FORMAL METHODS IN SOFTWARE ENGINEERING LABORATORY
FAKULTAS ILMU KOMPUTER UNIVERSITAS INDONESIA

Tutorial Feature Diagram Transformation to ABS

Muhammad Irfan Fadhillah

irfan.fadhillah10@gmail.com

Table of Contents

1. Installation

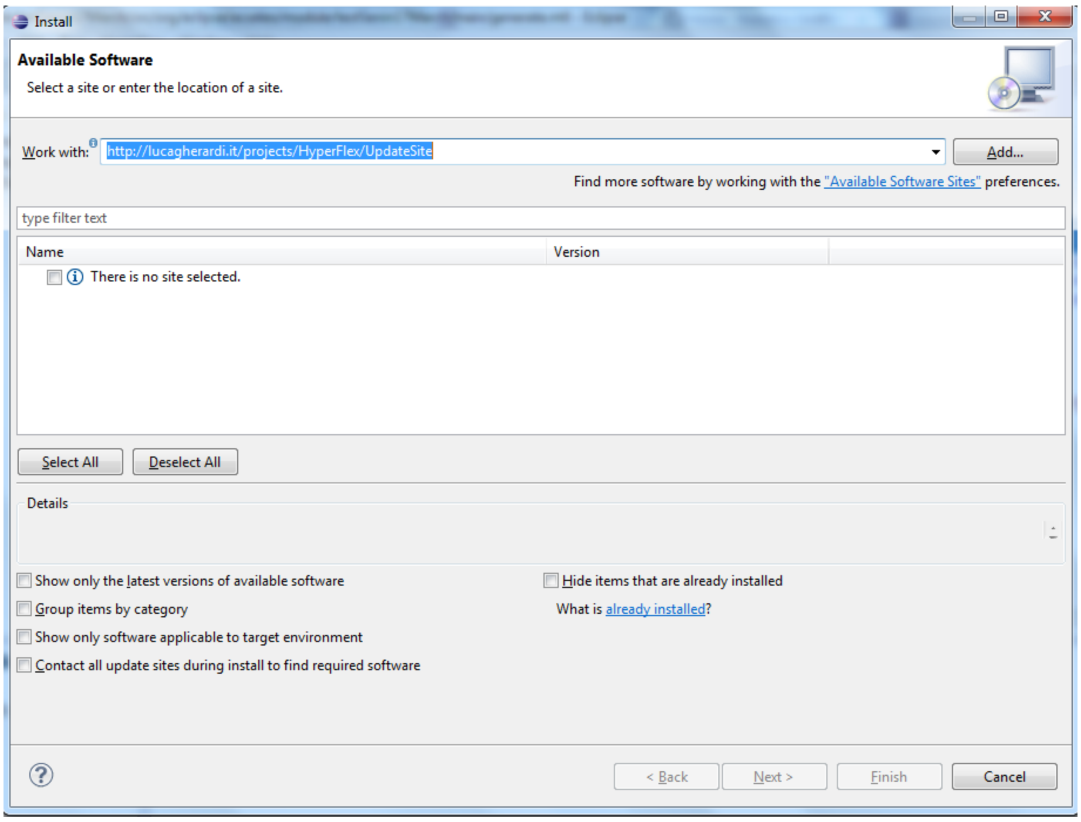
*note: **this procedure can be followed with the assumption we have had the Eclipse IDE Luna.**

Before we do the transformation process of feature diagram ABS in Eclipse IDE, there are several Eclipse plug-ins which are required to help the transformation process:

1.1 HyperFlex Feature Diagram Editor

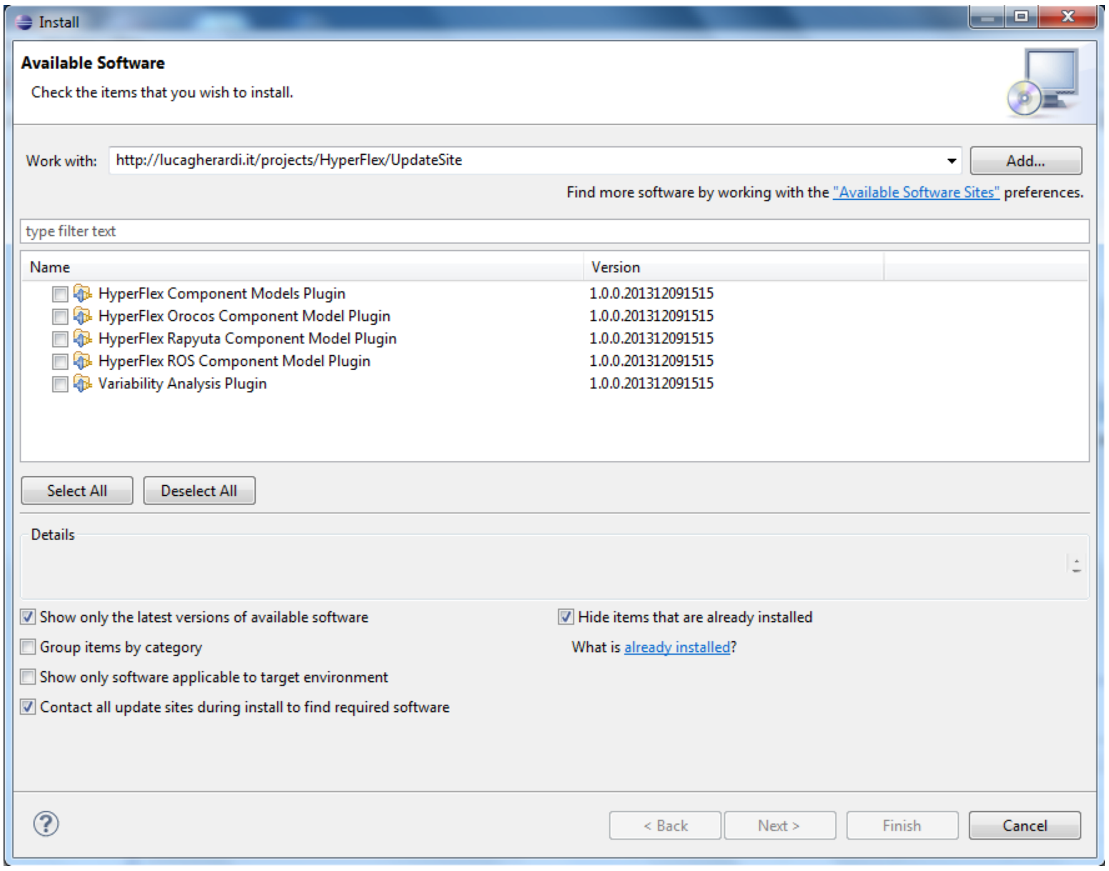
We need this tool to make a feature diagram that will become an input for the transformation process. We can install this plug-in via its update site.

After we have installed those plug-ins, we can continue to install HyperFlex. In Eclipse choose Menu **Help → Install New Software …** and put this link http://lucagherardi.it/projects/HyperFlex/UpdateSite on `Work with:` column then choose the `Add` button.



Install
Available Software
Select a site or enter the location of a site.
Work with:
http://lucagherardi.it/projects/HyperFlex/UpdateSite
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version
There is no site selected.
Select All
Deselect All
Details
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Show only software applicable to target environment
Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel

If there is no plug-ins showed on the window, unflag `Group item by category`, so the window will show the plug-ins like in this picture.


Install

Available Software

Check the items that you wish to install.

Work with: http://lucagherardi.it/projects/HyperFlex/UpdateSite

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☐ HyperFlex Component Models Plugin	1.0.0.201312091515
☐ HyperFlex Orocos Component Model Plugin	1.0.0.201312091515
☐ HyperFlex Rapyuta Component Model Plugin	1.0.0.201312091515
☐ HyperFlex ROS Component Model Plugin	1.0.0.201312091515
☐ Variability Analysis Plugin	1.0.0.201312091515

Details

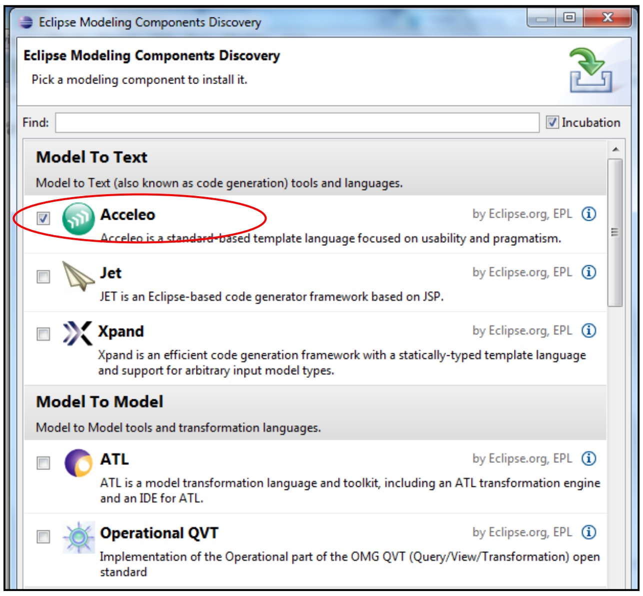
- [✓] Show only the latest versions of available software
- [ ] Group items by category
- [ ] Show only software applicable to target environment
- [✓] Contact all update sites during install to find required software
- [✓] Hide items that are already installed

What is already installed?

Please choose the `Variability Analysis Plugin` and then choose `Next`.

1.2 Acceleo Model to Text Transformation (M2T)

Rules for UML class diagram translation to ABS are implemented in Acceleo M2T. Thus, to run the translation process, we need to install plug-in Acceleo M2T. From Eclipse, choose Menu **Help →Install Modeling Components → Choose Acceleo in Model to Text Part → Click Finish.**


Eclipse Modeling Components Discovery

Eclipse Modeling Components Discovery

Pick a modeling component to install it.

Find: [✓] Incubation

Model To Text

Model to Text (also known as code generation) tools and languages.

- [✓] **Acceleo** — by Eclipse.org, EPL — Acceleo is a standard-based template language focused on usability and pragmatism.
- [ ] **Jet** — by Eclipse.org, EPL — JET is an Eclipse-based code generator framework based on JSP.
- [ ] **Xpand** — by Eclipse.org, EPL — Xpand is an efficient code generation framework with a statically-typed template language and support for arbitrary input model types.

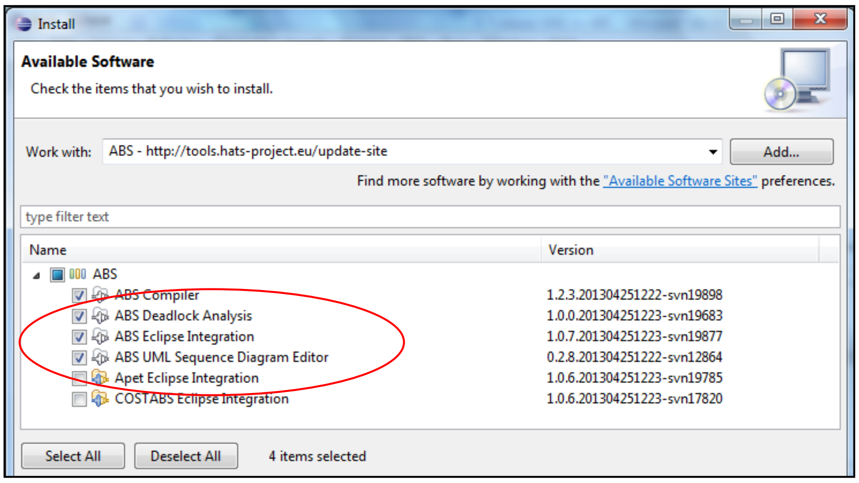
Model To Model

Model to Model tools and transformation languages.

- [ ] **ATL** — by Eclipse.org, EPL — ATL is a model transformation language and toolkit, including an ATL transformation engine and an IDE for ATL.
- [ ] **Operational QVT** — by Eclipse.org, EPL — Implementation of the Operational part of the OMG QVT (Query/View/Transformation) open standard

1.3 ABS Tool

Output from translation process is ABS model. To build and debug the ABS model, we need to install ABS tool eclipse plug-in from HATS Project. It is available ini http://docs.abs-models.org/update-site.



Install

Available Software

Check the items that you wish to install.

Work with: ABS - http://tools.hats-project.eu/update-site

Find more software by working with the "Available Software Sites" preferences.

type filter text

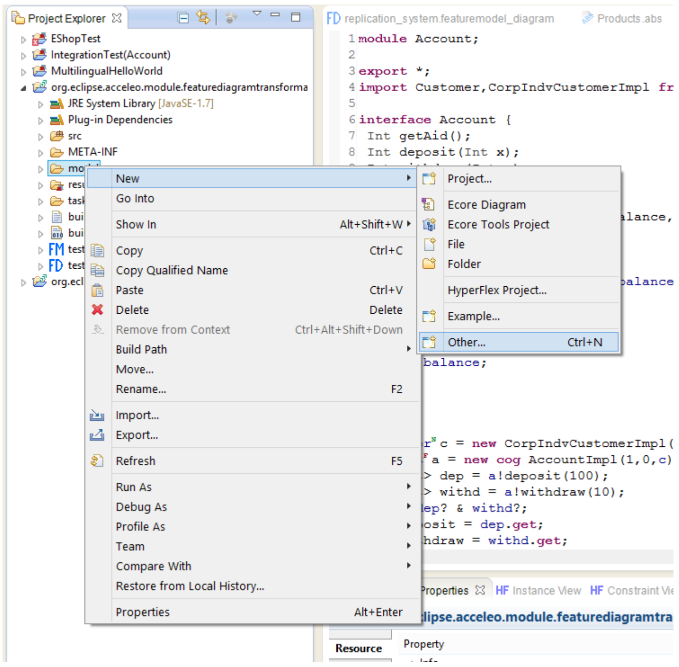
Name	Version
ABS	
☑ ABS Compiler	1.2.3.201304251222-svn19898
☑ ABS Deadlock Analysis	1.0.0.201304251223-svn19683
☑ ABS Eclipse Integration	1.0.7.201304251223-svn19877
☑ ABS UML Sequence Diagram Editor	0.2.8.201304251222-svn12864
☐ Apet Eclipse Integration	1.0.6.201304251223-svn19785
☐ COSTABS Eclipse Integration	1.0.6.201304251223-svn17820

2. Create Feature Diagram

This section will explain how to make a feature diagram on HyperFlex Feature Diagram Editor. To follow steps in this section you have to do the steps in section 1.1

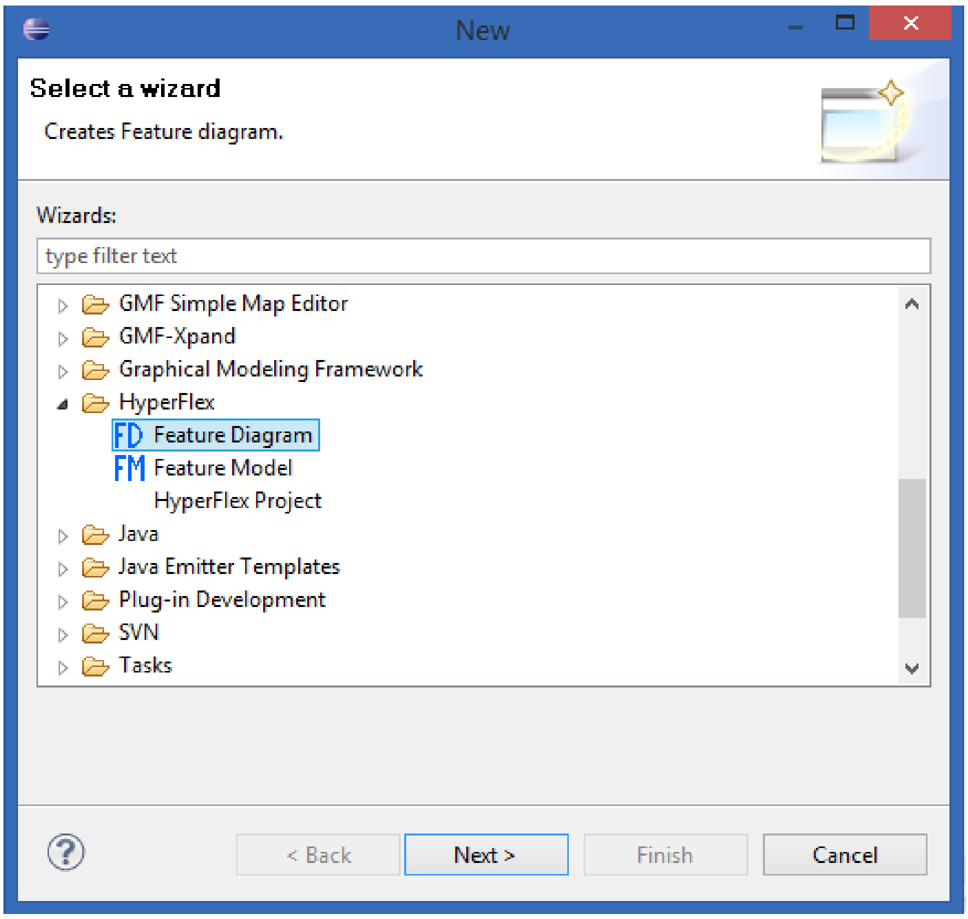
Steps:

1. [Optional] Open Perspective of HyperFlex, choose Menu `Window->Open Perspective->Other->HyperFlex->` Click `OK`
2. Right click on a project or folder that will contain this feature diagram then choose menu `New->Other` …

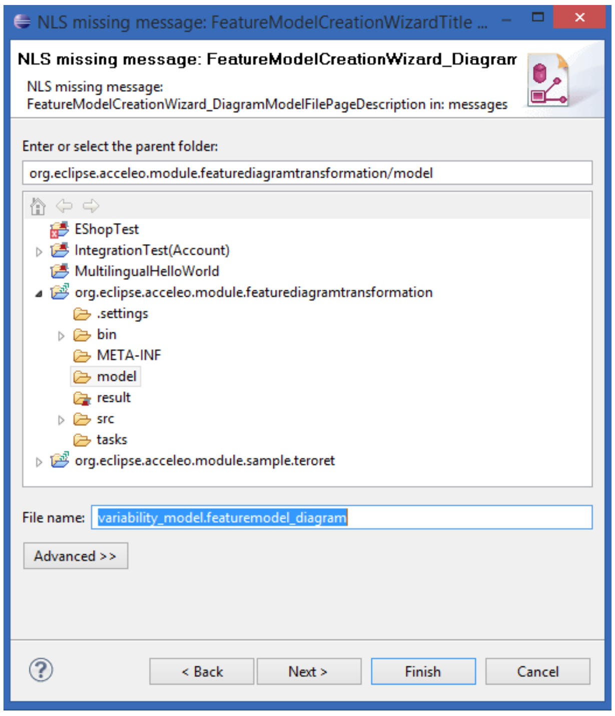


```
module Account;

export *;
import Customer,CorpIndvCustomerImpl fr

interface Account {
 Int getAid();
 Int deposit(Int x);
```

3. After that, the window will show up, then choose `HyperFlex->Feature Diagram->` Click `Next`.



4. On the wizard enter the feature diagram name (with its extension `.featuremodel_diagram`) and also its feature model (with its extension `.featuremodel`). And we can finish it with choosing `Finish`. This wizard will create two files in our workspace first is a diagram file with extension `.featuremodel_diagram` and second is a HyperFlex feature model file with extension `.featuremodel`. HyperFlex feature model file will be an input in the next transformation process.


NLS missing message: FeatureModelCreationWizardTitle ...

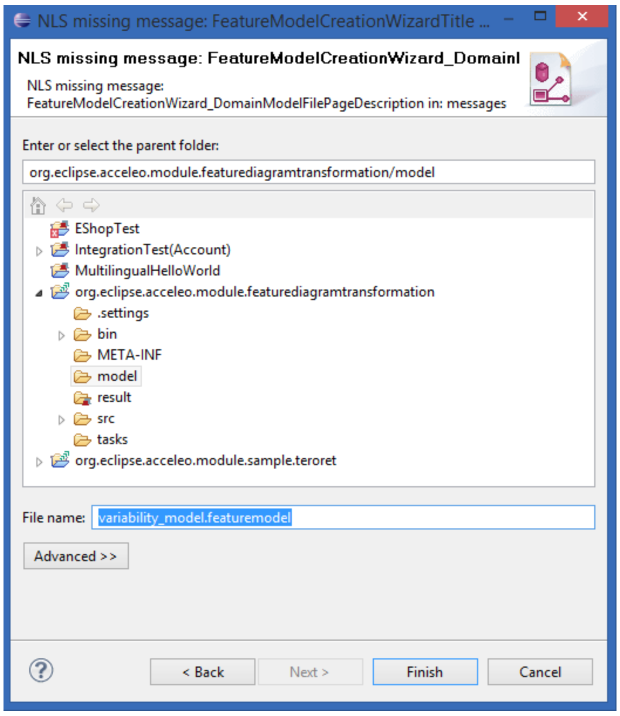
NLS missing message: FeatureModelCreationWizard_Diagram

NLS missing message: FeatureModelCreationWizard_DiagramModelFilePageDescription in: messages

Enter or select the parent folder: org.eclipse.acceleo.module.featurediagramtransformation/model

- EShopTest
- IntegrationTest(Account)
- MultilingualHelloWorld
- org.eclipse.acceleo.module.featurediagramtransformation
  - .settings
  - bin
  - META-INF
  - model
  - result
  - src
  - tasks
- org.eclipse.acceleo.module.sample.teroret

File name: variability_model.featuremodel_diagram


NLS missing message: FeatureModelCreationWizardTitle ...

NLS missing message: FeatureModelCreationWizard_DomainI

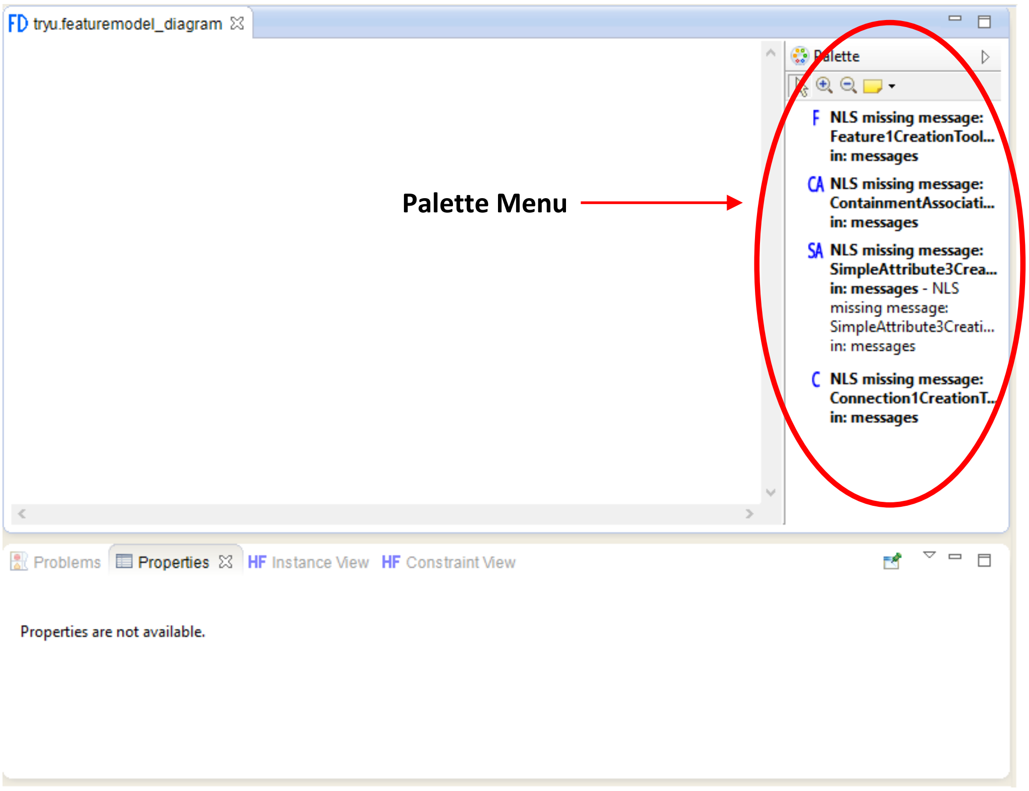
NLS missing message: FeatureModelCreationWizard_DomainModelFilePageDescription in: messages

Enter or select the parent folder: org.eclipse.acceleo.module.featurediagramtransformation/model

- EShopTest
- IntegrationTest(Account)
- MultilingualHelloWorld
- org.eclipse.acceleo.module.featurediagramtransformation
  - .settings
  - bin
  - META-INF
  - model
  - result
  - src
  - tasks
- org.eclipse.acceleo.module.sample.teroret

File name: variability_model.featuremodel

5. If the workspace has been showed like in the picture below then the making of feature diagram can be done with drag and drop for its component from the palette that has been provided.



F NLS missing message: Feature1CreationTool... in: messages

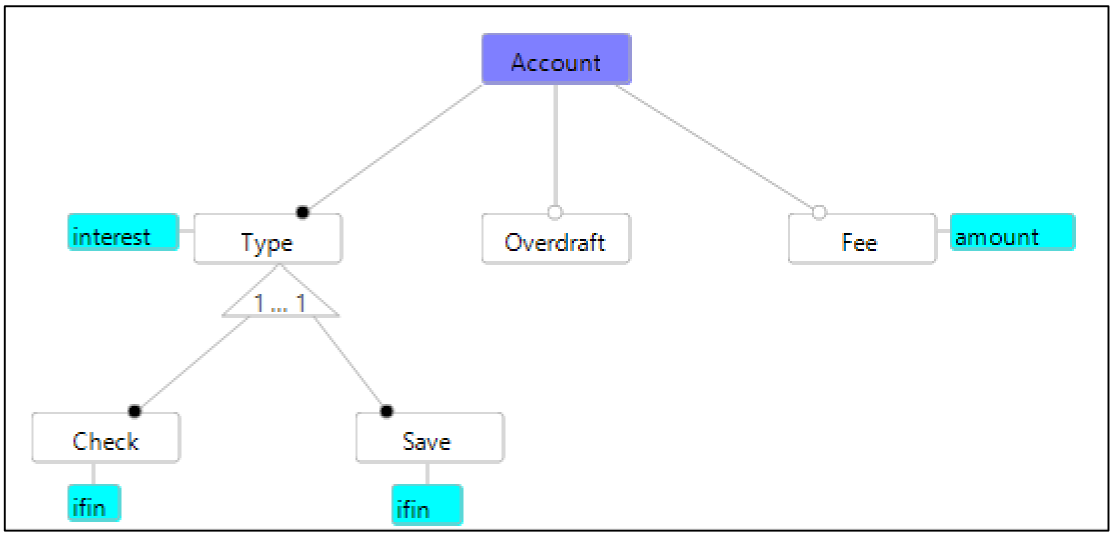
CA NLS missing message: ContainmentAssociati... in: messages

SA NLS missing message: SimpleAttribute3Crea... in: messages - NLS missing message: SimpleAttribute3Creati... in: messages

C NLS missing message: Connection1CreationT... in: messages

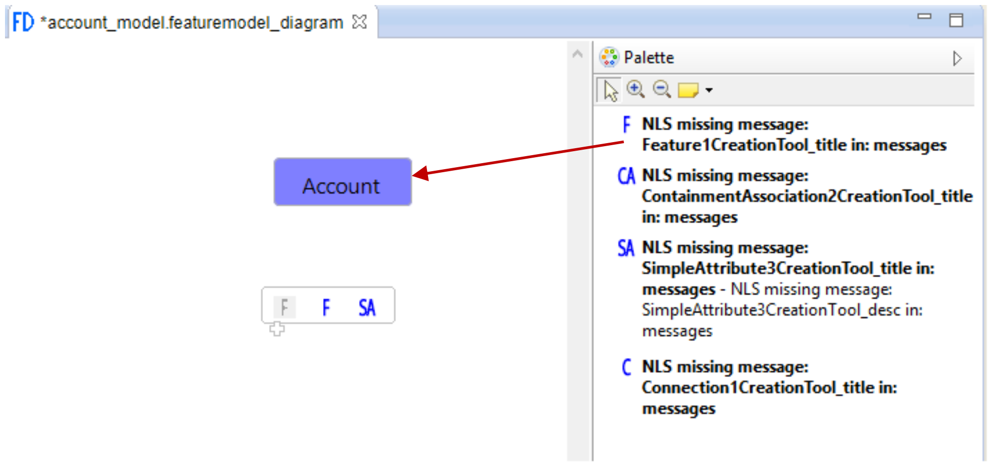
Properties are not available.

Supposed we want to create Feature diagram for Account Case Study, as shown in the picture below:

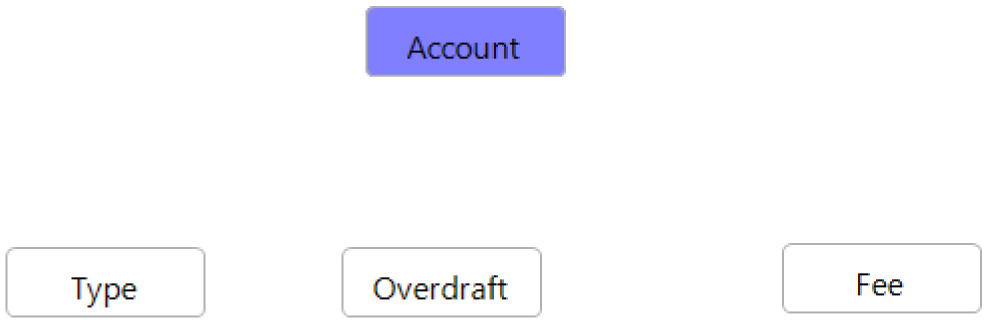


Steps:

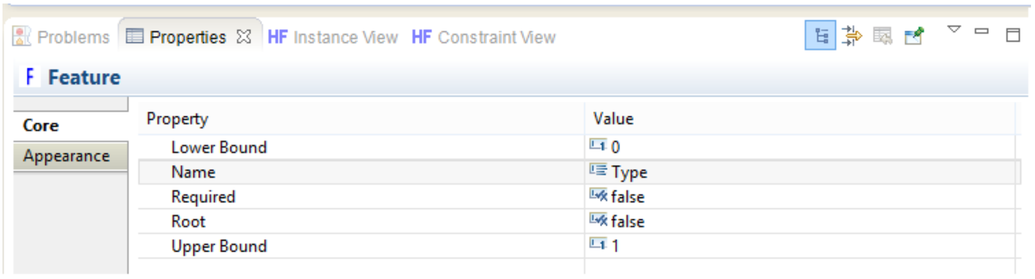
1. Drag nodes `Feature` from Palette menu to Workspace. This action will give a result of blue box which indicates this feature is a root feature.


FD *account_model.featuremodel_diagram
Account
F F SA
Palette
F NLS missing message: Feature1CreationTool_title in: messages
CA NLS missing message: ContainmentAssociation2CreationTool_title in: messages
SA NLS missing message: SimpleAttribute3CreationTool_title in: messages - NLS missing message: SimpleAttribute3CreationTool_desc in: messages
C NLS missing message: Connection1CreationTool_title in: messages

2. After we have created the root feature with the same way, we can create other features.



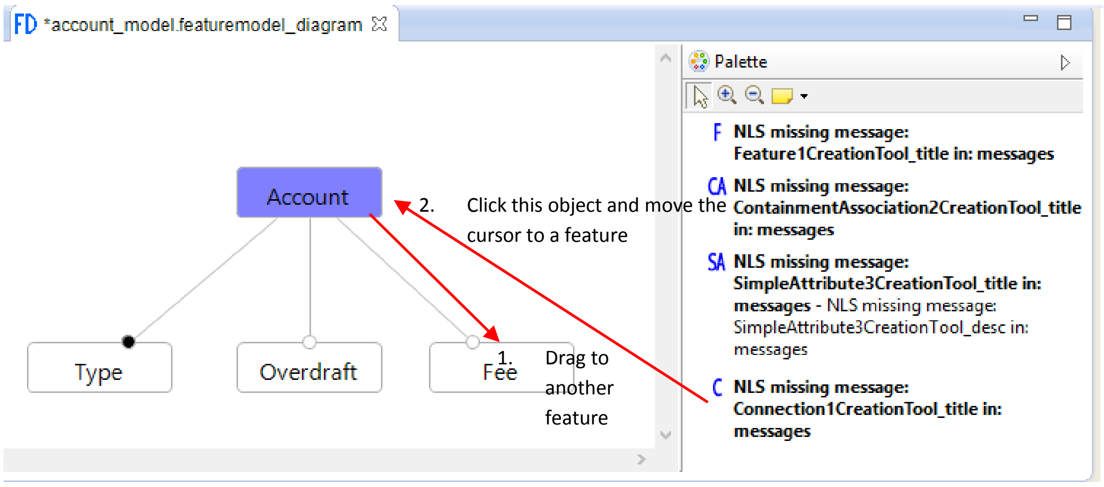
If we want to change characteristics of these non-root features, we can change it in their properties like in this picture.



Property	Value
Lower Bound	0
Name	Type
Required	false
Root	false
Upper Bound	1

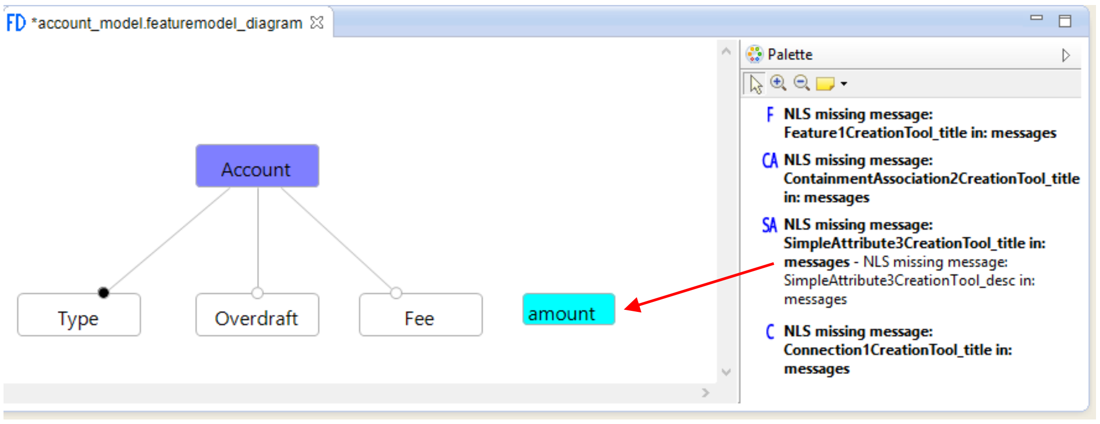
As a default condition these features are optional features. We can know it from `Required` field, if it has false condition then it is an optional feature. If it has true condition then it is a mandatory feature.

3. Click `Connection` from Palette menu and move cursor to a feature and drag it to another feature to make a connection between two features.



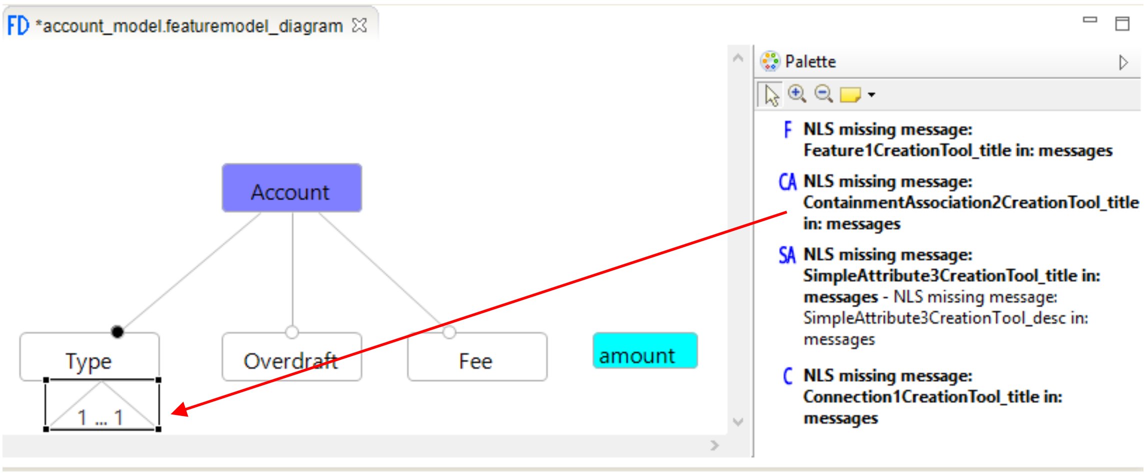
*account_model.featuremodel_diagram
Account
Type
Overdraft
Fee
2. Click this object and move the cursor to a feature
1. Drag to another feature
Palette
F NLS missing message: Feature1CreationTool_title in: messages
CA NLS missing message: ContainmentAssociation2CreationTool_title in: messages
SA NLS missing message: SimpleAttribute3CreationTool_title in: messages - NLS missing message: SimpleAttribute3CreationTool_desc in: messages
C NLS missing message: Connection1CreationTool_title in: messages

4. Drag `SimpleAttribute` node to create an attribute. This component is showed on a cyan box.



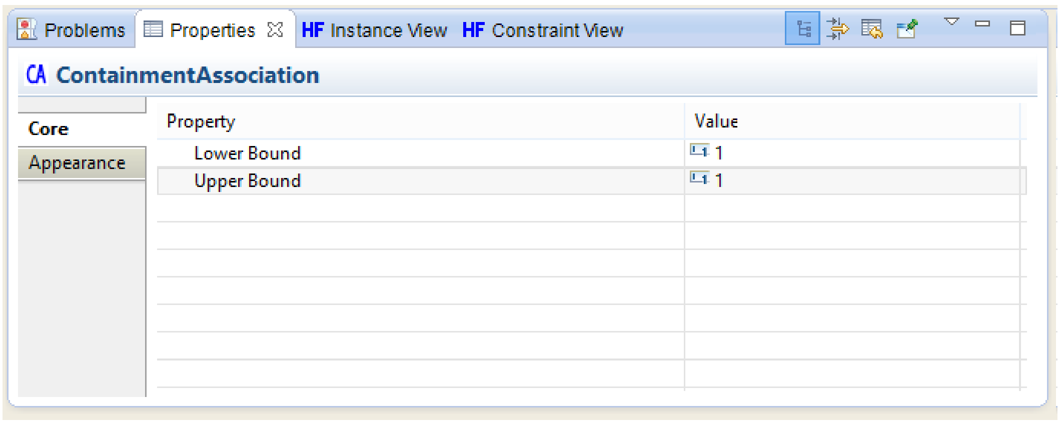
*account_model.featuremodel_diagram
Account
Type
Overdraft
Fee
amount
Palette
F NLS missing message: Feature1CreationTool_title in: messages
CA NLS missing message: ContainmentAssociation2CreationTool_title in: messages
SA NLS missing message: SimpleAttribute3CreationTool_title in: messages - NLS missing message: SimpleAttribute3CreationTool_desc in: messages
C NLS missing message: Connection1CreationTool_title in: messages

5. Click on `ContainmentAssociation` node to create a grouped feature and move your cursor to a feature. This is indicating that this feature will be a feature parent.



*account_model.featuremodel_diagram
Account
Type
Overdraft
Fee
amount
1 ... 1
Palette
F NLS missing message: Feature1CreationTool_title in: messages
CA NLS missing message: ContainmentAssociation2CreationTool_title in: messages
SA NLS missing message: SimpleAttribute3CreationTool_title in: messages - NLS missing message: SimpleAttribute3CreationTool_desc in: messages
C NLS missing message: Connection1CreationTool_title in: messages

We can change its properties to maintain lower and upper bound that will indicate its relation between parent and child features.


Problems | Properties | HF Instance View | HF Constraint View

ContainmentAssociation

Core
Appearance

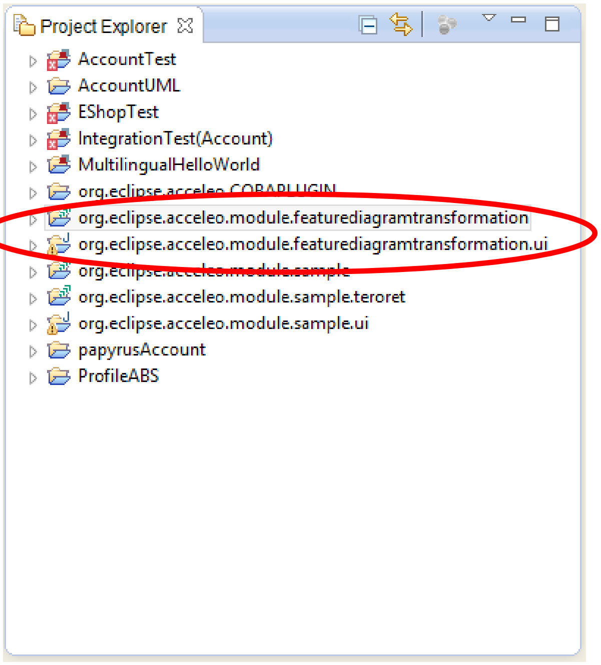
Property	Value
Lower Bound	1
Upper Bound	1

3. Deploying Transformation Rules as an Eclipse Plug-in

We need to import the transformation rules which is implemented in Acceleo template as an Eclipse project and deploy the project as a plug-in in Eclipse.

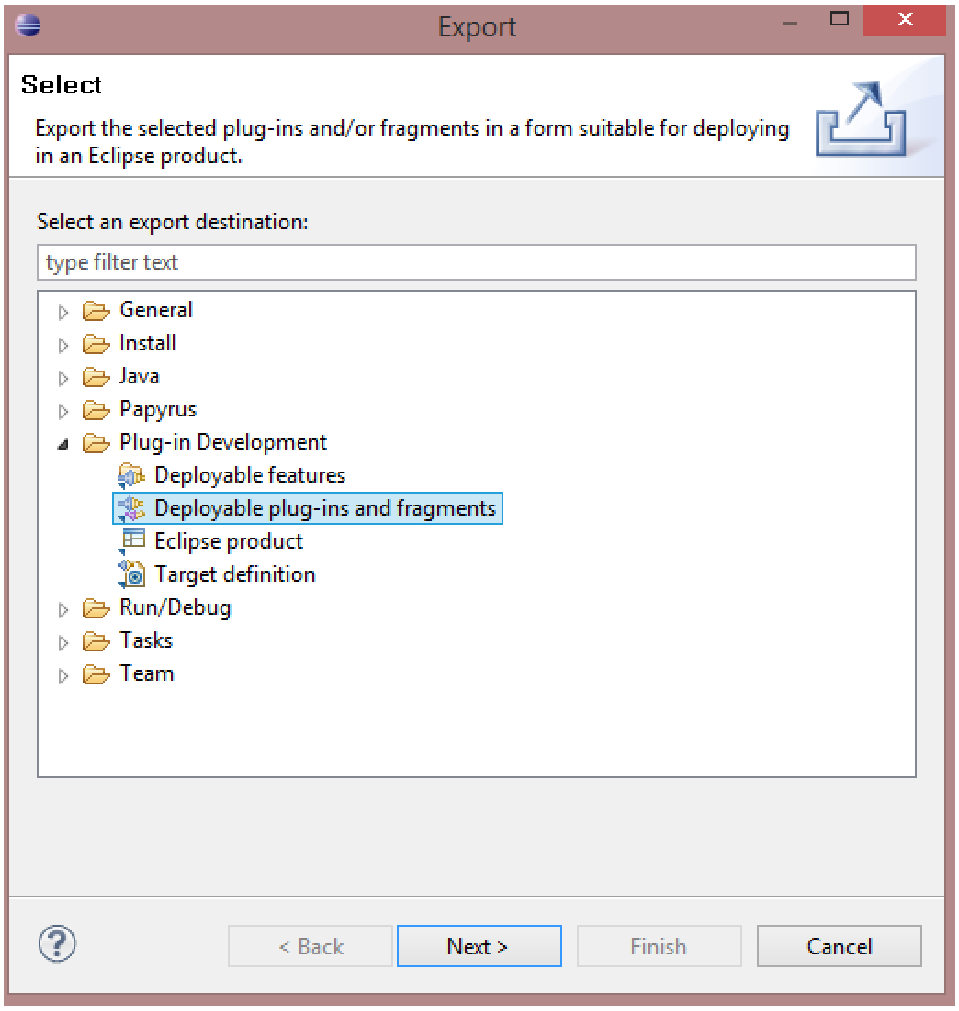
Steps:

1. Download file feature-ABS.zip from https://projects.ui.ac.id//svn/abs/PRICESTools/FeatureDiagram-ABS//
2. Import the project to Eclipse and make sure the **org.eclipse.acceleo.module.featurediagramtransformation** and **org.eclipse.acceleo.module.featurediagramtransformation.ui** projects are placed in Package Explorer and your workspace.

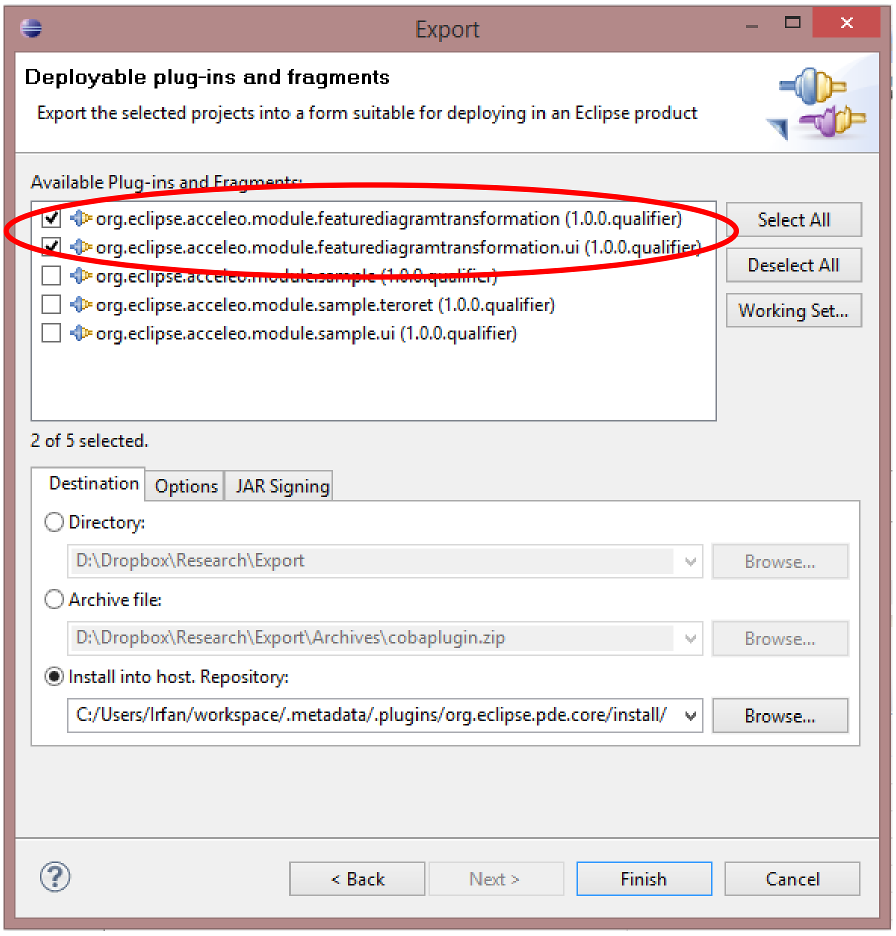

Project Explorer

- AccountTest
- AccountUML
- EShopTest
- IntegrationTest(Account)
- MultilingualHelloWorld
- org.eclipse.acceleo.CORAPLUGIN
- org.eclipse.acceleo.module.featurediagramtransformation
- org.eclipse.acceleo.module.featurediagramtransformation.ui
- org.eclipse.acceleo.module.sample
- org.eclipse.acceleo.module.sample.teroret
- org.eclipse.acceleo.module.sample.ui
- papyrusAccount
- ProfileABS

3. After you have imported the transformation rules in the projects, the next step is deploying the projects as a plug-in in Eclipse. What you need to do is export **org.eclipse.acceleo.module.featurediagramtransformation** and **org.eclipse.acceleo.module.featurediagramtransformation.ui** projects. **Right Click** in those projects **-> Export … -> Plug-in Development -> Deployable plug-ins and fragments -> Next**


Export
Select
Export the selected plug-ins and/or fragments in a form suitable for deploying in an Eclipse product.
Select an export destination:
type filter text
General
Install
Java
Papyrus
Plug-in Development
Deployable features
Deployable plug-ins and fragments
Eclipse product
Target definition
Run/Debug
Tasks
Team
< Back
Next >
Finish
Cancel

4. In the export wizard dialog select in this case ***Install into host. Repository***. This is depicted in the following screenshot. Make sure you choose **org.eclipse.acceleo.module.featurediagramtransformation** and **org.eclipse.acceleo.module.featurediagramtransformation.ui** projects in the wizard.



Export

Deployable plug-ins and fragments

Export the selected projects into a form suitable for deploying in an Eclipse product

Available Plug-ins and Fragments:

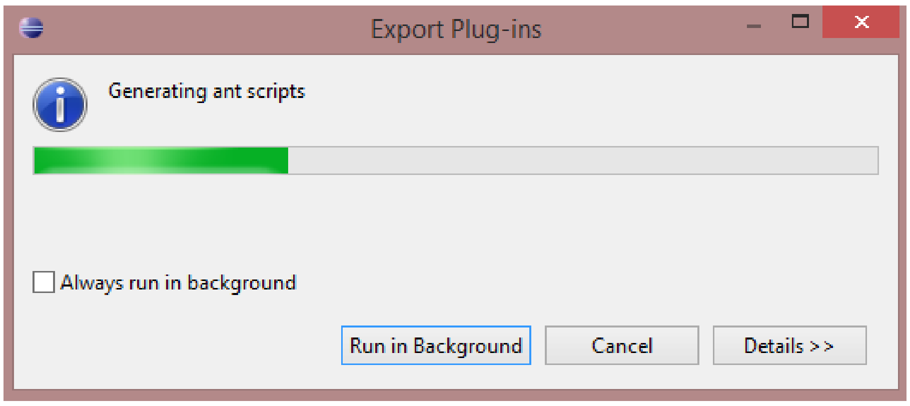
- [✓] org.eclipse.acceleo.module.featurediagramtransformation (1.0.0.qualifier)
- [✓] org.eclipse.acceleo.module.featurediagramtransformation.ui (1.0.0.qualifier)
- [ ] org.eclipse.acceleo.module.sample (1.0.0.qualifier)
- [ ] org.eclipse.acceleo.module.sample.teroret (1.0.0.qualifier)
- [ ] org.eclipse.acceleo.module.sample.ui (1.0.0.qualifier)

2 of 5 selected.

Destination | Options | JAR Signing

- ○ Directory: D:\Dropbox\Research\Export
- ○ Archive file: D:\Dropbox\Research\Export\Archives\cobaplugin.zip
- ● Install into host. Repository: C:/Users/Irfan/workspace/.metadata/.plugins/org.eclipse.pde.core/install/

5. Please wait while Eclipse install this plug-in

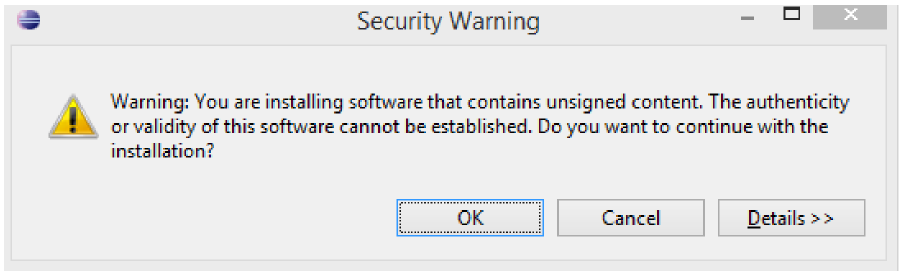


Export Plug-ins

Generating ant scripts

[ ] Always run in background

If there is a **Security Warning** like presented in this picture **Click OK.**


Security Warning

Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?

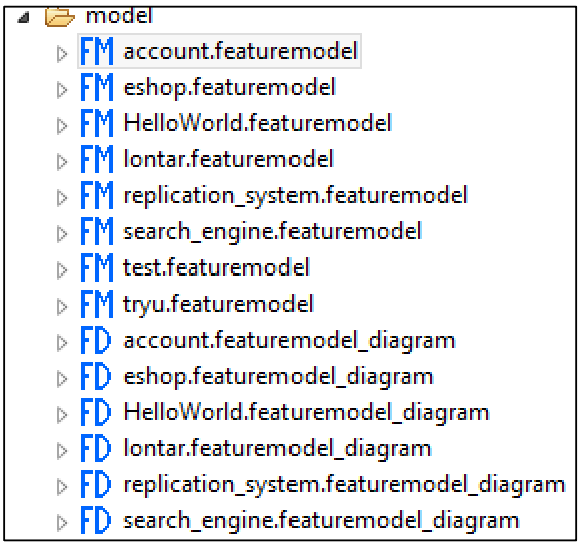
After the installation process is completed, you need to restart your Eclipse. After this restart your new plug-in is available in your Eclipse installation and ready to be used.

4. Transformation Process

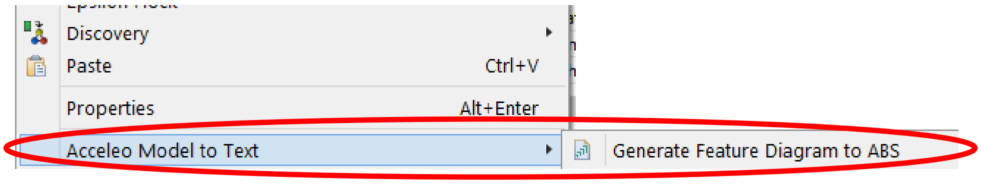
We have installed the transformation rules in the form of Eclipse plug-in. This step will explain how we can do the transformation process with the recent installed plug-in.

Steps:

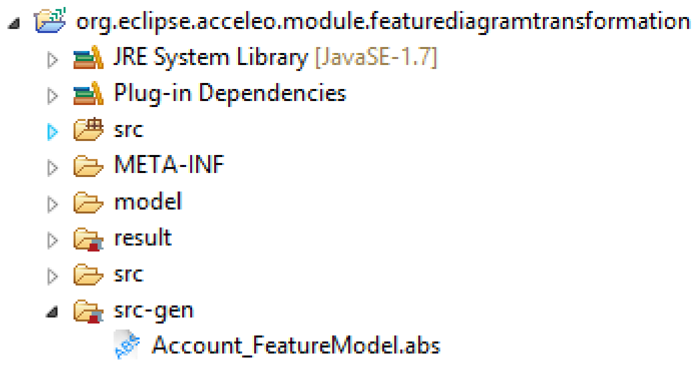
1. Assuming we have had a feature diagram which is made with HyperFlex Feature Diagram Editor in our workspace. Some examples of feature diagrams have been placed in **org.eclipse.acceleo.module.featurediagramtransformation** project in **model** folder like presented in picture below.


model
account.featuremodel
eshop.featuremodel
HelloWorld.featuremodel
lontar.featuremodel
replication_system.featuremodel
search_engine.featuremodel
test.featuremodel
tryu.featuremodel
account.featuremodel_diagram
eshop.featuremodel_diagram
HelloWorld.featuremodel_diagram
lontar.featuremodel_diagram
replication_system.featuremodel_diagram
search_engine.featuremodel_diagram

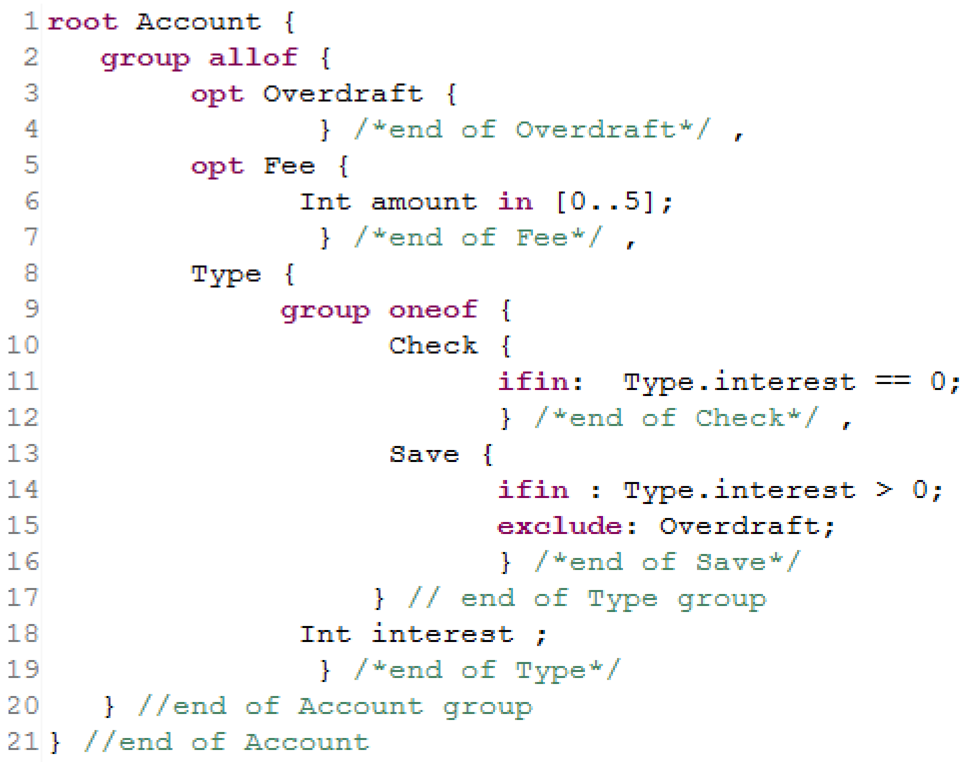
2. The first step to do is we have to know what file that we want to transform into ABS model. The input for this plug-in is a file that has **.featuremodel** extension. For example, we have `account.featuremodel` file.
We can transform that file into ABS model with following steps: **Right Click** on `account.featuremodel` file -> Acceleo Model to Text -> Generate Feature Diagram to ABS



3. The transformation process will create a folder named **src-gen** in **org.eclipse.acceleo.module.featurediagramtransformation** project on your workspace and this folder will contain the generated ABS model.


org.eclipse.acceleo.module.featurediagramtransformation
JRE System Library [JavaSE-1.7]
Plug-in Dependencies
src
META-INF
model
result
src
src-gen
Account_FeatureModel.abs

`Account_FeatureModel.abs` file is a generated ABS model which is a result from the transformation result with `account.featuremodel` as an input. This file contains ABS model as showed in the picture below.



```
root Account {
   group allof {
        opt Overdraft {
               } /*end of Overdraft*/ ,
        opt Fee {
              Int amount in [0..5];
               } /*end of Fee*/ ,
        Type {
             group oneof {
                   Check {
                         ifin:  Type.interest == 0;
                         } /*end of Check*/ ,
                   Save {
                         ifin : Type.interest > 0;
                         exclude: Overdraft;
                         } /*end of Save*/
                  } // end of Type group
              Int interest ;
               } /*end of Type*/
   } //end of Account group
} //end of Account
```