UML Profile for MOF

Author:
Martin Matula, NetBeans / Sun Microsystems

History:
Available in the CVS

Introduction

This document describes UML Profile for MOF supported by **UML2MOF conversion tool**. The profile defined here is based on OMG's UML Profile for MOF described in chapter 6 of **UML Profile for EDOC** standard (OMG document ad/01-08-19). A separate profile was created (i.e. the standard could not be fully reused) for the following reasons:

- The standard defines mapping from UML to MOF 1.3. Mapping from UML to MOF 1.4 was needed.
- The standard uses several UML elements that are not supported by known UML tools.
- There are several bugs in the standard.

For convenience, differences between OMG's standard UML Profile for MOF and profile defined herein are discussed at the end of each sub-section.

UML Representation of MOF Elements

Below is a simple table showing how each MOF element maps to a corresponding UML element in a profile (if there are several variants, they are separated by a semicolon). Click on a MOF element to go to a detailed description of the mapping.

MOF Element	UML Element
AliasType	DataType with <<alias>> stereotype
Association	Association
AssociationEnd	AssociationEnd
Attribute	Attribute
Class	Class
CollectionType	Class with <<collection>> stereotype
Constant	DataValue
Constraint	Constraint
EnumerationType	Enumeration; Class or DataType with <<enumeration>> stereotype
Exception	Exception; Class with <<exception>> stereotype
Import	Dependency with <<import>> or <<clustering>> stereotype
Operation	Operation
Parameter (Operation)	Parameter
Parameter (Exception)	Attribute
Package	Model or Package, both with <<metamodel>> stereotype
PrimitiveType	DataType
Reference	Attribute with <<reference>> stereotype; AssociationEnd (if implicitReferences is set to true)
StructureField	Attribute
StructureType	Class with <<structure>> stereotype
Tag	TaggedValue

Detailed Mapping

This sections describes mapping of all elements and their features in detail. The mapping decsribed herein is bidirectional, although it is defined from MOF-to-UML perspective. Detailed description of mapping for each of the elemenets contains of the following subsections:

- **Tags** - defines extra tags that are added to UML model in order to be able to represent values of some MOF attributes.
- **Feature Map** - table describing mapping between the particular features of a MOF element and UML elements.
- **Limitations** - names limitations of the MOF-to-UML mapping (if certain MOF constructs cannot be expressed in UML using this profile)
- **UML Model Constraints** - lists constraints to be preserved in UML model in order to be compliant with this profile (i.e. to be mappable to MOF)
- **OMG Profile** - describes differences (if any) between this profile and the standard MOF Profile for UML defined by OMG.

Here are the basic mapping rules shared by all the elements:

Tags

Tag	Value
documentation	any string

Feature Map

MOF Feature	UML
name	ModelElement.name
annotation	body of the first comment in ModelElement.comment; value of documentation tag
container	ModelElement.namespace
constraints	ModelElement.constraint

UML Model Constraints

Every UML ModelElement that maps to MOF ModelElement must have a name.

OMG Profile

Lacking support for ModelElement.comment - mapping of annotation attribute fully relies on the documetation tag. However most of the UML tools store documentation for model elements in instances of Comment classes.

Detailed mapping of other features specific to each MOF class follows.

AliasType

AliasType is represented by an instance of UML DataType with <<alias>> stereotype.

Tags

Tag	Value
org.netbeans.uml2mof.aliasFor	fully qualified name (with individual simple names separated by ".") of a classifier from the same model

Feature Map

MOF Feature	UML
type	classifier that the fully qualified name in value of org.netbeans.uml2mof.aliasFor tag resolves to

OMG Profile

Lacks definition of AliasType mapping as the standard profile is based on MOF 1.3.

Association

MOF Association maps directly to UML Association.
UML Association stereotyped as <<implicit>> is ignored by the profile and is not mapped to a MOF Association.

Tags

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

Feature Map

MOF Feature	UML
isRoot	isRoot

isLeaf	isLeaf
isAbstract	isAbstract
visibility	always set to `public_vis`
supertypes	generalization.parent
contents	ownedElement, connection, taggedValue
isDerived	value of org.omg.uml2mof.isDerived tag

Limitations

The order of contents reference are not fully preserved because UML has a separate associations for ownedElement and connection.

UML Model Constraints

An Association must have exactly two ends.

OMG Profile

There is no way how to map derived associations (isDerived tag is missing).

AssociationEnd

MOF AssociationEnd is directly mapped to UML AssociationEnd.

Tags

Tag	Value

Feature Map

MOF Feature	UML
type	participant
multiplicity	lower and upper is 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
isChangeable	changeability (`changeable maps to true`)

UML Model Constraints

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

OMG Profile

There is a bug in the profile which states that MOF feature "type" is mapped to UML feature "type". However UML AssociationEnd does not have a reference called "type" - it uses "participant" reference for this purpose.

Attribute

MOF Attribute is straightforwardly mapped to UML Attribute.

Tags

Tag	Value
org.omg.uml2mof.isDerived	`true` or `false` (default)
org.omg.uml2mof.isUnique	`true` or `false` (default)

Feature Map

MOF Feature	UML
multiplicity	lower and upper are determined by multiplicity.range, isOrdered is mapped to ordering (where `true` corresponds to `ordered`) and isUnique maps to org.omg.uml2mof.isUnique tag value.
isChangeable	changeability (`changeable` maps to `true`, otherwise `false`)
isDerived	value of org.omg.uml2mof.isDerived tag

Class

MOF Class maps to UML Class with no stereotypes.

Tags

Tag	Value
org.omg.uml2mof.isSingleton	`true` or `false`

Feature Map

MOF Feature	UML
isRoot	isRoot
isLeaf	isLeaf
isAbstract	isAbstract
visibility	always set to `public_vis`
supertypes	generalization.parent
singleton	value of org.omg.uml2mof.isSingleton tag (`false` by default)
contents	ownedElement followed by feature (in order) and taggedValue

Limitations

The order of contents reference is not fully preserved because UML has separate associations for features and other owned elements.

CollectionType

MOF CollectionType is represented in UML as a class with <<collection>> stereotype. Such class must contain attribute named `items` that determines the element type and multiplicity of the collection.

Feature Map

MOF Feature	UML
multiplicity	multiplicity of `items` attribute
type	type of `items` attribute

UML Model Constraints

Class representing a collection must contain an attribute named `items`.

OMG Profile

Lacks definition of CollectionType mapping as the standard profile is based on MOF 1.3.

Constant

TBD

Constraint

TBD

EnumerationType

MOF Enumerations can be represented by either UML Enumerations or classes with <<enumeration>> stereotype.

Feature Map

MOF Feature	UML
labels	list of names of literals contained by a corresponding UML enumeration or names of attributes if the enumeration is represented by UML class (attribute types, multiplicities, etc. are ignored)

OMG Profile

Lacks definition of EnumerationType mapping as the standard profile is based on MOF 1.3.

Exception

MOF Exception is maps to UML Class with <<exception>> stereotype.

Feature Map

MOF Feature	UML
contents	feature, ownedElement, taggedValue
scope	always `classifier_level`

visibility	always maps to `public_vis`

OMG Profile

Maps MOF Exception to UML Exception. UML Exception is however not supported in Class diagrams by most of the known UML tools.

Import

MOF Import is represented by UML dependency stereotyped as <<import>> or <<clustering>>.

Feature Map

MOF Feature	UML
container	client
importedNamespace	supplier
isClustered	`true` if this dependency is stereotyped as <<clustering>>, `false` if stereotyped as <<import>>
visibility	always `true`

UML Model Constraints

Element on client end of the dependency must be a UML Package/Model mapped to a MOF Package.

OMG Profile

Maps MOF Import to UML ElementImport that is not supported by most of the known tools.

Operation

MOF Operation directly maps to UML Operation.

Tags

Tag	Value
org.netbeans.uml2mof.raisedExceptions	Multiple values or one value containing comma-separated list of fully quialified names of classes corresponding to MOF Exceptions

Feature Map

MOF Feature	UML
contents	parameter, taggedValue
isQuery	isQuery
exceptions	Exceptions corresponding to the fully qualified names in values of org.netbeans.uml2mof.raisedExceptions tag
scope	Feature.ownerScope
visibility	always maps to `public_vis`

See **Feature mapping** for mapping of inherited features.

OMG Profile

Raised exceptions are supposed to be linked via raisedSignal reference (with no need for a separate tag for them). This reference is however not supported by most of the tools.

Parameter

MOF Parameter contained by an Operation is mapped to UML Parameter. MOF Parameter contained by an Exception is mapped to UML Attribute.

Tags

Tag	Value
org.omg.uml2mof.multiplicity	upper and lower multiplicity bound of the paramter values separated by two dots ("..") - e.g. 1..1 (default) or 0..* (zero to many)
org.omg.uml2mof.isOrdered	`true` or `false` (default)
org.omg.uml2mof.isUnique	`true` or `false` (default)

Feature Map

MOF Feature	UML
direction	kind for operation parameters, `out_dir` for exception parameters
multiplicity	lower and upper are mapped to multiplicity.range, however if the org.omg.uml2mof.multiplicity tag is present, it takes the precedence, isOrdered is mapped to ordering (for Exception parameters) or value of org.omg.uml2mof.isOrdered tag (for Operation parameters), isUnique is mapped to the value of org.omg.uml2mof.isUnique tag
type	type

OMG Profile

Introduces a new tag for multiplicity although both UML Attribute and UML Parameter have multiplicity parameter for that purpose. This tag is preserved as some UML tools do not support multiplicity on Parameters, so both the tag and the multiplicity attribute are taken into account by this profile, the tag value takes the precedence.
In case of UML Exception neither special tag for isOrdered is needed as this is provided by ordering attribute.

Package

Outermost MOF Package is mapped to UML Model or UML Package, both with <<metamodel>>, the nested MOF Package is represented by either UML Model or UML Package (i.e. the <<metamodel>> stereotype is not necessary in case of nested packages).

Tags

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

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.

Limitations

The order of contents are not fully preserved when rendered using the profile as UML has separate associations for ownedElement and taggedValue.

UML Model Constraints

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

OMG Profile

While the standard profile allows MOF Packages to be represented by UML Models only, UML Models are not supported by some existing UML tools. Also org.omg.uml2mof.clusteredImport tag is introduced for clustered imports declarations. This profile solves clustered imports differently (see **Import mapping**).
Package subtyping is handled by a standard generalization.parent reference which was replaced in this mapping by a stereotyped dependency as package subtyping is not supported by most of the known UML tools.

PrimitiveType

Mapping of PrimitiveTypes is based on their names. UML datatype with no stereotype and with the same name as one of the MOF primitive types in PrimitiveTypes package is mapped to that type (no matter what UML package it is in). All the other UML data types with no stereotype are mapped to instances of MOF PrimitiveType class (with a given name) residing in a MOF package corresponding to the UML package they are in.

Limitations

Modeling of primitive types distinct from the MOF standard primitive types (defined in PrimitiveTypes package) but with the same name is not supported by this profile.

OMG Profile

Lacks definition of PrimitiveType mapping as the standard profile is based on MOF 1.3.

Reference

MOF Reference can be represented by UML Attribute with <<reference>> stereotype.
Also, if the UML package corresponding to MOF outermost package does not have the value of tag org.omg.uml2mof.hasImplicitReferences set to `false`, then a MOF Reference is implied by each eligible UML AssociationEnd. An end is considered eligible if it is navigable, there is no explicit MOF Reference for that end within the same outermost package and the end's association is owned by the same package that owns its opposite end's type (to not create circular package dependencies).

Tags

Tag	Value
org.omg.uml2mof.referencedEnd	fully qualified name of an association end

Feature Map

MOF Feature	UML
scope	always maps to `instance_level`
visibility	always maps to `public_vis`
refenrecedEnd	AssociationEnd corresponding to the fully qualified name in the value of org.omg.uml2mof.referencedEnd tag.

Values for the inherited attributes of multiplicity, isChangeable and type are fully infered from the corresponding association end (values of these attributes for the corresponding UML Reference are ignored).
In case of implicit references, name and annotation are also taken from the association end.

StructureField

StructureField is represented in UML as an attribute of a UML class representing StructureType.

Tags

None.

Feature Map

MOF Feature	UML
type	type

Please note that StructureField does not have a multiplicity attribute, thus multiplicity specified on the corresponding UML attribute is ignored when mapping from UML to MOF.

OMG Profile

Lacks definition of StructureField mapping as the standard profile is based on MOF 1.3.

StructureType

StructureType is mapped to a class with <<structure>> stereotype.

Feature Map

MOF Feature	UML
contents	Namespace.ownedElement followed by Classifier.feature (in order)

OMG Profile

Lacks definition of StructureType mapping as the standard profile is based on MOF 1.3.

Tag

MOF Tag is mapped to UML TaggedValue.

Feature Map

MOF Feature	UML
tagId	name
values	value
elements	modelElement

Limitations

Tags without prefix or with org.omg.uml2mof and org.netbeans.uml2mof prefixes are ignored when mapping from UML to MOF and thus they should not be used in MOF

metamodels.