Community Home

Overview
User Zone
  Documentation
Developer Zone
  Architecture
  Documentation
  Ideas
Partner Zone
  Schools
Community
  Download
  Forums
  Mailing-lists
  Get the code

Index_modeler

Modeler
  News
  Overview
  Features
  Download
  Mailing-lists
  Model
  Screenshots
  Documentation
  Demo
  License
  Contribute

Index_mappingor

MappingOR
  News
  Overview
  Features
  Download
  Mailing-lists
  Model
  Screenshots
  Documentation
  Demo
  License
  Contribute

Index_distrib

Distrib
  News
  Overview
  Features
  Download
  Mailing-lists
  Model
  Screenshots
  Documentation
  Demo
  License
  Contribute

ATL Documentation

We use ATL to write our 2 transformations. The first transformation permit to get a XMI file conforms at the metamodel UML to integrate in other modeler. The second transformation is very similar and is used to integrate our models into AndroMDA3. We study the possibility to use AndroMDA4.

You can download our transformation and configuration : here !

To launch the transformation, you must :

- Add the input model IN with the meta-model MMUseCase
- Add the output model OUT with the meta-model UML
- Link the meta-model MMUseCase with the eCore file and indicate that we used EMF
- Link the meta-model UML with the XMI file and indicate that we used MDR.
- Set the input file and the output file.

ATL (Atlas Transformation Language) has been defined to perform general transformations within the MDA framework (Model Driven Architecture) recently proposed by the OMG.

Example of helper :

```
-- This helper permits to know if a package is the model. To
-- know if it's, we select the only package not contained by
-- other packages.
-- CONTEXT: MMUseCas!Package
helper def: firstPackage : MMUseCase!Package =
     MMUseCase!Package.allInstances()->reject(pa |
         MMUseCase!Package.allInstances()->collect(p | p.packageSet)-
        exists(ptmp | ptmp = pa))->asSequence()->first();
```

Example of rule :

```
-- Rule 'Model'
-- This rule permit to create the model. We initialize all data types
rule Model {
    from
        po : MMUseCase!Package  (
            po = thisModule.firstPackage
        )
    to
        pu : UML!Model (
            name <- po.name,
            ownedElement <- Set{po.packageSet, po.classSet, po.stereo
                po.associationSet, po.comments,datatype}
        ),
        datatype : UML!Package (
            name <- 'datatype',
            ownedElement <- Set{int,double,float,boolean,byte,
                char,short,void,date,long,string}
        ),
        int : UML!DataType(
            name <- 'int'
        ),
        double : UML!DataType(
            name <- 'double'
        ),
        float : UML!DataType(
            name <- 'float'
        ),
        boolean : UML!DataType(
            name <- 'boolean'
        ),
        byte : UML!DataType(
```

```
        ),
        char : UML!DataType(
            name <- 'char'
        ),
        short : UML!DataType(
            name <- 'short'
        ),
        void : UML!DataType(
            name <- 'void'
        ),
        date : UML!DataType (
            name <- 'Date'
        ),
        long : UML!DataType (
            name <- 'Long'
        ),
        string : UML!DataType (
            name <- 'String'
        )
}
```

Need help ?

Eclipse project : http://www.eclipse.org/m2m/atl/

Team project : http://www.sciences.univ-nantes.fr/lina/atl/