OCL conf checker
SOFTWARE USER MANUAL

Prepared by	***Date***	***Signature***
Patricia SEGONDS – Atos **Anne HAUGOMMARD - Atos**	12/03/2021	

Accepted by	***Date***	***Signature***

Approved by	***Date***	***Signature***

TABLE OF CONTENTS

1. INTRODUCTION

1.1. OBJECTIVE OF THE DOCUMENT

This document is the OCL configuration checker User Manual.

The document describes recommendations and guidance on how to use the tool.

1.2. DOCUMENT ORGANISATION

This document is divided into main sections:

- Presentation
- Installation
- Instantiation on new configuration
- Usage

2. APPLICABLE AND REFERENCE DOCUMENTS

2.1. APPLICABLE DOCUMENTS

None

2.2. REFERENCE DOCUMENTS

Reference	Title
RD1	Object Constraint Language https://www.omg.org/spec/OCL/

3. TERMS, DEFINITIONS AND ABBREVIATED TERMS

Acronym	Meaning
EMF	Eclipse Modeling Framework
OCL	Object Constraint Language
XSD	XML Schema Definition

4. OCL CONF CHECKER - PRESENTATION

OCL Checker tool is a tool used in the development process of IMA space projects.

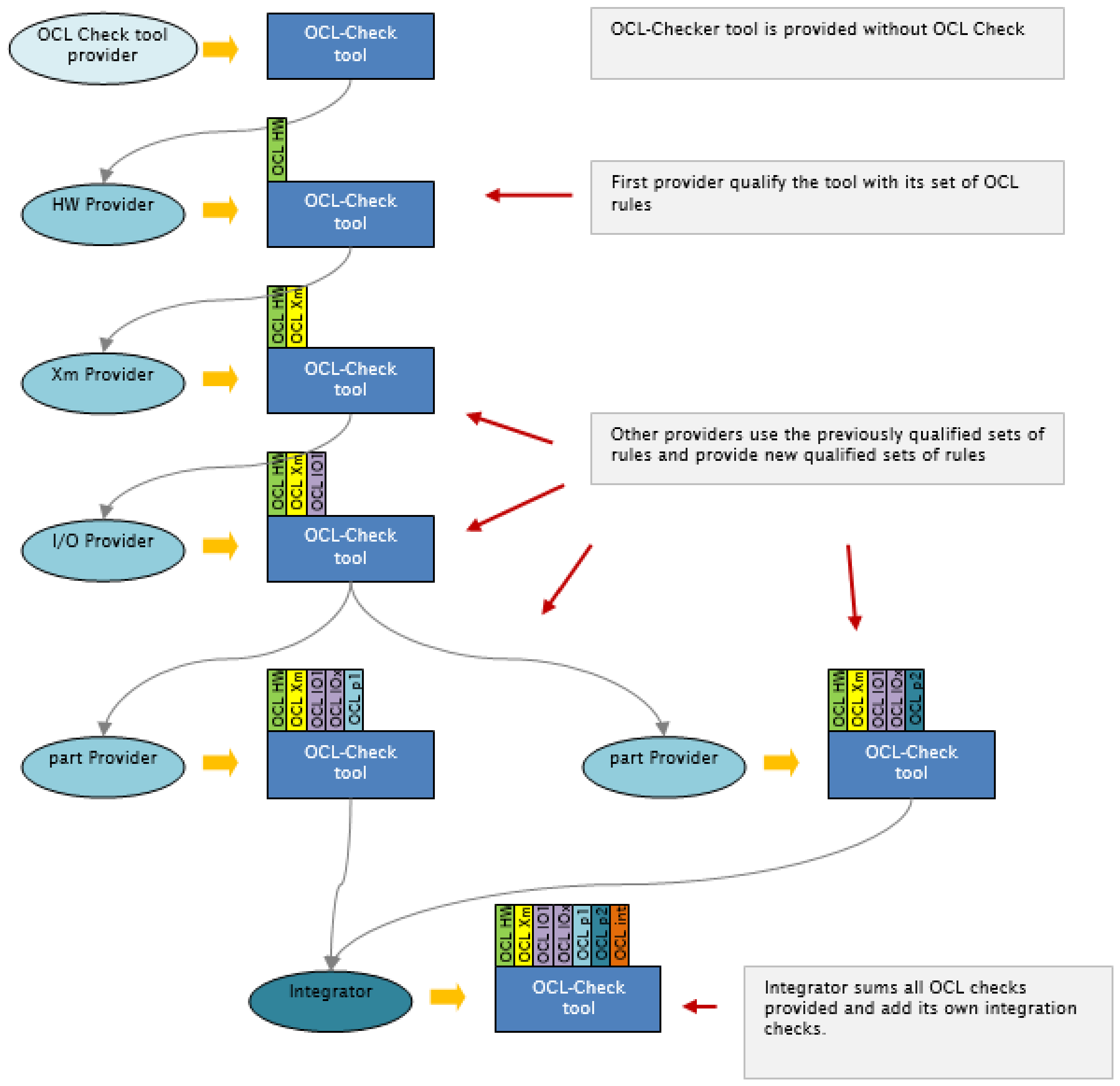
4.1. TOOL CONTEXT

The provider will deliver the tools with a configuration of checks implemented by OCL rules and XSD(s) files.

The user will use the tool in the configuration delivered by the provider and performs the checks on its configuration data.

A user can also be a provider for the next users.

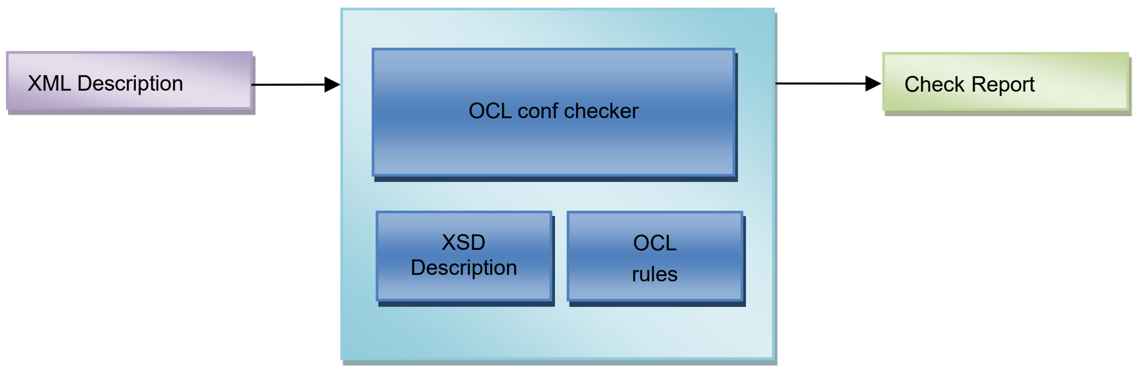
Global interactions of different user/providers are illustrated in the following diagram:



This diagram highlights an enrich rules pattern. Details of this pattern are:

1. Get the OCL Checker with OCL rules updated by the previous provider
2. Add your part of the product (a partition)
3. Check that OCL rules provided are checked
4. Define your own set of OCL rules relative to the new part
5. Qualify the new set of OCL rules
6. Deliver the new development, new configuration file and the qualified new set of OCL rules

4.2. CHECKTOOL INTERFACES



Input definition

- XML description: one or several XML files defining model elements

/!\ CAUTION /!\
➔ Only one namespace is supported, the same must be used in all XSD.
➔ Defined types must be unique (no duplication authorized), an XSD specific ‘types.xsd’ can be used.
➔ Complex types referenced in an xsd must be present in the xsd.

Output Definition

- Check Report: Result of the check

5. OCL CONF CHECKER - INSTALLATION

As a prerequisite Java 1.8 must be installed.

The tool is running on following Eclipse environment. Following steps are necessary to run it:
- Eclipse installation
- [optional] Suggestion of Eclipse plugins

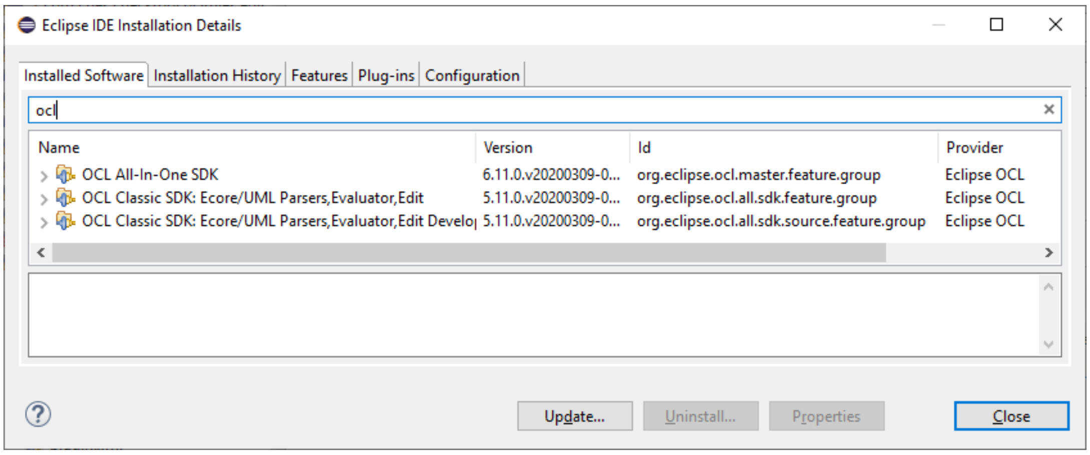
5.1. ECLIPSE INSTALLATION

Environment	Version	Source
Eclipse	2020-03-modeling-tools	Download zip file from: https://www.eclipse.org/downloads/packages/release/2020-03/r/eclipse-modeling-tools

1. Download zip file
2. Unzip it in a required application directory
3. Launch eclipse.exe

5.2. SUGGESTION OF ECLIPSE PLUGINS

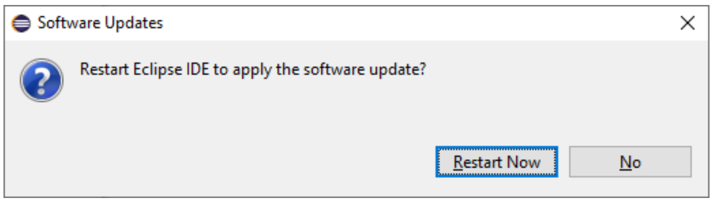
Install OCL Editor
Following elements must also be installed:
1. Go on Help > Install new software
2. From repository http://download.eclipse.org/releases/2020-03, install OCL following elements:



ocl

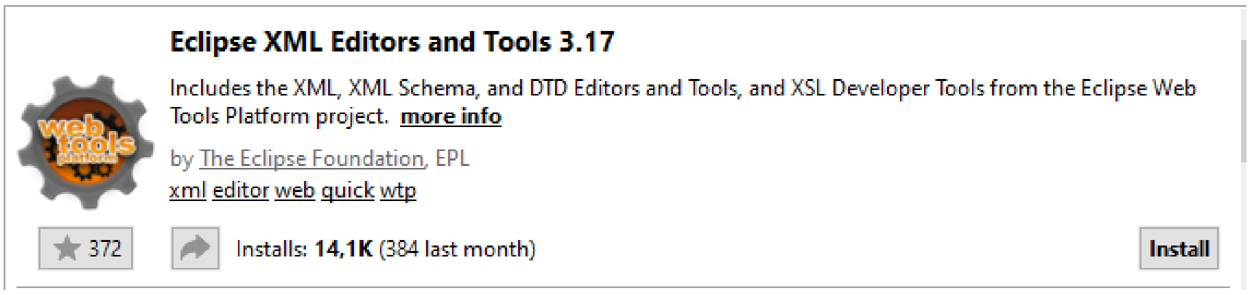
Name	Version	Id	Provider
OCL All-In-One SDK	6.11.0.v20200309-0...	org.eclipse.ocl.master.feature.group	Eclipse OCL
OCL Classic SDK: Ecore/UML Parsers,Evaluator,Edit	5.11.0.v20200309-0...	org.eclipse.ocl.all.sdk.feature.group	Eclipse OCL
OCL Classic SDK: Ecore/UML Parsers,Evaluator,Edit Develo|	5.11.0.v20200309-0...	org.eclipse.ocl.all.sdk.source.feature.group	Eclipse OCL

1. Click on Next, wait for installation
2. Click on Next, accept License text
3. Click on Finish
4. Restart Eclipse as requested:


Software Updates

Restart Eclipse IDE to apply the software update?

Other suggestion :


Eclipse XML Editors and Tools 3.17

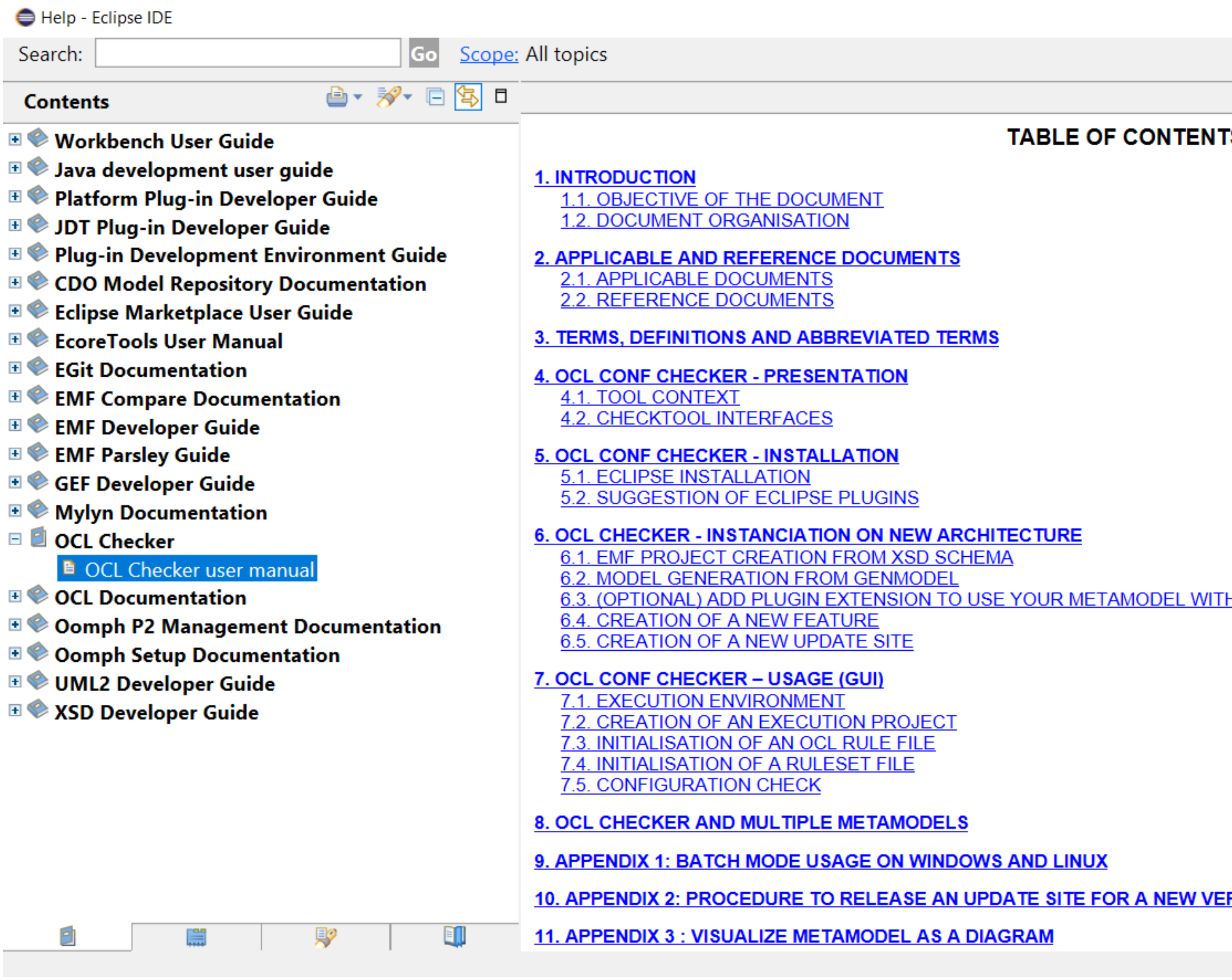
Includes the XML, XML Schema, and DTD Editors and Tools, and XSL Developer Tools from the Eclipse Web Tools Platform project. more info

by The Eclipse Foundation, EPL

xml editor web quick wtp

5.3. ONLINE DOCUMENTATION INSIDE ECLIPSE

After installation on Eclipse, this OCL Checker User Manual document is also available within the tool, and accessible after tool installation on Eclipse Help> Help contents



Help - Eclipse IDE

Search: Go Scope: All topics

Contents

- Workbench User Guide
- Java development user guide
- Platform Plug-in Developer Guide
- JDT Plug-in Developer Guide
- Plug-in Development Environment Guide
- CDO Model Repository Documentation
- Eclipse Marketplace User Guide
- EcoreTools User Manual
- EGit Documentation
- EMF Compare Documentation
- EMF Developer Guide
- EMF Parsley Guide
- GEF Developer Guide
- Mylyn Documentation
- OCL Checker
  - OCL Checker user manual
- OCL Documentation
- Oomph P2 Management Documentation
- Oomph Setup Documentation
- UML2 Developer Guide
- XSD Developer Guide

TABLE OF CONTENT

Note : The OCL Documentation is also available within Eclipse.

6. OCL CHECKER - INSTANCIATION ON NEW ARCHITECTURE

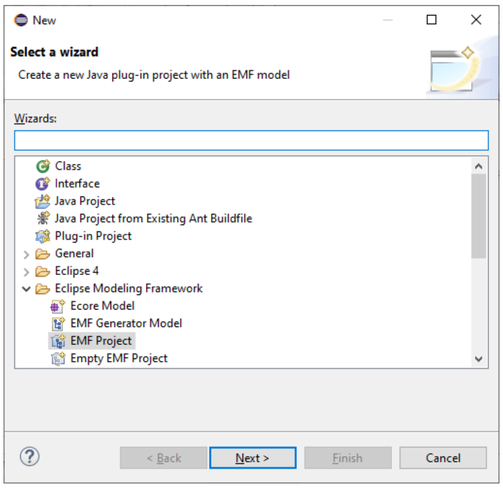
To check an OCL configuration on a new architecture, specific XSD schema configuration files defining this architecture must be used in the tool regarding the following 3 steps procedure:

1. EMF project creation from XSD schema
2. Model generation from genmodel (of previous created project)
3. Creation of a new feature (to package all previous created elements)
4. Create a dedicated update site (with the previous created feature) to enable the installation of the metamodel into an Eclipse environment
5. Then this update site can be installed into the environment and the XSD and associated metamodel is known from OCLChecker tool.

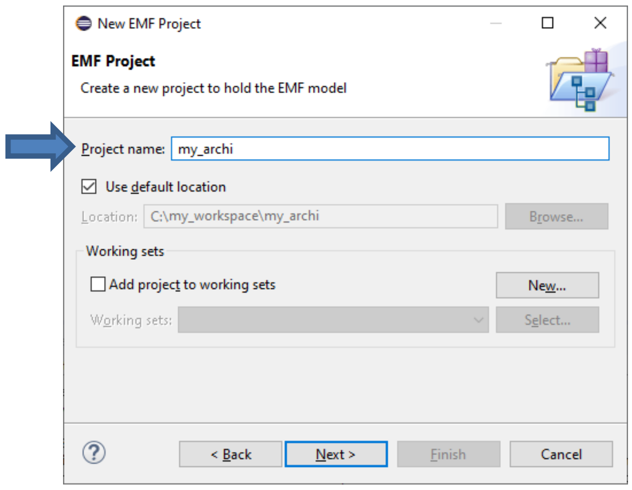
Hereafter are described all those steps.

6.1. EMF PROJECT CREATION FROM XSD SCHEMA

1. Go on File → New → Other… in the main menu
2. Select EMF Project:



6. Click on Next button
7. Give a name to the new project:


New EMF Project

EMF Project

Create a new project to hold the EMF model

Project name: my_archi

☑ Use default location

Location: C:\my_workspace\my_archi

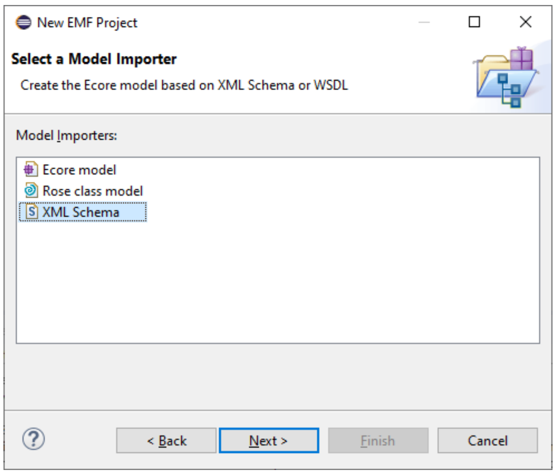
Working sets

☐ Add project to working sets

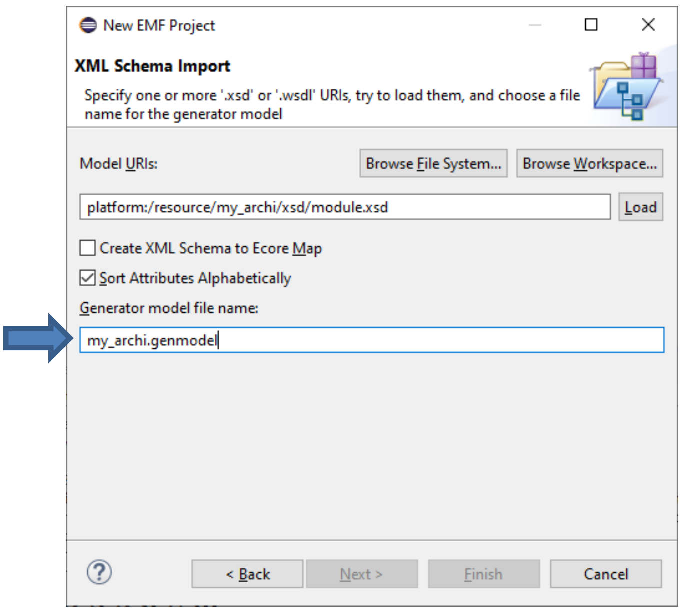
Working sets:

8. Click on Next button

9. Select XML Schema as Model Importer:



10. Click on Next
11. Click on Browse File System and select the XMS schema, and give a name to the new generator model file:


New EMF Project

XML Schema Import

Specify one or more '.xsd' or '.wsdl' URIs, try to load them, and choose a file name for the generator model

Model URIs: platform:/resource/my_archi/xsd/module.xsd

☐ Create XML Schema to Ecore Map

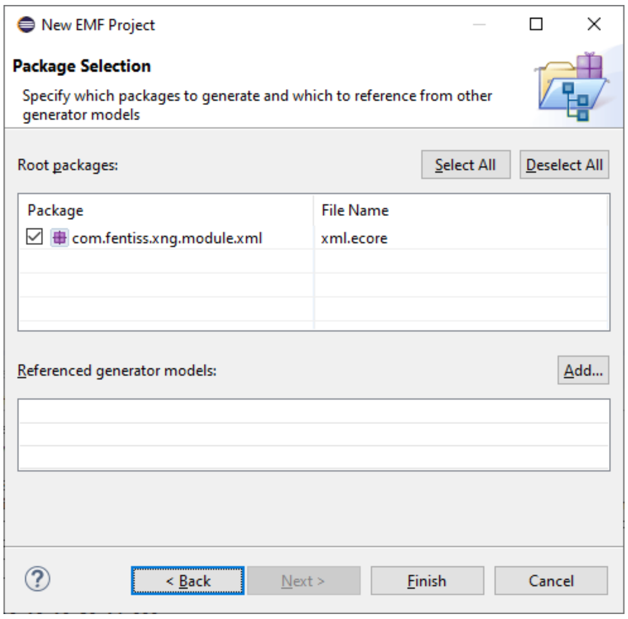
☑ Sort Attributes Alphabetically

Generator model file name: my_archi.genmodel

Note:

Please note that in this step a check is done on the XML schema, this check can also be done by clicking on the Load button.

12. Click on Next. The ecore filename can be modified at this step.



New EMF Project

Package Selection

Specify which packages to generate and which to reference from other generator models

Root packages:

Package	File Name
☑ com.fentiss.xng.module.xml	xml.ecore

Referenced generator models:

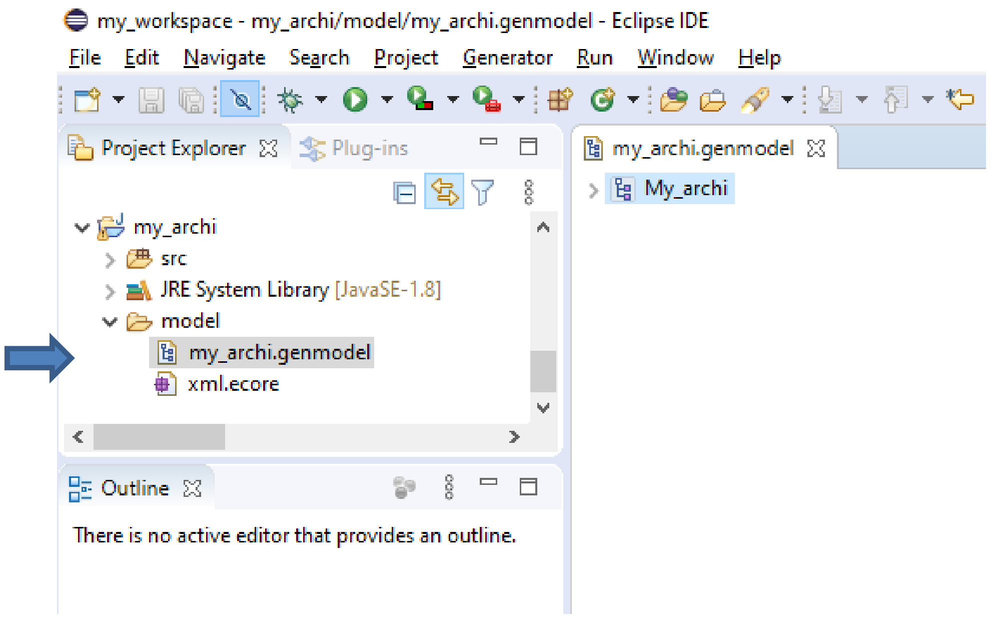
13. Click on Finish
14. The new project has been created and can be seen in the project view explorer.

As an additional feature interesting to mention, you can use Eclipse plugins to visualize the generated Ecore as a diagram (see 11 - APPENDIX 3 : Visualize metamodel as a diagram).

6.2. MODEL GENERATION FROM GENMODEL

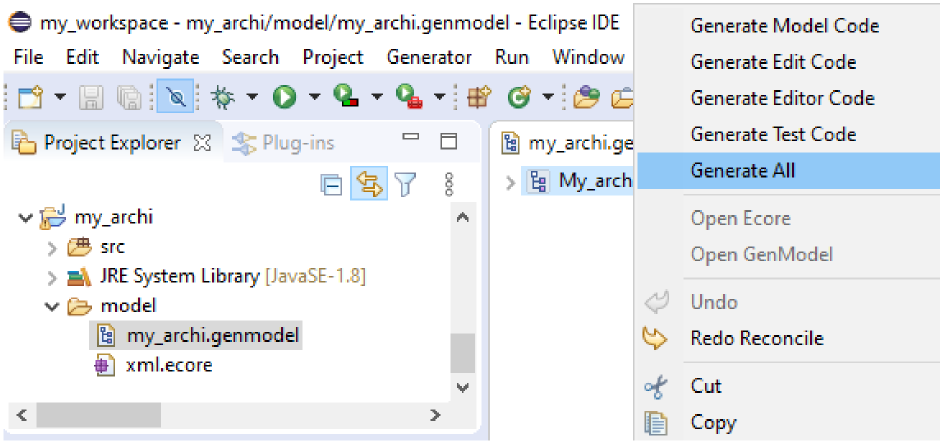
From previously created project

1. Double-click on the .genmodel file in the model folder to open the .genmodel:

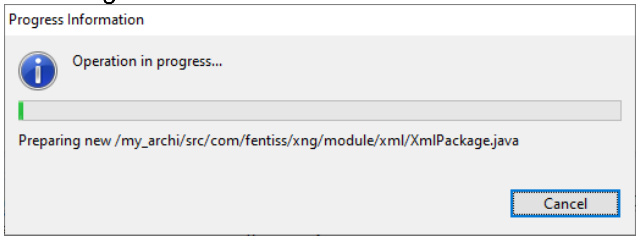

my_workspace - my_archi/model/my_archi.genmodel - Eclipse IDE
File Edit Navigate Search Project Generator Run Window Help
Project Explorer
Plug-ins
my_archi
src
JRE System Library [JavaSE-1.8]
model
my_archi.genmodel
xml.ecore
Outline
my_archi.genmodel
My_archi

There is no active editor that provides an outline.

2. Right-click on the root node and select Generate Model, Generate Edit and Generate Editor :



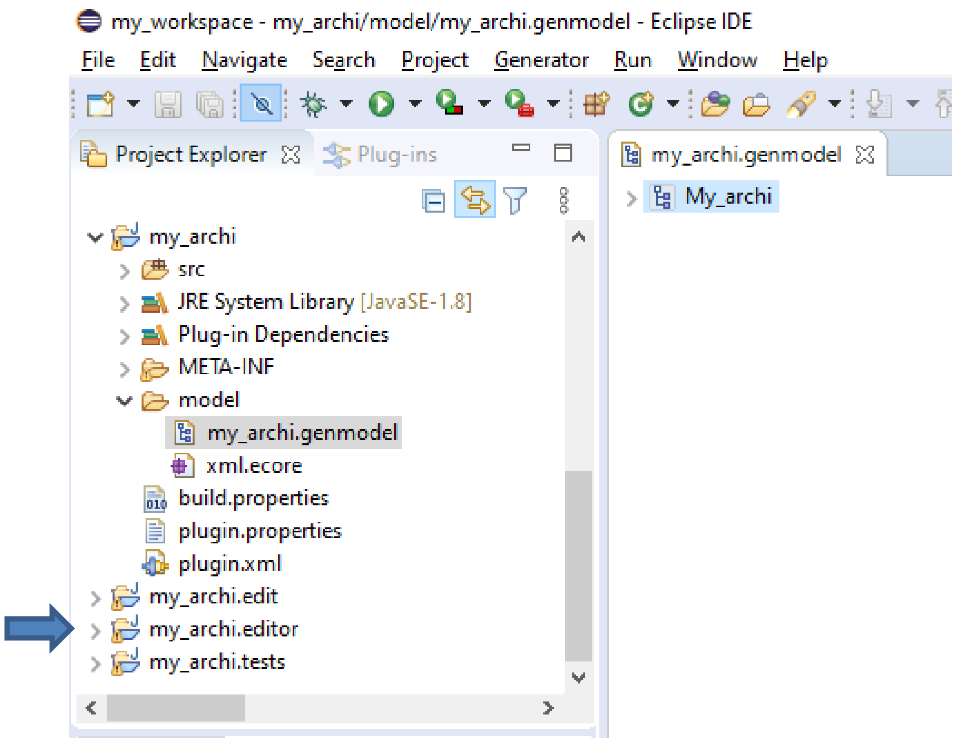
3. Wait for generation:



Operation in progress...

Preparing new /my_archi/src/com/fentiss/xng/module/xml/XmlPackage.java

4. 3 new projects are created:
   - <project_name>.edit
   - <project_name>.editor
   - <project_name>.tests



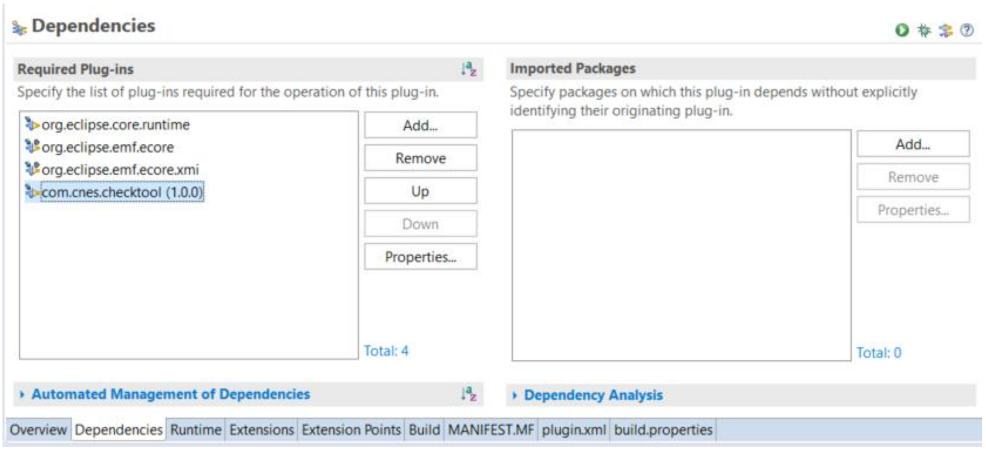
my_workspace - my_archi/model/my_archi.genmodel - Eclipse IDE

```
my_archi
  src
  JRE System Library [JavaSE-1.8]
  Plug-in Dependencies
  META-INF
  model
    my_archi.genmodel
    xml.ecore
  build.properties
  plugin.properties
  plugin.xml
my_archi.edit
my_archi.editor
my_archi.tests
```

6.3. (OPTIONAL) ADD PLUGIN EXTENSION TO USE YOUR METAMODEL WITH MULTIPLE METAMODELS

If you plan to use your XML files along with other metamodels / XML depending on other XSD described in section 8 - OCL Checker and Multiple metamodels below , you need to add an OCL Checker extension to your plug-in.

1. Open file plugin.xml from plugin <project-name>
2. On tab “Dependencies”, click on “Add..” and select “`com.cnes.checktool`”


Dependencies

Required Plug-ins
Specify the list of plug-ins required for the operation of this plug-in.

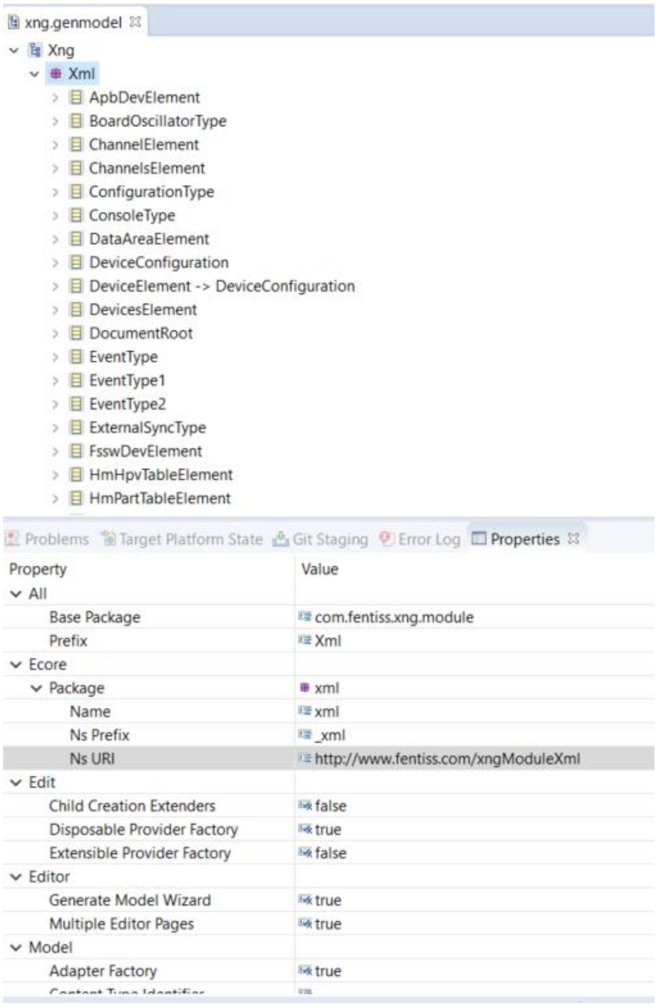
org.eclipse.core.runtime
org.eclipse.emf.ecore
org.eclipse.emf.ecore.xmi
com.cnes.checktool (1.0.0)

Total: 4

Imported Packages
Specify packages on which this plug-in depends without explicitly identifying their originating plug-in.

Total: 0

3. Keep plugin.xml opened
4. On the same project, open the .genmodel with the editor
   For each ePackage (node with symbol), select it and open Properties view to get property Ecore>Package>NsURI



xng.genmodel

- Xng
  - Xml
    - ApbDevElement
    - BoardOscillatorType
    - ChannelElement
    - ChannelsElement
    - ConfigurationType
    - ConsoleType
    - DataAreaElement
    - DeviceConfiguration
    - DeviceElement -> DeviceConfiguration
    - DevicesElement
    - DocumentRoot
    - EventType
    - EventType1
    - EventType2
    - ExternalSyncType
    - FsswDevElement
    - HmHpvTableElement
    - HmPartTableElement

Problems | Target Platform State | Git Staging | Error Log | Properties

Property	Value
All	
Base Package	com.fentiss.xng.module
Prefix	Xml
Ecore	
Package	xml
Name	xml
Ns Prefix	_xml
Ns URI	http://www.fentiss.com/xngModuleXml
Edit	
Child Creation Extenders	false
Disposable Provider Factory	true
Extensible Provider Factory	false
Editor	
Generate Model Wizard	true
Multiple Editor Pages	true
Model	
Adapter Factory	true

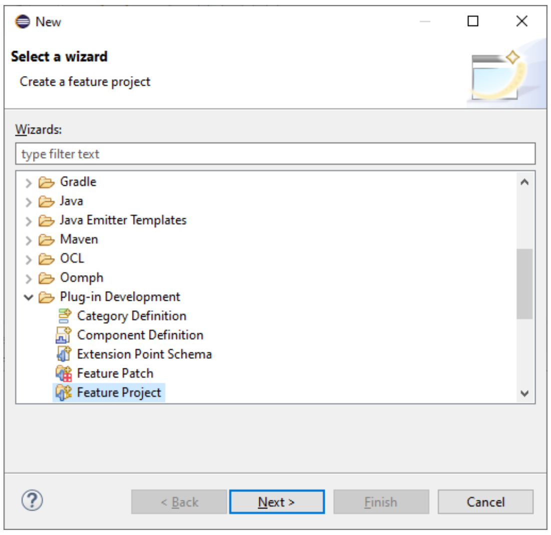
5. For each EPackage Ns URI found on the .genmodel,
   add one extension in plugin.xml, before last line `</plugin>` with the following content :

```
<extension
       point="com.cnes.checktool.metamodel.epackage">
       <EPackage
          EPackageURI="<<EPackage URI>>">
    </EPackage>
 </extension>
```

6. Save and close plugin.xml

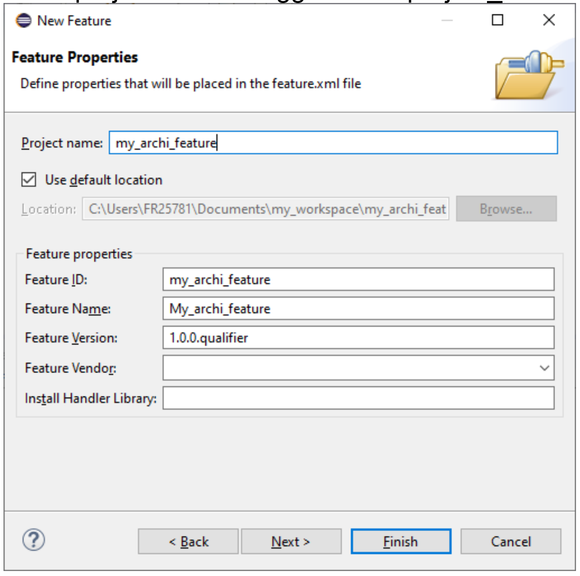
6.4. CREATION OF A NEW FEATURE

1. Go on File → New → Other… in the main menu
2. Select Feature Project:



Create a feature project

3. Click on Next
4. Enter project name. Suggestion: <project_name>.feature



Define properties that will be placed in the feature.xml file

Project name: my_archi_feature

☑ Use default location

Location: C:\Users\FR25781\Documents\my_workspace\my_archi_feat

Feature ID: my_archi_feature

Feature Name: My_archi_feature

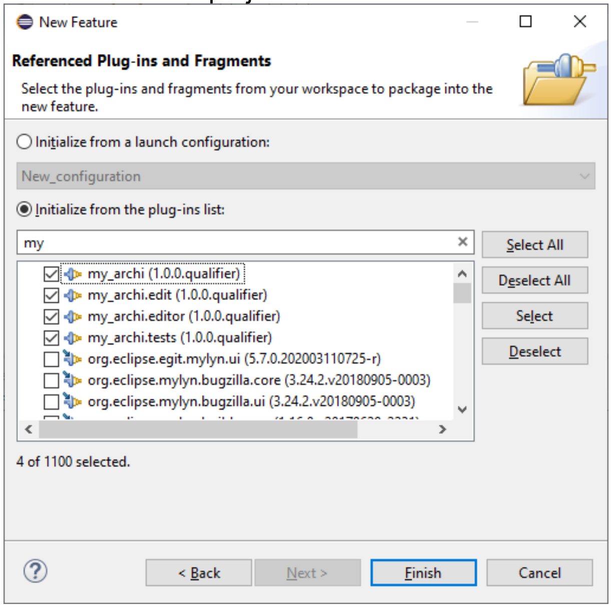
Feature Version: 1.0.0.qualifier

Feature Vendor:

Install Handler Library:

5. Click on Next

6. Select all related projects:


New Feature

Referenced Plug-ins and Fragments

Select the plug-ins and fragments from your workspace to package into the new feature.

○ Initialize from a launch configuration:

New_configuration

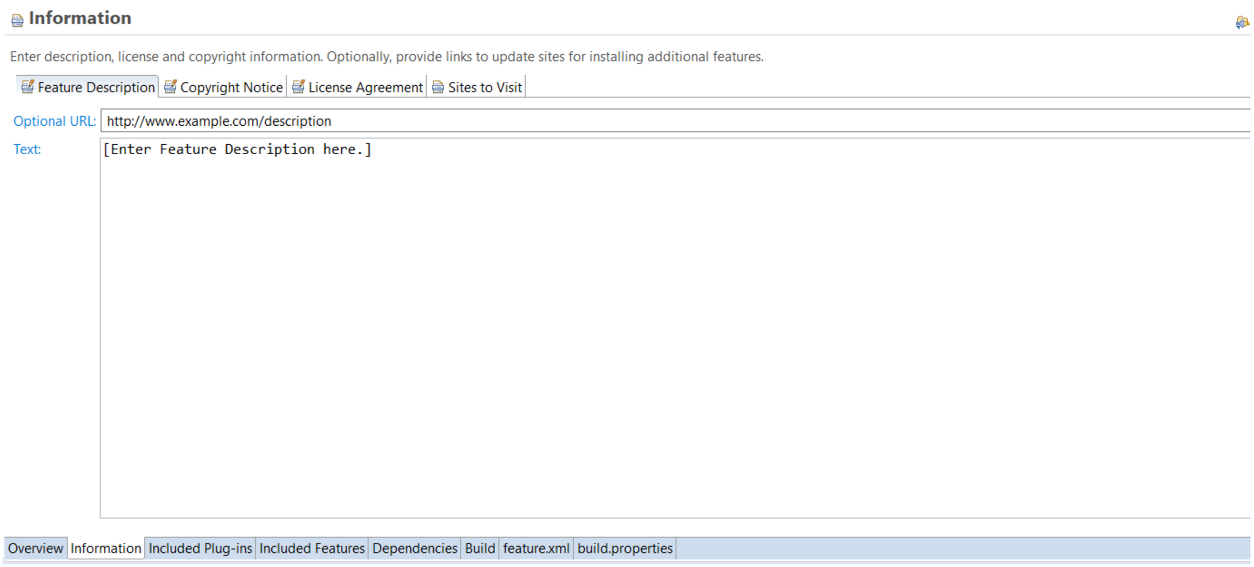
● Initialize from the plug-ins list:

my

- ☑ my_archi (1.0.0.qualifier)
- ☑ my_archi.edit (1.0.0.qualifier)
- ☑ my_archi.editor (1.0.0.qualifier)
- ☑ my_archi.tests (1.0.0.qualifier)
- ☐ org.eclipse.egit.mylyn.ui (5.7.0.202003110725-r)
- ☐ org.eclipse.mylyn.bugzilla.core (3.24.2.v20180905-0003)
- ☐ org.eclipse.mylyn.bugzilla.ui (3.24.2.v20180905-0003)

4 of 1100 selected.

7. Click on Finish
8. New feature project appears in the project explorer.
9. Open file “feature.xml”, on “Information” tab


Information

Enter description, license and copyright information. Optionally, provide links to update sites for installing additional features.

Optional URL: http://www.example.com/description

Text: [Enter Feature Description here.]

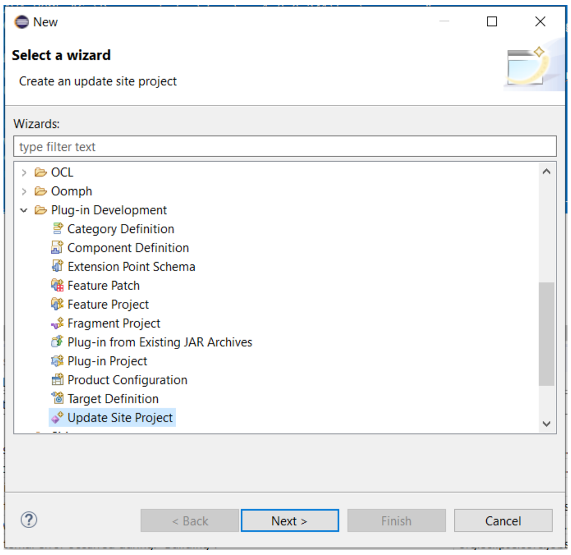
10. Fill “Feature Description”, “Copyright notice” and “License Agreement”

Note : the “License Agreement” will be displayed to any user installing this feature, for acceptation.

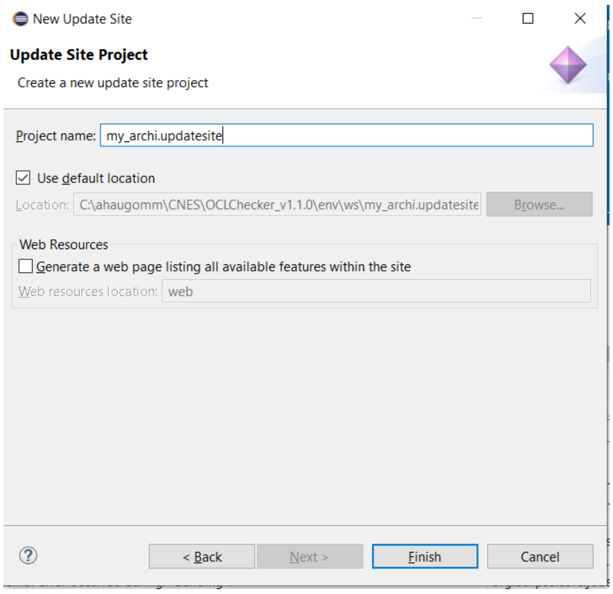
It is HIGHLY recommended to fill these three tabs.

6.5. CREATION OF A NEW UPDATE SITE

1. Go on File → New → Other… in the main menu
2. Plug-in development → Update site project



3. Fill the name on next page "<project-name>.updatesite"


New Update Site

Update Site Project

Create a new update site project

Project name: my_archi.updatesite

☑ Use default location

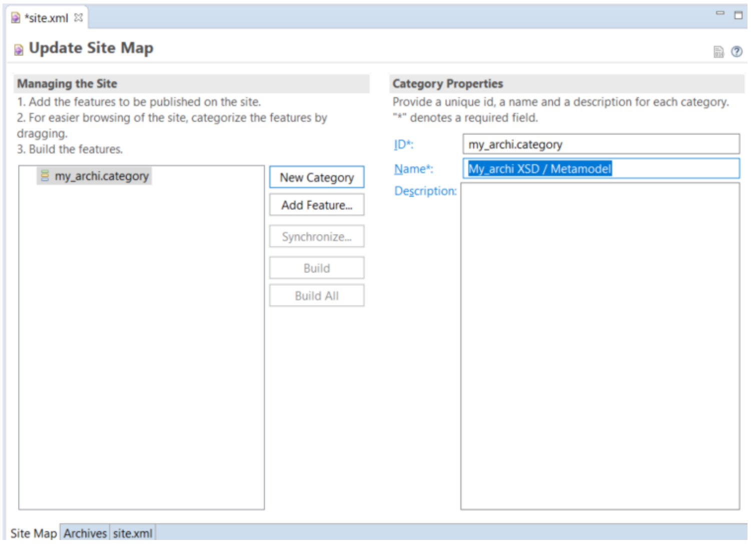
Location: C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\ws\my_archi.updatesite

Web Resources

☐ Generate a web page listing all available features within the site

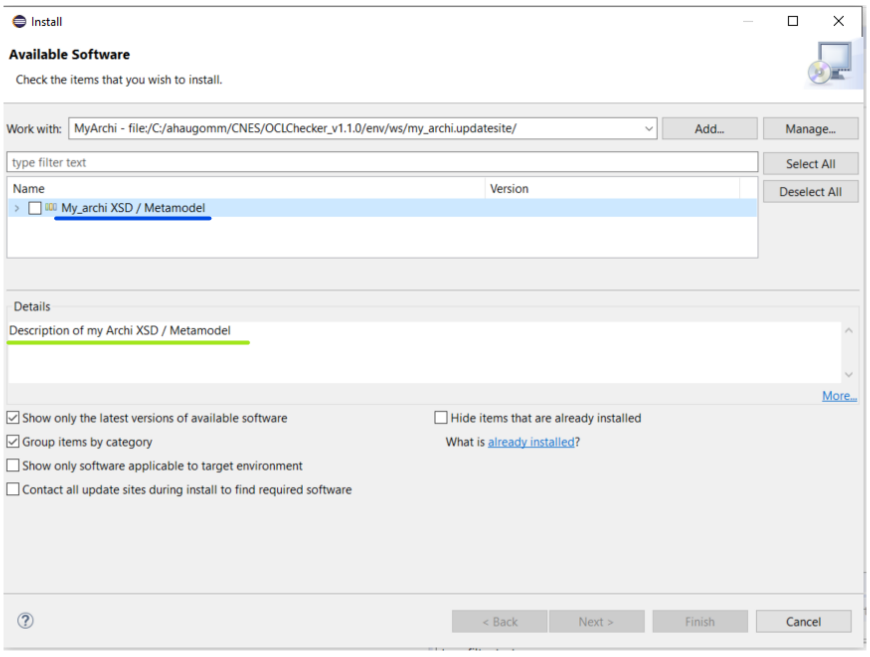
Web resources location: web

4. Open site.xml file and select "New Category"

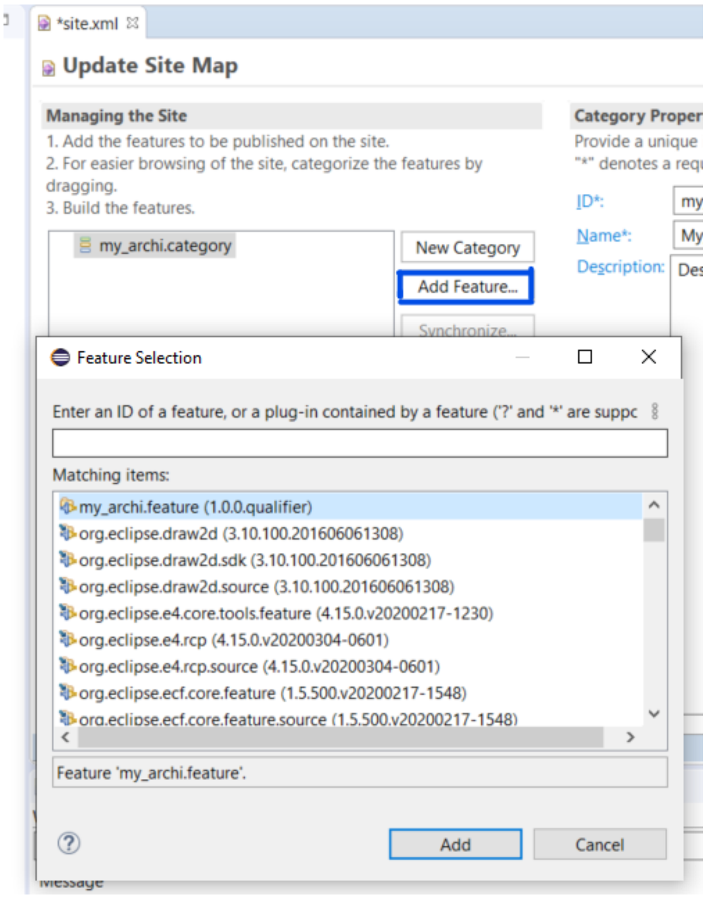

*site.xml
Update Site Map
Managing the Site
1. Add the features to be published on the site.
2. For easier browsing of the site, categorize the features by dragging.
3. Build the features.
my_archi.category
New Category
Add Feature...
Synchronize...
Build
Build All
Category Properties
Provide a unique id, a name and a description for each category.
"*" denotes a required field.
ID*: my_archi.category
Name*: My_archi XSD / Metamodel
Description:
Site Map
Archives
site.xml

5. Fill ID with <project-name>.category, and Name with something meaningful for the user to know what is being installed. Description can be filled to add more details on the category.

   Note : when users will be installing content from this update site, the category is displayed as a group of items to be installed (in blue) and the description appears below (green).


Install
Available Software
Check the items that you wish to install.
Work with: MyArchi - file:/C:/ahaugomm/CNES/OCLChecker_v1.1.0/env/ws/my_archi.updatesite/
Add...
Manage...
type filter text
Select All
Deselect All
Name
Version
My_archi XSD / Metamodel
Details
Description of my Archi XSD / Metamodel
More...
Show only the latest versions of available software
Group items by category
Show only software applicable to target environment
Contact all update sites during install to find required software
Hide items that are already installed
What is already installed?
< Back
Next >
Finish
Cancel

6. In site.xml, select category and click on “Add feature” and select <project-name>.feature (version 1.0.0.qualifier) from the list.



Update Site Map

Managing the Site
1. Add the features to be published on the site.
2. For easier browsing of the site, categorize the features by dragging.
3. Build the features.

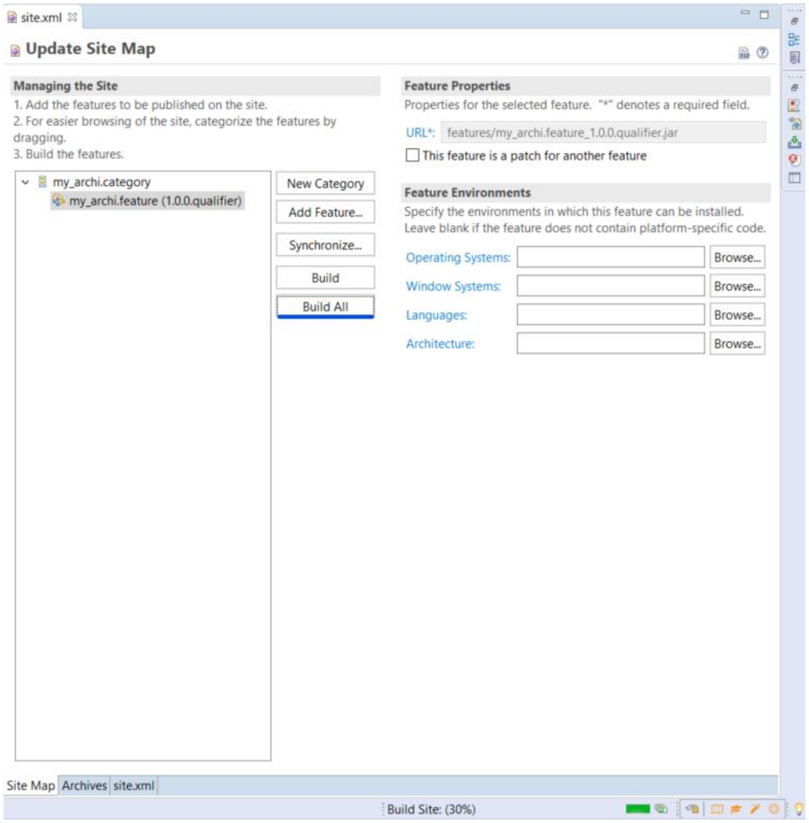
Feature Selection

Enter an ID of a feature, or a plug-in contained by a feature ('?' and '*' are suppc

Matching items:
- my_archi.feature (1.0.0.qualifier)
- org.eclipse.draw2d (3.10.100.201606061308)
- org.eclipse.draw2d.sdk (3.10.100.201606061308)
- org.eclipse.draw2d.source (3.10.100.201606061308)
- org.eclipse.e4.core.tools.feature (4.15.0.v20200217-1230)
- org.eclipse.e4.rcp (4.15.0.v20200304-0601)
- org.eclipse.e4.rcp.source (4.15.0.v20200304-0601)
- org.eclipse.ecf.core.feature (1.5.500.v20200217-1548)
- org.eclipse.ecf.core.feature.source (1.5.500.v20200217-1548)

Feature 'my_archi.feature'.

7. Save site.xml and click on “Build all”



Update Site Map

Managing the Site
1. Add the features to be published on the site.
2. For easier browsing of the site, categorize the features by dragging.
3. Build the features.

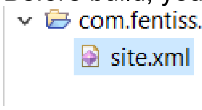
Feature Properties
Properties for the selected feature. "*" denotes a required field.
URL*: features/my_archi.feature_1.0.0.qualifier.jar
☐ This feature is a patch for another feature

Feature Environments
Specify the environments in which this feature can be installed.
Leave blank if the feature does not contain platform-specific code.
Operating Systems:
Window Systems:
Languages:
Architecture:

Build Site: (30%)

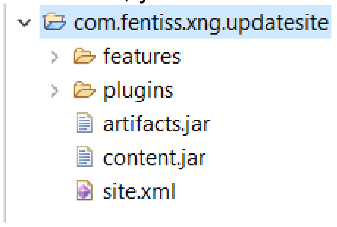
8. Wait until Build is 100%

   Before build, you have only one file



```
com.fentiss.xng.updatesite
    site.xml
```

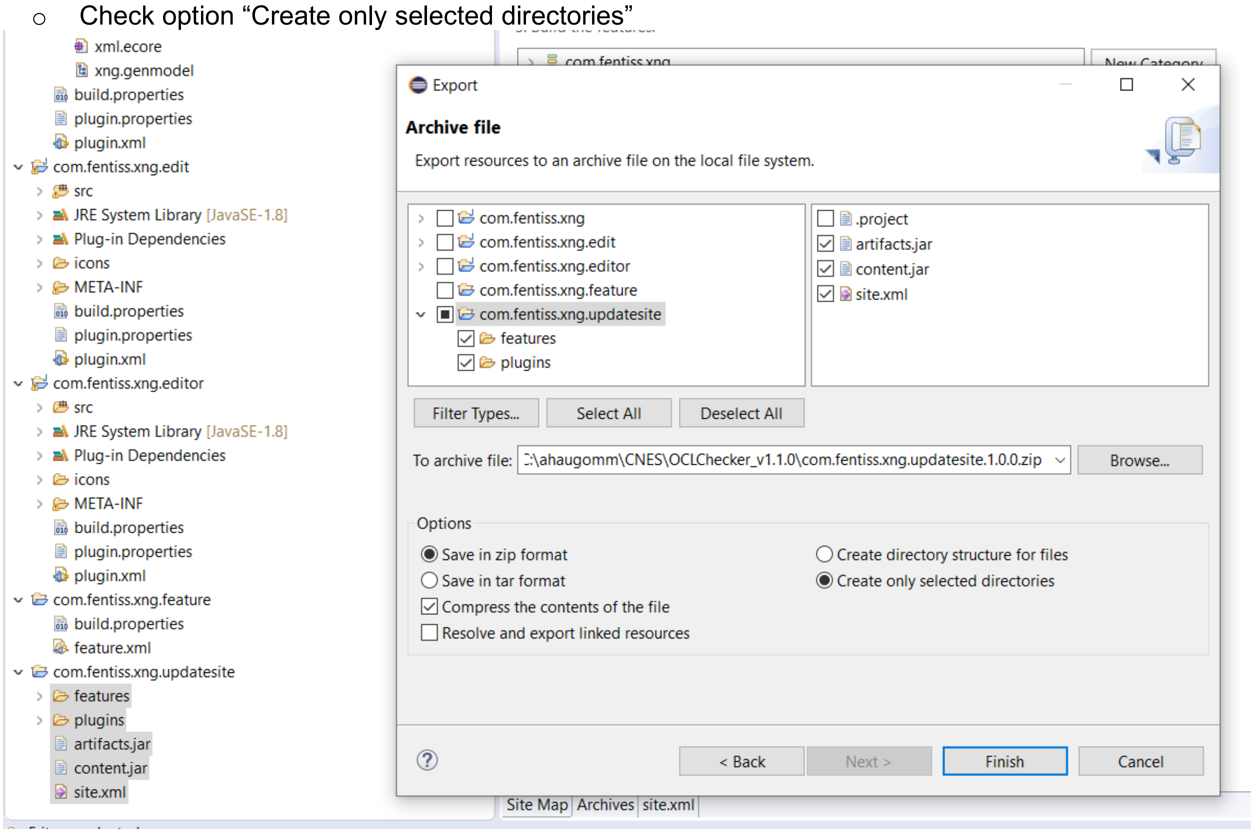
   After build, you have 2 folders and 3 files



```
com.fentiss.xng.updatesite
  > features
  > plugins
    artifacts.jar
    content.jar
    site.xml
```

9. In order to share this update site, create an archive of all files in the project
   WARNING : The archive must contain directly the site.xml file, no upper level folder.

   If you create the archive from Eclipse
   - Select the files (site.xml, artifacts.jar, contents.jar, features folder , plugins folder) and click File > Export > Archive file > Next
   - Check option "Create only selected directories"



```
xml.ecore
xng.genmodel
build.properties
plugin.properties
plugin.xml
com.fentiss.xng.edit
  > src
  > JRE System Library [JavaSE-1.8]
  > Plug-in Dependencies
  > icons
  > META-INF
    build.properties
    plugin.properties
    plugin.xml
com.fentiss.xng.editor
  > src
  > JRE System Library [JavaSE-1.8]
  > Plug-in Dependencies
  > icons
  > META-INF
    build.properties
    plugin.properties
    plugin.xml
com.fentiss.xng.feature
    build.properties
    feature.xml
com.fentiss.xng.updatesite
  > features
  > plugins
    artifacts.jar
    content.jar
    site.xml
```

Export

Archive file

Export resources to an archive file on the local file system.

```
> [ ] com.fentiss.xng                 [ ] .project
> [ ] com.fentiss.xng.edit            [✓] artifacts.jar
> [ ] com.fentiss.xng.editor          [✓] content.jar
  [ ] com.fentiss.xng.feature         [✓] site.xml
v [■] com.fentiss.xng.updatesite
      [✓] features
      [✓] plugins
```

Filter Types... | Select All | Deselect All

To archive file: :\ahaugomm\CNES\OCLChecker_v1.1.0\com.fentiss.xng.updatesite.1.0.0.zip   Browse...

Options
- (●) Save in zip format
- ( ) Save in tar format
- [✓] Compress the contents of the file
- [ ] Resolve and export linked resources
- ( ) Create directory structure for files
- (●) Create only selected directories

< Back | Next > | Finish | Cancel

   - Finish

10. Open the resulting archive to ensure that you have site.xml at top level.
11. Test the installation from your eclipse platform
    - Help > Install new software > Add.. > Archive ..
    - Select archive of the new update site

- Check that the two names in the check boxes correspond to
    - For highest level , the label of the category from site.xml in update site project
    - For lower level, the feature name (in feature.xml)

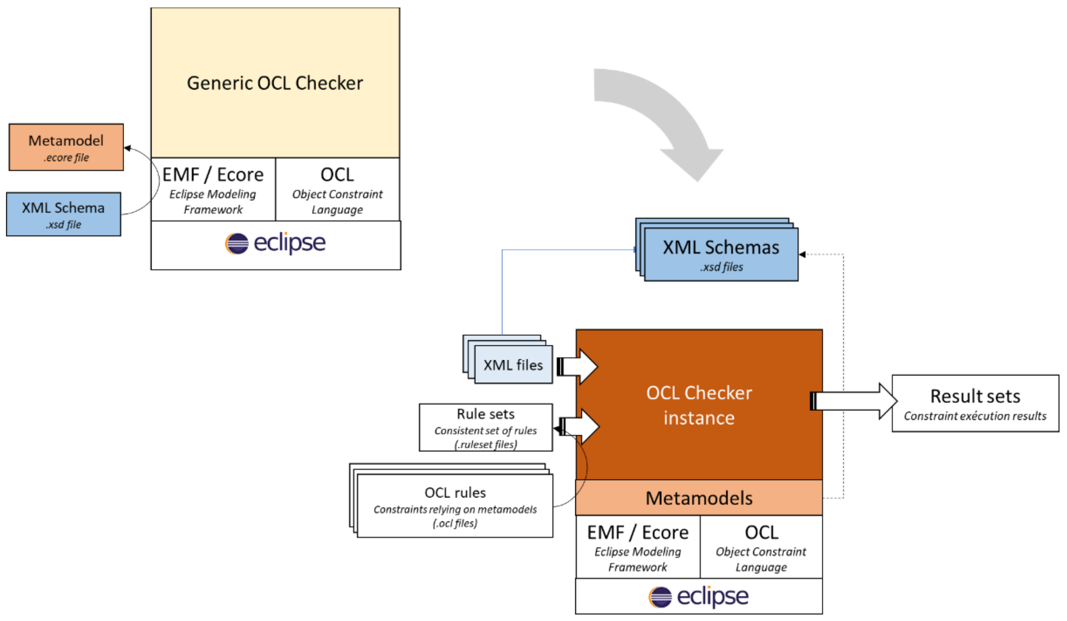
7. OCL CONF CHECKER – USAGE (GUI)

The tool must be used in Eclipse running environment.
Only ruleset files can be executed on xml files to validate rules.
To use the tool the following steps must be followed:

1. Running environment setup
2. Creation of an execution project
3. Initialisation of a rule file
4. Initialisation of a ruleset file
5. Configuration check

Hereafter are described all those steps.

7.1. EXECUTION ENVIRONMENT


Generic OCL Checker
Metamodel
.ecore file
XML Schema
.xsd file
EMF / Ecore
Eclipse Modeling Framework
OCL
Object Constraint Language
eclipse
XML Schemas
.xsd files
XML files
Rule sets
Consistent set of rules (.ruleset files)
OCL rules
Constraints relying on metamodels (.ocl files)
OCL Checker instance
Result sets
Constraint exécution results
Metamodels
EMF / Ecore
Eclipse Modeling Framework
OCL
Object Constraint Language
eclipse

As presented in section 4 - OCL conf checker - Presentation above, in order to use OCL Checker tool with XML files compliant with a specific XSD , you need to “install this XSD as a metamodel in the Eclipse environment”.

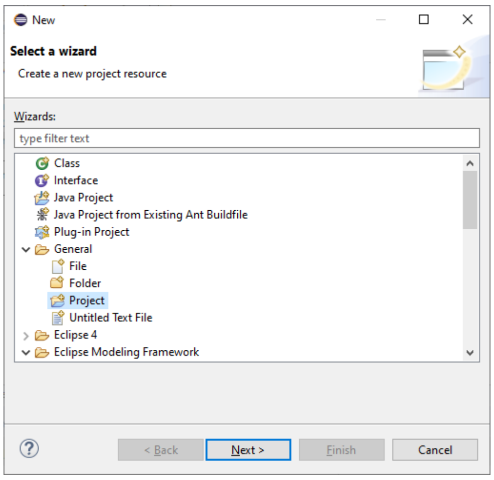
This corresponds to :

- Install “Generic” OCL Checker on top of the Eclipse installation (see section 5 - OCL conf checker - Installation)
- Install required metamodels (from XSD) for the XML models to check in the same Eclipse environment (see section 6 - OCL checker - Instanciation on new architecture)

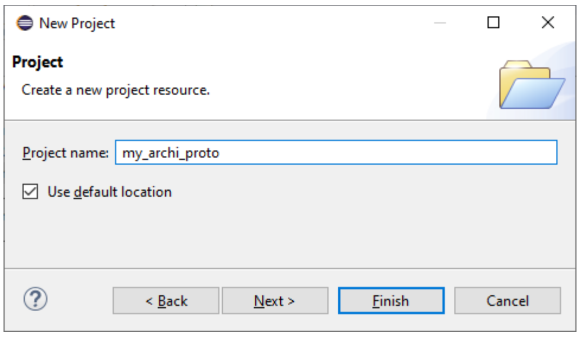
Then the OCL Checker knows how to handle such XML models for check.

7.2. CREATION OF AN EXECUTION PROJECT

1. Go on File → New → Other… in the main menu
2. Select General Project:



3. Click on Next
4. Precise name project



Project name: my_archi_proto

☑ Use default location

5. Click on Finish
6. New empty project appears in the project view explorer.

7.3. INITIALISATION OF AN OCL RULE FILE

A rule file is an OCL file associated to the namespace from an ecore file.

/!\ CAUTION /!\
Please be aware that creation of OCL file from another manner can lead to a bad ocl file (due to a bad first import line in the created file).

1. In your project, click on New File
2. Precise the name of your new ocl rule file, with .ocl extension
3. Once created, fill in the file with the minimum content:

```
import 'http://www.fentiss.com/xngModuleXml'

package xml

-- OCL rules

endpackage
```

   Here completion, with CTRL + J, can be used to precise the namespace to be imported (corresponding to the ecore file).

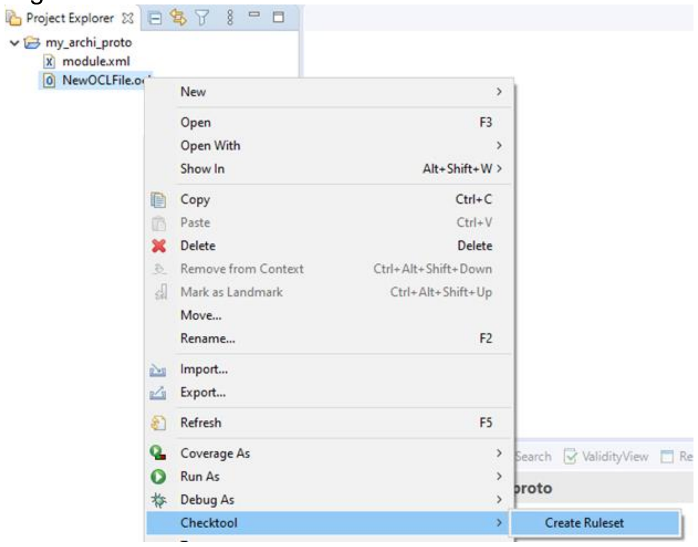
4. Fill in the file with the relevant ocl rules.

Note : The full OCL Syntax is available at the following location : https://www.omg.org/spec/OCL/2.4/PDF
.

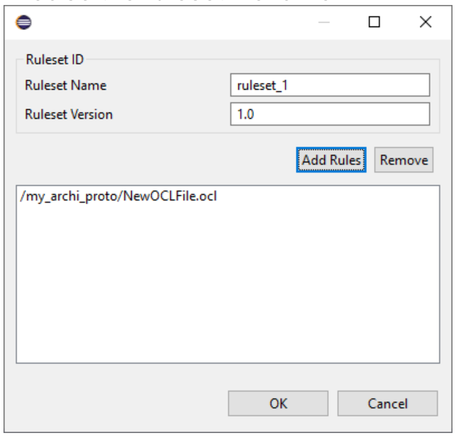
7.4. INITIALISATION OF A RULESET FILE

A ruleset file is set of rules issued from one or several ocl files.

1. Click on an ocl file.
2. Right click on it and select Checktoool → Create Ruleset



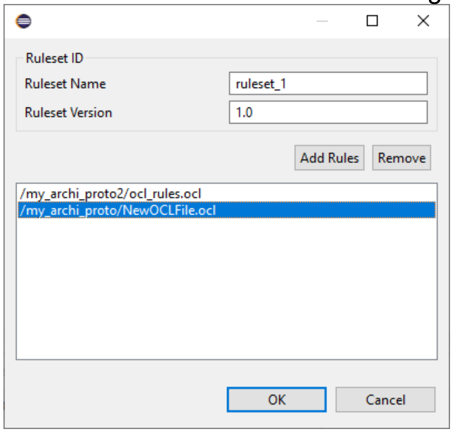
3. Precise the ruleset filename:



Ruleset ID
Ruleset Name: ruleset_1
Ruleset Version: 1.0

/my_archi_proto/NewOCLFile.ocl

Several ocl files can be added through Add Rules:



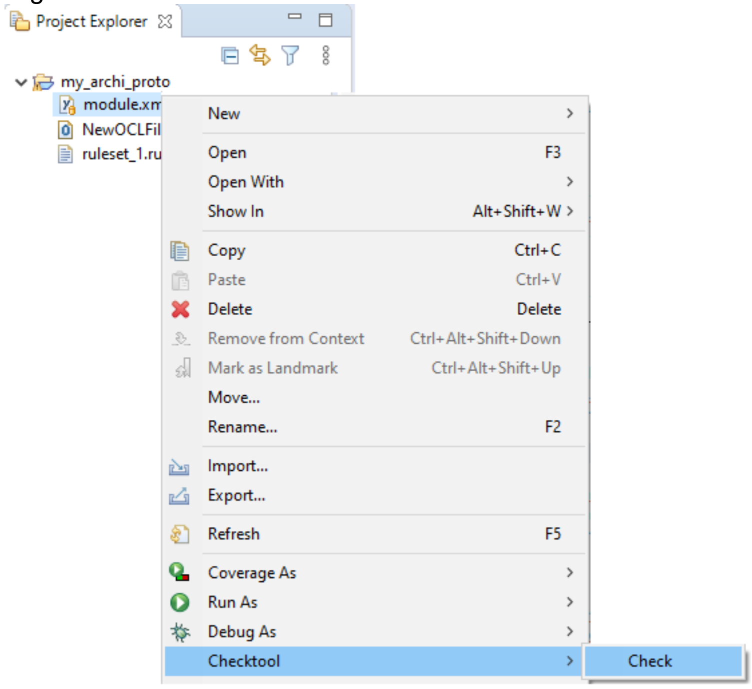
Ruleset ID
Ruleset Name: ruleset_1
Ruleset Version: 1.0

/my_archi_proto2/ocl_rules.ocl
/my_archi_proto/NewOCLFile.ocl

4. Click on OK
5. Ruleset file is created

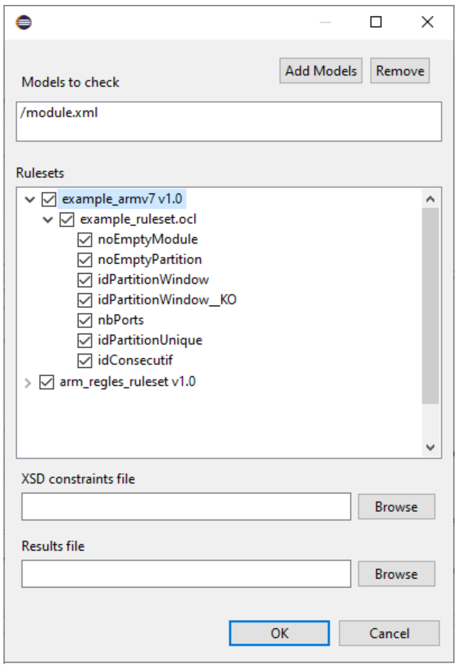
7.5. CONFIGURATION CHECK

1. Import XML files in the execution project.
2. Click on XML configuration file to be checked
3. Right click on it and select Checktool → → Check



4. Select the ruleset
   And click on Browse to precise result file:

> /!\ CAUTION /!\
> Output result file must be filled to launch execution.



Models to check

/module.xml

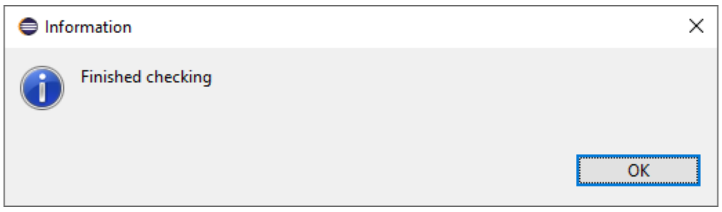
Rulesets

- ☑ example_armv7 v1.0
  - ☑ example_ruleset.ocl
    - ☑ noEmptyModule
    - ☑ noEmptyPartition
    - ☑ idPartitionWindow
    - ☑ idPartitionWindow_KO
    - ☑ nbPorts
    - ☑ idPartitionUnique
    - ☑ idConsecutif
- ☑ arm_regles_ruleset v1.0

XSD constraints file

Results file

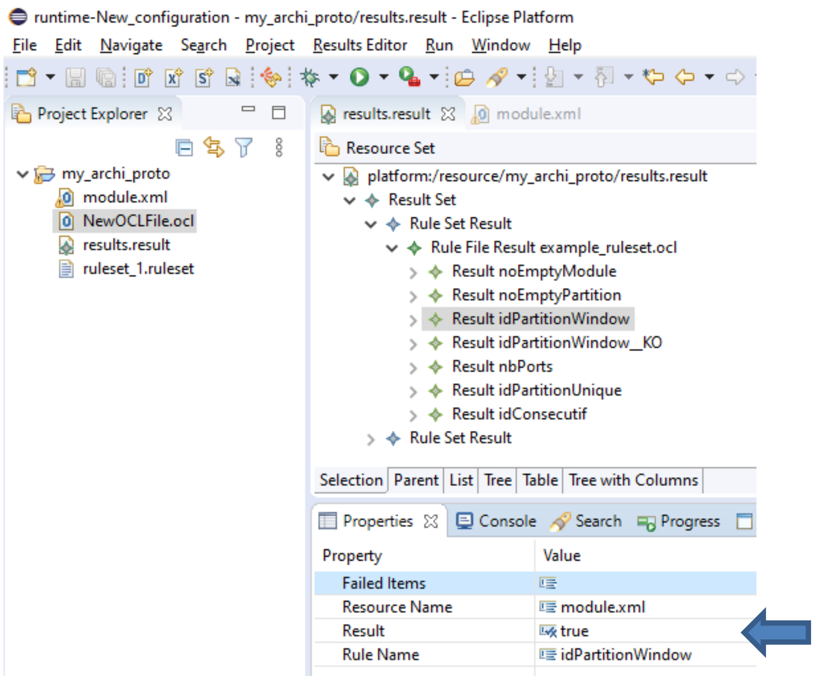
5. Click on OK to launch the check, wait for execution end



Information

Finished checking

6. The result file has been created.



runtime-New_configuration - my_archi_proto/results.result - Eclipse Platform

Project Explorer

- my_archi_proto
  - module.xml
  - NewOCLFile.ocl
  - results.result
  - ruleset_1.ruleset

results.result — Resource Set

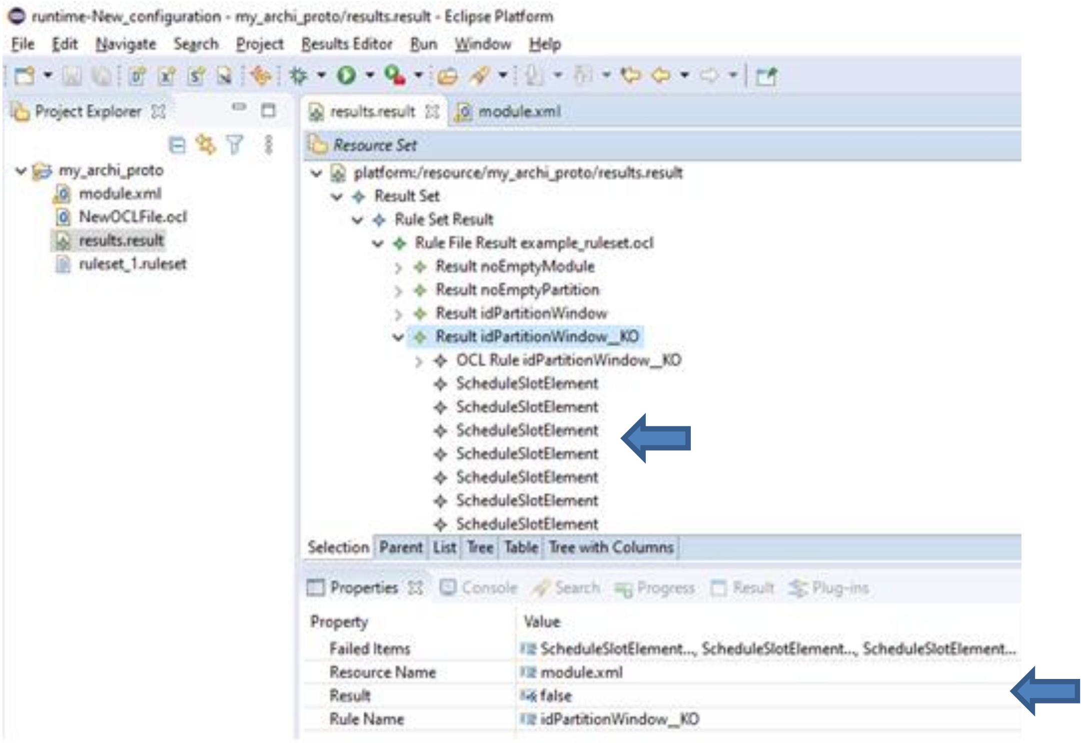
- platform:/resource/my_archi_proto/results.result
  - Result Set
    - Rule Set Result
      - Rule File Result example_ruleset.ocl
        - Result noEmptyModule
        - Result noEmptyPartition
        - Result idPartitionWindow
        - Result idPartitionWindow_KO
        - Result nbPorts
        - Result idPartitionUnique
        - Result idConsecutif
    - Rule Set Result

Property	Value
Failed Items	
Resource Name	module.xml
Result	true
Rule Name	idPartitionWindow

7. Click on it and expand the resource set in the right view.

For each rule you have its result: true or false (see the arrow above).

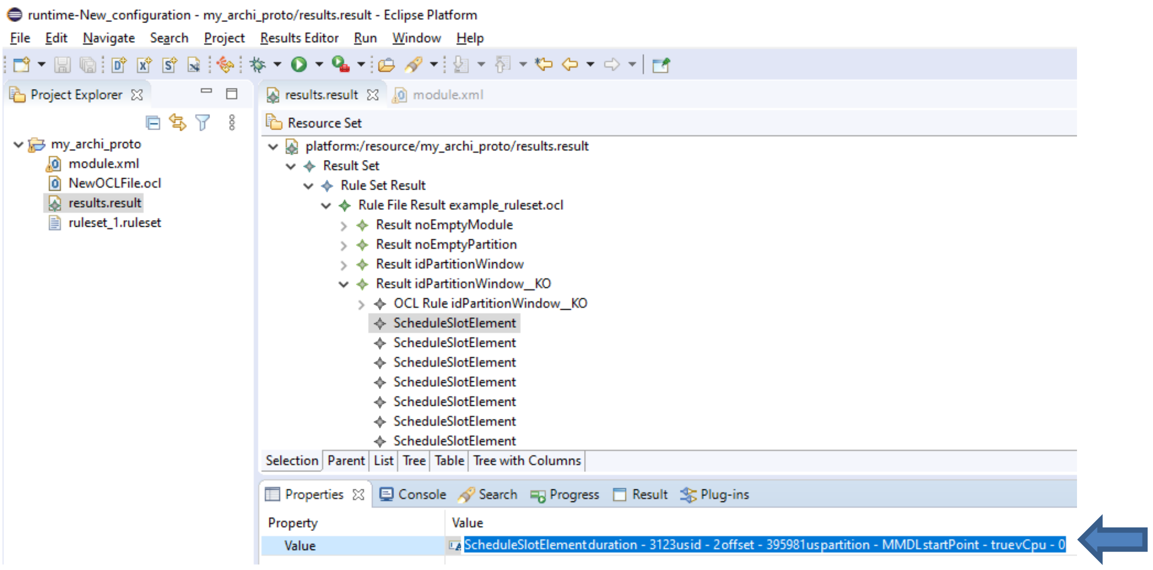
In case of a rule has failed, as follow, the list of failed items can be accessed:


runtime-New_configuration - my_archi_proto/results.result - Eclipse Platform

Resource Set
- platform:/resource/my_archi_proto/results.result
  - Result Set
    - Rule Set Result
      - Rule File Result example_ruleset.ocl
        - Result noEmptyModule
        - Result noEmptyPartition
        - Result idPartitionWindow
        - Result idPartitionWindow_KO
          - OCL Rule idPartitionWindow_KO
          - ScheduleSlotElement
          - ScheduleSlotElement
          - ScheduleSlotElement
          - ScheduleSlotElement
          - ScheduleSlotElement
          - ScheduleSlotElement
          - ScheduleSlotElement

Property	Value
Failed Items	ScheduleSlotElement..., ScheduleSlotElement..., ScheduleSlotElement...
Resource Name	module.xml
Result	false
Rule Name	idPartitionWindow_KO

Select each element in the upper view to display its properties (no element is displayed when the rule is OK). Double click on ‘Value’ in Properties tab to display it:


runtime-New_configuration - my_archi_proto/results.result - Eclipse Platform

Property	Value
Value	ScheduleSlotElementduration - 3123usid - 2offset - 395981uspartition - MMDLstartPoint - truevCpu - 0

Those information leads to identify the element who caused the failure.

8. OCL CHECKER AND MULTIPLE METAMODELS

With OCL evaluation used inside OCLChecker, there is a limitation : only one metamodel declaration is allow on the first lines of .ocl files.

Thus the ocl file to use rules with multiple metamodel should look like the following :

```
import 'http://www.fentiss.com/xngModuleXml'
-- import ‘metamodel2 NS_URI’

package xml

context PartitionElement
inv my_rule : <<combine elements from NS_URI 1 and NS_URI 2>>


endpackage
```

The second metamodel is imported through the extension point declared in section 6.3 - (optional) Add plugin extension to use your metamodel with multiple metamodels .

9. APPENDIX 1: BATCH MODE USAGE ON WINDOWS AND LINUX

Tool can be used in batch mode, as an Eclipse application with a set of arguments .

```
java
-cp <<Eclipse Equinox launcher>>
org.eclipse.core.launcher.Main
-data <<Folder to use as a workspace>>
-application com.cnes.checktool.application
  --provider=<<Folder to use as workspace>>
  --rulesets=<<RuleSet files>>
  --ocls=<<OCL rule Files>>
  --xmlmodels=<<XML models to evaluate>>
  --output=<<Result file path>>
```

Hereafter are tool batch arguments for application “`com.cnes.checktool.application`” :

Argument	Required	Description
provider	yes	Folder to use as a workspace (Batch mode launches an Eclipse instance). Contains all ruleset files and XSD constraints.
rulesets	no	The path, relative to the provider workspace, of the rulesets to be applied.
xsdpath		The path, relative to the provider workspace, of the selected XML schema file.
xmlmodels	yes	The absolute path of all xml models to be evaluated, separated by ; Use quotes “” around the full parameter value if your are using multiple values. `--xmlmodels=”path/to/m1.xml;path/to/m2.xml;path/to/m3.xml”`
ocls	No	The path, relative to the provider workspace, of the selected constraint file.
output	yes	The absolute path of the file containing the results.

Note:
Tool will not run if rulesets, ocls and xsdpath are both null.

- Sample command line (windows) with ruleset file (replace blue values by local paths):

```
@echo off
set JAVA_HOME="C:\Program Files\Java\jre1.8.0_241\bin\java.exe"
set EQUINOX_LAUNCHER_PATH="C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\eclipse-modeling-2020-03-R-win32-x86_64\eclipse\plugins\org.eclipse.equinox.launcher_1.5.700.v20200207-2156.jar"
set TEST_WORKSPACE_PATH=C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\ws
set XML_MODELS=%TEST_WORKSPACE_PATH%\OCLChecker_Tests\OCLCHECKER-TV-030-001\file1.xml;%TEST_WORKSPACE_PATH%\OCLChecker_Tests\OCLCHECKER-TV-030-001\file2.xml
set RULE_SETS=\OCLChecker_Tests\OCLCHECKER-TV-030-001\OCLCHECKER-TV-030-001_v1.0.ruleset
set OUTPUT_FILE=%TEST_WORKSPACE_PATH%\OCLChecker_example\res.result

echo Executing CheckTool
echo with rulesets: %RULE_SETS%
echo on models: %XML_MODELS%
echo result in: %OUTPUT_FILE%
```

```
%JAVA_HOME% -cp %EQUINOX_LAUNCHER_PATH% org.eclipse.core.launcher.Main -data
%TEST_WORKSPACE_PATH% -application com.cnes.checktool.application --
provider=%TEST_WORKSPACE_PATH% --rulesets=%RULE_SETS%  --xmlmodels=%XML_MODELS% --
output=%OUTPUT_FILE%
```

- Sample command line (windows) with OCL files directly (replace blue values by local paths): :

```
@echo off
set JAVA_HOME="C:\Program Files\Java\jre1.8.0_241\bin\java.exe"
set EQUINOX_LAUNCHER_PATH="C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\eclipse-modeling-2020-03-
R-win32-x86_64\eclipse\plugins\org.eclipse.equinox.launcher_1.5.700.v20200207-2156.jar"
set TEST_WORKSPACE_PATH=C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\ws
set XML_MODELS="%TEST_WORKSPACE_PATH%\OCLChecker_Tests\OCLCHECKER-TV-040-
001\file1.xml;%TEST_WORKSPACE_PATH%\OCLChecker_Tests\OCLCHECKER-TV-040-001\file2.xml"
set OCL_RULES="/OCLChecker_Tests/OCLCHECKER-TV-040-001/OCLCHECKER-TV-040-
001.ocl;/OCLChecker_Tests/OCLCHECKER-TV-040-001/NewOCLFile.ocl"
set OUTPUT_FILE=%TEST_WORKSPACE_PATH%\OCLChecker_Tests\OCLCHECKER-TV-040-001\OCLCHECKER-TV-
040-001.result

echo Executing CheckTool
echo with ocls: %OCL_RULES%
echo on models: %XML_MODELS%
echo result in: %OUTPUT_FILE%

%JAVA_HOME% -cp %EQUINOX_LAUNCHER_PATH% org.eclipse.core.launcher.Main -data
%TEST_WORKSPACE_PATH% -application com.cnes.checktool.application --
provider=%TEST_WORKSPACE_PATH% --ocls=%OCL_RULES%  --xmlmodels=%XML_MODELS% --
output=%OUTPUT_FILE%
```

- Sample command line (linux) with ruleset file (replace blue values by local paths)::

```
#!/bin/bash
# This script launches OCL Checker tool on models
# Usage : ./oclChecker_launcher.sh

export JAVA_HOME=/usr/bin/java
export
EQUINOX_LAUNCHER_PATH="/home/lv/Tools/Eclipse/eclipse/plugins/org.eclipse.equinox.launcher_1.5
.700.v20200207-2156.jar"
export WORKSPACE_PATH=/home/lv/Tools/Eclipse/ws_test
export
XML_MODELS="$WORKSPACE_PATH/example/example_sparcv8/module.xml;$WORKSPACE_PATH/example/example
_sparcv8/module2.xml;$WORKSPACE_PATH/example/example_sparcv8/module3.xml"
export
RULE_SETS="/example/example_sparcv8/example_sparcv8.ruleset;/example/example_sparcv8/another.r
uleset"
export OUTPUT_FILE=$WORKSPACE_PATH/example/example_sparcv8/output.result

echo Executing CheckTool
echo with rulesets: $RULE_SETS
echo on models: $XML_MODELS
echo result in: $OUTPUT_FILE

$JAVA_HOME  -cp  $EQUINOX_LAUNCHER_PATH org.eclipse.core.launcher.Main  -data  $WORKSPACE_PATH  -
application com.cnes.checktool.application --provider=$WORKSPACE_PATH --rulesets=$RULE_SETS   -
-xmlmodels=$XML_MODELS --output=$OUTPUT_FILE
```

- Sample output of the command line

```
Executing CheckTool
with                    ocls:                   "/OCLChecker_Tests/OCLCHECKER-TV-040-001/OCLCHECKER-TV-040-
001.ocl;/OCLChecker_Tests/OCLCHECKER-TV-040-001/NewOCLFile.ocl"
on          models:          "C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\ws\OCLChecker_Tests\OCLCHECKER-TV-040-
001\file1.xml;C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\ws\OCLChecker_Tests\OCLCHECKER-TV-040-001\file2.xml"
result in: C:\ahaugomm\CNES\OCLChecker_v1.1.0\env\ws\OCLChecker_Tests\OCLCHECKER-TV-040-001\OCLCHECKER-TV-
040-001.result
10:07:36 INFO   Started checking process
10:07:36 INFO   Preparing temporary working space
10:07:37 INFO   Importing rulesets
10:07:37 INFO   Importing model files
10:07:37 INFO   Checking files
10:07:37 INFO   Evaluating rule file OCLCHECKER-TV-040-001.ocl
10:07:42 INFO   Evaluating rule file NewOCLFile.ocl
10:07:42 INFO   Done
[WARN] Problem while deactivating CDOViewProviderRegistryImpl
java.lang.InterruptedException
        at        java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireSharedInterruptibly(Unknown
Source)
        at java.util.concurrent.Semaphore.acquire(Unknown Source)
        at org.eclipse.net4j.util.lifecycle.Lifecycle.lock(Lifecycle.java:310)
        at org.eclipse.net4j.util.lifecycle.Lifecycle.internalDeactivate(Lifecycle.java:118)
        at
org.eclipse.net4j.util.lifecycle.ShareableLifecycle.internalDeactivate(ShareableLifecycle.java:52)
        at org.eclipse.net4j.util.lifecycle.Lifecycle.deactivate(Lifecycle.java:168)
        at org.eclipse.net4j.util.lifecycle.LifecycleUtil.deactivate(LifecycleUtil.java:235)
        at org.eclipse.net4j.util.lifecycle.LifecycleUtil.deactivate(LifecycleUtil.java:225)
        at org.eclipse.net4j.util.lifecycle.LifecycleUtil.deactivate(LifecycleUtil.java:251)
        at org.eclipse.emf.internal.cdo.bundle.Activator$Implementation.doStop(Activator.java:129)
        at org.eclipse.emf.internal.cdo.bundle.Activator$Implementation.stop(Activator.java:99)
        at org.eclipse.osgi.internal.framework.BundleContextImpl$4.run(BundleContextImpl.java:899)
        at org.eclipse.osgi.internal.framework.BundleContextImpl$4.run(BundleContextImpl.java:1)
        at java.security.AccessController.doPrivileged(Native Method)
        at org.eclipse.osgi.internal.framework.BundleContextImpl.stop(BundleContextImpl.java:891)
        at org.eclipse.osgi.internal.framework.EquinoxBundle.stopWorker0(EquinoxBundle.java:1029)
        at
org.eclipse.osgi.internal.framework.EquinoxBundle$EquinoxModule.stopWorker(EquinoxBundle.java:370)
        at org.eclipse.osgi.container.Module.doStop(Module.java:658)
        at org.eclipse.osgi.container.Module.stop(Module.java:520)
        at
org.eclipse.osgi.container.ModuleContainer$ContainerStartLevel.decStartLevel(ModuleContainer.java:1885)
        at
org.eclipse.osgi.container.ModuleContainer$ContainerStartLevel.doContainerStartLevel(ModuleContainer.java:
1760)
        at org.eclipse.osgi.container.SystemModule.stopWorker(SystemModule.java:275)
        at
org.eclipse.osgi.internal.framework.EquinoxBundle$SystemBundle$EquinoxSystemModule.stopWorker(EquinoxBundl
e.java:202)
        at org.eclipse.osgi.container.Module.doStop(Module.java:658)
        at org.eclipse.osgi.container.Module.stop(Module.java:520)
        at org.eclipse.osgi.container.SystemModule.stop(SystemModule.java:207)
        at
org.eclipse.osgi.internal.framework.EquinoxBundle$SystemBundle$EquinoxSystemModule$1.run(EquinoxBundle.jav
a:220)
        at java.lang.Thread.run(Unknown Source)
```

Note : The large warning message (in blue above) comes from another plugin and is systematically displayed. It does not correspond to OCLChecker execution , please ignore it.

10. APPENDIX 2: PROCEDURE TO RELEASE AN UPDATE SITE FOR A NEW VERSION OF XSD

5 projects should be available in the ed

- 3 plugins for the metamodel
  - com.xx.yy <= *containing the metamodel*
  - com.xx.yy.edit <= *generated from metamodel*
  - com.xx.yy.editor <= *generated from metamodel*
- One feature (= installable unit, referencing the plugins)
  - com.xx.yy.feature <= *description of the feature to install (list of plugins, versions, dependencies, copyright, license)*
- One update site (= installer)
  - com.xx.yy.updatesite <= *Eclipse installation unit. To be built and distributed for a new release of the metamodel.*

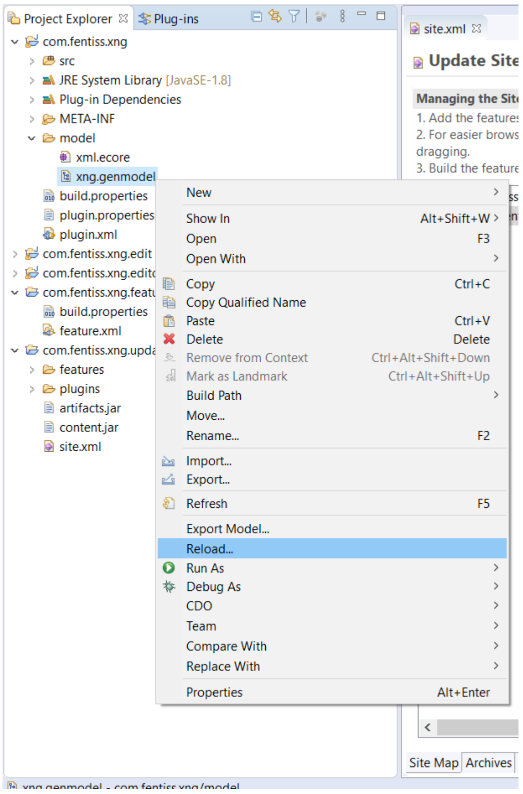
For each evolution of the XSD , a new update site of the metamodel should be released and install into ECLIPSE in order to be taken into account in OCL rules used by OCL Checker.
This update site build is done is several steps :

- Reload the metamodel from the new version of XSD
- Regenerate associated code
- Increase all version numbers (plugins, feature, update site)
- Re-build the update site for new version
- Export of the update site

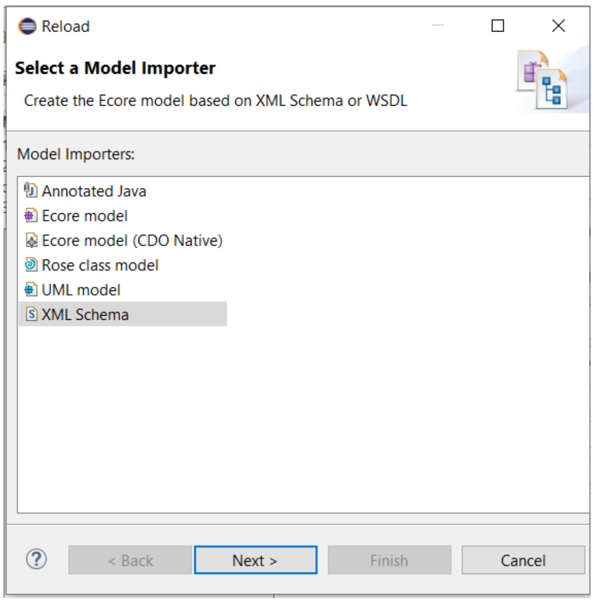
These steps are detailed below.

- **Import the updated XSD into the .genmodel file**

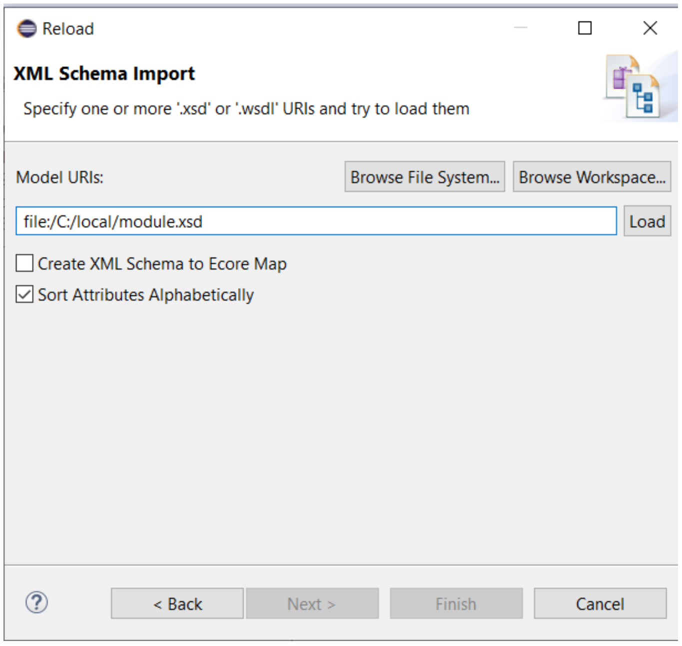
1. From the plugin containing the ecore and genmodel file : right click on .genmodel file > Reload



2. Select XML schema > Next



3. Browse your local updated XSD file
Click Load, then Next > Finish



Reload

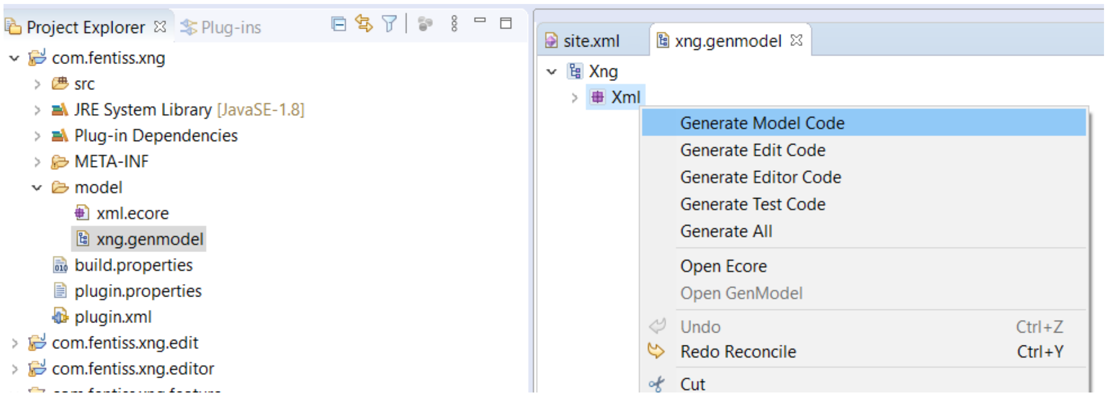
XML Schema Import

Specify one or more '.xsd' or '.wsdl' URIs and try to load them

Model URIs: file:/C:/local/module.xsd

☐ Create XML Schema to Ecore Map
☑ Sort Attributes Alphabetically

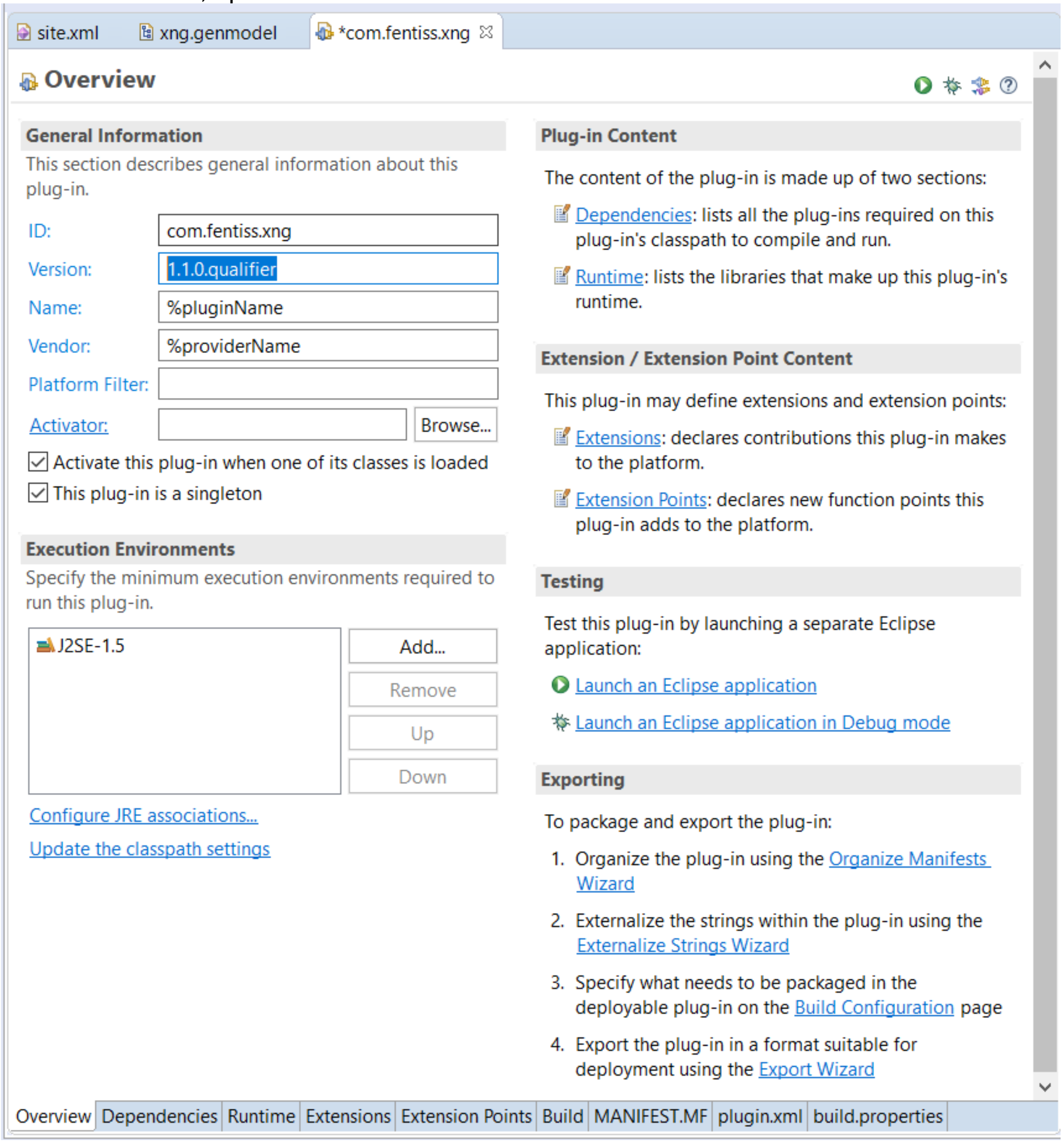
- **Regenerate code from .genmodel file**
  1. Open .genmodel file. From main ePackage > right click and press successively :
     - Generate model code,
     - Generate Edit code,
     - Generate editor code.



- **Increase version numbers in plugins, feature and update site**

  1. On each of the 3 plugins XX, XX.edit and XX.editor :
     open plugin.xml file

In “Overview” tab, update version number and save



site.xml | xng.genmodel | *com.fentiss.xng

Overview

General Information

This section describes general information about this plug-in.

ID: com.fentiss.xng
Version: 1.1.0.qualifier
Name: %pluginName
Vendor: %providerName
Platform Filter:
Activator:

☑ Activate this plug-in when one of its classes is loaded
☑ This plug-in is a singleton

Execution Environments

Specify the minimum execution environments required to run this plug-in.

J2SE-1.5

Configure JRE associations...
Update the classpath settings

Plug-in Content

The content of the plug-in is made up of two sections:

Dependencies: lists all the plug-ins required on this plug-in's classpath to compile and run.

Runtime: lists the libraries that make up this plug-in's runtime.

Extension / Extension Point Content

This plug-in may define extensions and extension points:

Extensions: declares contributions this plug-in makes to the platform.

Extension Points: declares new function points this plug-in adds to the platform.

Testing

Test this plug-in by launching a separate Eclipse application:

Launch an Eclipse application
Launch an Eclipse application in Debug mode

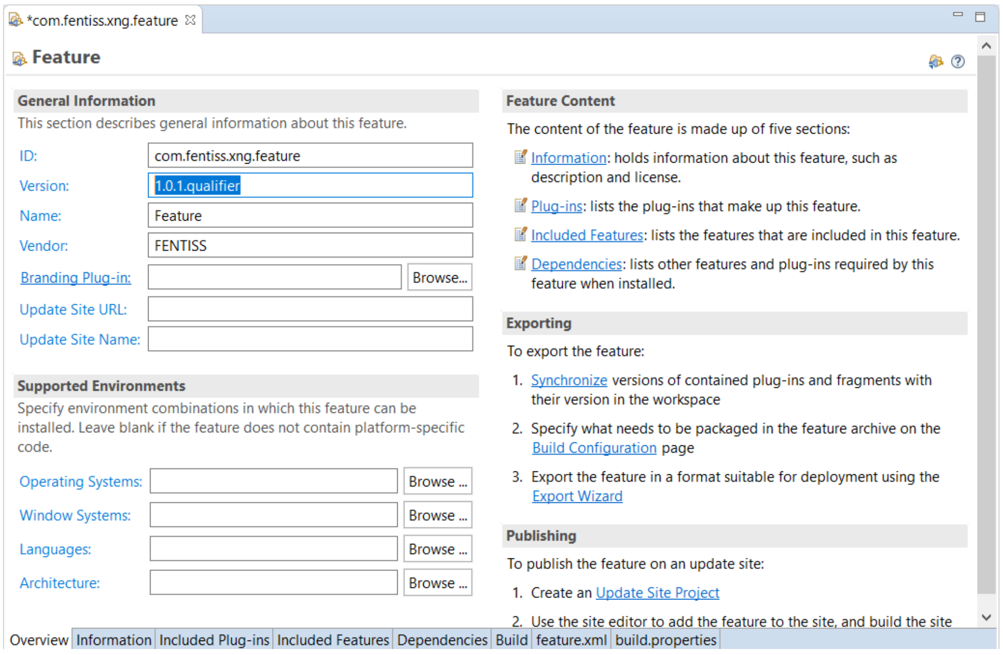
Exporting

To package and export the plug-in:

1. Organize the plug-in using the Organize Manifests Wizard
2. Externalize the strings within the plug-in using the Externalize Strings Wizard
3. Specify what needs to be packaged in the deployable plug-in on the Build Configuration page
4. Export the plug-in in a format suitable for deployment using the Export Wizard

2. Open feature.xml file in the feature project

- In “Overview” tab, increase version number



*com.fentiss.xng.feature

Feature

General Information
This section describes general information about this feature.

ID:	com.fentiss.xng.feature
Version:	1.0.1.qualifier
Name:	Feature
Vendor:	FENTISS
Branding Plug-in:	
Update Site URL:	
Update Site Name:	

Supported Environments
Specify environment combinations in which this feature can be installed. Leave blank if the feature does not contain platform-specific code.

Operating Systems: / Window Systems: / Languages: / Architecture:

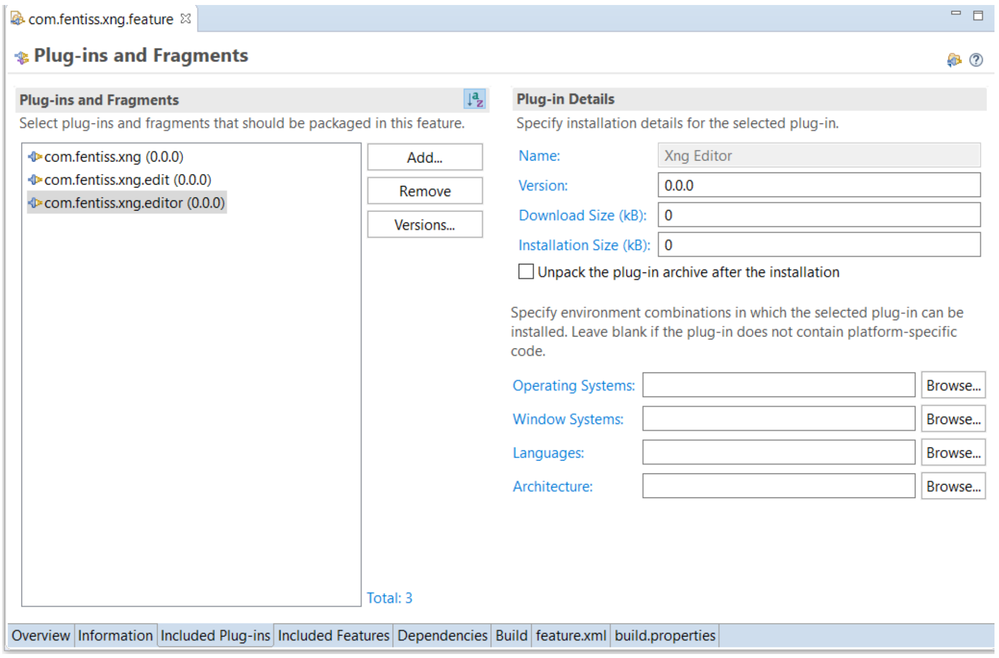
Feature Content
The content of the feature is made up of five sections:
- Information: holds information about this feature, such as description and license.
- Plug-ins: lists the plug-ins that make up this feature.
- Included Features: lists the features that are included in this feature.
- Dependencies: lists other features and plug-ins required by this feature when installed.

Exporting
To export the feature:
1. Synchronize versions of contained plug-ins and fragments with their version in the workspace
2. Specify what needs to be packaged in the feature archive on the Build Configuration page
3. Export the feature in a format suitable for deployment using the Export Wizard

Publishing
To publish the feature on an update site:
1. Create an Update Site Project
2. Use the site editor to add the feature to the site, and build the site

Overview | Information | Included Plug-ins | Included Features | Dependencies | Build | feature.xml | build.properties

- In “Included Plug-ins” tab, check that no version is mentioned



com.fentiss.xng.feature

Plug-ins and Fragments

Plug-ins and Fragments
Select plug-ins and fragments that should be packaged in this feature.

- com.fentiss.xng (0.0.0)
- com.fentiss.xng.edit (0.0.0)
- com.fentiss.xng.editor (0.0.0)

Total: 3

Plug-in Details
Specify installation details for the selected plug-in.

Name:	Xng Editor
Version:	0.0.0
Download Size (kB):	0
Installation Size (kB):	0

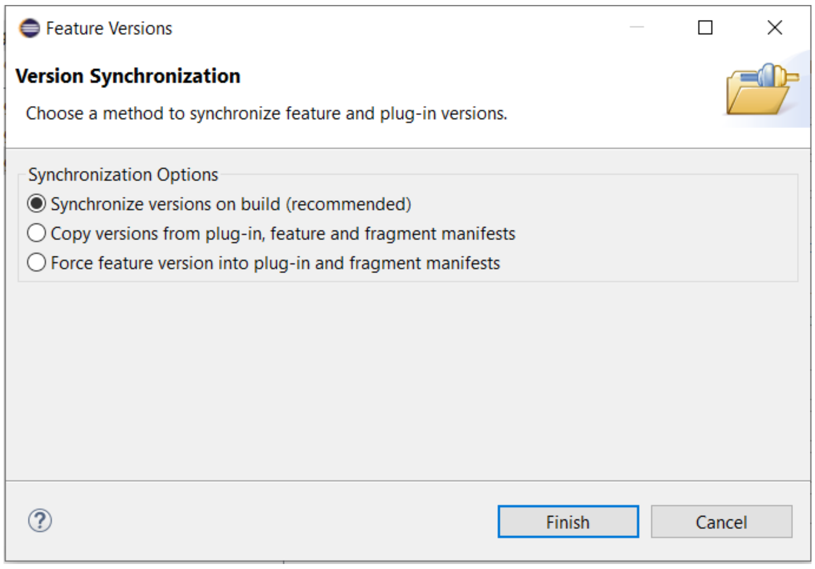
☐ Unpack the plug-in archive after the installation

Specify environment combinations in which the selected plug-in can be installed. Leave blank if the plug-in does not contain platform-specific code.

Operating Systems: / Window Systems: / Languages: / Architecture:

Overview | Information | Included Plug-ins | Included Features | Dependencies | Build | feature.xml | build.properties

In case the version is not 0.0.0 for each plugin,
- Select each plugin in the list
- Click “Versions…”
- Select “Synchronize versions on build”



Feature Versions

Version Synchronization

Choose a method to synchronize feature and plug-in versions.

Synchronization Options

[✓ Synchronize versions on build (recommended)]

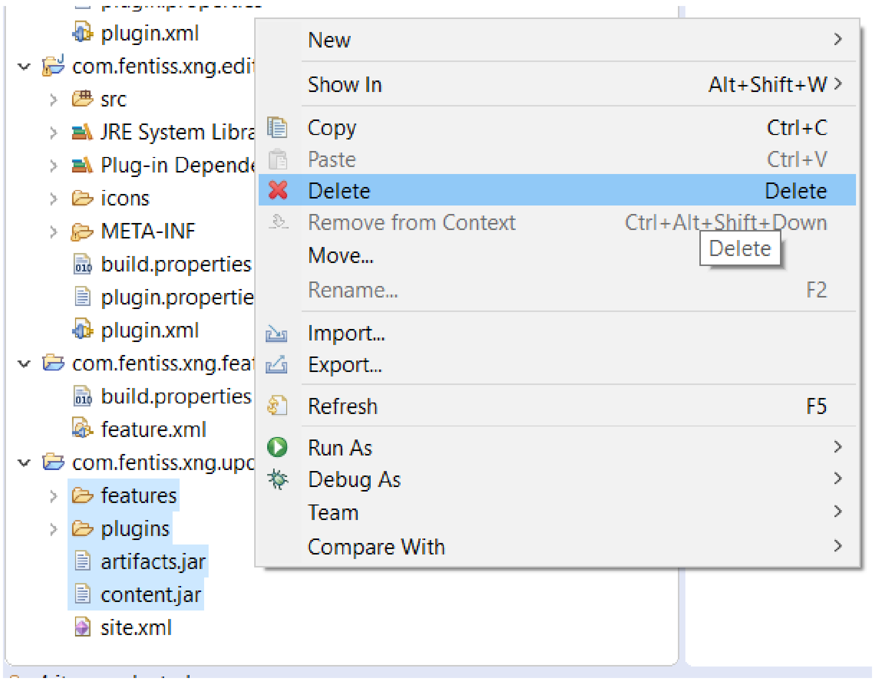
Copy versions from plug-in, feature and fragment manifests

Force feature version into plug-in and fragment manifests

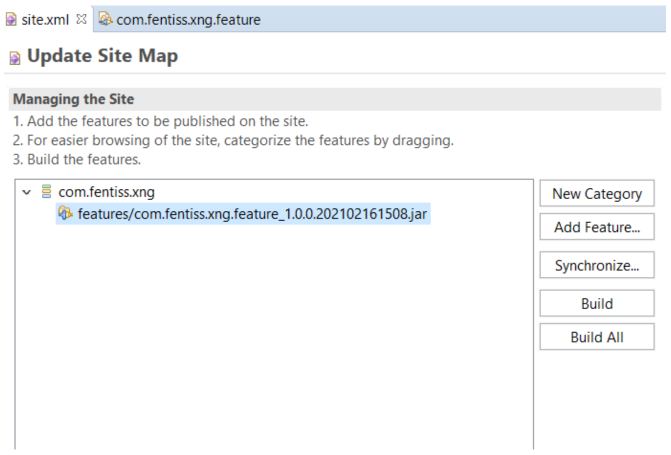
- Finish

3. From the update site project created previously

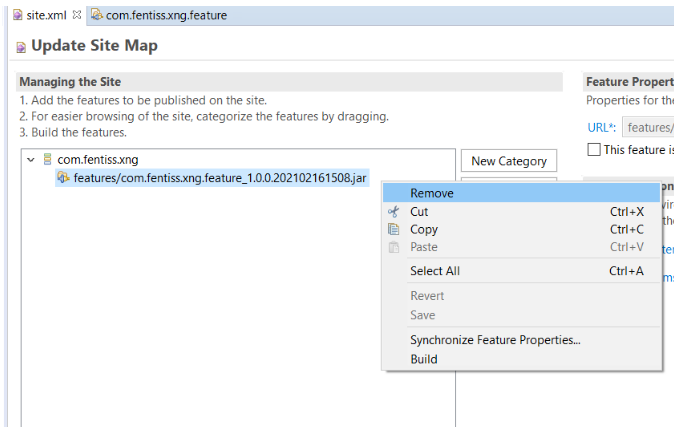
- If the update site project contains more than site.xml file, delete all other files (they are generated during BUILD)



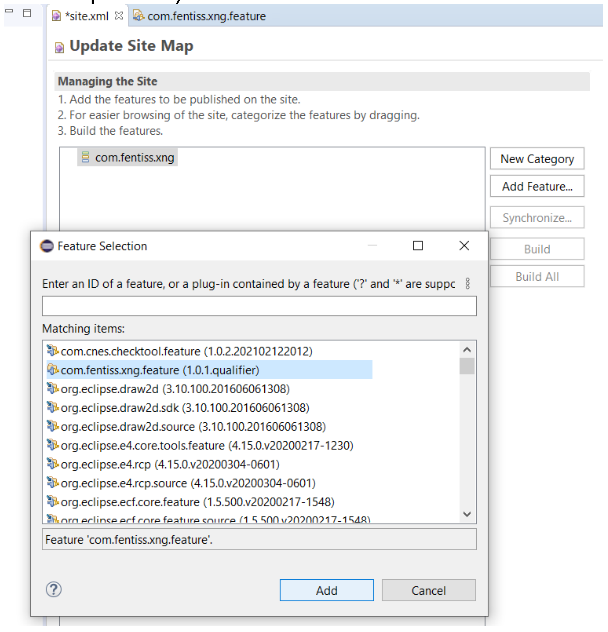
- Open site.xml
  The content should look like the following : a category, and a feature with a version number in which the ".qualifier" is replaced by the build date (year month day time).


site.xml
com.fentiss.xng.feature
Update Site Map
Managing the Site
1. Add the features to be published on the site.
2. For easier browsing of the site, categorize the features by dragging.
3. Build the features.
com.fentiss.xng
features/com.fentiss.xng.feature_1.0.0.202102161508.jar
New Category
Add Feature...
Synchronize...
Build
Build All

- Select the feature and right click > remove (to remove this previous version of the feature from last build)



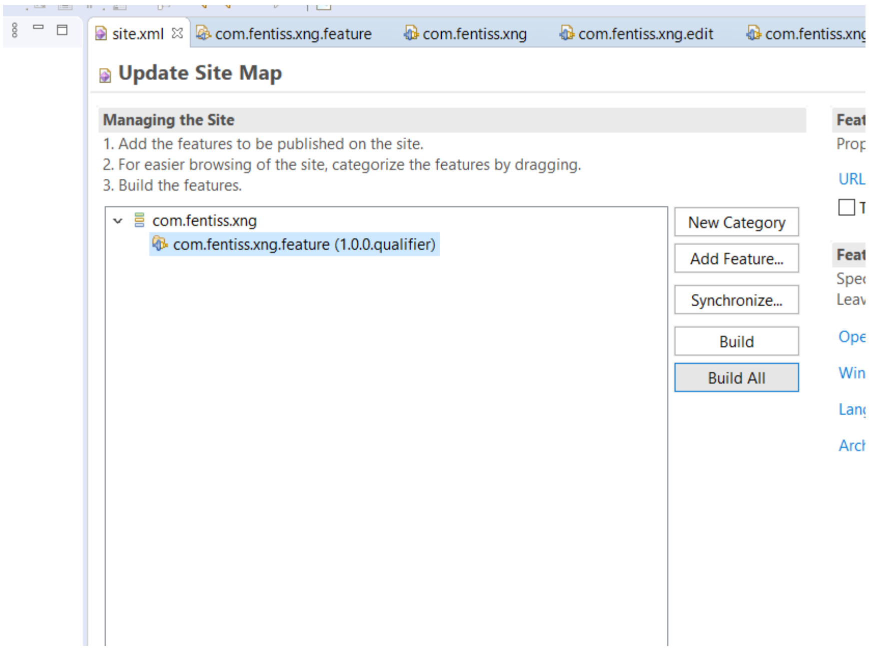
- Select the category and click “Add feature…” to select the new version of the feature (with suffix .qualifier)


*site.xml
com.fentiss.xng.feature
Update Site Map
Managing the Site
1. Add the features to be published on the site.
2. For easier browsing of the site, categorize the features by dragging.
3. Build the features.
com.fentiss.xng
New Category
Add Feature...
Synchronize...
Build
Build All
Feature Selection
Enter an ID of a feature, or a plug-in contained by a feature ('?' and '*' are suppc
Matching items:
com.cnes.checktool.feature (1.0.2.202102122012)
com.fentiss.xng.feature (1.0.1.qualifier)
org.eclipse.draw2d (3.10.100.201606061308)
org.eclipse.draw2d.sdk (3.10.100.201606061308)
org.eclipse.draw2d.source (3.10.100.201606061308)
org.eclipse.e4.core.tools.feature (4.15.0.v20200217-1230)
org.eclipse.e4.rcp (4.15.0.v20200304-0601)
org.eclipse.e4.rcp.source (4.15.0.v20200304-0601)
org.eclipse.ecf.core.feature (1.5.500.v20200217-1548)
org.eclipse.ecf.core.feature.source (1.5.500.v20200217-1548)
Feature 'com.fentiss.xng.feature'.
Add
Cancel

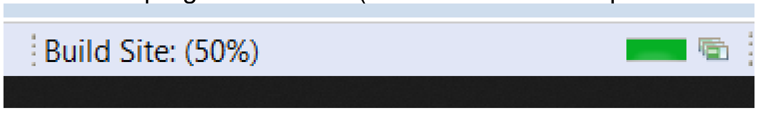
- Save the site.xml
- If you store in version control system , this is the version to be stored (before BUILD)

- **Build the update site and release**

12. Open site.xml from the update site project
13. Click "Build All"

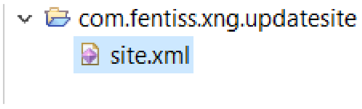


14. Wait until the progress is 100% (on the bottom of Eclipse main window)

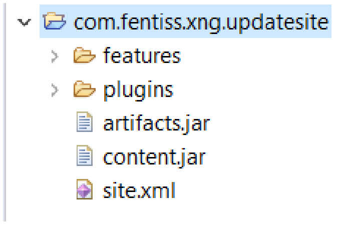


15. Look at update site project in ProjectExplorer view

Before build

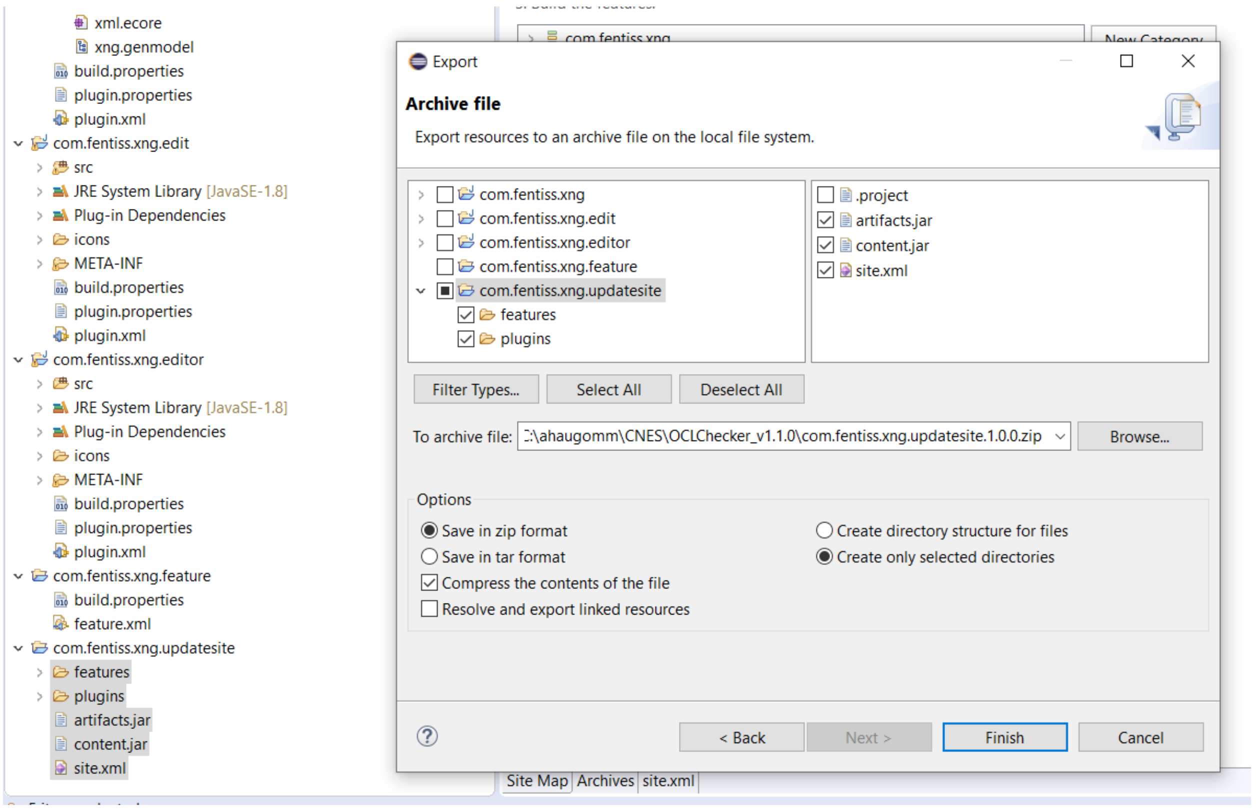


After build



16. In order to share this update site, create an archive of all files in the project
    WARNING : The archive must contain directly the site.xml file, no upper level folder.

    If you create the archive from Eclipse
    - Select the files (site.xml, artifacts.jar, contents.jar, features folder , plugins folder) and click File > Export > Archive file > Next
    - Check option "Create only selected directories"



```
xml.ecore
xng.genmodel
build.properties
plugin.properties
plugin.xml
com.fentiss.xng.edit
  src
  JRE System Library [JavaSE-1.8]
  Plug-in Dependencies
  icons
  META-INF
  build.properties
  plugin.properties
  plugin.xml
com.fentiss.xng.editor
  src
  JRE System Library [JavaSE-1.8]
  Plug-in Dependencies
  icons
  META-INF
  build.properties
  plugin.properties
  plugin.xml
com.fentiss.xng.feature
  build.properties
  feature.xml
com.fentiss.xng.updatesite
  features
  plugins
  artifacts.jar
  content.jar
  site.xml
```

Export

Archive file

Export resources to an archive file on the local file system.

☐ com.fentiss.xng	☐ .project
☐ com.fentiss.xng.edit	☑ artifacts.jar
☐ com.fentiss.xng.editor	☑ content.jar
☐ com.fentiss.xng.feature	☑ site.xml
■ com.fentiss.xng.updatesite	
☑ features	
☑ plugins	

Filter Types... Select All Deselect All

To archive file: :\ahaugomm\CNES\OCLChecker_v1.1.0\com.fentiss.xng.updatesite.1.0.0.zip Browse...

Options

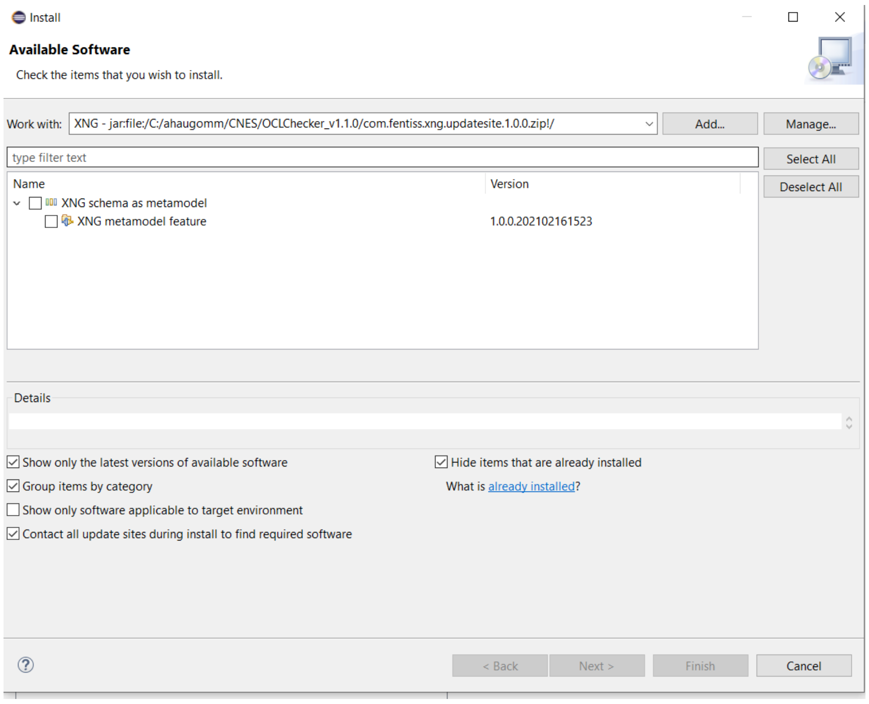
- ● Save in zip format
- ○ Save in tar format
- ☑ Compress the contents of the file
- ☐ Resolve and export linked resources
- ○ Create directory structure for files
- ● Create only selected directories

< Back Next > Finish Cancel

Site Map | Archives | site.xml

- ○ Finish

17. Open the resulting archive to ensure that you have site.xml at top level.
18. Test the installation from your eclipse platform
    - Help > Install new software > Add.. > Archive ..
    - Select archive of the new update site
    - Check that the two names in the check boxes correspond to
        - For highest level , the label of the category from site.xml in update site project
        - For lower level, the feature name (in feature.xml)



Work with: XNG - jar:file:/C:/ahaugomm/CNES/OCLChecker_v1.1.0/com.fentiss.xng.updatesite.1.0.0.zip!/

type filter text

Name	Version
☐ XNG schema as metamodel	
☐ XNG metamodel feature	1.0.0.202102161523

☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed
What is already installed?

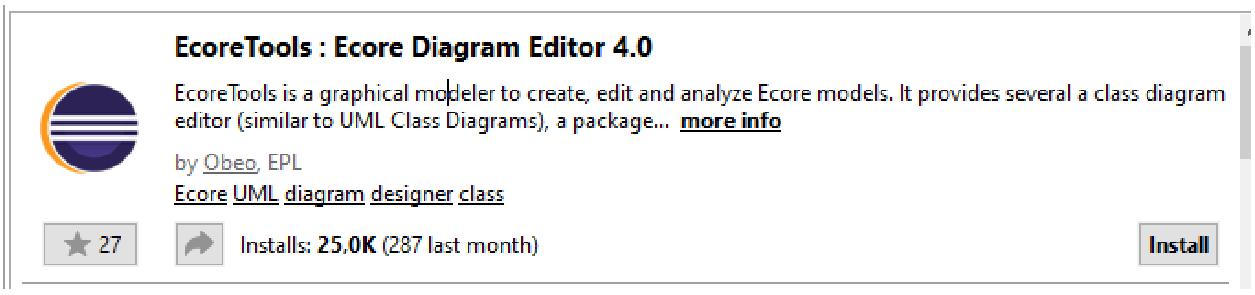
- Click next twice
- The text from the license correspond to what has been provided in feature.xml > information tab.

19. Share the archive file of the update site to users.

    Note : the update site can also be shared as a URL.
    Extract the archive in a folder and provide the URL pointing to this folder (again, the folder containing directly site.xml file)

11. APPENDIX 3 : VISUALIZE METAMODEL AS A DIAGRAM

Prerequisite:



EcoreTools : Ecore Diagram Editor 4.0

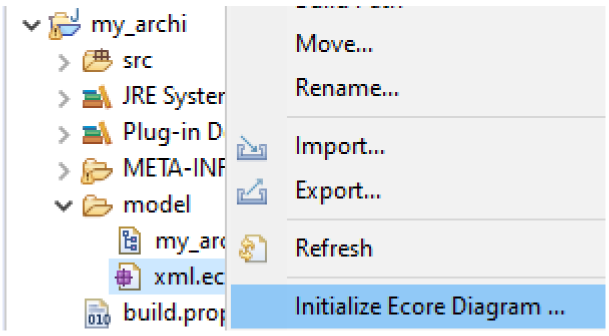
EcoreTools is a graphical modeler to create, edit and analyze Ecore models. It provides several a class diagram editor (similar to UML Class Diagrams), a package... more info

by Obeo, EPL

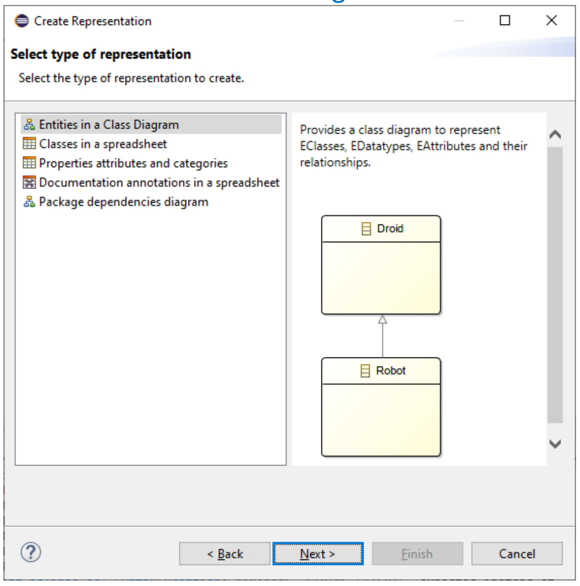
Ecore UML diagram designer class

Here is the plugin documentation: https://www.eclipse.org/ecoretools/doc/.

1. Select the .ecore file
2. Right click on it and go on Initialize Ecore Diagram …

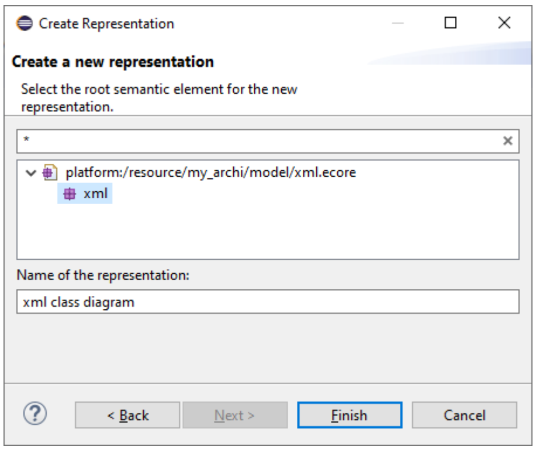


3. Enter diagram name
4. Click on Next
5. Select Entities in a Class Diagram


Create Representation
Select type of representation
Select the type of representation to create.
Entities in a Class Diagram
Classes in a spreadsheet
Properties attributes and categories
Documentation annotations in a spreadsheet
Package dependencies diagram
Provides a class diagram to represent EClasses, EDatatypes, EAttributes and their relationships.
Droid
Robot
< Back
Next >
Finish
Cancel

6. Click on Next

7. Select the related ecore



Create Representation

Create a new representation

Select the root semantic element for the new representation.

*

platform:/resource/my_archi/model/xml.ecore
xml

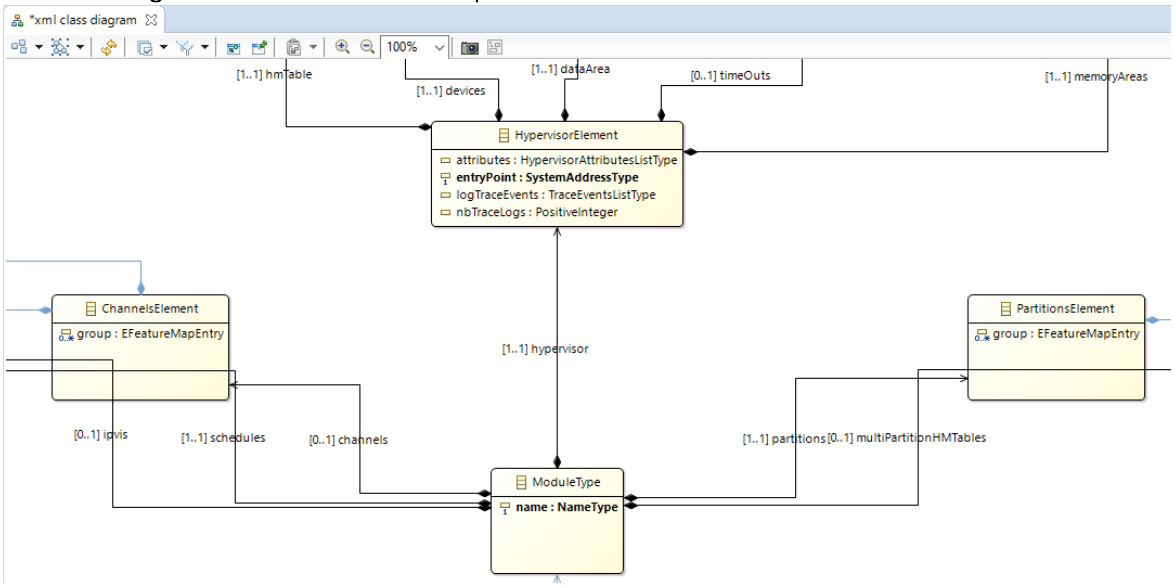
Name of the representation:

xml class diagram

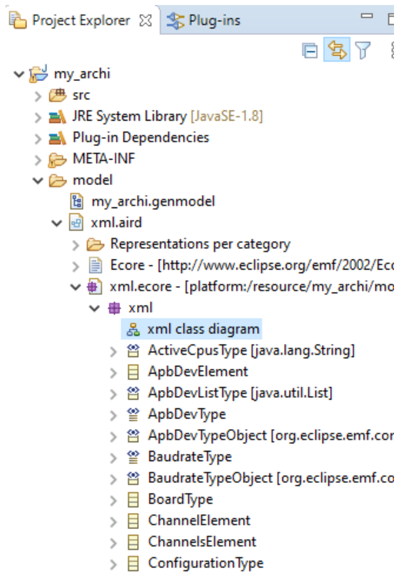
< Back | Next > | Finish | Cancel

8. Click on Finish
9. A new .aird file appear in the project, click on it.
10. Double-click in the tab to initialize diagram, wait. Move in the diagram to see different elements.

    Here is a diagram extract from an example:



Navigation in the different elements is also available in the project explorer:



```
my_archi
  src
  JRE System Library [JavaSE-1.8]
  Plug-in Dependencies
  META-INF
  model
    my_archi.genmodel
    xml.aird
      Representations per category
      Ecore - [http://www.eclipse.org/emf/2002/Ec
      xml.ecore - [platform:/resource/my_archi/mo
        xml
          xml class diagram
          ActiveCpusType [java.lang.String]
          ApbDevElement
          ApbDevListType [java.util.List]
          ApbDevType
          ApbDevTypeObject [org.eclipse.emf.cor
          BaudrateType
          BaudrateTypeObject [org.eclipse.emf.co
          BoardType
          ChannelElement
          ChannelsElement
          ConfigurationType
```

VERSIONS

Issue	*Date*	*Relevant Information*
1.0	2020-10-12	• First issue for the OCL conf checker tool User Guide
1.1	2021-03-12	OCL Conf checker v1.1 (update installation prerequisites, appendices)