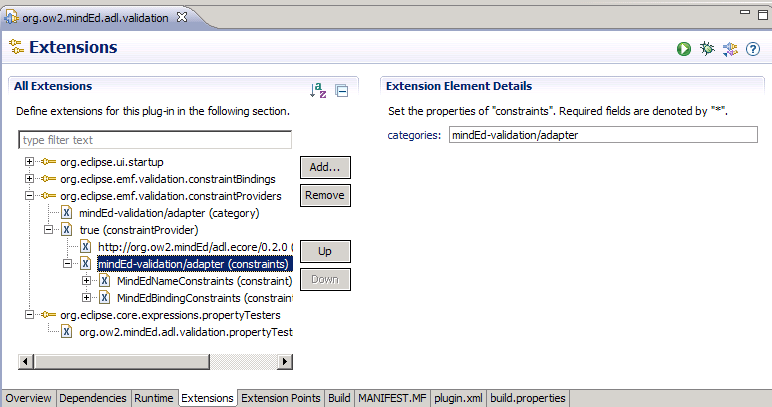
# In Open Source Code:

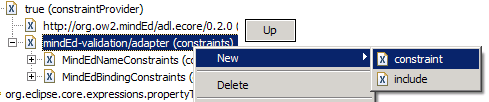
The Plug-in org.ow2.mindEd.adl.validation implements the default MinEd Constraints. We can add, remove or modify some constraints.

## Create à new constraint:

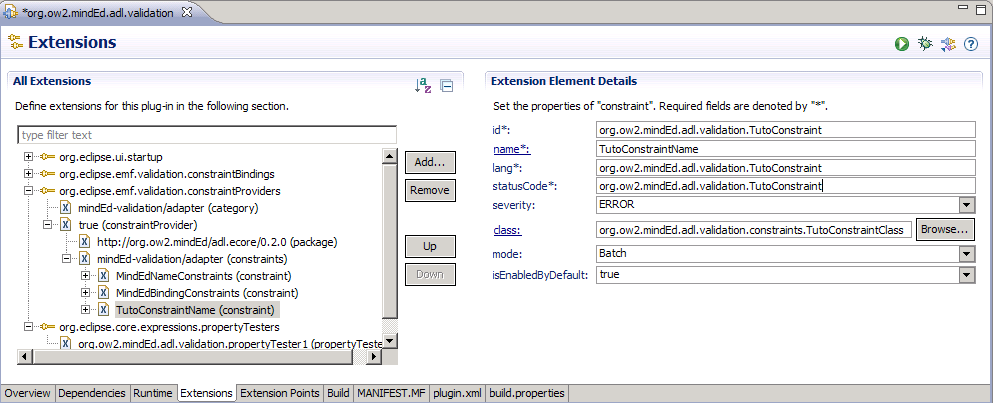
Open plugin.xml in org.ow2.mindEd.adl.validation project and select “Extensions” tab. Open “org.eclipse.emf.validation.constraintProviders” extension then ”constraintProvider” and “constraints”. The constraints “MindEdNameConstraints” and “MindEdBindingConstraints” are defaults constraints for MindEd. We can modify or remove it.



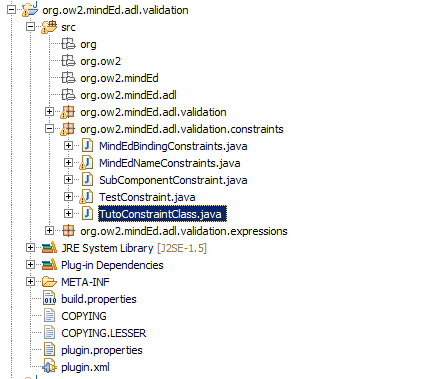
Add a new constraint with a right click on “minded-validation/adapter (constraints)”>>New>>constraint.



Fill the different channel. For this example, we create a new TutoConstraint. This constraint point on a the TutoConstraintClass which create in the next point.

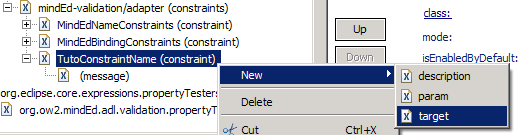


Create a new class in “or.ow2.mindEd.adl.validation.constraints” package in “or.ow2.mindEd.adl.validation” project:



And extend this class with “AbstractModelConstraint” class. In this class, you can make your own constraint. Now in the pulgin.xml file, you must select your target for this constraint.

Make a right click on TutoConstraintName in plugin.xml and select New>>target:



This target is a class from ECORE model. Choose “CompositeComponentDefinition” for example. When you validate the diagram, TutoConstraintClass will be called if they are a Composite Component. You can define more than one target and check in “TutoConstraintClass” from which target the validation is called.

In your “TutoConstraintClass”, add this code:

**public** **class** TutoConstraintClass **extends** AbstractModelConstraint{

@Override

**public** IStatus validate(IValidationContext ctx) {

// Check if validation come from the define class

**if**(ctx.getTarget().eClass().getClassifierID()==AdlPackage.*COMPOSITE\_COMPONENT\_DEFINITION*)

{

// Convert context Target in model class

CompositeComponentDefinition compositeComponent = (CompositeComponentDefinition) ctx.getTarget();

//

// Insert your test code

//

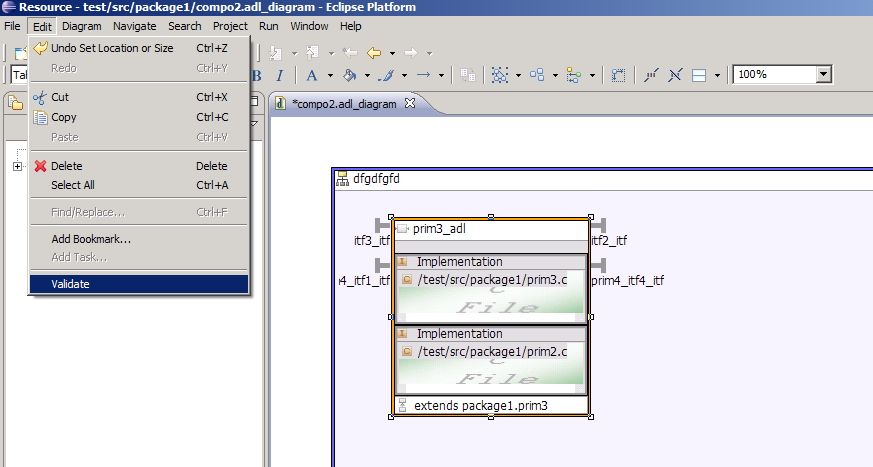
}

**return** ctx.createSuccessStatus();

}

}

Your constraint are execute when you try to validate the model. Currently, this constraint is define in BATCH mode. You must make a manual validation. In a MindEd diagram, click on Edit > Validate. This operation checks your constraint.



If you want that the constraint will be test automatically, you must set the constraint in live mode. Find your constraint in plugin.xml file and set mode in Live:

