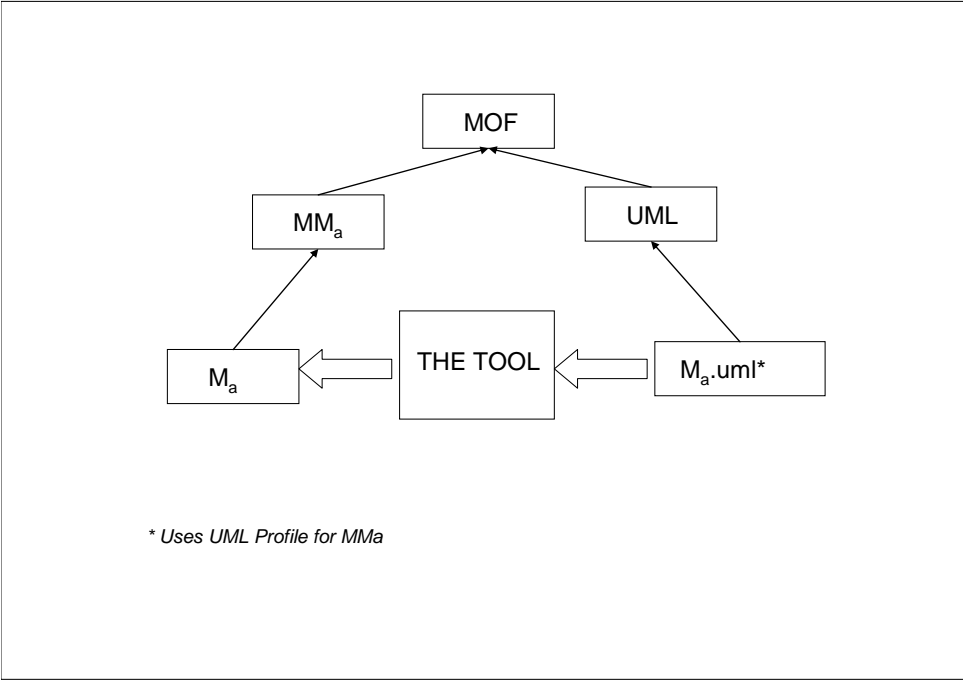


Figure 2


1.5.2. Mappings details

Mappings details specify links between UML elements and the elements from the profiled metamodel. This information should not be redundant even if the transformation is bidirectional. This requires finding a way to represent it separately of the two models in the input of the tool (the profile model and the profiled metamodel). A separated representation of mapping links, as well as it removes redundancy, facilitates the implementation of a bidirectional transformation because it can be obtained from the same representation.

Modelling mappings details require also some structures. We need to express conditional mapping, design new types, define OCL expressions, and navigate both elements of profile and profiled metamodel. We illustrate some examples below. Tables below show the mappings details for the MOF AssociationEnd which is profiled by an UML AssociationEnd.

Feature Map:

MOF Feature	UML
type	participant
multiplicity	lower and upper are determined by multiplicity.range, isOrdered is mapped to ordering (where true corresponds to ordered) and isUnique maps always to true
aggregation	aggregation (UML aggregate matches MOF shared)
isNavigable	isNavigable

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

isChangeable	changeability (changeable maps to true)
--------------	---

Table 6

Constraints:

An association must have exactly two ends. UML changeability must be either changeable or frozen. Multiplicity must have a single range.
--

Table 7

The attribute "multiplicity" is an example of the construction of a new type since it does not correspond exactly to an UML element. It is designed from several different elements as shown in Figure 3 below. The attribute "isOrdered" is an example of a conditional mapping.

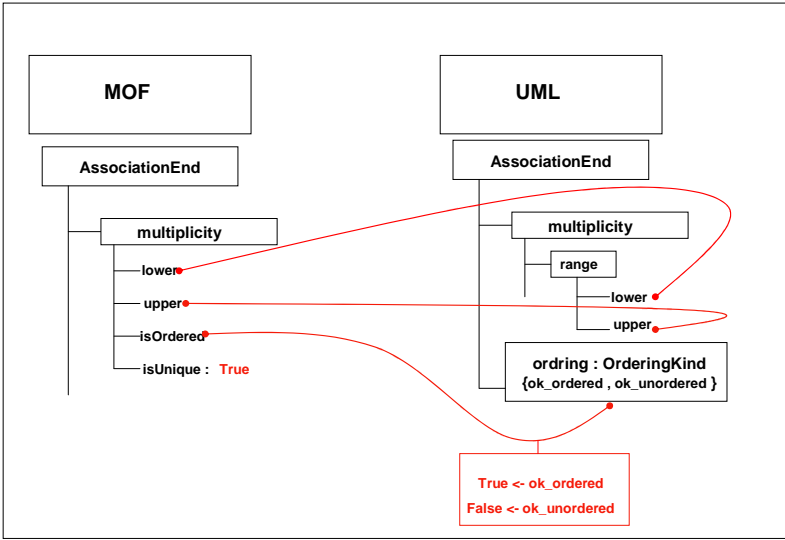



Figure 3

Table 7 shows an example of constraints which can be described with OCL expressions.

1.6. A solution using ATL (Atlas Transformation Language) & AMW (ATLAS Model Weaver)

We decided to implement the tool using ATL transformations and AMW (ATLAS Model Weaver). The AMW is a tool for representing correspondences between models. The correspondences are stored in

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

a model, called weaving model. It is created conforming to a weaving metamodel. This way allows us to generate two transformations as shown in Figure 4 below. The first, from UML models to models conforming to the metamodel profiled. The second, from models conforming to the metamodel profiled to UML models.

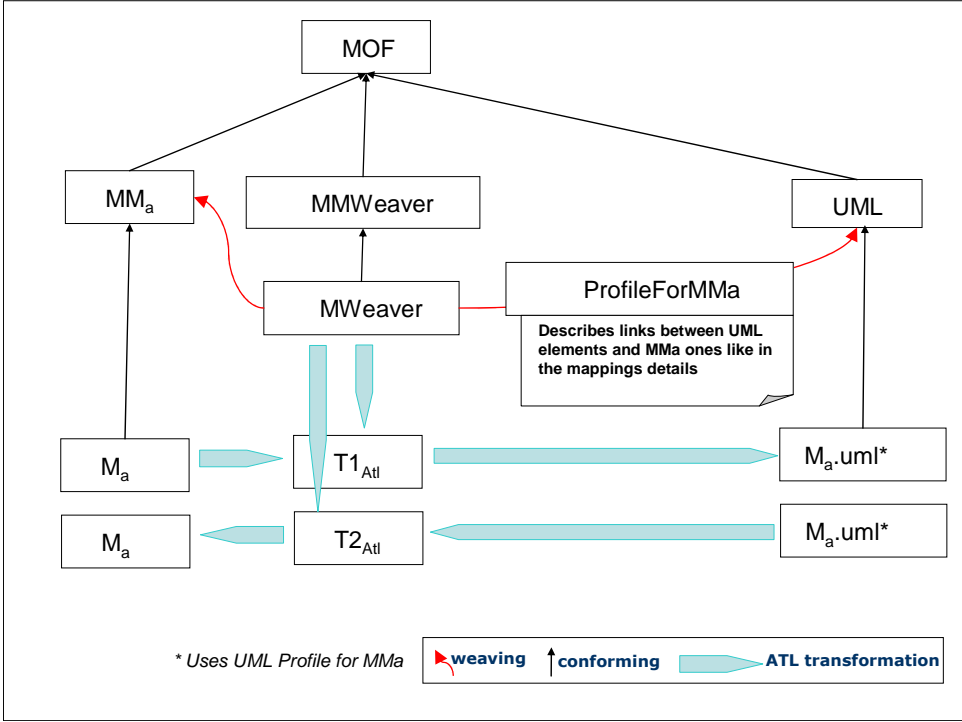



Figure 4

MMa is a metamodel for which a profile already exists (ProfileForMMa). Ma.uml is an UML model designed using the profile for MMA. MMWeaver is a metamodel that extends the base weaving metamodel to add other mapping semantics. This way there is the possibility of creating specific mapping details requirements. MWeaver is a weaving model conforming to MMWeaver. It represents links between MMA elements and UML ones conforming to the ProfileForMMA. Using two ATL transformations the tool generates automatically transformations $T1_{Atl}$ and $T2_{ATL}$. $T1_{Atl}$ transforms models conforming to the metamodel MMA to UML models and $T2_{Atl}$ transforms UML models to models conforming to MMA.

As shown in the figure below, we can use the UML metamodel to specify every profile, however, it presents some disadvantages:

- The UML modelling tools use several versions of UML (UML2.0, UML1.4 ...) which are not necessary compatible.
- Model Weaver is only compatible with Ecore models.
- We have to put all the UML metamodel on the weaving tool to describe each profile, whereas we will just use some of its elements each time.

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

For those reasons we decided to conceive a profile metamodel. This way there is the possibility to specify each profile separately as a model with the require elements. We also resolve the problem of UML versions thanks to the profile metamodel. The new configuration of the tool will be like in Figure 5 below.

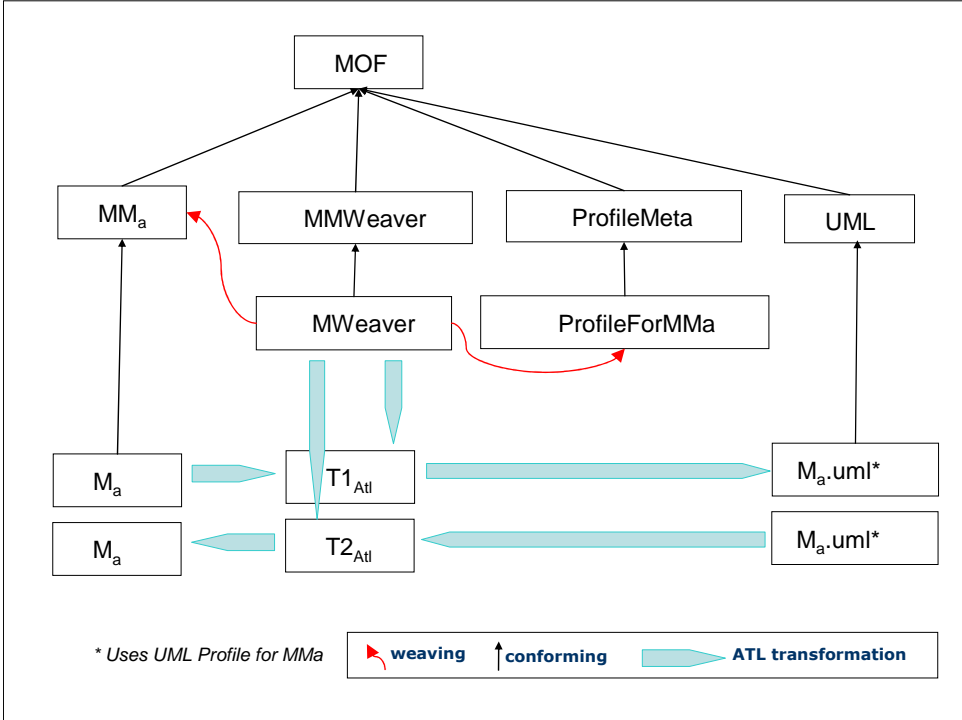


Figure 5

1.6.1. The profile metamodel

To define a Profile metamodel we have to take care of some points:

- The definition of each virtual UML subclass, which corresponds to a definition of a Profile element (in particular a class of the metamodel profiled), can be done in several ways. In fact, each element of the profiled metamodel may be represented with more than one UML element. In our example (Profile for MOF), a Package can be represented by an UML Model or an UML Package.
- An UML element can take more than one stereotype to define the same element of the metamodel profiled. However, it can also take different stereotypes to define different elements.
- We should have the possibility to navigate the UML elements used by the Profile and have access to some of their attributes and values.

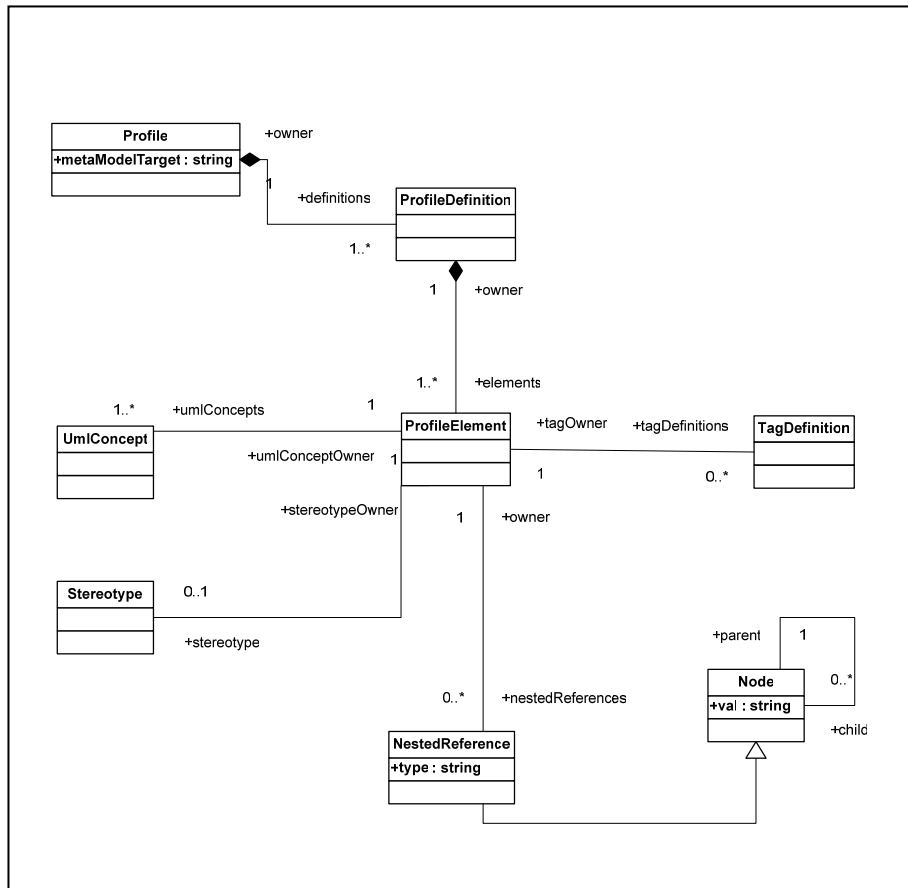

Figure 6


Figure 6 shows the profile metamodel. A ProfileDefinition corresponds to the definition of a metamodel element with the profile. As we said, this definition can be made by different ways, so each way is described by a ProfileElement. A ProfileElement contains one or more UmlConcept which specify the UML elements associated to the profile element, a stereotype, and the NestedReferences that we will need to define mapping links. Nodes will allow the possibility to navigate the UML elements.

With TCS (Textual Concrete Syntax) we define a textual syntax to define profiles. Below, an example shows a part of the textual file defining the profile for MOF. The tables Table 8 and Table 9 explain the example.

```
profile ProfileForMOF : MOF {
```

```
.....
```

```
define AssociationEnd {  
UML!AssociationEnd {
```

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

```

tag documentation ;
  -- AssociationEnd.annotation
  }
nested{
  ref name;
  ref namespace;
  ref constraint;
  ref participant;
  ref multiplicity {range { lower , upper } };
  ref ordering : literal {ok_ordered (true) , ok_unordered (false)} ;

  -- isOrdered is mapped to ordering (where true corresponds to ordered)

  ref isUnique : boolean = true;
  ref aggregation : literal {ak_composite , ak_aggregate ,ak_none};
  ref isNavigable;
  ref changeability : literal {ck_changeagle (true), ck_frozen(false) ,
                                ck_addOnly(false)};
  }
}
.....
}

```

Tag	Value
<i>documentation</i>	<i>any string</i>

Table 8


MOF Feature	UML
<i>name</i>	<i>ModelElement</i>
<i>annotation</i>	<i>Body of the first comment in ModelElement.comment; value of documentation</i>
<i>container</i>	<i>ModelElement.namespace</i>
<i>constraints</i>	<i>ModelElement.constraint</i>

Table 9

1.6.2. Implementation of mapping links with AMW

We have the possibility to define links and correspondences with AMW. We use it to define our mapping links.

The weaving model takes the profile model as left input and the profiled metamodel as right one. We have defined five kinds of correspondences:

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

- **LinkDef:** it defines a link between the ProfileDefinition and the metamodel element refers to. It has a left element, a right element and childrens which can be LinkElements.
- **LinkElem:** it defines a link between the ProfileElement and the metamodel element refers to. It has a left element, a right element and children which can be of type: Nested, IfNested or New type.
- **Nested:** it defines direct mapping between UML elements attributes and metamodel elements ones. It has just a left element and a right one.
- **IfNested:** it's a kind of mapping table. We use it when a metamodel element attributes don't corresponds exactly to UML element ones. It has a left element, a right element and Nested links, IfNested links and NewType links which are needed to define mappings.
- **NewType:** We use this kind of correspondence to define a metamodel element with UML attributes which appertain to different UML elements. It has a right element and Nested links, IfNested links and NewType links which are needed to define mappings.

Figure 7 below shows a screen shot of a part of the weaving model for MOF.

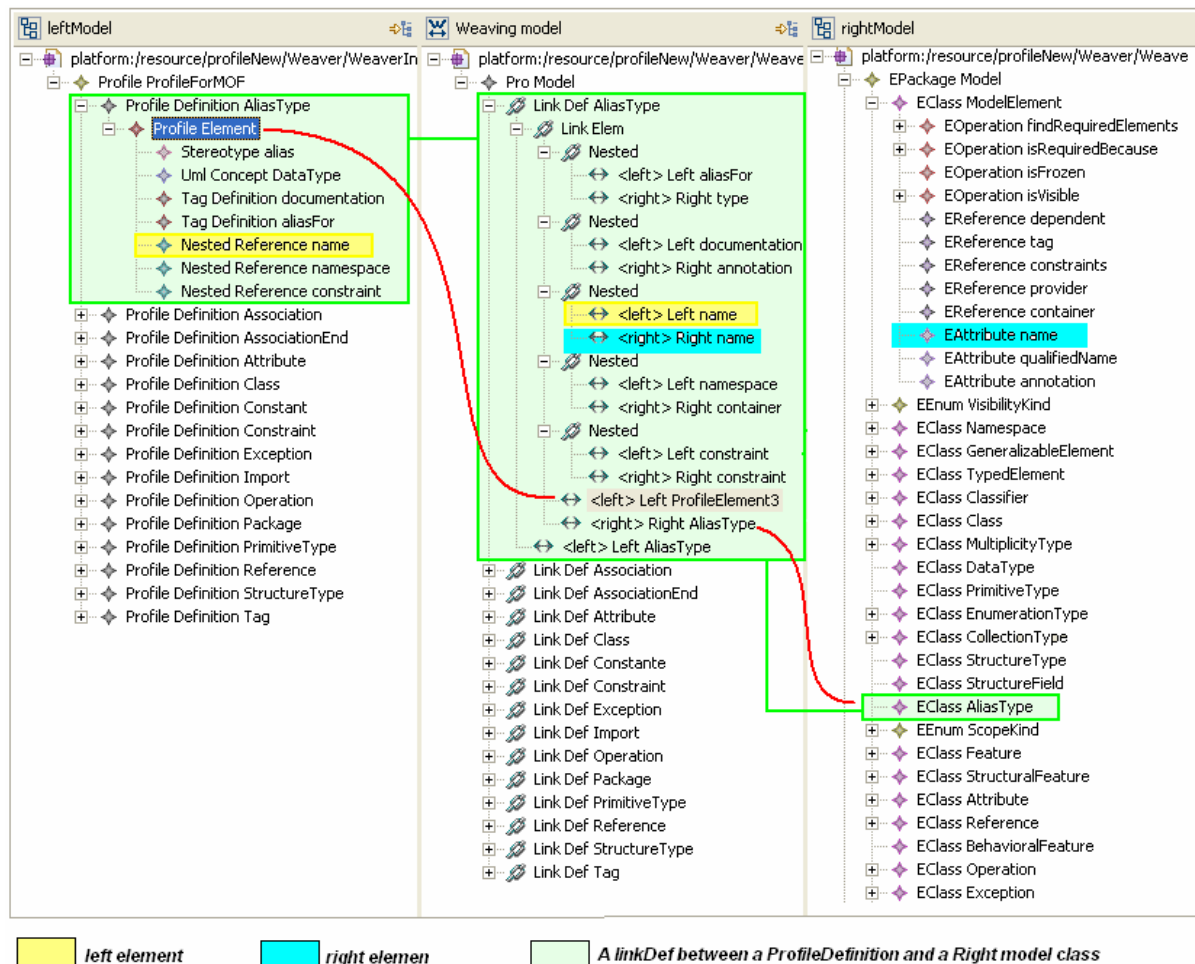
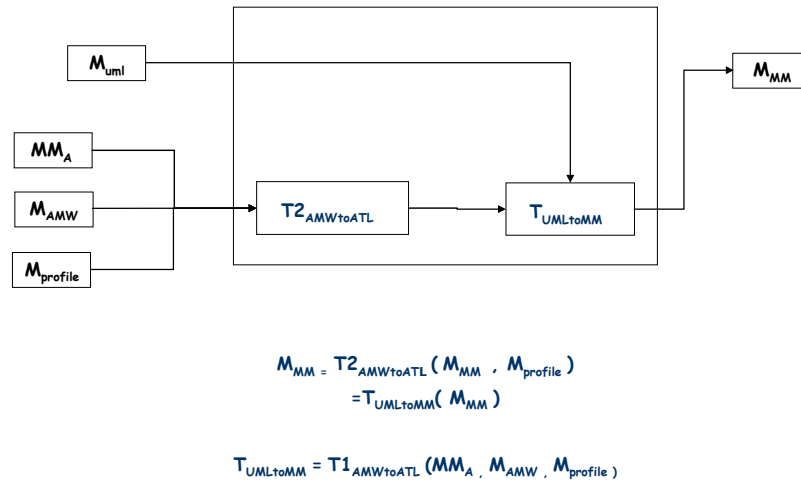
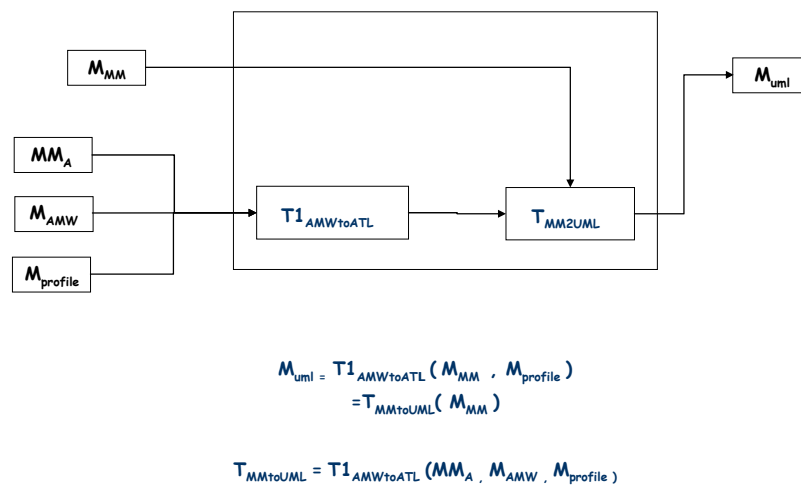



Figure 7

1.6.3. Transformations

The tool uses two ATL transformations $T1_{AMWtoATL}$ and $T2_{AMWtoATL}$ to automatically generate the two resulted ATL models. As shown in the figures Figure 8 and Figure 9, $T1_{AMWtoATL}$ and $T2_{AMWtoATL}$ take as input the profiled metamodel, the profile model and the weaving model. The first output transformation is $T_{MMtoUML}$ which transforms UML profiled models to a model conforming to the profiled metamodel. The second transformation output is $T_{UMLtoMM}$ which transforms models conforming to the profiled metamodel to UML profiled models.


Figure 8

Figure 9

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

Rules Specification:

These are some of the major rules of the transformations:


- The ATL model has to be created from the AMW model:
 - Its inModel has to be created from the left model of the weaving for $T1_{AMWtoATL}$. Its outModel has to be created from the right one and vice versa for $T2_{AMWtoATL}$.
- For each ProfileElement, a MachedRule has to be created:
 - The inPattern has to contain the right or the left element of the LinkElem according to the transformation we are implementing ($T1_{AMWtoATL}$ or $T2_{AMWtoATL}$).
 - The outPattern has to contain, in addition of the right or the left element of the LinkElem, other outPatterns created for the MachedRule: from NewType links, TagDefinitions ...
 - The bindings has to be created from all the children of the LinkElem refers to the ProfileDefinition which outputs the MatchedRule: Nested links, IfNested links, NewType links ...

1.6.4. Limitations

The tool implemented at the moment is still a prototype. Some functionalities are not implemented yet. These are some of them:

For $T1_{AMWtoATL}$:

- A ProfileElement can have more then one UML concept. We have to choose the good one when we transform a metamodel element to an UML one. In the profile for MOF for example, UML Package and Model are both mapped to MOF Package. However, in the inverse direction a MOF Package must be mapped only to UML Package.
- A weaving link can have several left elements and one right. In this transformation we have to filter those elements and affect good ones. Let's take the example of the UML Class which maps to MOF Class on the profile for MOF. The attribute contents of the MOF Class maps to ownedElement followed by feature (in order) and taggedValue of the UML Class. The Nested link which expresses this mapping has three left elements and one right. For $T2_{AMWtoATL}$, we have to select appropriate elements from the attribute contents of the MOF Class which have to be affected to each attribute from ownedElement, feature, and taggedValue of the UML Class.
- If the right metamodel of the weaving doesn't have a root element, the ATL result model of the transformation will not have a rule to create a global UML Model. Every UML model must have a Model element which contains all the Packages; it's like a root of the UML model. However, we need a metamodel element which can be linked with this UML element. When the metamodel doesn't have one, we should add it manually at the result UML model (see the user guide).

	Bridging UML profiles and Domain Specific Languages	ABOUZAHRA Anas
	A tool documentation	Date 28/07/2005

For T2_{AMWtoATL}:

- When the UML attribute is a sequence or a set, the weaving model doesn't specify it. For example, in the profile for MOF, the attribute multiplicity.range of an UML AssociationEnd is a set. For the mapping, we need the attribute upper of it to map it to multiplicity.upper of a MOF AssociationEnd. The problem is that the profile model doesn't specify that this attribute is a set. So, in the result transformation we find :

```

rule AssociationEnd2AssociationEnd {
  from
    uml : UML!AssociationEnd
  to
    targ : MOF!AssociationEnd (
      .....
      multiplicity <- _multiplicity
    ),
    _multiplicity : MOF!MultiplicityType (
      upper <- uml.multiplicity.range.upper,
      lower <- uml.multiplicity.range.lower,
      .....
    )
}

```

- Our implementation allows affecting one stereotype by ProfileElement, whereas an UML element can have more.