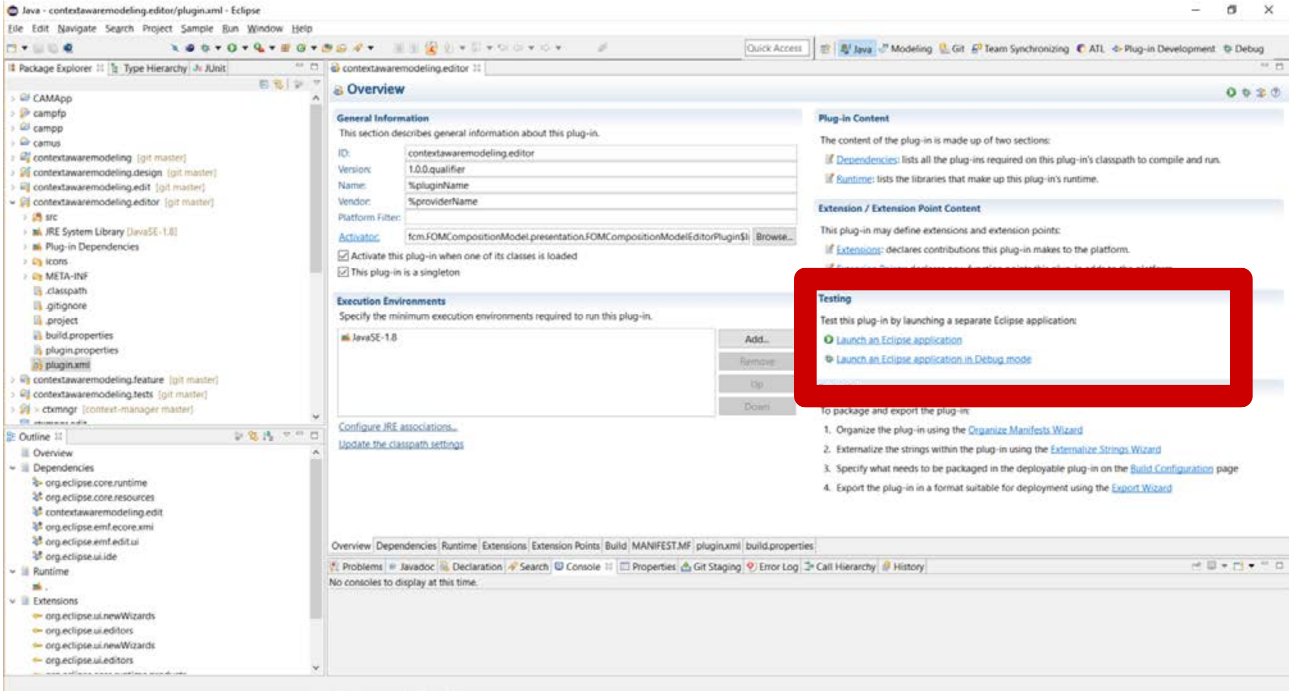
To work with the editor import the projects contextawaremodeling + .edit, .editor, .design into your workspace (if you want to edit the sirius view-specification, it is easier to have it in the Eclipse Instance started from this workspace. To not get confused as to what viewpoint to select it is a good idea to close the contextawaremodeling.design-project in the initial workspace).

You can run the application from the plugin-overview tab of the editor project (actually it does not matter which project you use, the Runtime Eclipse Application is always launched with alle the plugins from the projects in your workspace)


Java - contextawaremodeling.editor/plugin.xml - Eclipse
File Edit Navigate Search Project Sample Run Window Help
Quick Access
Java
Modeling
Git
Team Synchronizing
ATL
Plug-in Development
Debug
Package Explorer
Type Hierarchy
JUnit
CAMApp
campfp
campp
camus
contextawaremodeling [git master]
contextawaremodeling.design [git master]
contextawaremodeling.edit [git master]
contextawaremodeling.editor [git master]
src
JRE System Library [JavaSE-1.8]
Plug-in Dependencies
icons
META-INF
.classpath
.gitignore
.project
build.properties
plugin.properties
plugin.xml
contextawaremodeling.feature [git master]
contextawaremodeling.tests [git master]
ctxmngr [context-manager master]
Outline
Overview
Dependencies
org.eclipse.core.runtime
org.eclipse.core.resources
contextawaremodeling.edit
org.eclipse.emf.ecore.xmi
org.eclipse.emf.edit.ui
org.eclipse.ui.ide
Runtime
Extensions
org.eclipse.ui.newWizards
org.eclipse.ui.editors
org.eclipse.ui.newWizards
org.eclipse.ui.editors
contextawaremodeling.editor

Overview

General Information

This section describes general information about this plug-in.

ID: contextawaremodeling.editor
Version: 1.0.0.qualifier
Name: %pluginName
Vendor: %providerName
Platform Filter:
Activator: fcm.FOMCompositionModel.presentation.FOMCompositionModelEditorPlugin$I Browse...

☑ 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.

JavaSE-1.8

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.

Testing

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

Launch an Eclipse application

Launch an Eclipse application in Debug mode

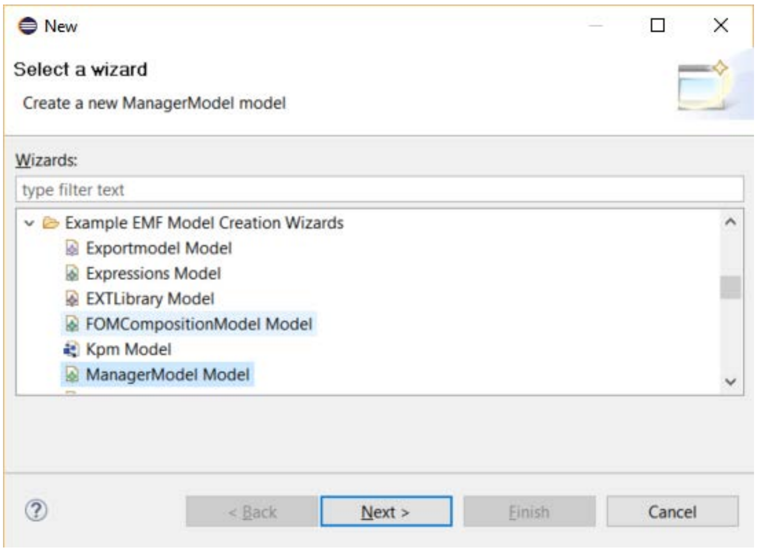
To package and export the plug-in:

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

No consoles to display at this time.

To test if your Runtime-Eclipse is working import the project contextawaremodeling.test which contains some FOM, FOMC and HOM models and their representation.

To start from scratch, create a modeling Project and then create a new Model (New → Other). Here you can find the ManagerModel (used for .fom and .hom) and the FOMCompositionModel (.fomc). Click on Next and select Manager (for .fom and .hom) or Context Manager (for .fomc) as the model object and hit finish.



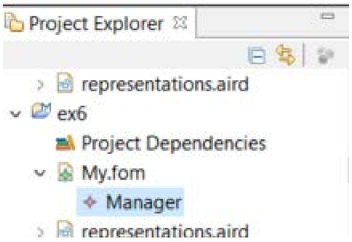
Select a wizard

Create a new ManagerModel model

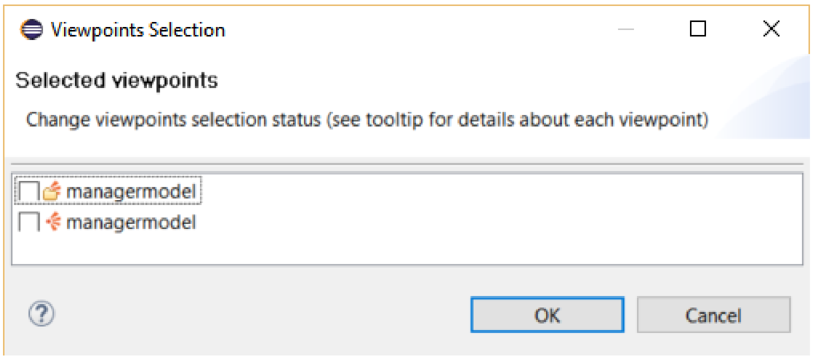
Wizards:

type filter text

Now you should be able to expand the model file in the project explorer



Now right click on the project and select a viewpoint (you will only have two like me here if you have the .design project in open in both the initial Eclipse and the Runtime Eclipse, the first one is the one from the Initial Eclipse and the second one that from the Runtime Eclipse). Select the viewpoint.


Viewpoints Selection

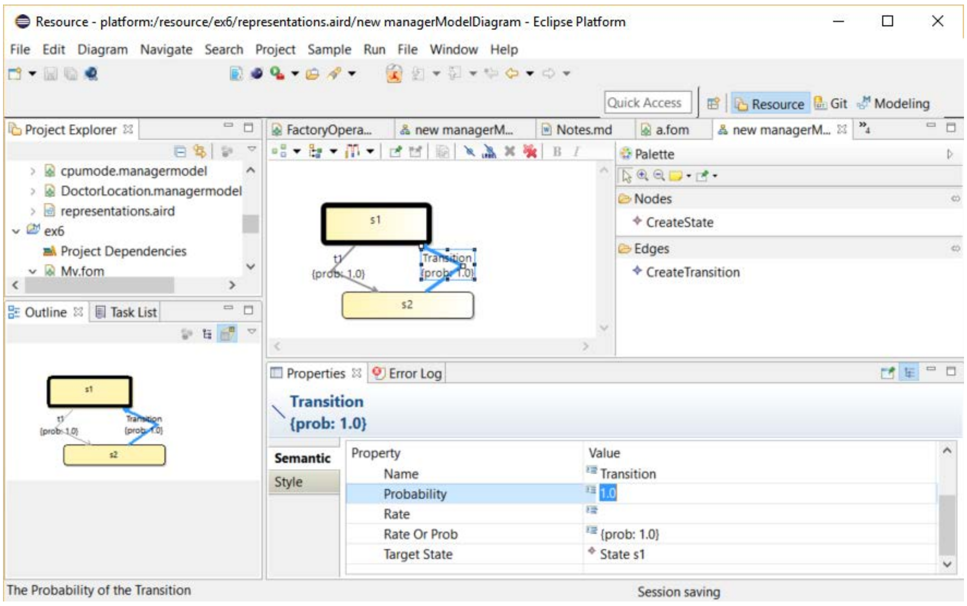
Selected viewpoints

Change viewpoints selection status (see tooltip for details about each viewpoint)

□ managermodel
□ managermodel

Now you can create a representation of the model. Right click on the Manager (or ContextManager) Element of your model and select „New Representation“ → „New managerModelDiagram“.

Start by naming the model in the Semantic tab of the properties view. Next create some states. Set one as the start state. Create Transitions, name them and give them either a rate or a probability


Resource - platform:/resource/ex6/representations.aird/new managerModelDiagram - Eclipse Platform

Transition
{prob: 1.0}

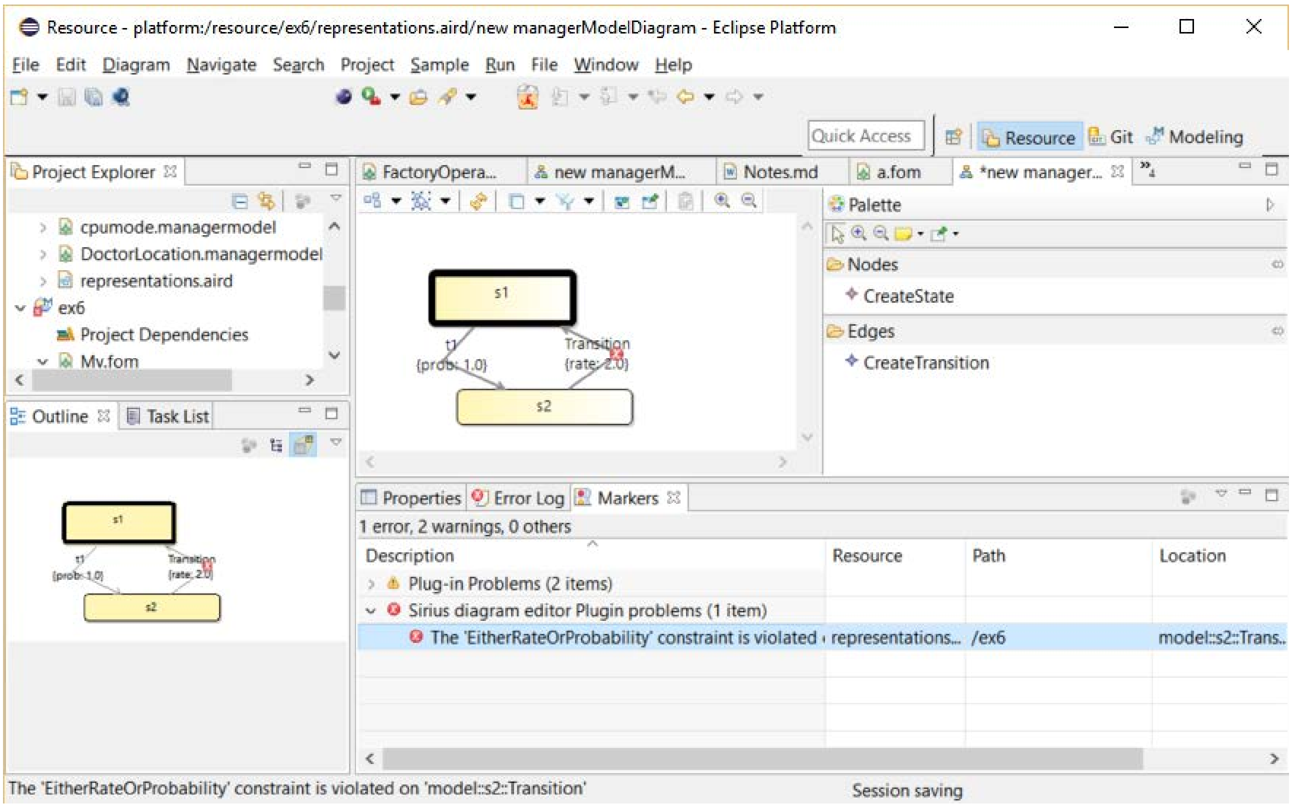
Property	Value
Name	Transition
Probability	1.0
Rate	
Rate Or Prob	(prob: 1.0)
Target State	State s1

The Probability of the Transition

Session saving

Note that you if you want to change from probability to rate or vice versa you will have to right-click the respective field and select **„Restore Default Value“** because otherwise it will sometimes not be nulled out correctly if you just remove the value using backspace. This can lead to violation beeing shown after you validate the diagram (right-click → valitate diagram) that you can't verify.

Heres an example of a violation when both rate an probability are provided by a user.



1 error, 2 warnings, 0 others

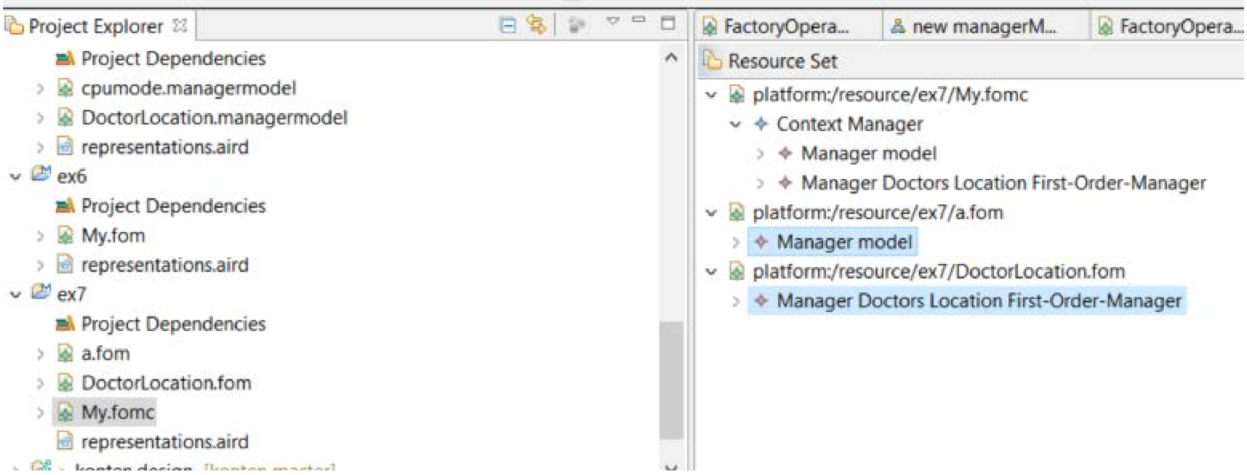
Description	Resource	Path	Location
Plug-in Problems (2 items)			
Sirius diagram editor Plugin problems (1 item)			
The 'EitherRateOrProbability' constraint is violated ‹	representations...	/ex6	model::s2::Trans..

The 'EitherRateOrProbability' constraint is violated on 'model::s2::Transition'

Session saving

If you have 2 or more FOMs you can create a FOMCompositionModel. After creation the tree-editor will open. Right click and select „Load Resources ...“ → „Browse Workspace“ and select the fom-files you need. The models will be shown in this editor now but note that this is not persisted in the file (you can see that in the xml-content) so when you restart the Runtime-Eclipse instance they will be gone.

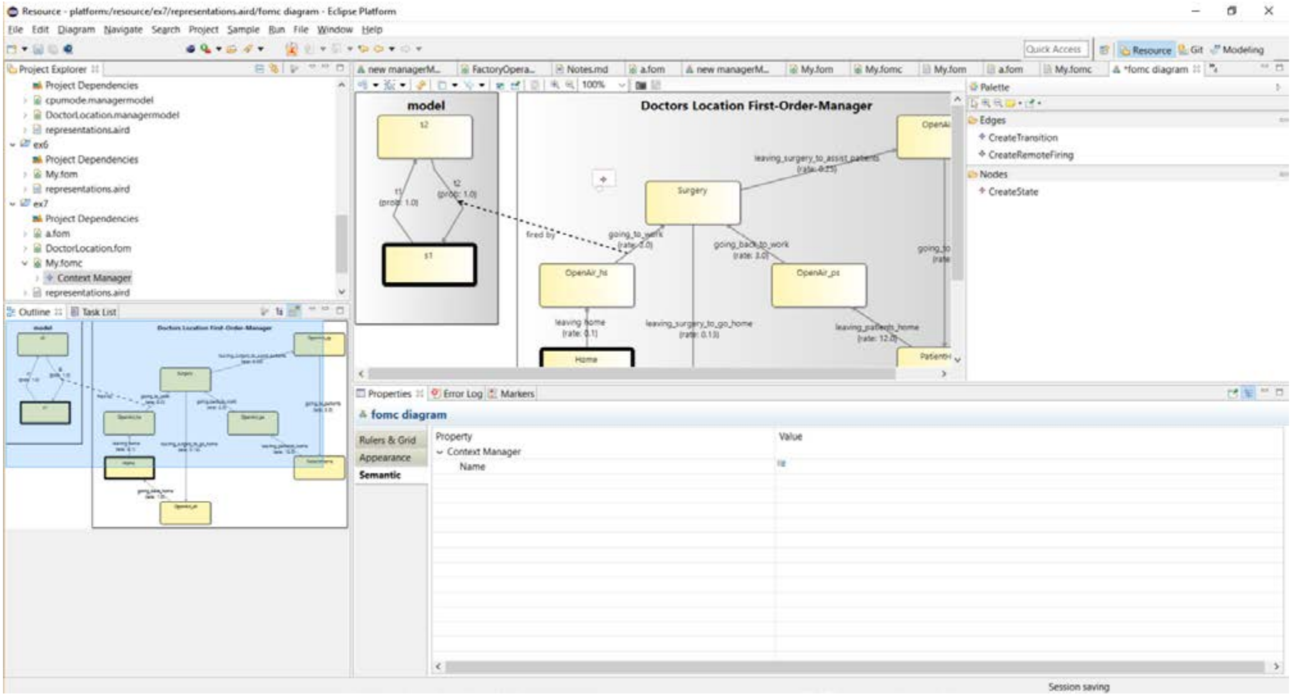
Drag and Drop the Managers in the ContextManager Element of your fomc-model to copy it.



Project Dependencies
cpumode.managermodel
DoctorLocation.managermodel
representations.aird
ex6
Project Dependencies
My.fom
representations.aird
ex7
Project Dependencies
a.fom
DoctorLocation.fom
My.fomc
representations.aird

Resource Set
platform:/resource/ex7/My.fomc
Context Manager
Manager model
Manager Doctors Location First-Order-Manager
platform:/resource/ex7/a.fom
Manager model
platform:/resource/ex7/DoctorLocation.fom
Manager Doctors Location First-Order-Manager

To edit it with the diagram editor you will have to add the „fomcompositionmodel“-viewpoint to your project (see description above). You can edit everything you could in the FOM-Editor plus you can create the Remote firings

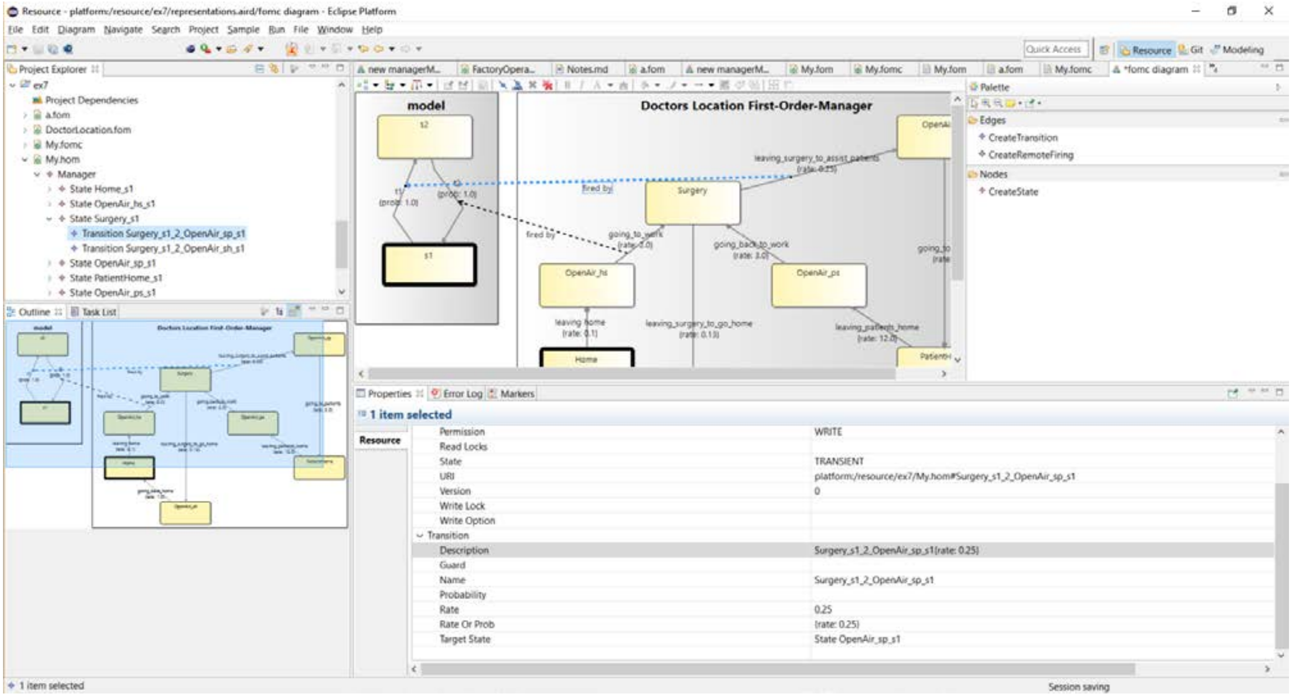


fomc diagram

Property	Value
Context Manager	
Name	

Do your stuff and validate the diagram at the end to make sure it is correct, otherwise the next steps won't work. Make sure you saved the file as well.

When you are finished create a HOM from your fomc-file by using the context-menu in the project-explorer and select „HOM“ → „create HOM“. A new hom-file will be created with the name of the same name as the fomc-file (delete the hom file or rename it if you want recreate it).

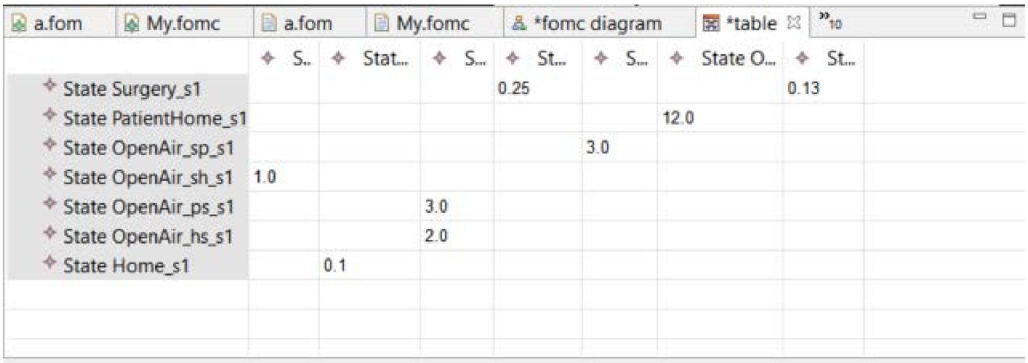

Resource - platform:/resource/ex7/representations.aird/fomc diagram - Eclipse Platform
File Edit Diagram Navigate Search Project Sample Run File Window Help
Project Explorer
ex7
Project Dependencies
a.fom
DoctorLocation.fom
My.fomc
My.hom
Manager
State Home_s1
State OpenAir_hs_s1
State Surgery_s1
Transition Surgery_s1_2_OpenAir_sp_s1
Transition Surgery_s1_2_OpenAir_sh_s1
State OpenAir_sp_s1
State PatientHome_s1
State OpenAir_ps_s1
Outline
Task List
model
Doctors Location First-Order-Manager
s2
s1
t1 (prob: 1.0)
t2 (prob: 1.0)
fired by
Surgery
OpenAir_hs
OpenAir_ps
Home
PatientH
leaving_surgery_to_assist_patients (rate: 0.25)
going_to_work (rate: 2.0)
going_back_to_work (rate: 3.0)
leaving_home (rate: 0.1)
leaving_surgery_to_go_home (rate: 0.13)
leaving_patients_home (rate: 12.0)
Palette
Edges
CreateTransition
CreateRemoteFiring
Nodes
CreateState
Properties
Error Log
Markers
Resource
1 item selected
Session saving

1 item selected

Property	Value
Permission	WRITE
Read Locks	
State	TRANSIENT
URI	platform:/resource/ex7/My.hom#Surgery_s1_2_OpenAir_sp_s1
Version	0
Write Lock	
Write Option	
Transition	
Description	Surgery_s1_2_OpenAir_sp_s1(rate: 0.25)
Guard	
Name	Surgery_s1_2_OpenAir_sp_s1
Probability	
Rate	0.25
Rate Or Prob	(rate: 0.25)
Target State	State OpenAir_sp_s1

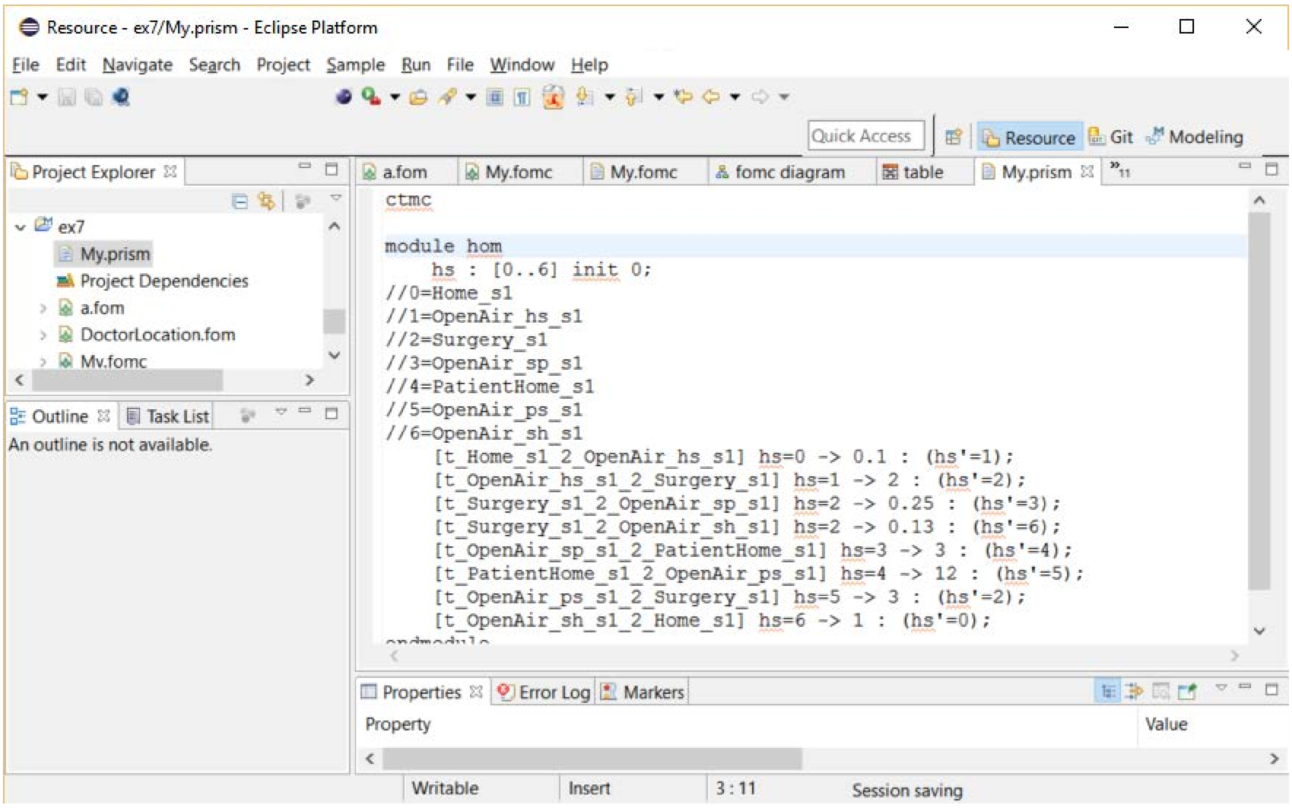
You can create a representation of it the same way you can do this for a hom-model (it uses the same meta-model). If you need to remove impossible states (guards are not yet implemented) i recommend the tree-editor to do that because it gives you a better overview (i generally don't see the diagram-representation of the HOM as being of much value because it gets quite huge and therefore confusing quite easily)

Another possible representation i added is the Cross-Table representation which gives you the matrix representation of the model. From there you can easily export the model to CSV via the context menu.



	S..	Stat...	S...	St...	S...	State O...	St...
State Surgery_s1				0.25			0.13
State PatientHome_s1						12.0	
State OpenAir_sp_s1					3.0		
State OpenAir_sh_s1	1.0						
State OpenAir_ps_s1			3.0				
State OpenAir_hs_s1			2.0				
State Home_s1		0.1					

To create the PRISM file open the context menu of the hom-file and select „CreatePRISMCode“ → „create“. (To recreate it rename or delete the file first).



```
ctmc

module hom
    hs : [0..6] init 0;
//0=Home_s1
//1=OpenAir_hs_s1
//2=Surgery_s1
//3=OpenAir_sp_s1
//4=PatientHome_s1
//5=OpenAir_ps_s1
//6=OpenAir_sh_s1
    [t_Home_s1_2_OpenAir_hs_s1] hs=0 -> 0.1 : (hs'=1);
    [t_OpenAir_hs_s1_2_Surgery_s1] hs=1 -> 2 : (hs'=2);
    [t_Surgery_s1_2_OpenAir_sp_s1] hs=2 -> 0.25 : (hs'=3);
    [t_Surgery_s1_2_OpenAir_sh_s1] hs=2 -> 0.13 : (hs'=6);
    [t_OpenAir_sp_s1_2_PatientHome_s1] hs=3 -> 3 : (hs'=4);
    [t_PatientHome_s1_2_OpenAir_ps_s1] hs=4 -> 12 : (hs'=5);
    [t_OpenAir_ps_s1_2_Surgery_s1] hs=5 -> 3 : (hs'=2);
    [t_OpenAir_sh_s1_2_Home_s1] hs=6 -> 1 : (hs'=0);
endmodule
```