SFIT– User guide

The SFIT development team

March 17, 2016

Contents

1 Installation

To install SFIT on Windows, do the following:

1. Go to `http://sfit.fortiss.org/download/`
2. Download the "Windows X86_64" version.
   We recommend to use the latest Release version in the upper download section.[1].
3. Unzip the file wherever you want – if you do not know where, unzip it on your desktop
4. Go to the webpage of Z3 : `https://github.com/Z3Prover/z3/releases`
5. Download the first windows package ("z3-4.4.1-x64-win.zip" at the moment of the writing)
6. Extract Z3 to the place you wish it to be installed in
7. Make sure the path to the Z3 binary is declared in the System's "Path" variable:
   - In Windows' Control Panel go to System – Advanced System Settings (on the right of the window)
   - In the System Properties box select the "Advanced" tab and then "Environment Variables"
   - Look for the "Path" variable and add to it the path to the "z3" executable ("your–z3–install–path\z3-4.4.1-x64-win\bin" at the time of the writing).
     The variable "Path" should already exist; you can append to it by separating with a ";".

On Mac and Linux apply the same steps but download the relevant executables[2].

[1] The nightly build below is automatically created every night from the latest source code under development and might contain major bugs and is not tested to be completely functional.

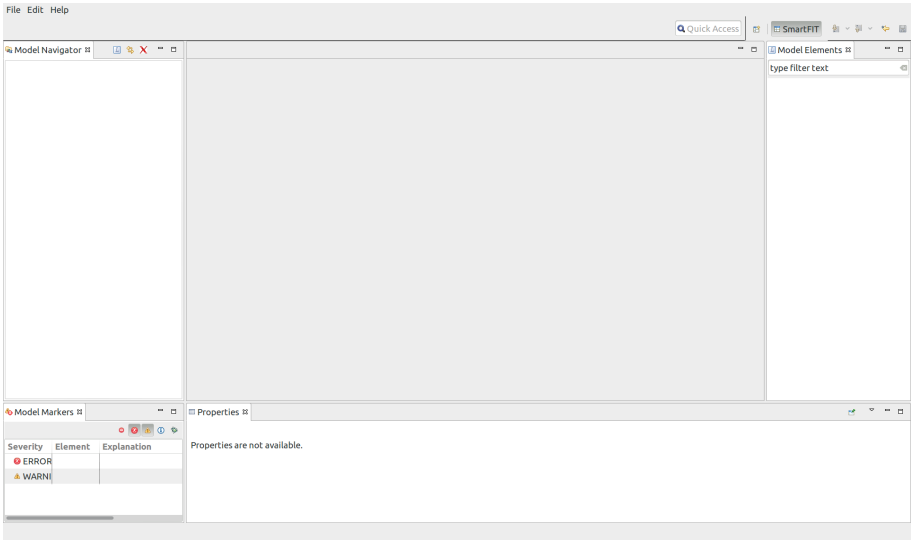
[2] Note, that this tutorial is only tested for Windows, and certain installation steps - like the Z3 installation - are different for Mac and Linux.

2 Tutorial: how to model an assembly line?

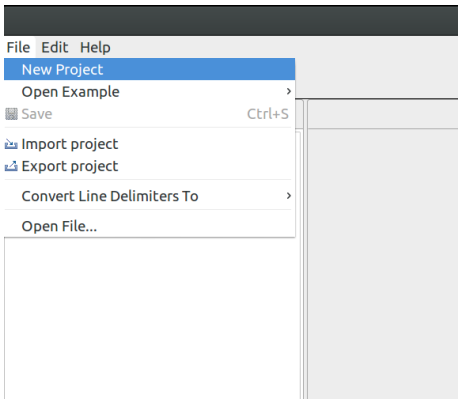
This tutorial will show you how to model an assembly line step by step. If you want to skip the step by step and go directly to a pre-made example, see Section 4.

2.1 Create a project

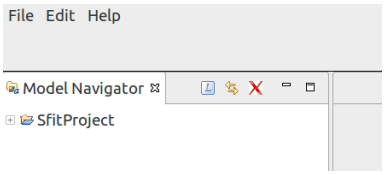
1. Double-click on `sfit` to launch the application. You should now see a window similar to the following:



2. Create a project by going in the "File" menu, then choose "New project"

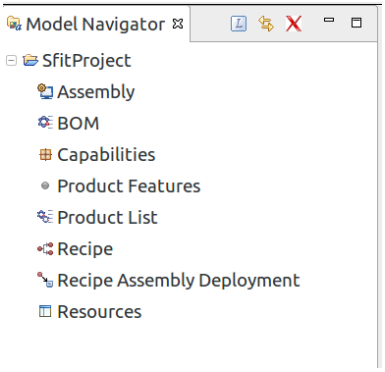


3. You should then see the following:

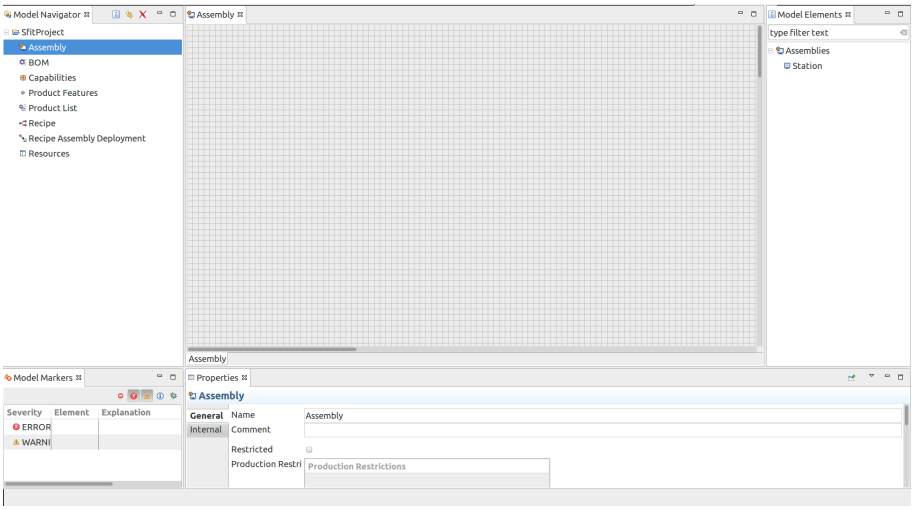


2.2 Create an assembly line

1. Open the created project

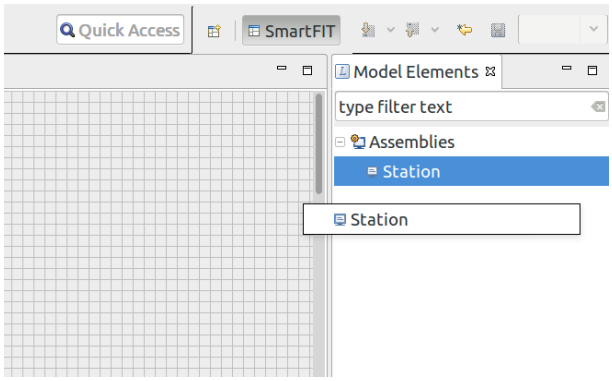


2. In the "Model Navigator" Tab, double click on "Assembly"

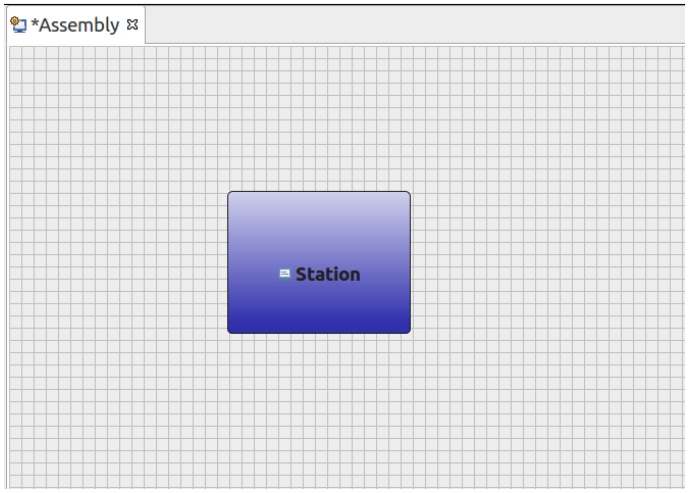


3. Add stations to your assembly by dragging from the "Model Elements"

tab on the right and dropping into the main editor:

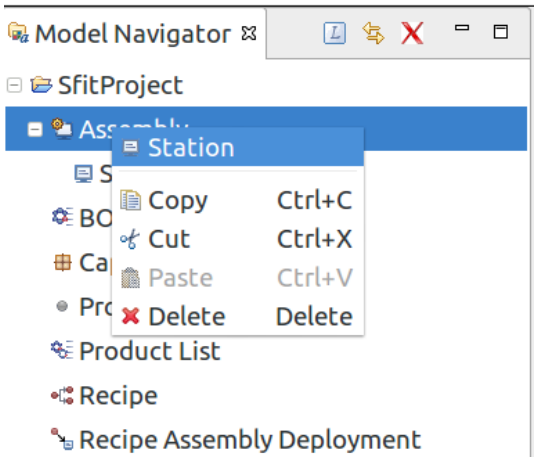


4. You should then obtain something like:

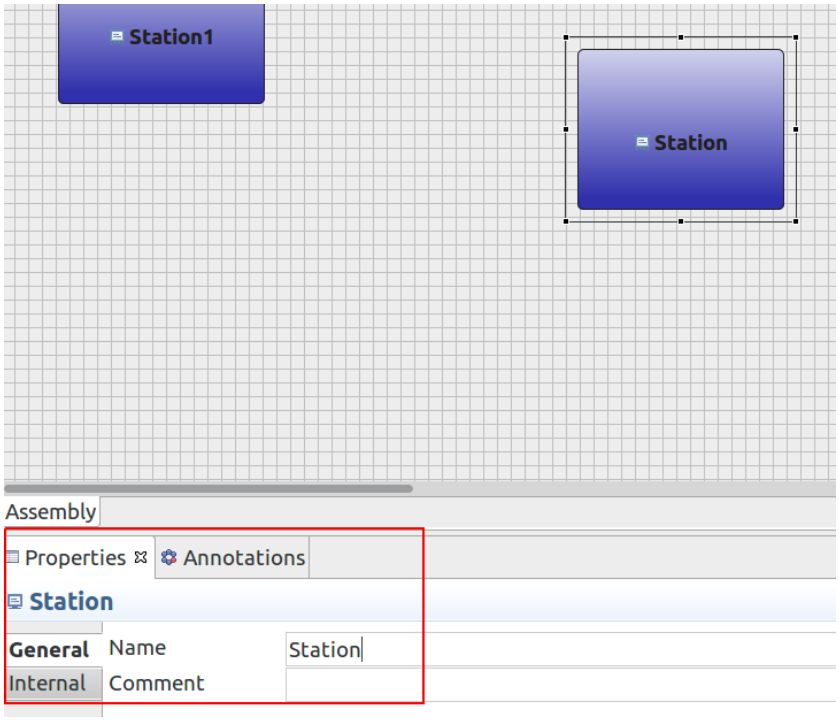


5. Another way of creating a station is by right clicking in the Model Navi-

gator:



6. You can move stations around by clicking them and moving them with the mouse, or by clicking them and moving them with the arrows of your keyboard (hint: maintain "Shift" while pressing the arrows to move bigger steps). You can also resize the stations. Finally you can change their name in the "Properties" View at the bottom of the screen:



7. You can connect a station "S" to a station "T" as follows:

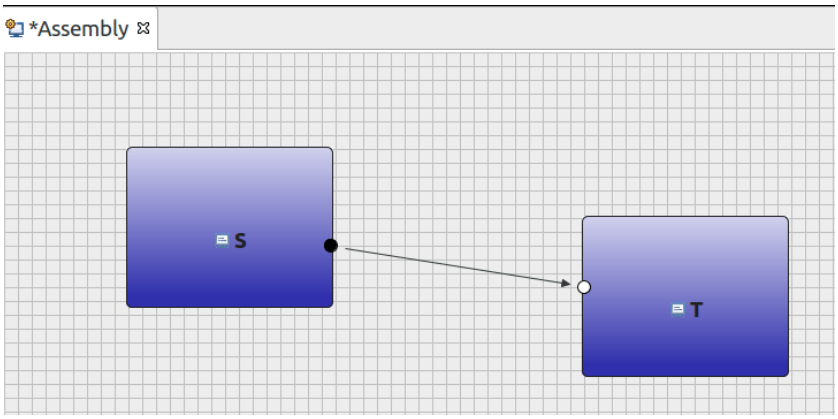
(a) Press Alt **on the keyboard** (or Ctrl on Linux)

(b) **Click (without releasing the button!)** on station "S"

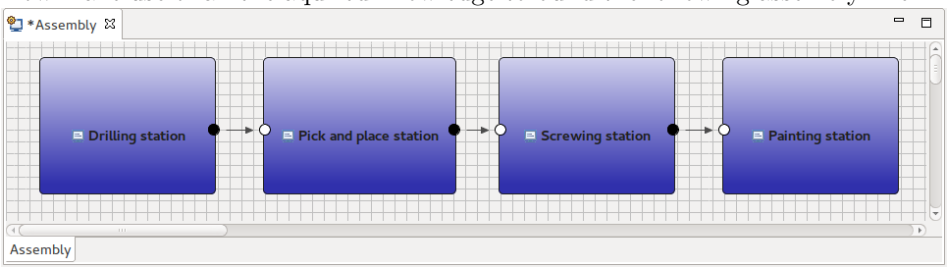
(c) Move the mouse to station "T"

(d) **Now you can release the button**

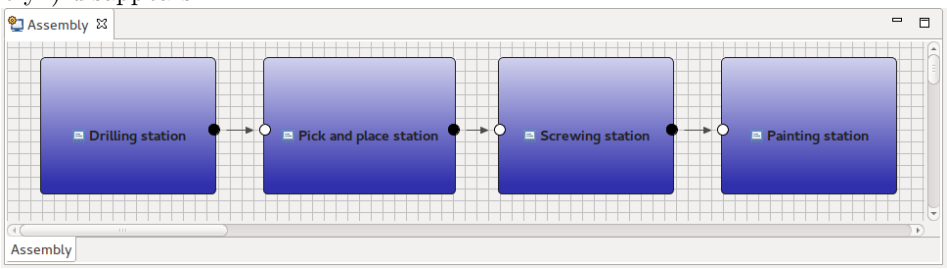
You should then see something like:



Now make use of all the aquired knowledge to build the following assembly line:



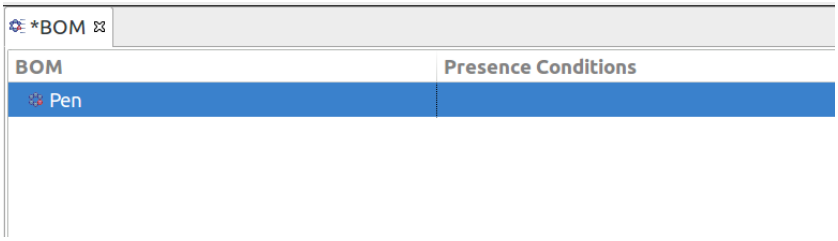
Save by going in the File menu and clicking "Save" or simply by typing "Ctrl+S" on your keyboard. Observe how the little star next to the tab name ("Assembly") disappears:



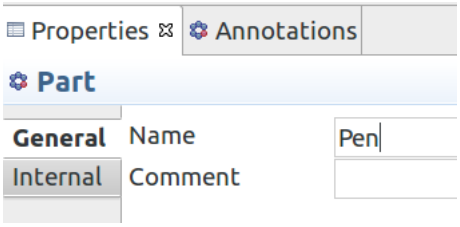
2.3 Create a bill of material

1. In the "Model Navigator" Tab, double click on "BOM":

2. Drag a "Part" and drop it in the empty space:

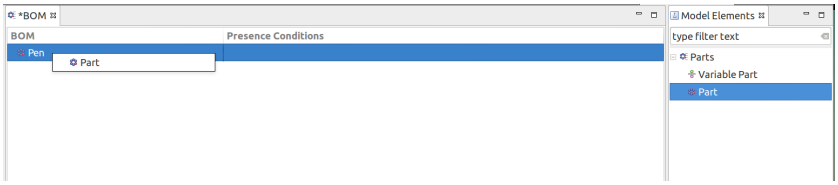


3. You can change the name of the part in the Properties tab below the screen:

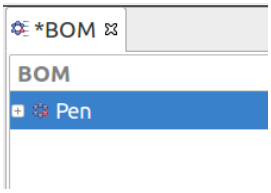


4. Like previously for stations, you can also right-click on "BOM" in the Model Navigator

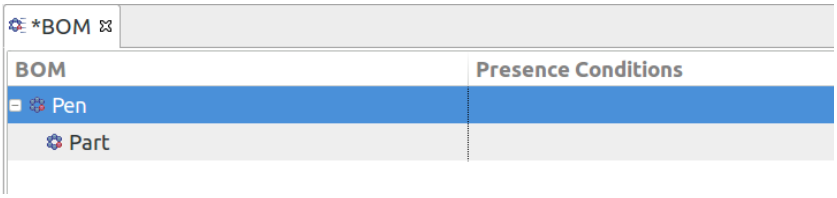
5. You can add further parts to a part by drag and dropping **on** this part:



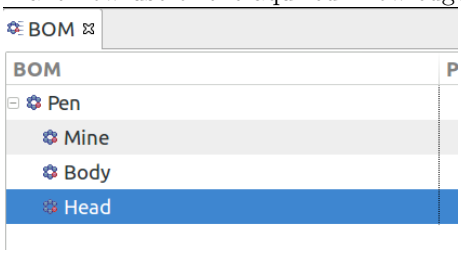
6. You should then obtain:



7. Open the part to see the newly added sub-part:



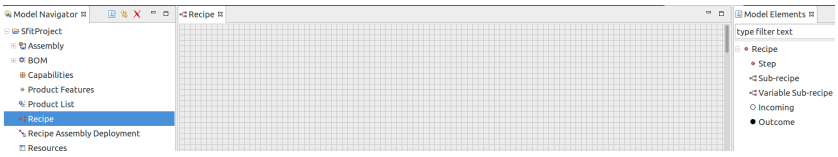
Make now use of the aquired knowledge to build the following BOM:



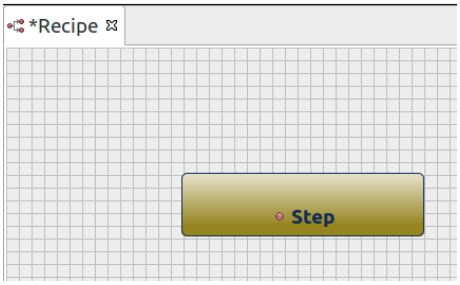
Do not use the "Variable part" for this tutorial by now.

2.4 Create a recipe

1. In the "Model Navigator" Tab, double click on "Recipe":



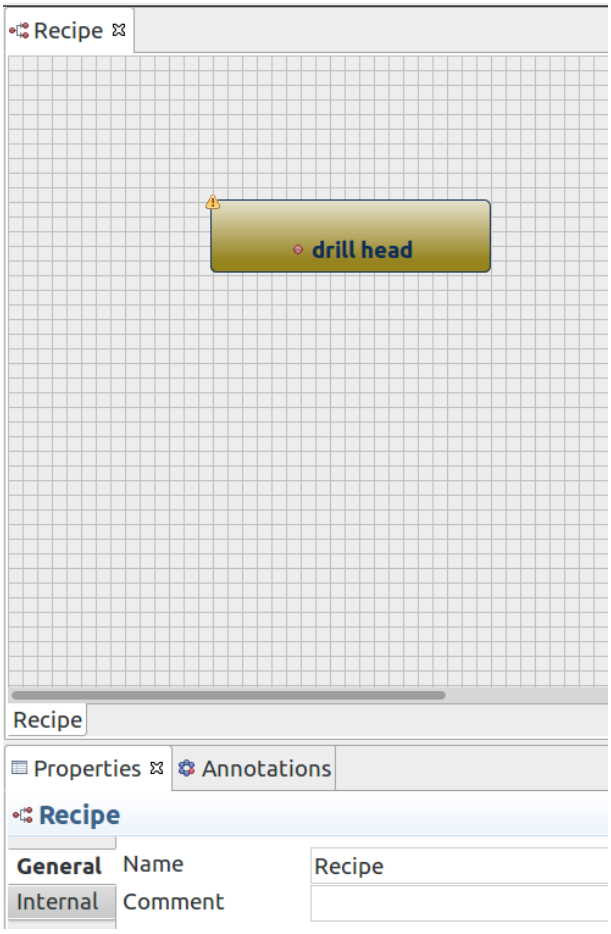
2. Drag a "Step" and drop it in the empty space:



3. Now save and observe how a warning appears on the step:

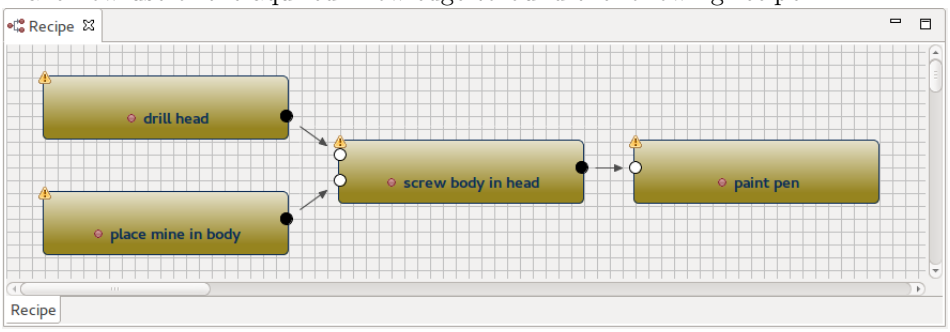
Ignore this warning for now.

4. Rename the step by selecting it and modifying the name in the properties as now usual:



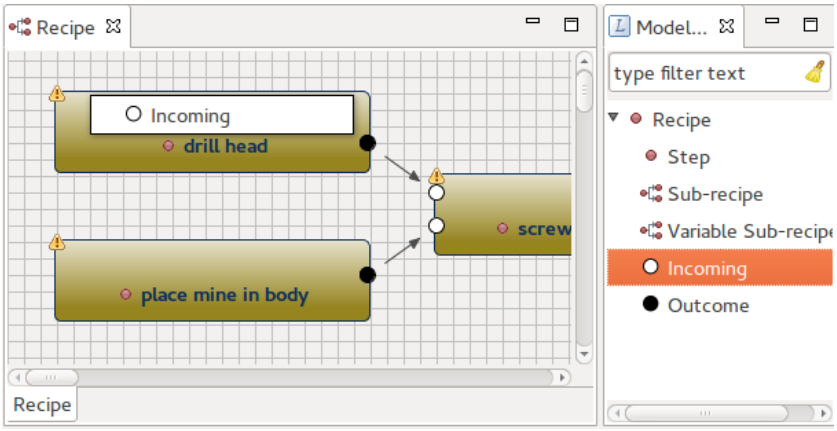
Creating further steps can be done as usual either by drag and dropping or through the context menu in the Model Navigator. Connecting steps between each other can be done like for stations by drag and dropping maintaining the Alt key.

Make now use of the aquired knowledge to build the following recipe:

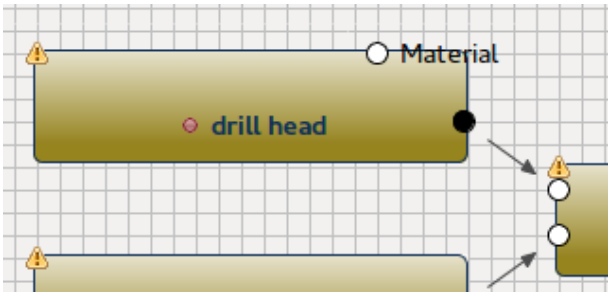


Do not use the "Sub-recipe" or the "Variable sub-recipe" for now.
We will now connect the recipe to the BOM:

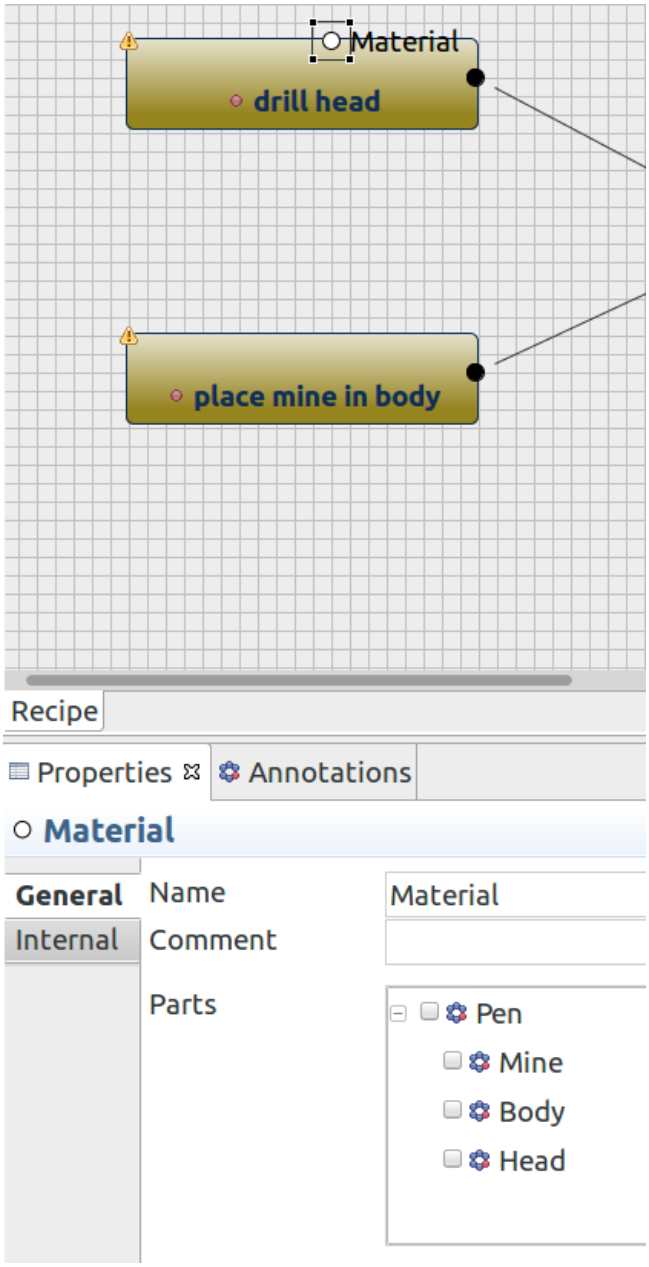
1. Drag an "Incoming" port from the "Model Elements" tab:



2. And drop it on the "drill head" step:

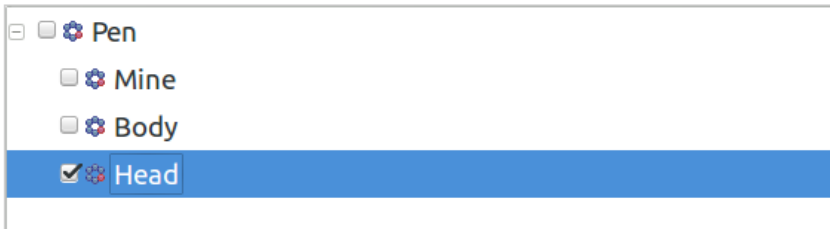


3. Then select the newly added port and observe the “Properties” tab:



The BOM is recalled here.

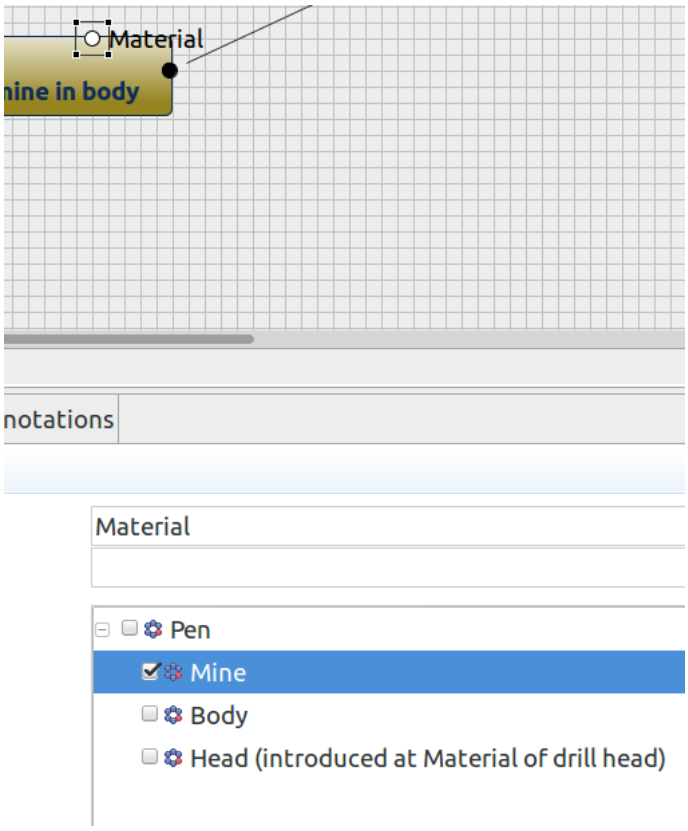
4. Select the “Head” part in the BOM and save:



→ this models the fact that the “Head” material is expected to be added when the “drill head” step is applied

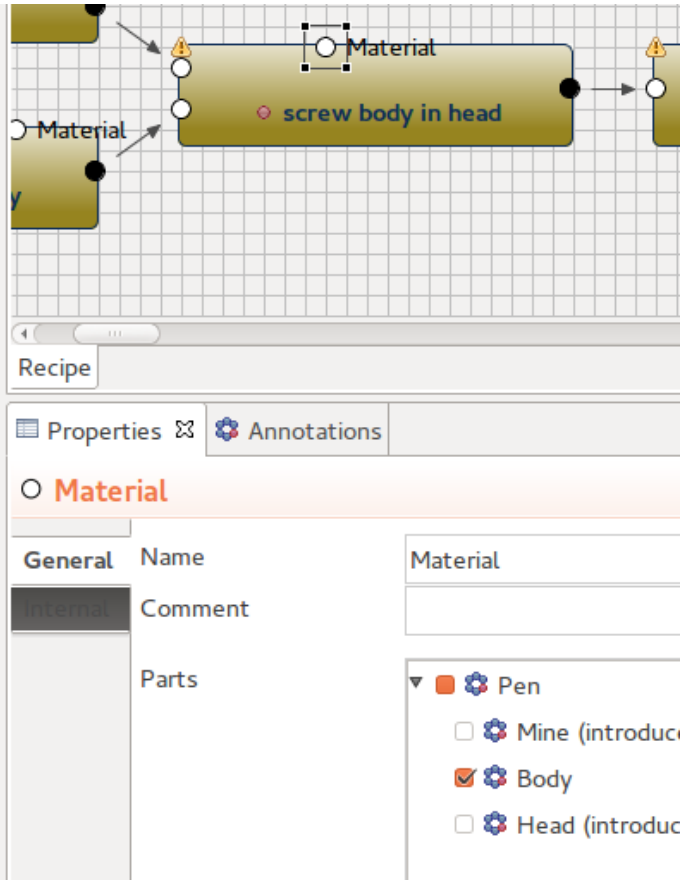
5. Add now an incoming port on the step “place mine in body” (either by drag and drop, or, again, by right-clicking on the step in the Model Navigator)

6. And select the “Mine” part in the properties:

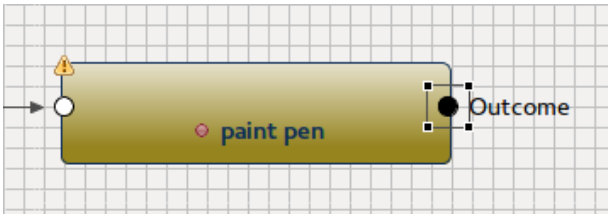


Note how the propertiers inform you that “head” has already been introduced in the other step. As a consequence, this material cannot be added again.

7. Repeat the operation by adding an incoming port on the step “screw body in head” and attach the “body” to it.



8. Finally, add now an *outcoming* port (and not incoming!) on the step “paint head”:

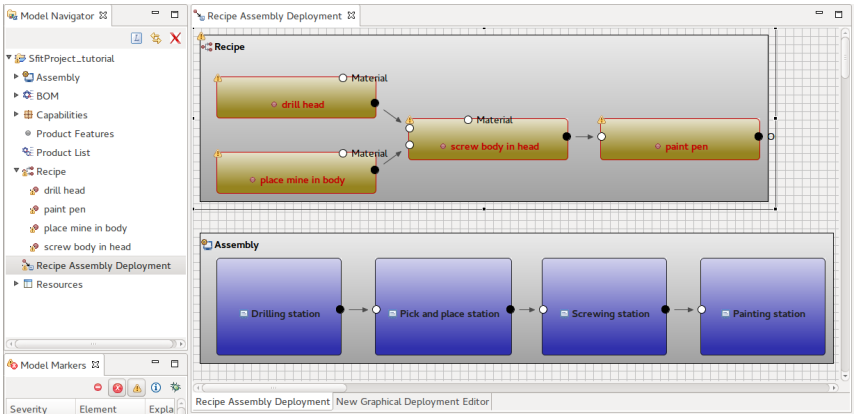


When selecting this outcoming port, you can see now, that all parts of the “Pen” are assembled after this step.

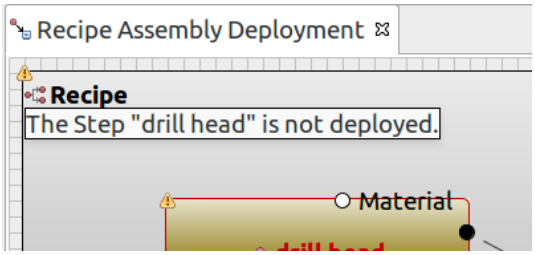
2.5 Deployment of the recipe on the assembly line

We will now deploy the recipe on the assembly line.

1. Close the recipe and open the “Recipe Assembly Deployment”:



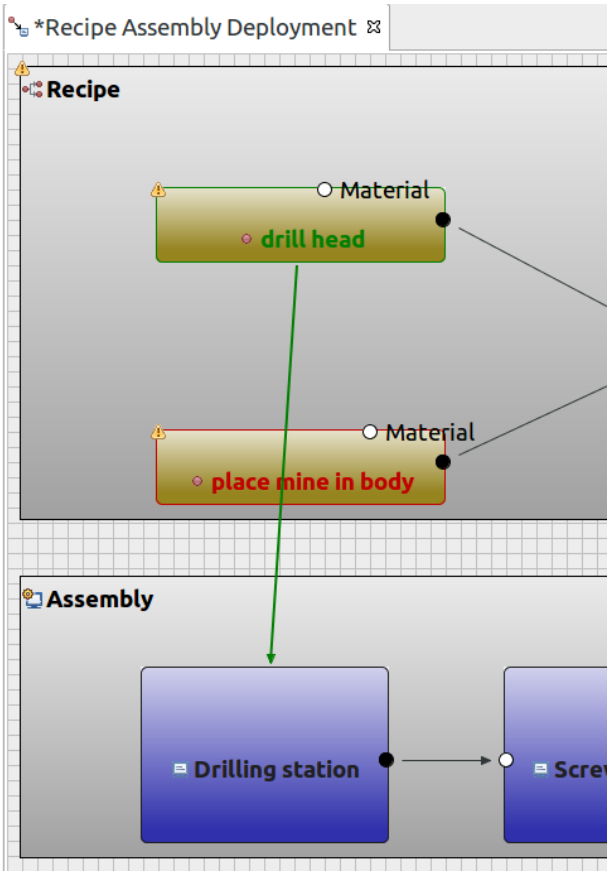
2. Observe how the deployment itself also has a warning, you can hover over it to know more:



SFIT complains because the step “drill head” is not deployed on any station.

3. To fix this, maintain “Alt” (or Ctrl on Linux), click on “drill head”, **without releasing the click** move the mouse to the *station* “Drilling station”,

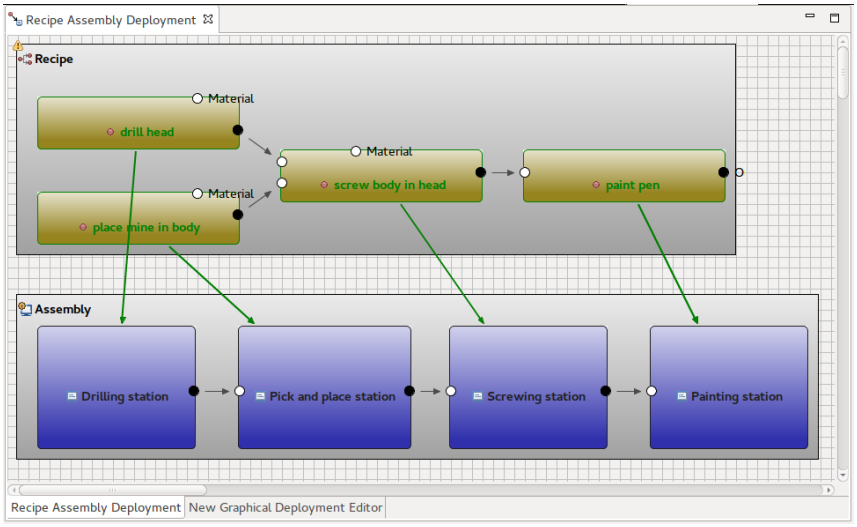
and only now release the button. You should obtain the following:



Observe how the step "drill head" is now green: this indicates you that the step is deployed.

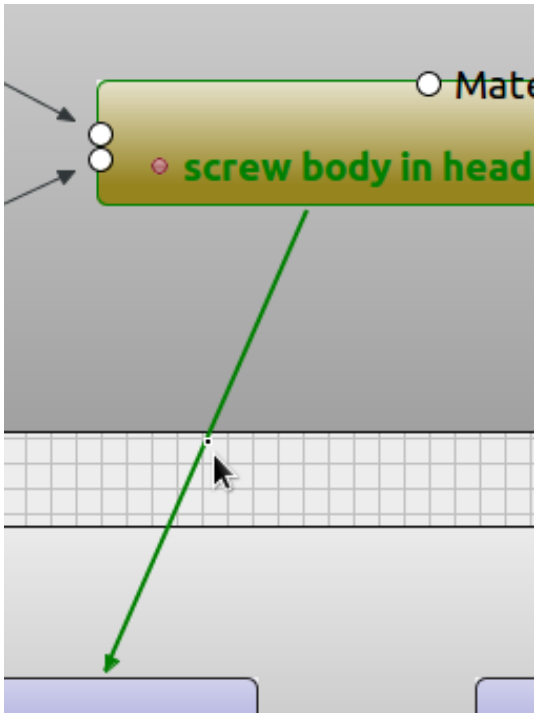
4. Save and observe how the warning on "drill head" disappeared.

5. Deploy now the two other steps:

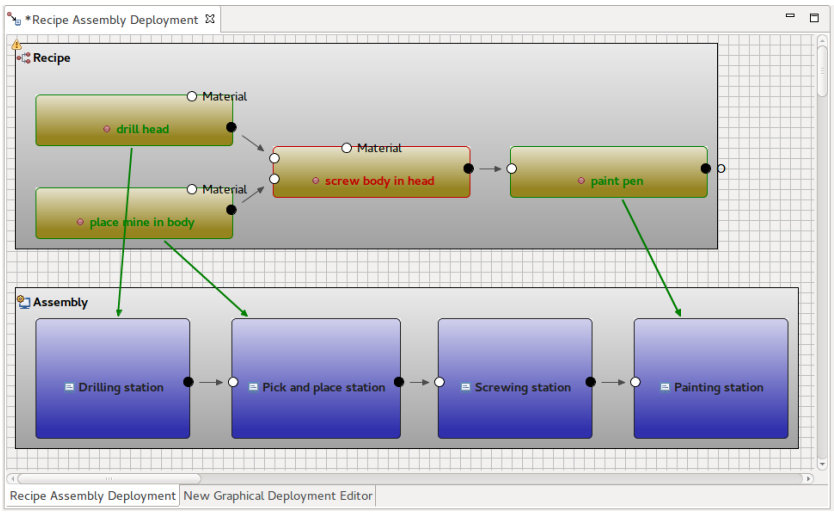


You can notice that the top warning is still there. If you hoover, you should see that it complains about the step “drill head” not having a “required Capability” selected. We will fix this soon, but to understand why this is needed, try to do the following:

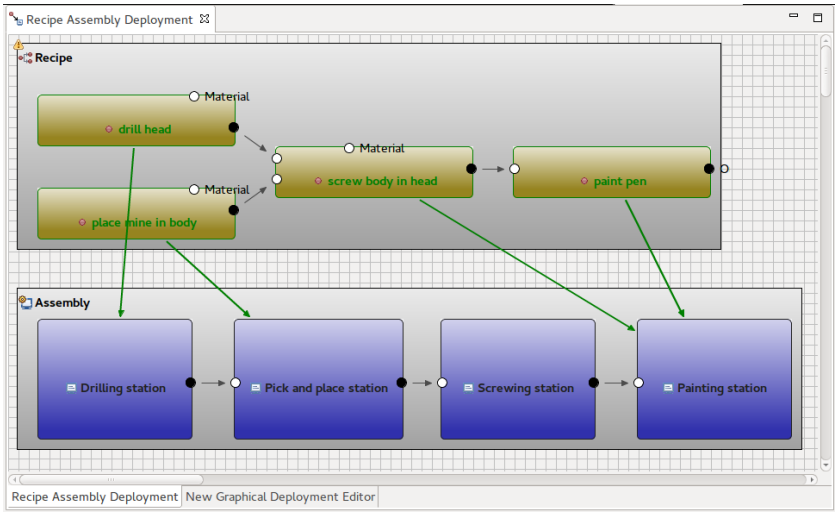
1. Select the deployment arrow of “screw body in head”:



2. Press “Delete” to remove it. You should now be back to the following situation:



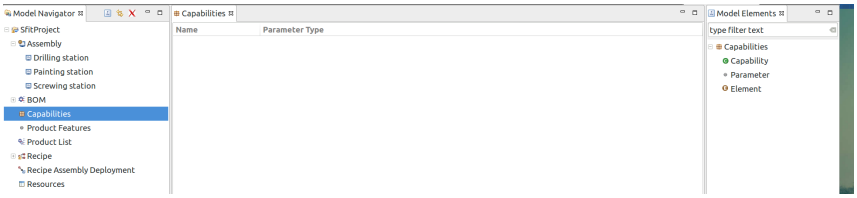
3. Deploy now “screw body in head” on the “painting station” and save:



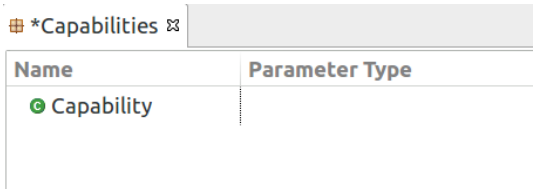
→ **Sfit accepts this blindly!** However, this does not make sense in reality: the painting station is probably not capable of drilling. On the other hand the screwing station was most probably a good choice to “screw body in head”.
→ Expressing such restrictions is the purpose of “capabilities”. Cancel the last change (Ctrl-Z) and go to the next section.

2.6 Capabilities

1. Close the deployment and open the “Capabilities”:

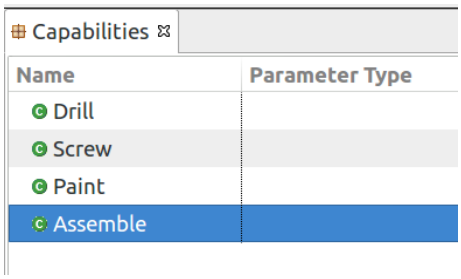


2. As always, you can add items either through the context menu in the model navigator or through the “Model Elements” tab. Add now a new capability:

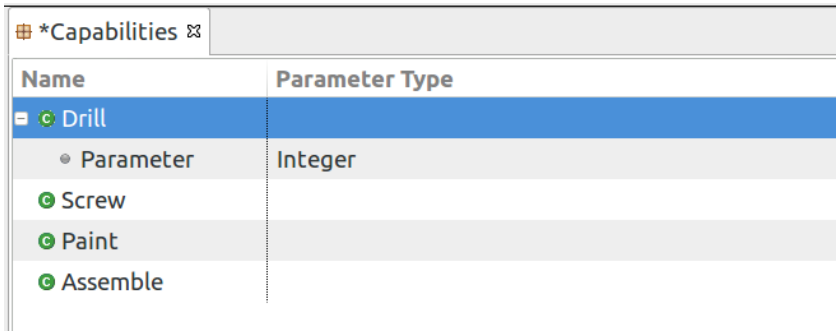


3. Rename this capability by clicking on it. Call it “Drill”. Similarly, add three other capabilities called “Screw”, “Paint” and “Assemble”. You

should obtain the following:

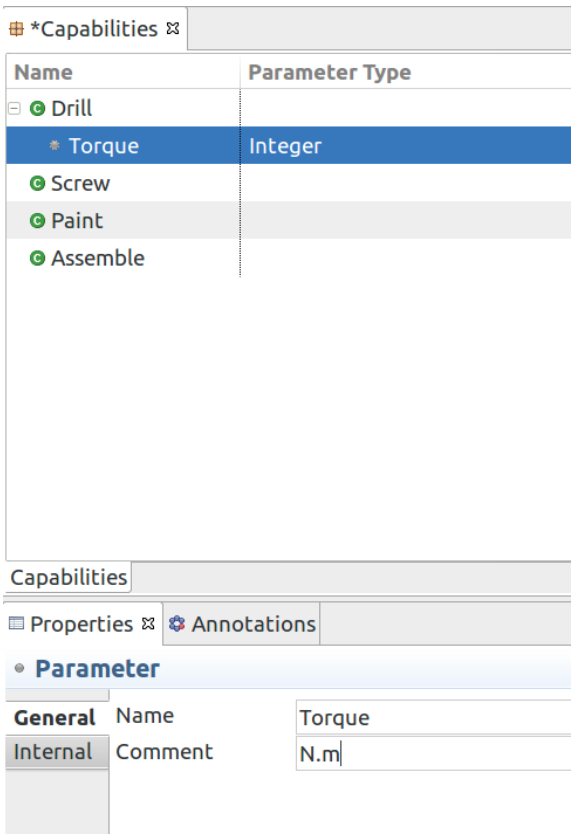


4. Add a “Parameter” to the “Drill” capability:

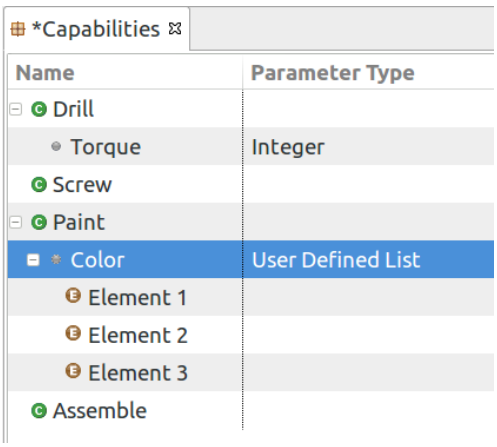


5. Rename it into “Torque”, and add as a comment “N.m” to indicate the

unit of the parameter.

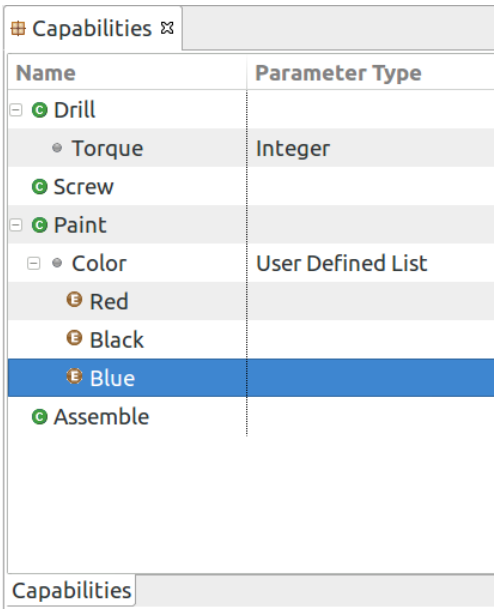


6. Add a “Parameter” to the “Paint” capability, rename it “Color”, click the parameter type “Integer”, and select “User defined List”:



7. Change “Element1” into “Red”, “Element2” into “Black” and “Element3”

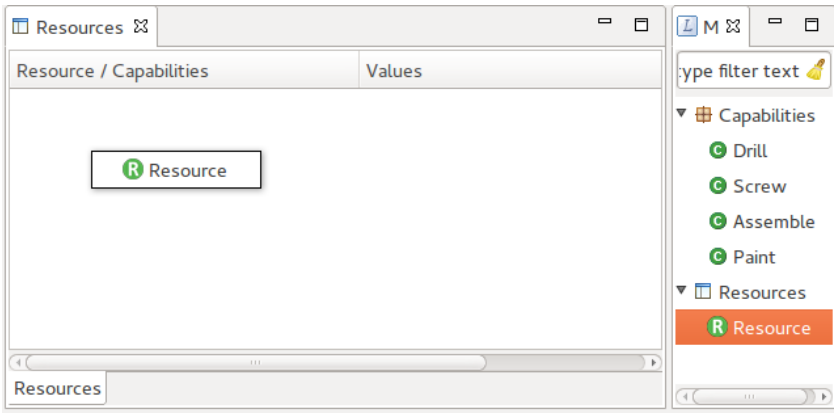
into “Blue”:



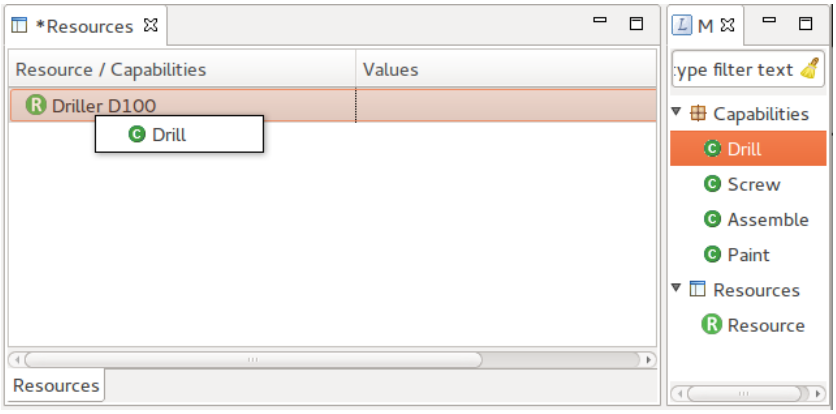
We will now create resources offering these capabilities:

1. Close the capabilities and open “Resources”.

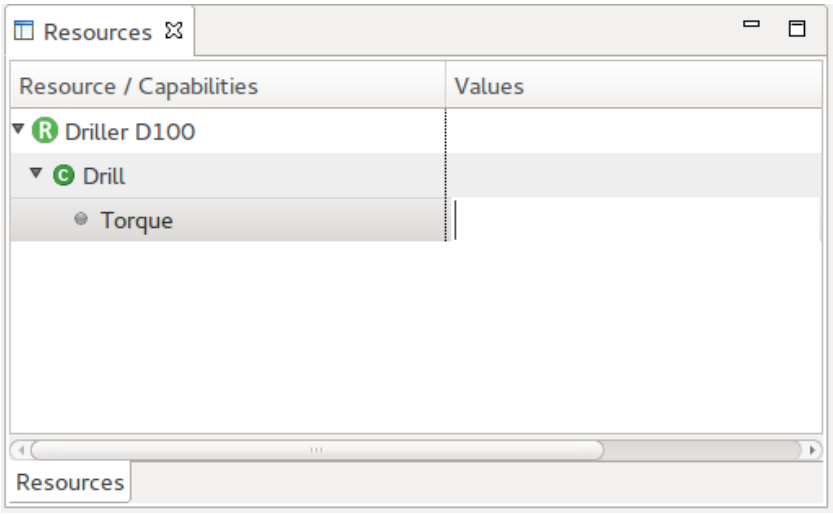
2. Create a new resource by drag and dropping “Resource” from the Model Elements view on the right.



3. Change the name to “Driller D100” and attach to it the capability “Drill”, that you can also drag and drop from the right:

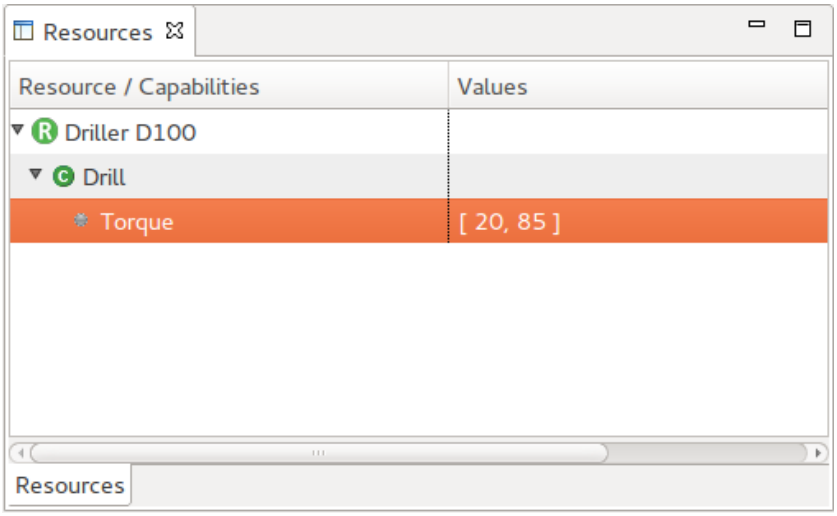


4. When unfolding “Driller D100” and the Capability “Drill”, you can see the Parameter “Torque”, that we added to “Drill”, previously. In the value field, there is the value 0 by default.
Let’s change this value by clicking into the value field of “Torque” (that contains the 0):



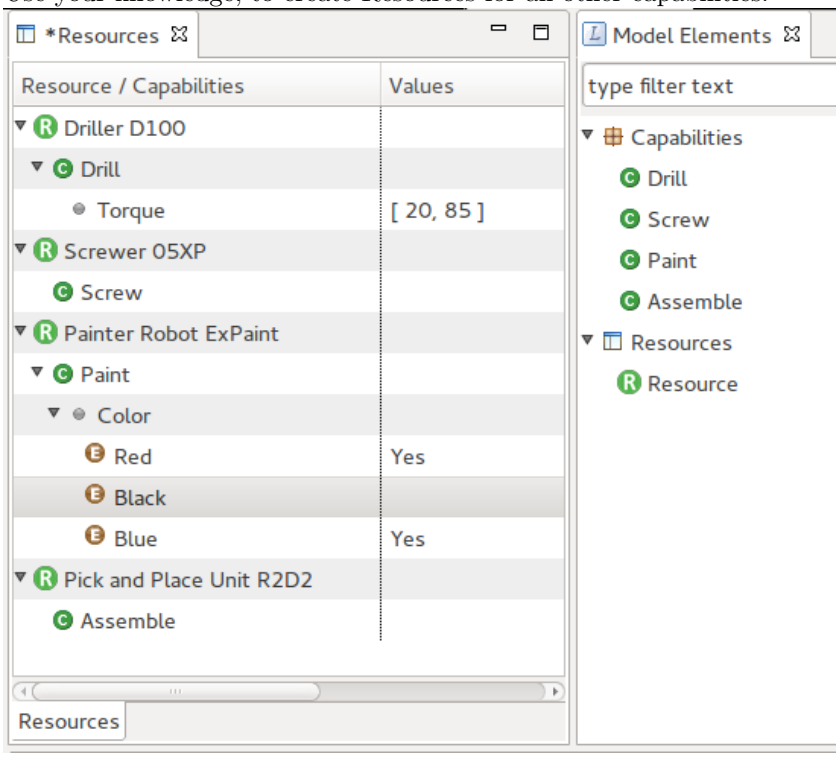
5. Type in “[20, 85]” to specify, that “Driller D100” can “Drill” with any torque within the range between 20 and 85.

The Result should look like this:



6. Save the model.

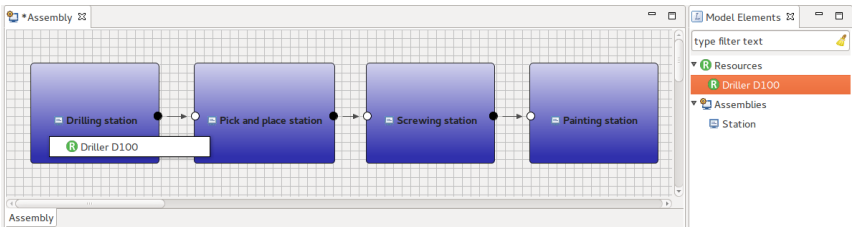
Use your knowledge, to create Resources for all other capabilities:



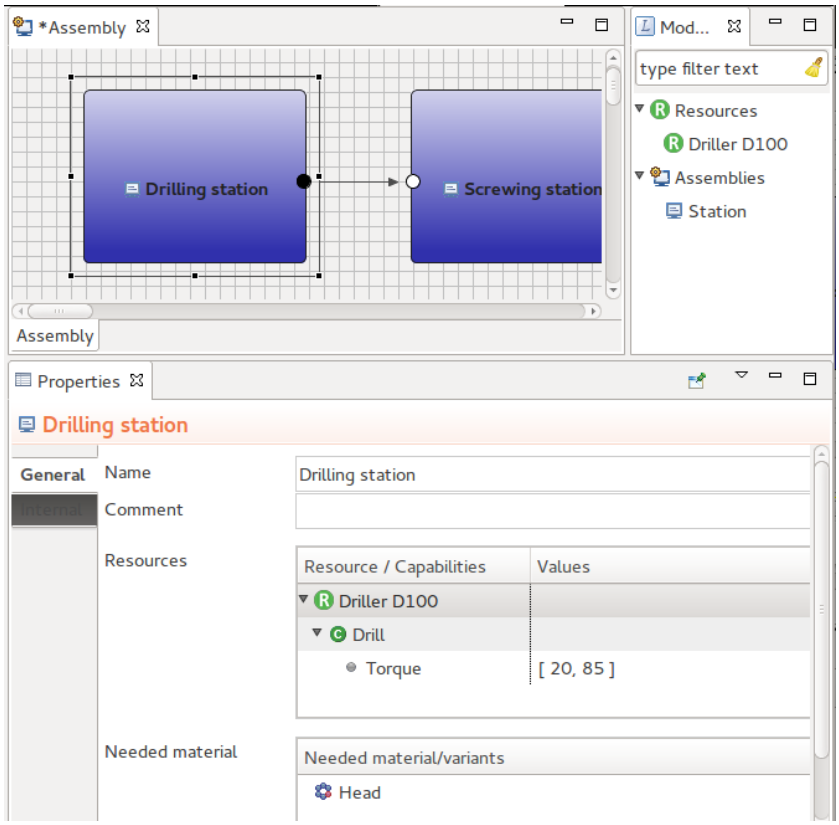
Now, that there is the "Driller D100" defined, we can use it in the Assembly Line:

1. Open the Assembly Line model.

2. In the Model Elements view on thr right, you can now also see the Resource "Driller D100".
   Drag and drop it to the "Drilling Station", to add it there:



3. The Station "Drilling Station" is now equiped with a "Driller D100". You can see this in the "Properties view", when selecting "Drilling Station":



4. Observe, how you can inspect the "Driller D100" also here to see it's Capabilities.

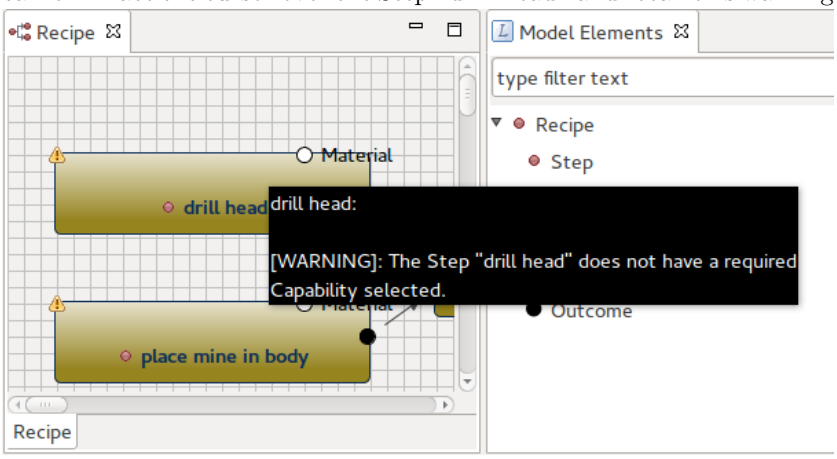
5. Below the Resources entry in this Property view, there is another field "Needed material". Here you can already see, that the SFIT tool can tell you, that on the "Drilling Station" the Part "Head" will be needed.[3]

To complete the Assembly Line model, add the other Resources to the respective Station:

Station	**Resource**
Screwing station	Screwer 05XP
Pick and place station	Pick and Place Unit R2D2
Painting Station	Painter Robot ExPaint

You saw, that Resources in the Stations can **provide** Capabilities. A further aspect is, that Recipes usually **require** these Capabilities. To model this, perform the following steps:

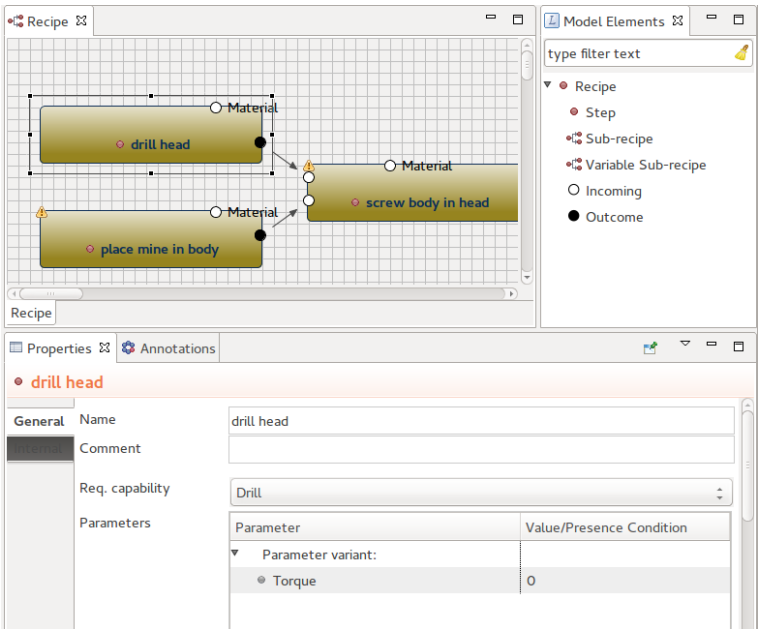
1. Open the Recipe. (We can come back here to the Warning, we ignored earlier. Place the cursor over the Step "drill head" and recall this warning:)



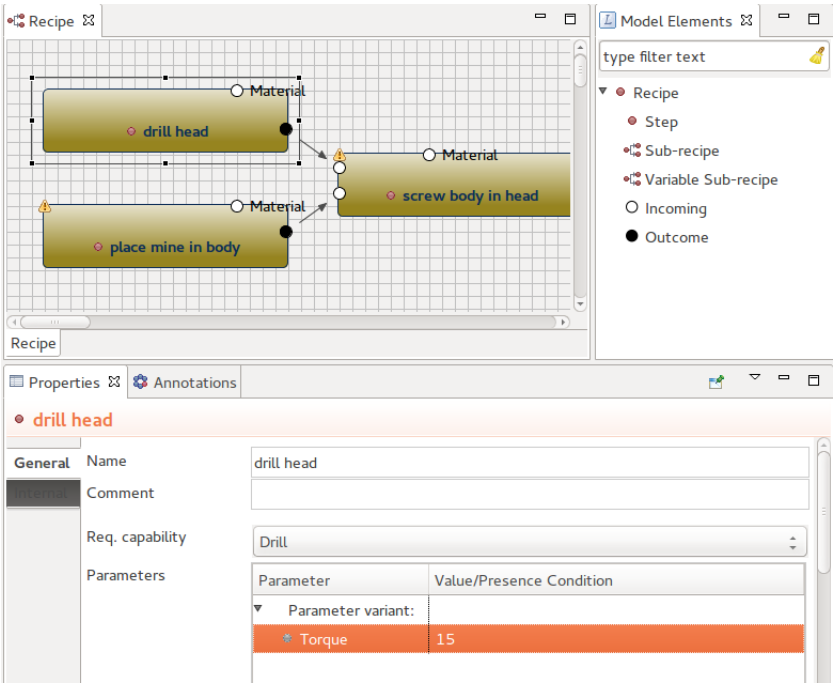
2. To attach a Capability and solve the warning, select the Step "drill head".

[3]The SFIT tool retrieves this information from the Recipe you created and the mapping to the Stations that you did in the "Recipe Assembly Deployment".

3. In the Property view, go to the entry “Req. capability” and select “Drill”:



4. Set the value for “Torque” to 15 and save the model:



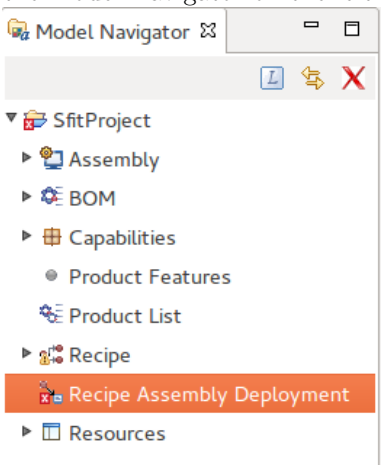
5. Note, that the warning for “drill head” now has disappeared.

Make use of you knowledge to configure the remaining process steps as follows:

Step	Capability	Parameters
place mine in body	Assemble	-
screw body in head	Screw	-
paint pen	Paint	{Blue = yes, Red = (No), Black = (No)}

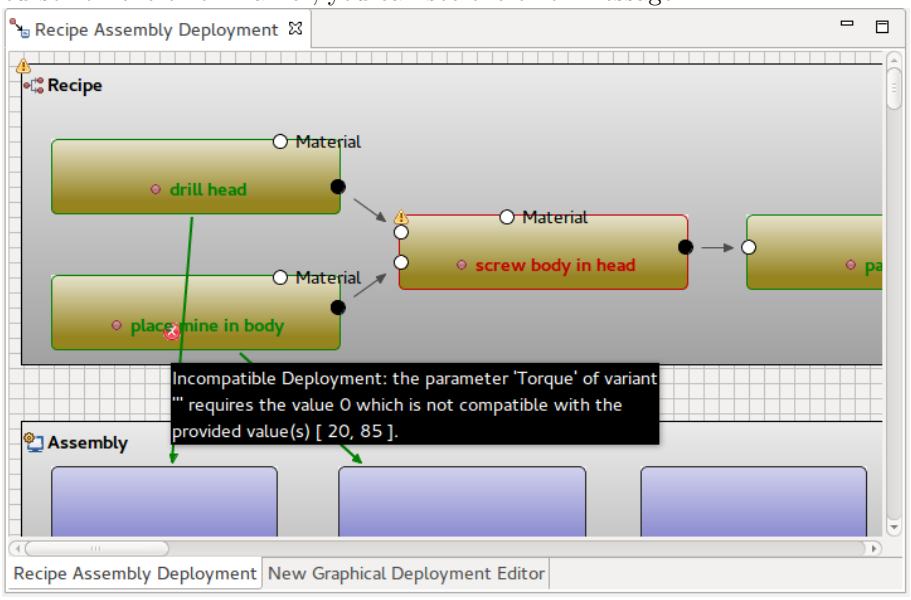
2.7 Compatibility Check

As you might already have realized when adding the Capability "Drill" to the Resource and to the Step, what we modelled contains an inconsistency. The SFIT tool also realizes this and gives as feedback in form of an error marker (on the Recipe Assembly Deployment and on the project itself. You can see this in the Model Navigator on the left:

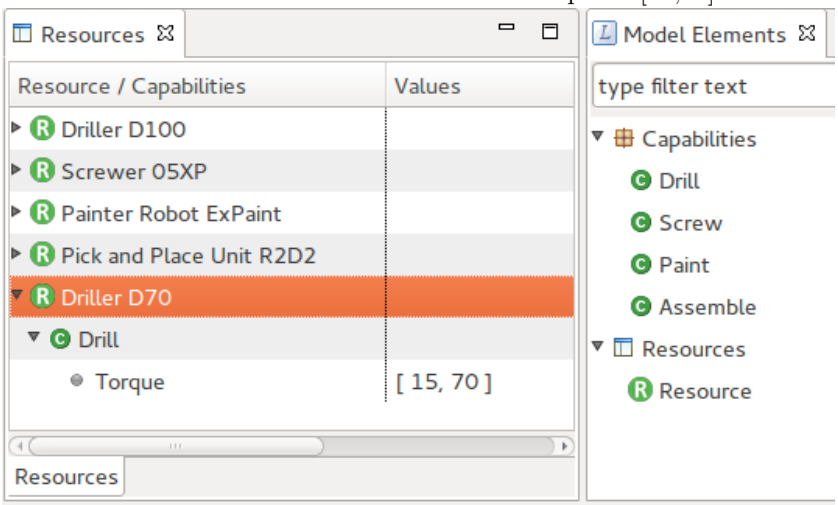


When opening the Deployment, there also is an error marker on the Deployment Arrow that deploys "drill head" to the "Drilling station". When placing the

cursor on the error marker, you can see the error message:

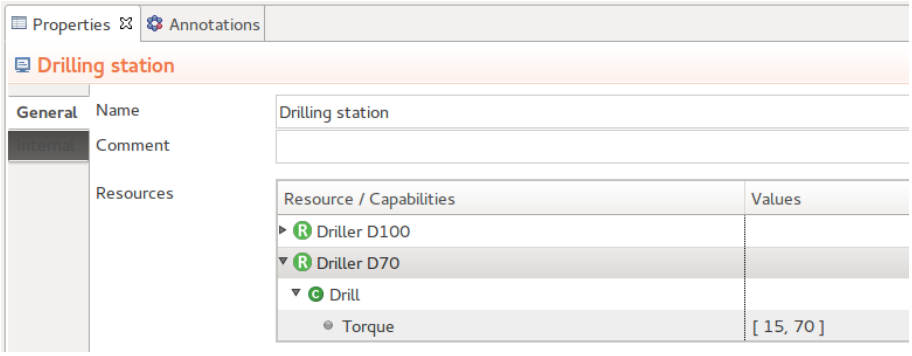


The problem here is, that we required for the Step “drill head” to “Drill” with a small torque of only 15Nm. The “Drilling Station”, to which this step is deployed (i.e. on which it is supposed to be performed) can only provide torques in the range between 20Nm and 85Nm (since this is the torque range of the “Driller D100”, that we installed in the “Drilling station”. With this, the “Drilling station! is not capable of performing the Step “drill head”.
One possible solution here would be to add another Resource to the station, that can provide the correct “Torque” Parameter.
Create a new Resource “Driller D70” with a torque of [15,70]:

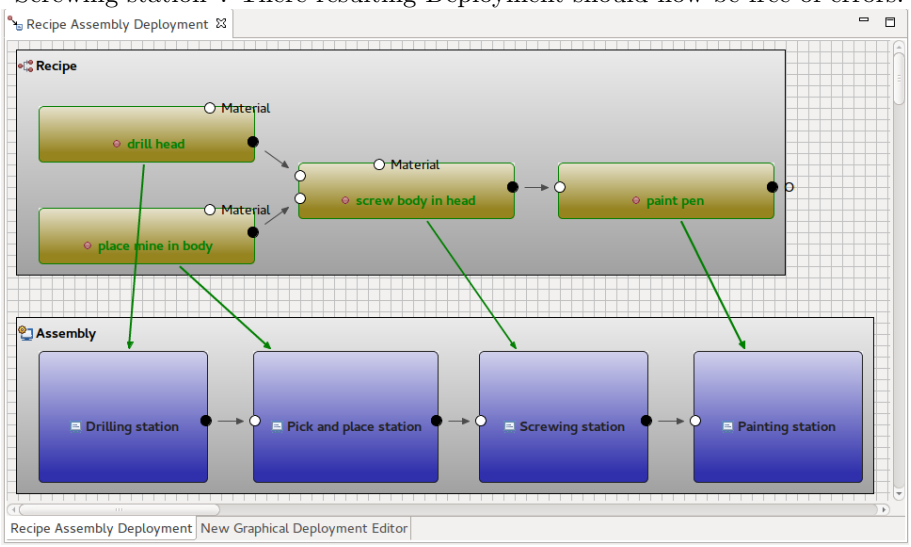


Add it to the “Drilling station”. Now, the “Drilling station” should contain two

Resources:



This is fine - both Drillers are now available at this Station. And this solves the compatibility problem, since the "Drilling station" now has a Resource available that is able to provide the Capability "Drill" with a "Torque" of 15Nm as required by the problematic Recipe Step "Drill head".
You can now also deploy the remaining Step "screw head to body" again to the "Screwing station". There resulting Deployment should now be free of errors:



3 Tutorial: how to deal with variability?

In this tutorial you will learn, how to deal with variability in the SFIT tool.
This tutorial builds on the result of the basic tutorial 2. If you do not have the result of the basic tutorial available, you can import the "Tutorial Result" via the example mechanism (see 4).
In the following this existing project is modified to add variability to certain elements. Please note, that it would of course also be possible to create variable elements from the beginning.
In the basic tutorial of section 2, you saw how to model manufacturing in SFIT. The product there was a single kind of pen. As scenario for this variability tutorial, the manufacturer decides to provide two versions of this pen to cope with the requirements of the market:

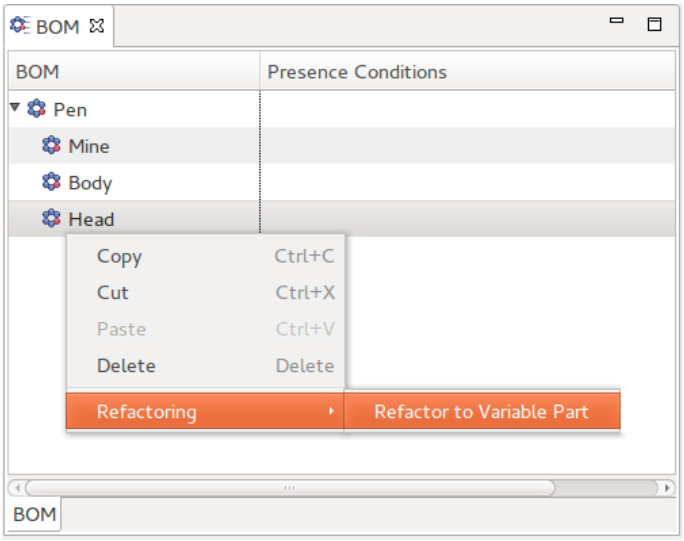
- A low cost version of plastic.
- A high end version of metal.

The engineers decide to replace the "Head" for the low cost version with a part variant, that is cheaper to be manufactured - it comes without a screw thread and is not screwed on the "Body", but glued. In turn the high end variant will be of metal instead of the plastic, used so far.

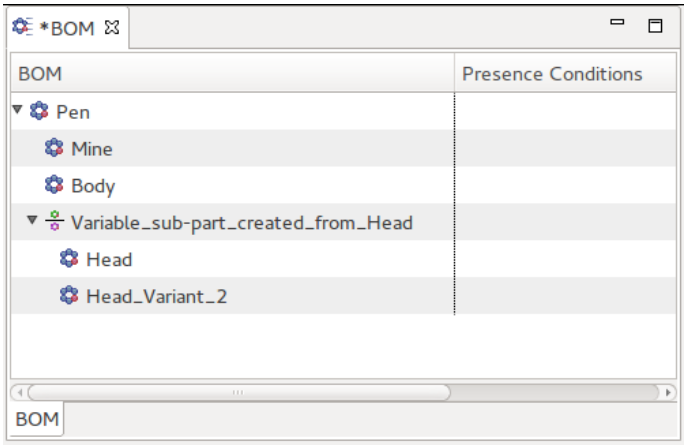
3.1 Variability in the BOM

Let's adopt the BOM to reflect this:

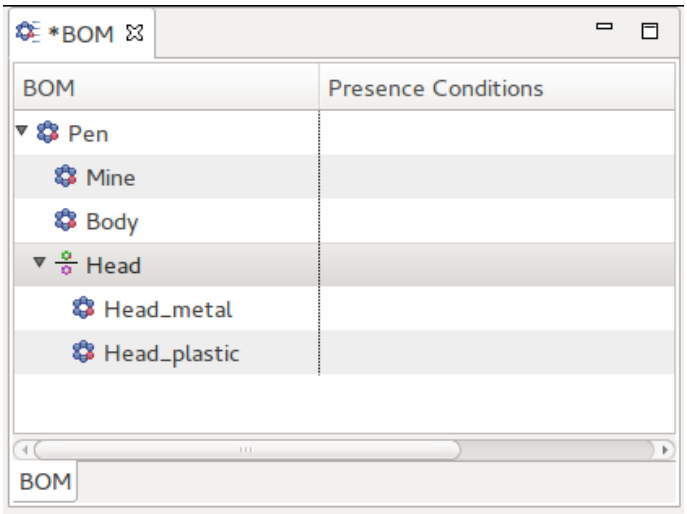
1. Open the BOM.
2. Right-click on "Head":



3. The result will be:



4. Rename the part "Head" to "Head_metal". Then also rename the other elements to this (you can also rename in the editor by clicking the names):



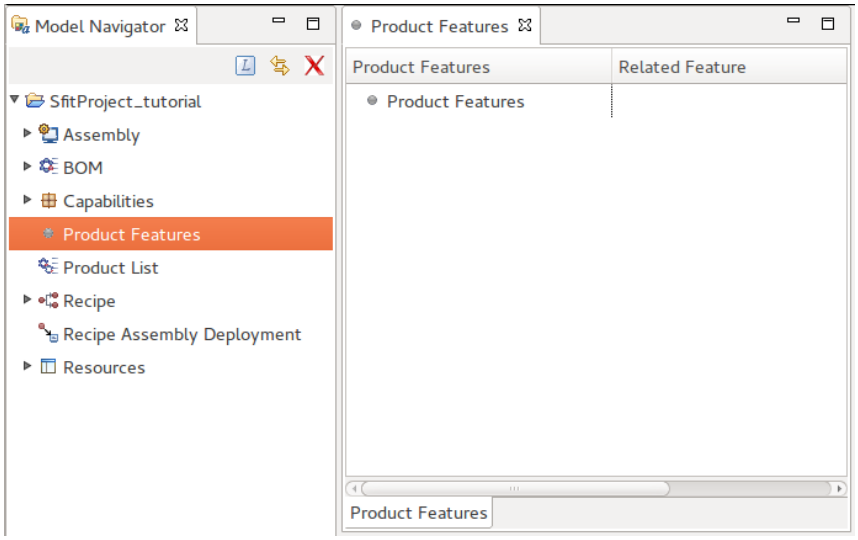
What this means, is the following: For the Part "Head", there are now actually two part variants - "Head_ metal" and "Head_ plastic". The new *Variable Part* "Head" hereby refers abstractly to any of the two.

3.2 The Feature Model

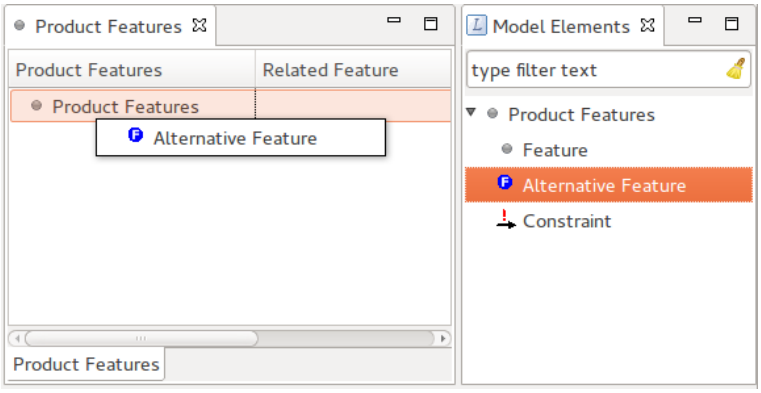
The SFIT tool, is also capable of managing when to use which of these part variants. To this occasion, the SFIT tool maintains information about the characteristics of products. And - most important - how these characteristics can differ between different product variants. This is done in a feature model. In this section you will learn what this means and how to use this functionality.

This is, how you can model the feature model for the pen product line:

1. Open the Product Features:

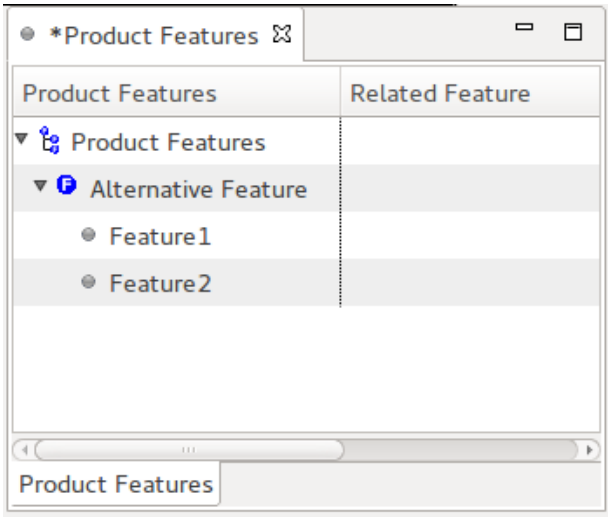


2. Add an Alternative Feature to "Product Features" in the editor:

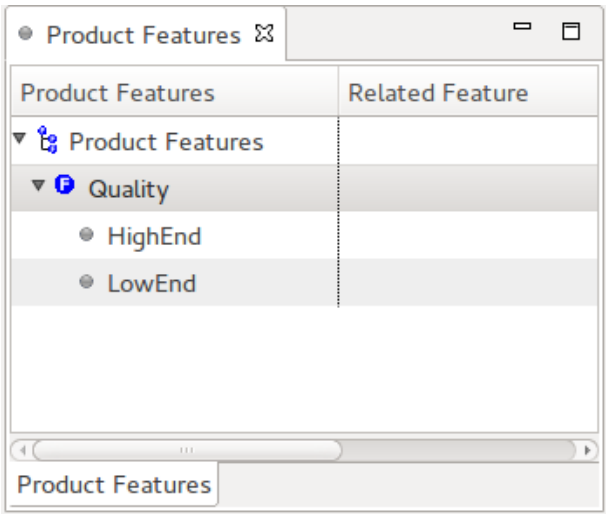


3. Note, how the icon of "Product Features" has changed. This tree like icon you see now, indicates that the "Product Features" is composed of other

Features - in this case the new “Alternative Feature”:



4. Now rename the Features to:

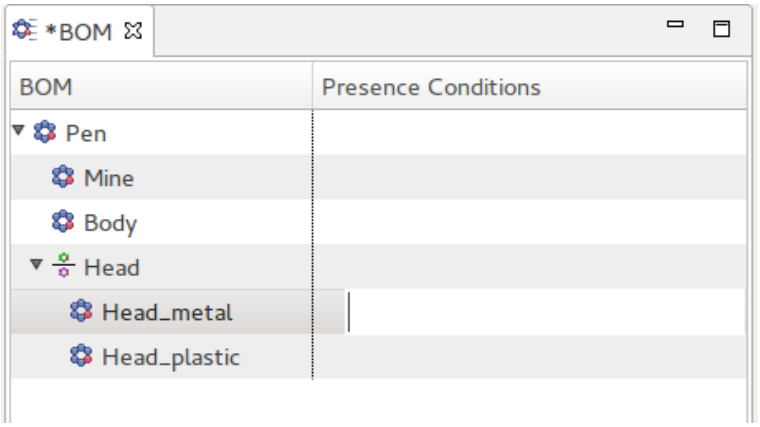


5. You might have noticed, that “Quality” does not have the tree icon, as “Product Features”. This is, because Quality is an Alternative Feature. This means, that for “Quality” only exactly one Sub-Feature - either “HighQuality” or “LowQuality” - can become part of a product.

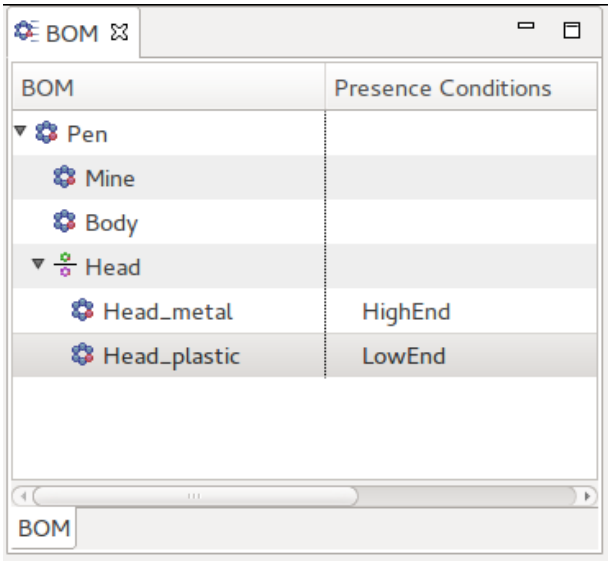
These features you have just created, can now be used, to document how the decision “HighQuality” vs. “LowQuality” affects the manufacturing. To document this in the BOM, do the following:

1. Open the BOM and notice the column *Presence Condition* next to the column with the Parts.

2. When clicking into the Presence Condition fields right of the Parts, there is the possibility to add text. These *Presence Conditions* allow to define dependencies to features:



3. For "Head_ metal" type "HighQuality" and for "Head_ plastic" type "LowQuality":

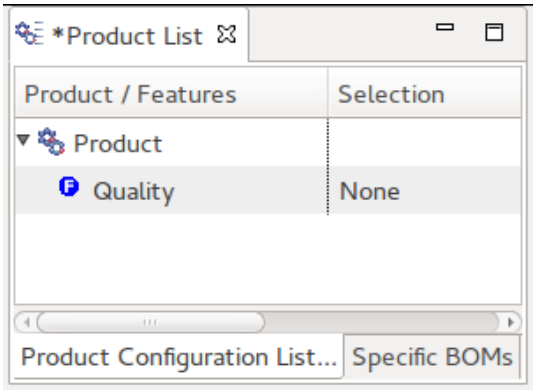


Like this, we specified, that the "Head_metal" is used for pens with the "HighEnd" feature, while the "Head_plastic" is chosen for "LowEnd" pens.

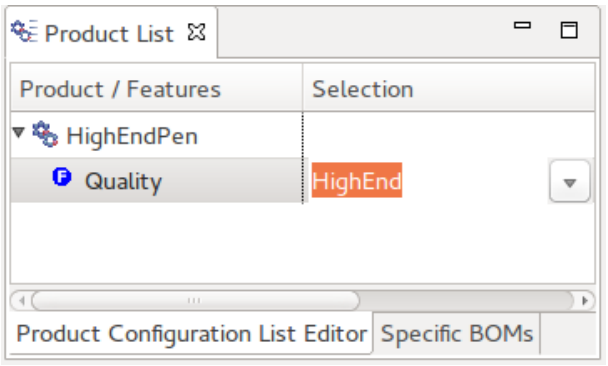
3.3 Configuring Products and Product Specific BOMs

To make the meaning of this more clear, we we will have a look at the BOMs for specific products now:

1. Open the Product List.
2. Add a new Products (again by drag-and-drop, as before):

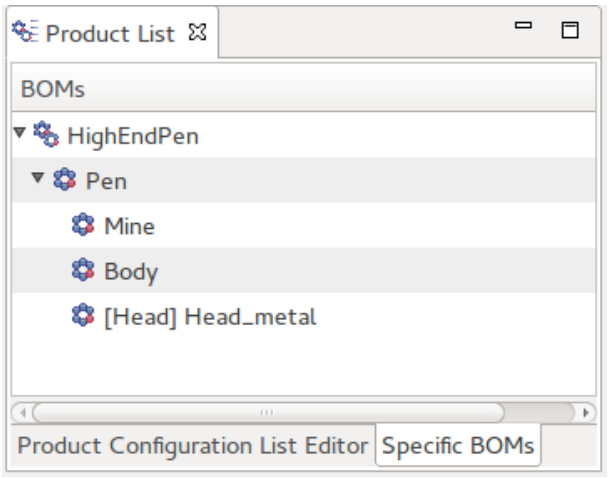


3. Give it a name and select a Quality:



4. Save and select *Specific BOMs* (the second tab under the editor). Now you can see, that for a pen with "Quality" "HightEnd", the BOM contains

the Part “Head_ metal”:



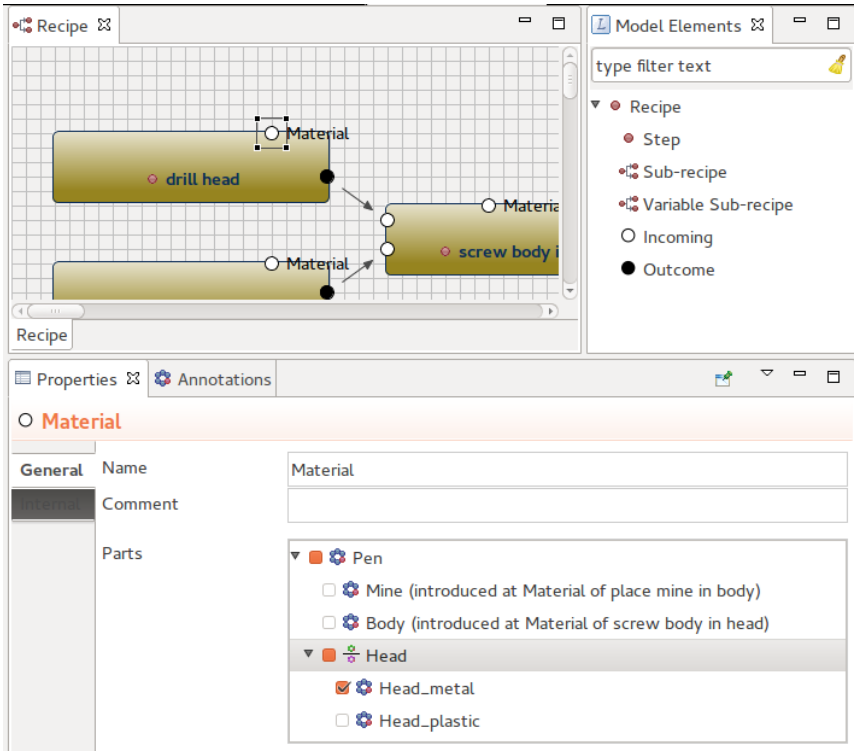
Feel free, to create another product with “LowEnd” “Quality” if you want.

3.4 Variability in Recipes

As already mentioned in the beginning of this part of the tutorial, the change in the product also requires a little different production processes. Hence the Recipe needs to be adopted:

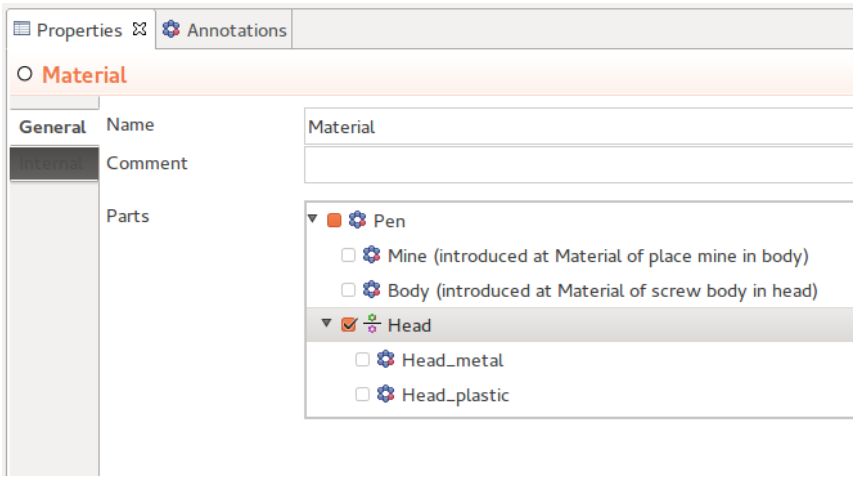
1. Open the recipe.

2. Select the “Material” income of “drill head”:



3. See, that “Head_metal” is selected as incoming material (recall, that you can find this information in the properties view). The reason for this is, that during the refactoring of “Head” to an Alternative Part, the original part “Head” was maintained as one of the alternatives (we renamed it afterwards to “Head_metal”). However, drilling is required for both “Head” variants now. Deselect “Head_metal” and then select the Alternative Part

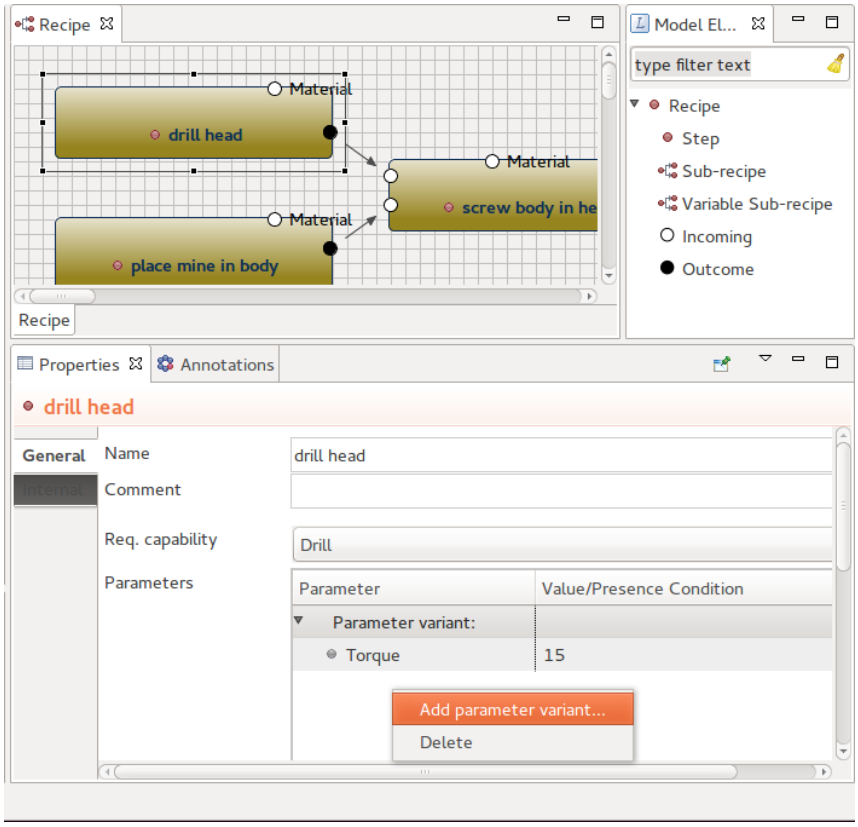
"Head" instead:



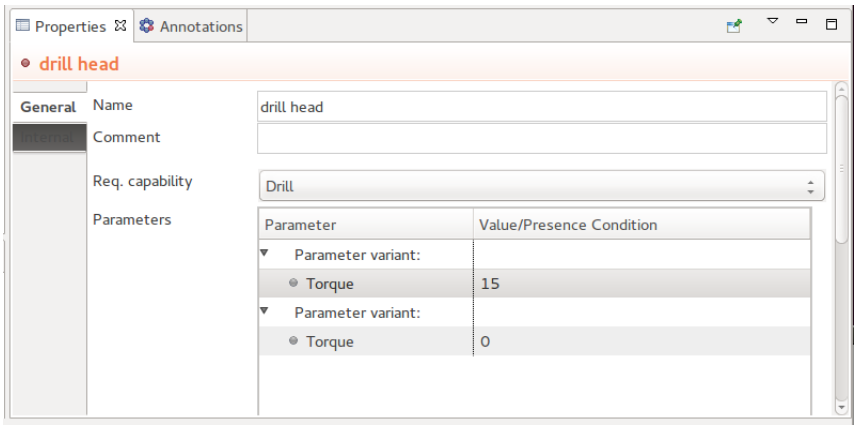
4. With this, the Step "drill head" is supposed to drill whatever "Head" there comes. I.e. no matter whether for the current product a "Head_metal" or a "Head_plastic" is required, it shall be drilled.

5. The new metal part "Head_metal" needs to be drilled with a higher Torque then the original part. For the "Head_plastic" however, the current selected Torque of 15Nm however is the maximum possible to drill without damaging the plastic material - two different Torques need to be used. To solve this problem, the SFIT tool can specify different torques for the different materials. To do this, select the step "drill head" and do a right-click into the *Parameters* field of the Properties view. In the opening

context menu, select *Add parameter variant...*:

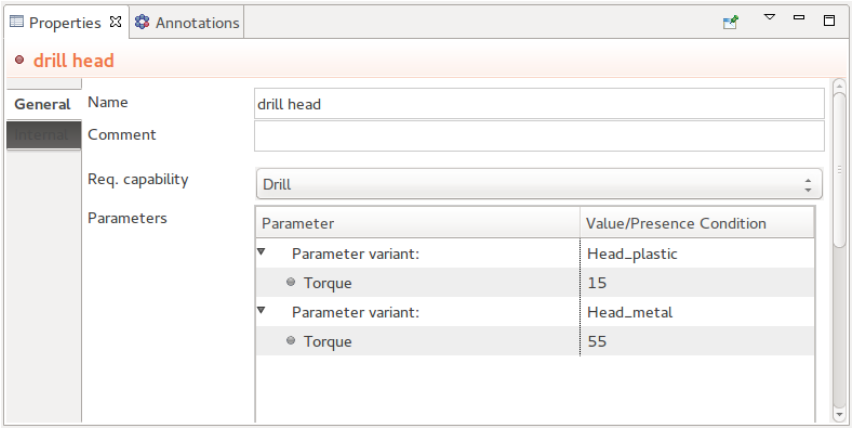


6. As a result you will see another entry for the "Torque" Parameter:



7. To specify for which Part which torque is relevant, add Presence Conditions like you did in the BOM previously and set the Torque for "Head_metal"

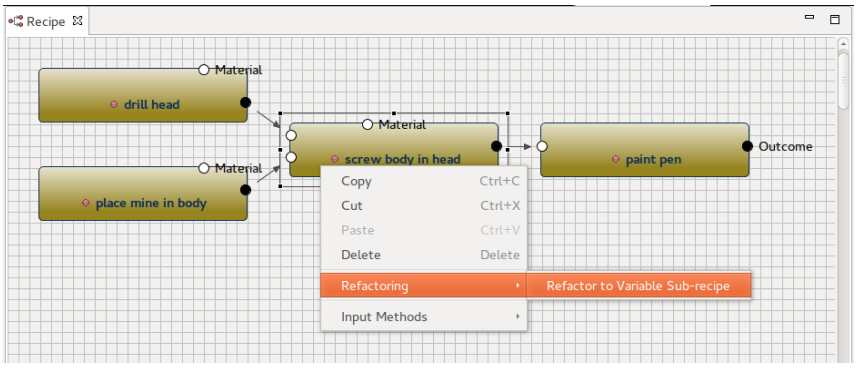
to 55:



Note, that here we used Parts ("Head_metal" and "Head_plastic") in Presence Conditions. Before - in case of the BOM - it had been Features ("HighEnd" and "LowEnd"). Features could have been used here as well. However talking about Parts is more appropriate in this context here. Even complex entries like "PartA OR FeatureX", "NOT FeatureY" and similar would be possible.

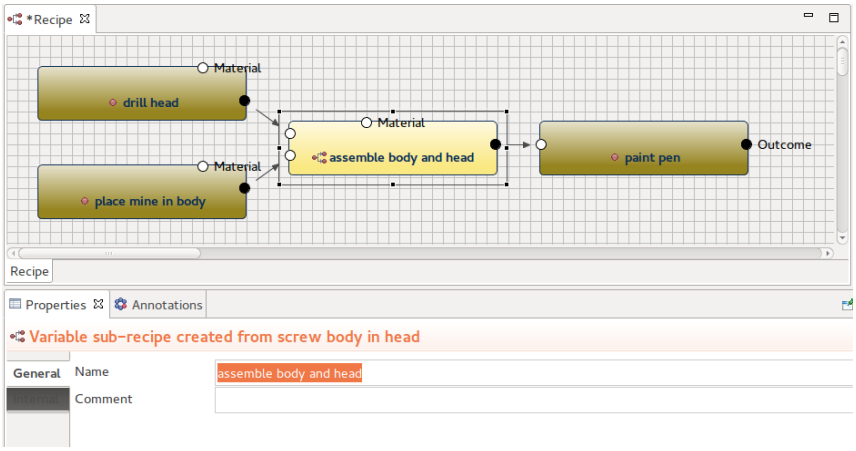
With this, the Step "drill head" is adopted to be able to deal with all variant of the product. Now remember that for the low cost pen the head and body parts shall be glued instead of being screwed. At the same time, the high quality metal pen shall still be screwed (as it was for the old single pen). Let's adopt the Recipe to take that into account:

1. Select the Step "screw body in head".
2. Right-click on the step[4] and select *Refactoring* → *Refactor to Variable Sub-recipe*:

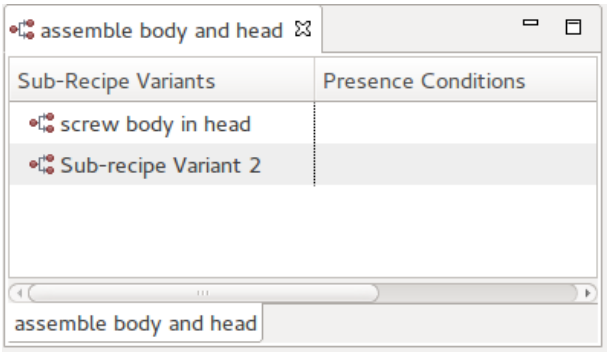


[4]Due to a technical problem in the software, it is required to select the step first. This is subject to be fixed in the future.

3. As a result, the Step will be refactored to a *Variable Sub-recipe*. Rename it to "assemble body and head":

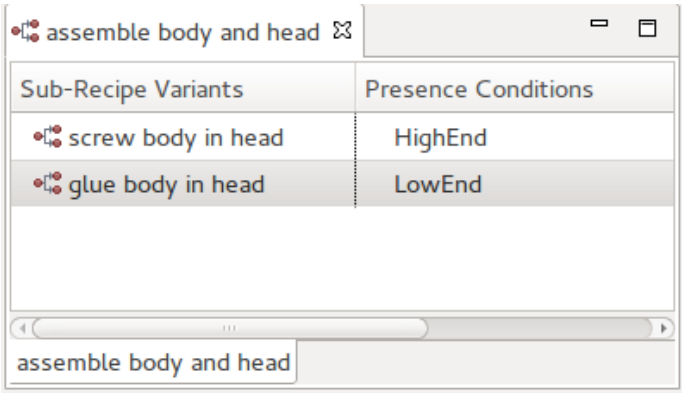


4. Perform a double click on "assemble body and head". A new editor for Variable Sub-recipes opens:

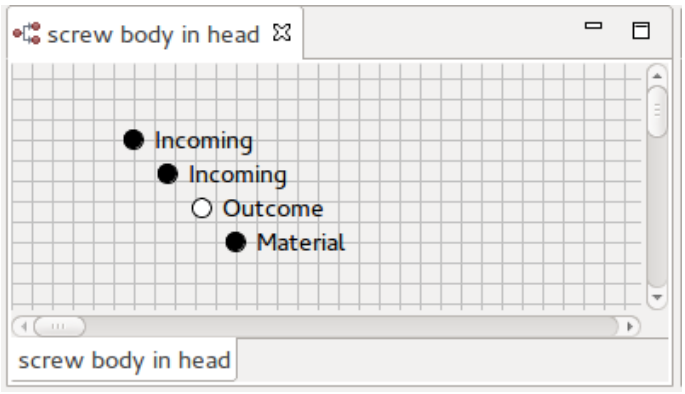


5. The two entries you can see, are the Variants for the Step "assemble body and head". As you can see, on entry has the name of the old step and a second one is already added for convenience. (Note, that you could add more then two Variants. But we won't need this here.)

Rename the second entry and add these Presence Conditions:

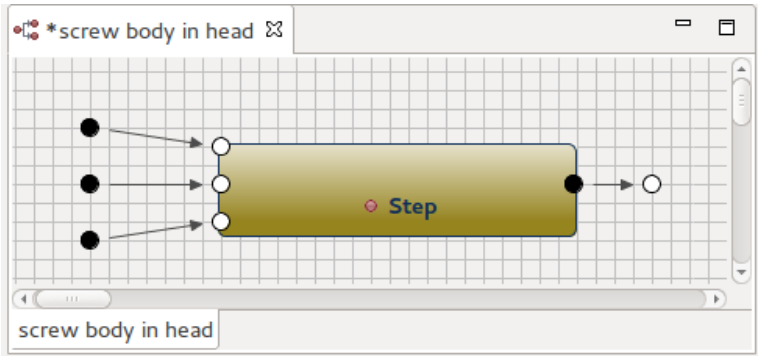


6. To see, how such a Variant for a Variable Sub-recipe looks like, perform a double click on "screw body in head". The Variant will get opened:

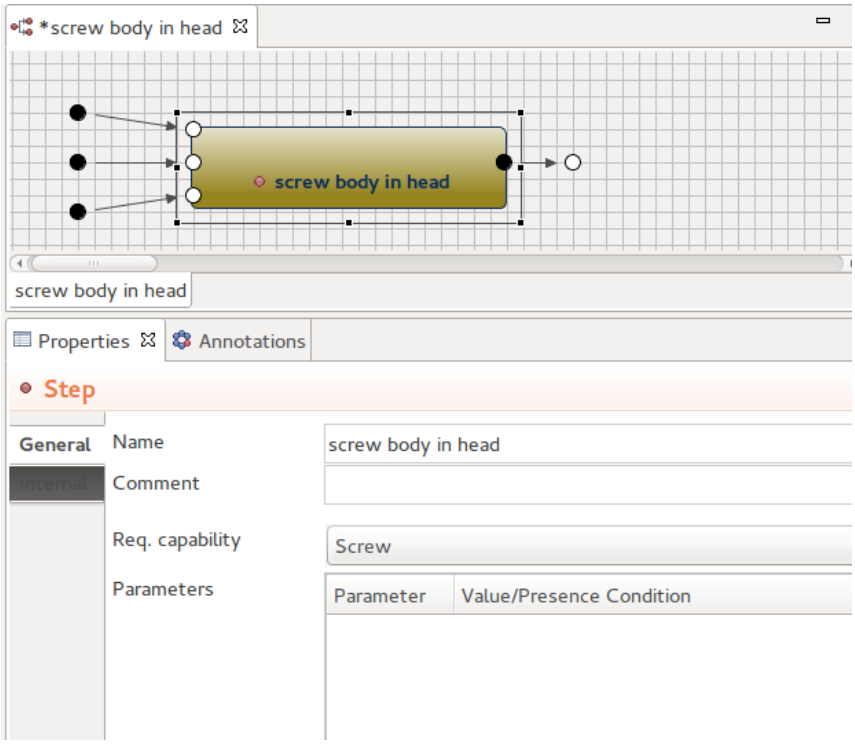


7. Here you can see exactly the Incoming and Outgoing Ports of the Variable Sub-recipe "assemble body and head". (Only the color is switched, since here you look on them from the inside of the Variable Sub-recipe; and where material was going *into* a Port from the outside, it will come *out* that Port from the inside. You can imagine this as it would be for a hole in the Variable Sub-recipe shell.) To specify, how "**screw** body in head" is done, add a Step and connect it. Connecting the Ports to the Step is done like between Steps - Ctrl + mouse-drag (Alt + mouse-drag on Mac).

You can rearrange all elements, such that it can look like this:



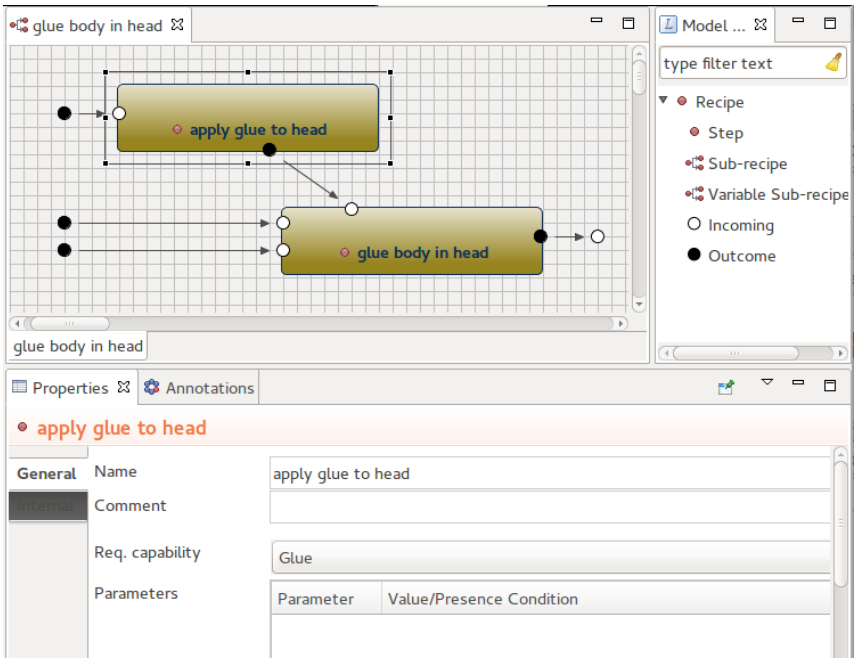
8. Rename Step to “screw body in head” and select the Capability “Screw” for the Step:



9. Use you knowledge to also specify the other Sub-recipe Variant “glue body in head” like in the following screenshot.
Note hereby:

   - You will have to create the Capability “Glue”.
   - Use the Capability “Assemble” for the Step “glue body in head”

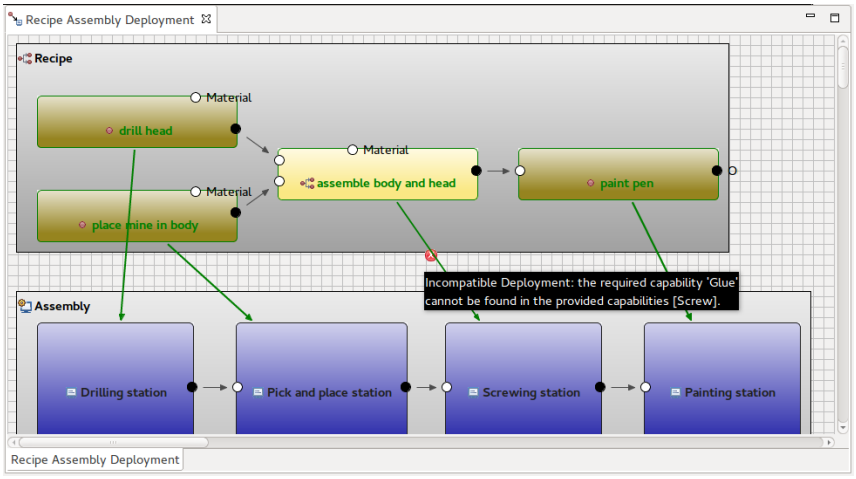
The result here should look similar to this:



3.5 Variability in Assembly and Deployment

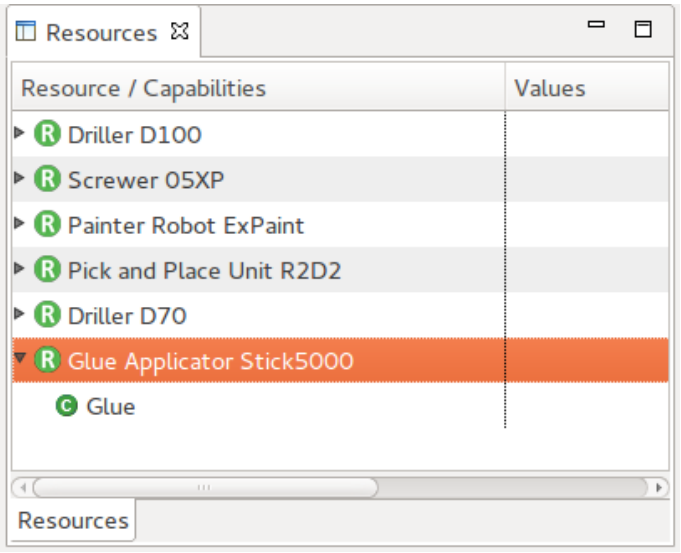
As you might have noticed earlier, the Recipe Assembly Deployment is not consistent anymore. Let's fix it:

1. Open the Deployment and notice the error in the deployment of "assemble body and head":

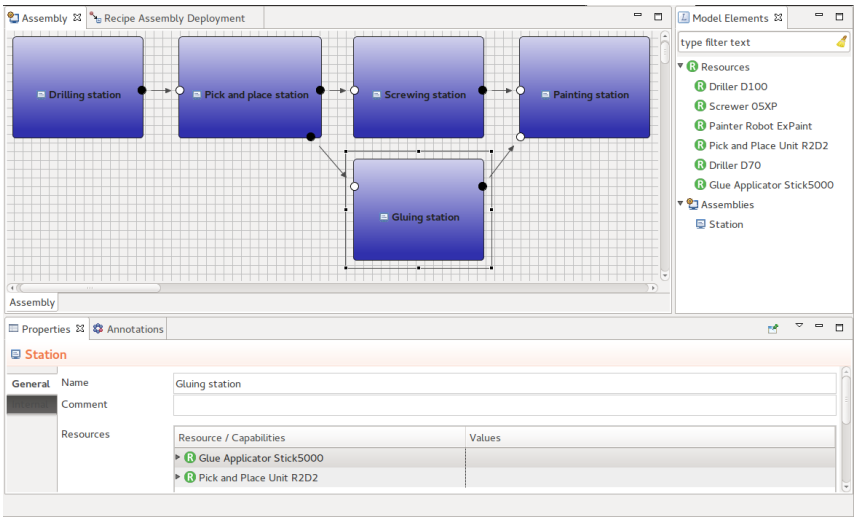


2. The problem is, that the screwing station is not capable of “Glue” - there is no resource with Capability “Glue” in the “Screwing station”.

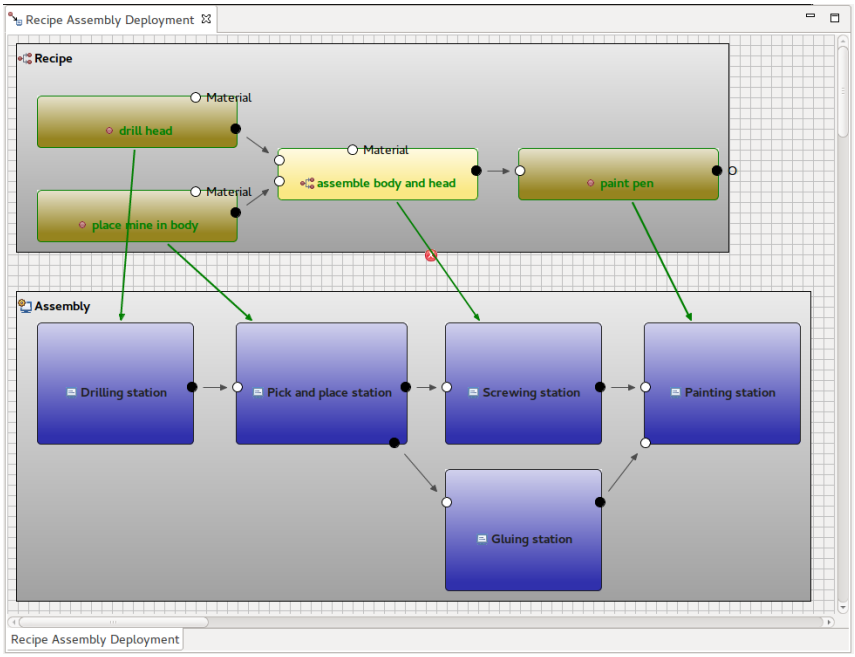
3. So far, there was no gluing machine in the factory. A new “Glue Applicator Stick5000” is bought to bring this functionality. Create it also in the SFIT tool with the capability “Glue”:



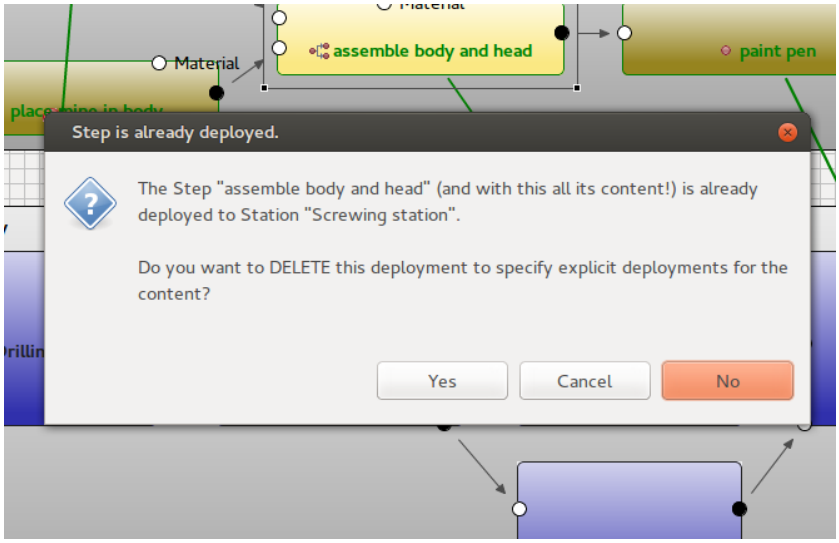
4. Unfortunately in the “Screwing station” there is no room for the new machine. The decision is made to build a new station for gluing. Enhance the Assembly Line with the new station and add the “Glue Applicator Stick5000” to it. The station will also need a “Pick and Place Unit R2D2” - also add it. The result should look similar to this:



5. Save and open the Recipe Assembly Deployment. Of course, there still is the error, but we can see the new station now:

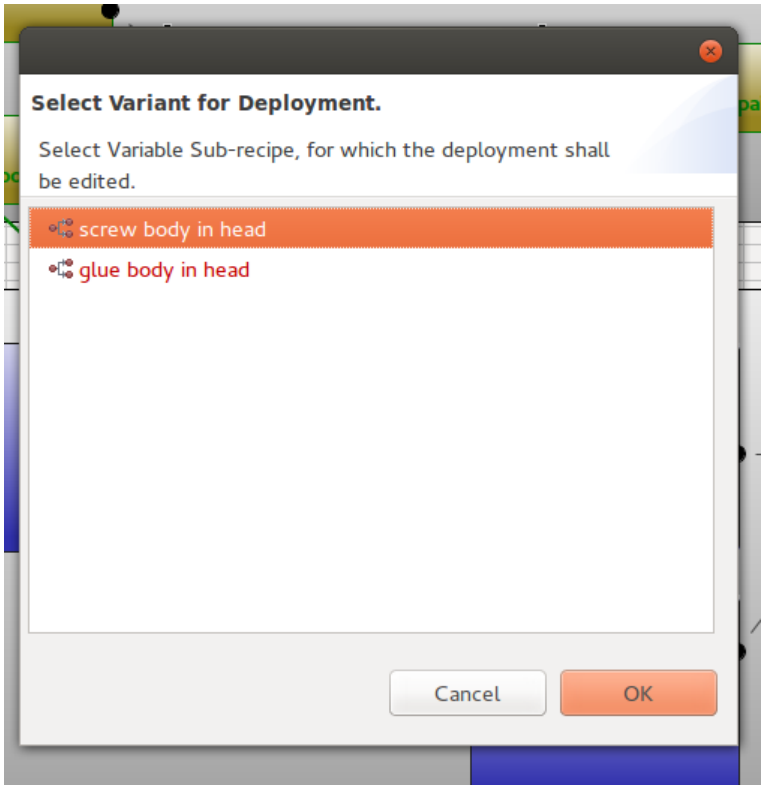


6. The gluing Variant inside “assemble body and head” should be done on the new “Gluing station”. To deploy it there, double click on “assemble body and head”. The following warning will show up:



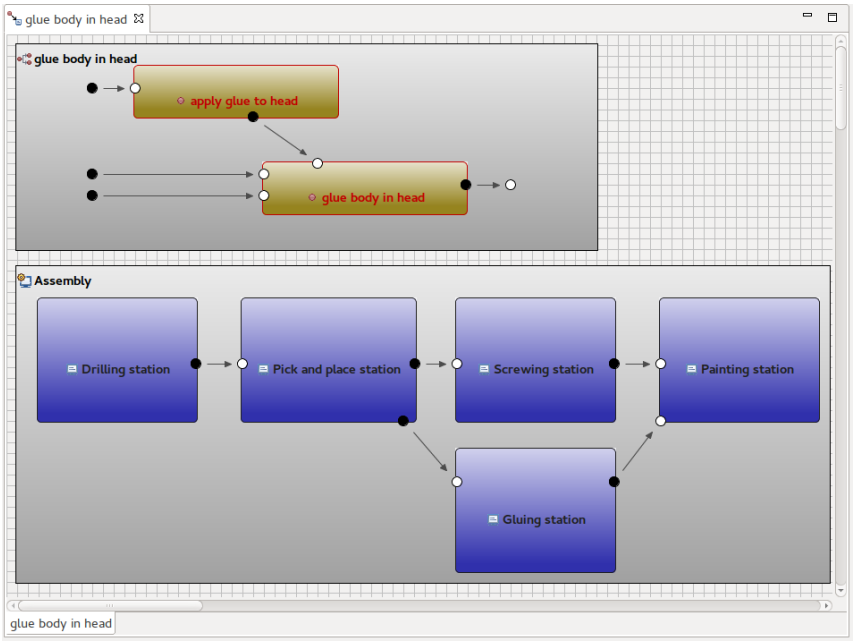
7. What this means is, that when we want to define deployments for steps

inside of a Variable Sub-recipe, we have to delete the existing deployment arrow first. The tool can do this - click *Yes*. Now you can select the Variant:

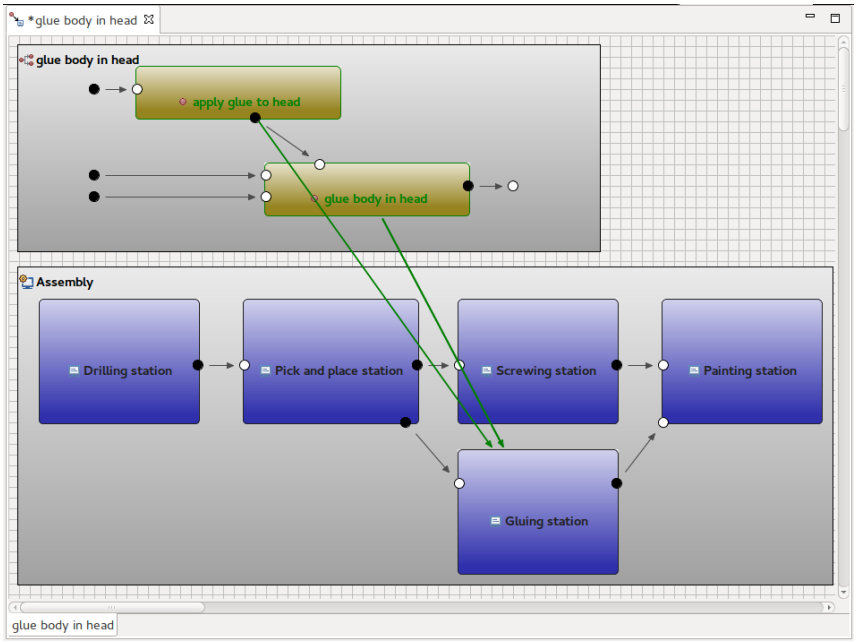


8. Select "glue body in head" and click *OK*. A Sub-deployment window will

open:

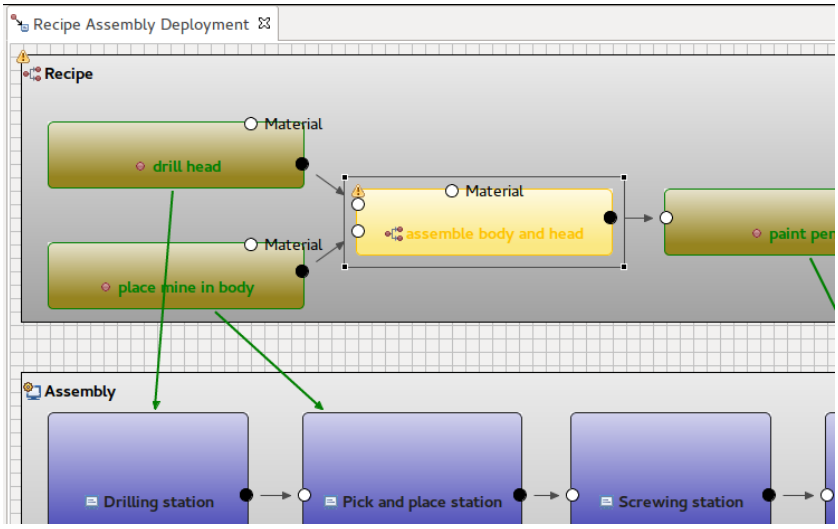


9. Deploy both steps to the “Gluing station”:

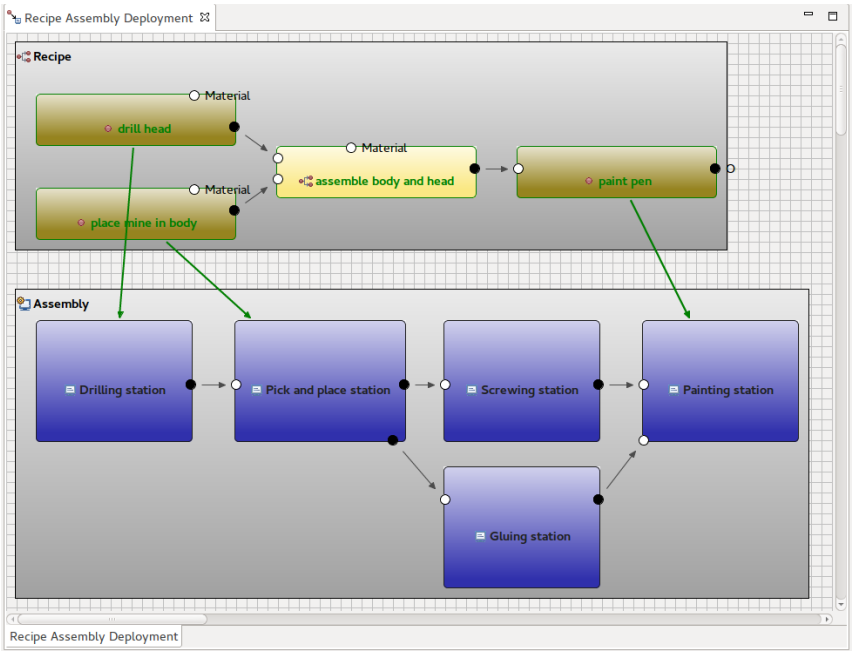


10. Save and open the Recipe Assembly Deployment again. Note, how the

Step got yellow now:

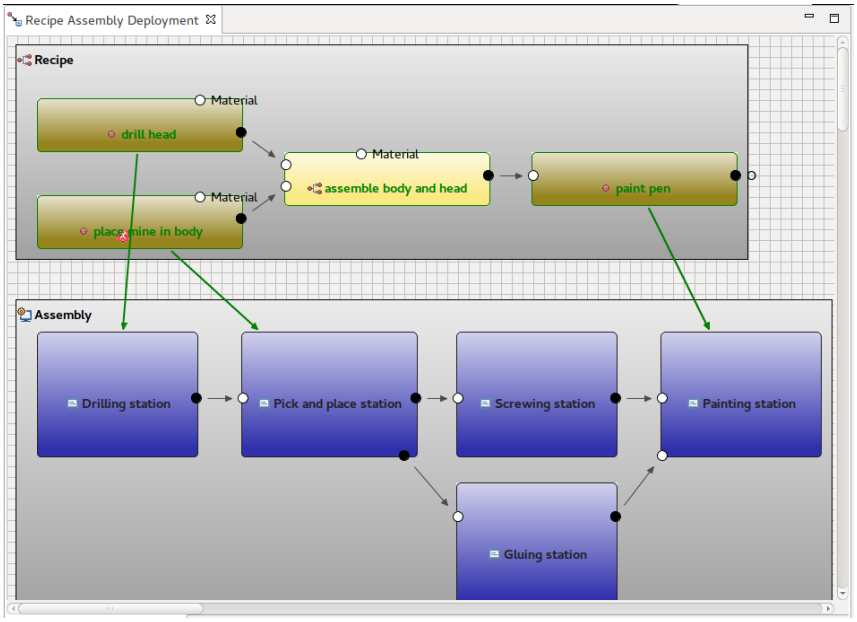


11. The yellow color indicates, that parts of the Step are deployed. Complete the deployment by deploying the Variant “screw body in head”:



12. Go back to the Recipe Assembly Deployment and note, that the Variable

Sub-recipe is completely deployed now[5] and the all steps are green again:



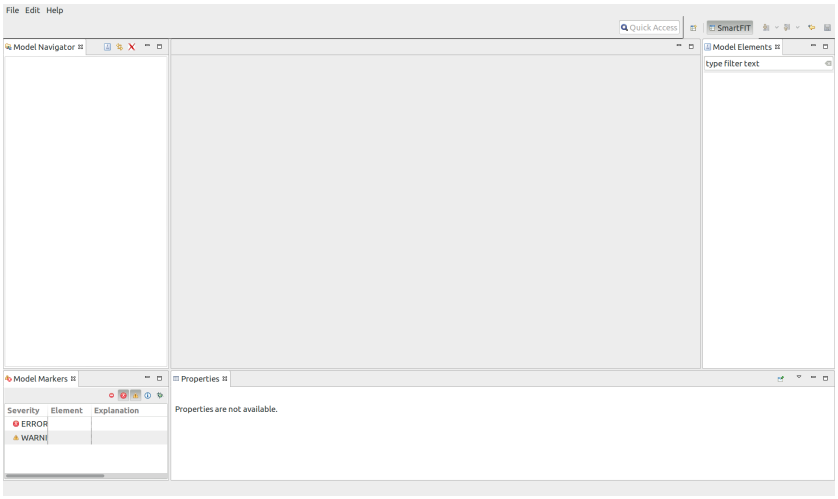
With this, this tutorial is complete. You have successfully enhanced your manufacturing plan to deal with multiple products of a product line.

[5] In fact, here there would also have been an easier solution: The Variable Sub-recipe itself could have been deployed to "both" stations at the top level at the same time. The tool would have figured out by itself, which station needs to be used for which variant.

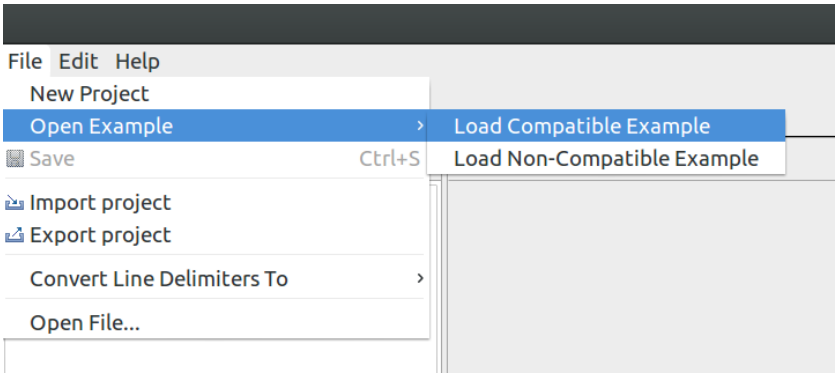
4 Standard Examples

SFIT comes with some pre-made examples:

1. Double-click on `sfit` to launch the application. You should now see a window similar to the following:



2. In the File menu, select the entry "Open Example", and choose then "Load compatible example"



3. Use the project explorer to explore the model