Tutorial 3 — Using SysML

Overview

This second tutorial will show you how to:

1. Add a new FMU in a SysML model
2. Generate a new multi-model configuration
3. Associate an FMU with a multi-model configuration
4. Execute a co-simulation using the new multi-model configuration

Requirements

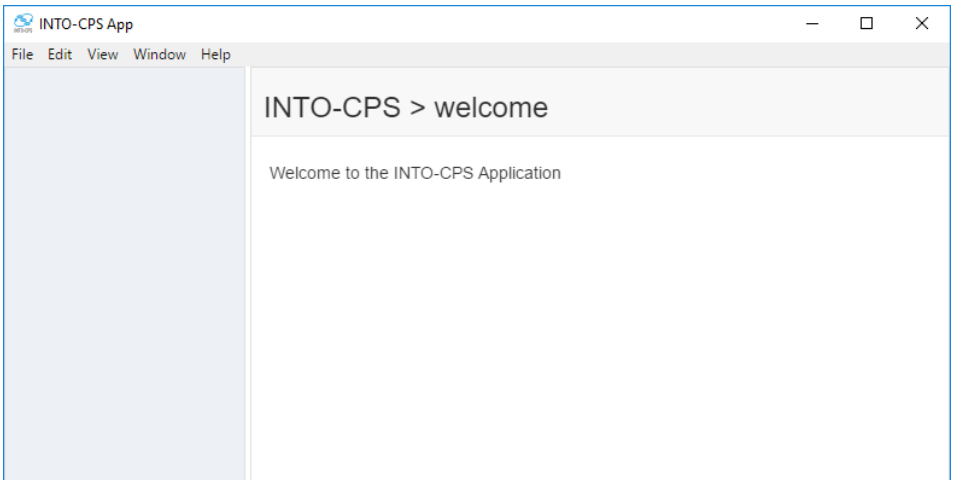
This tutorial requires the following tools from the INTO-CPS tool chain to be installed:

- INTO-CPS Application
- COE (Co-simulation Orchestration Engine) accessible to the Application
- Modelio v3.4.1

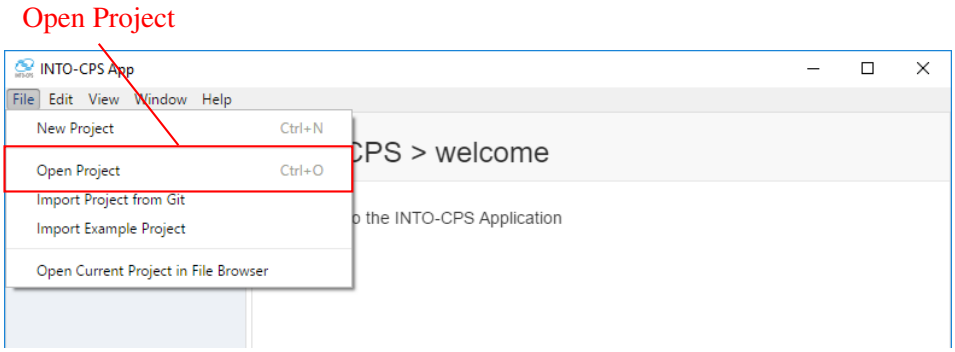
You may have been provided with tools on a USB drive at your training session. Otherwise the INTO-CPS Application can be downloaded from `https://into-cps.github.io/download/` and tools can be downloaded from there through *Window* > *Show Download Manager* to your *into-cps-projects* install downloads directory. Please ask if you are unsure.

1 Opening a Project

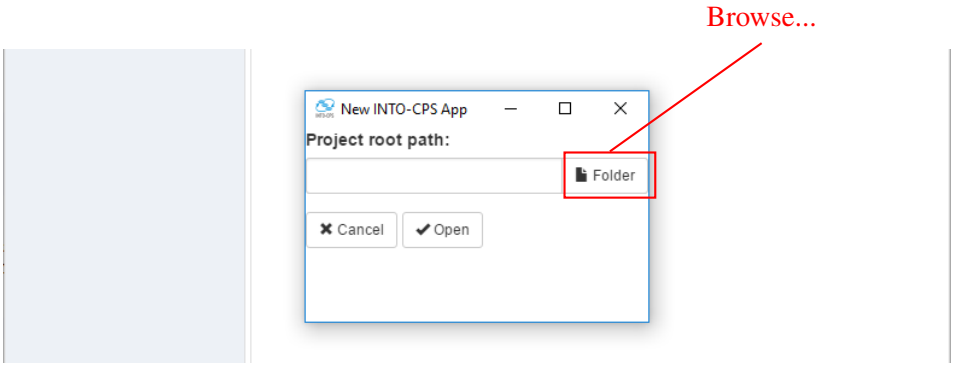
Step 1. Launch the *INTO-CPS Application*. On first loading, it will look like the screenshot below. If you have opened a project previously, that project will be opened automatically.



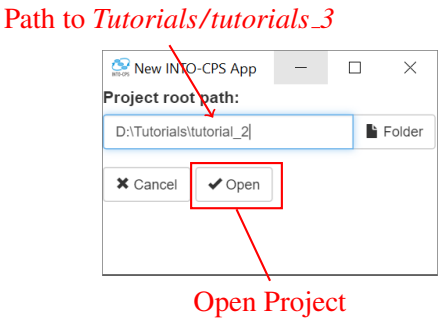
Step 2. To open a project, select *File* > *Open Project*.



Step 3. Set the *Project root path* to the location of *Tutorials/tutorials_3*. You can browse using the *Folder* button.



Step 4. Click *Open Project*.



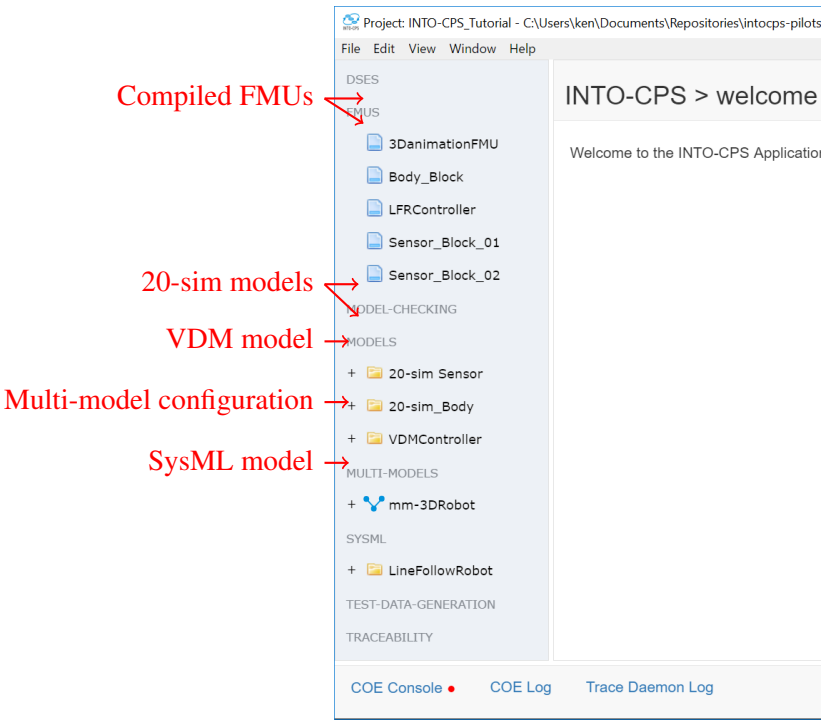
Step 5. Once the project is opened, you will see that project browser on the left of the INTO-CPS Application window is now populated. The entries in the project browser correspond to folders and files in the *Tutorials/tutorials_3* folder. These are:

FMUs Compiled FMUs (with file extension .fmu) that are used in co-simulation.

Models Source models used to generate the FMUs. The icon of each entry shows which tool created the model. In this case Overture and 20-sim.

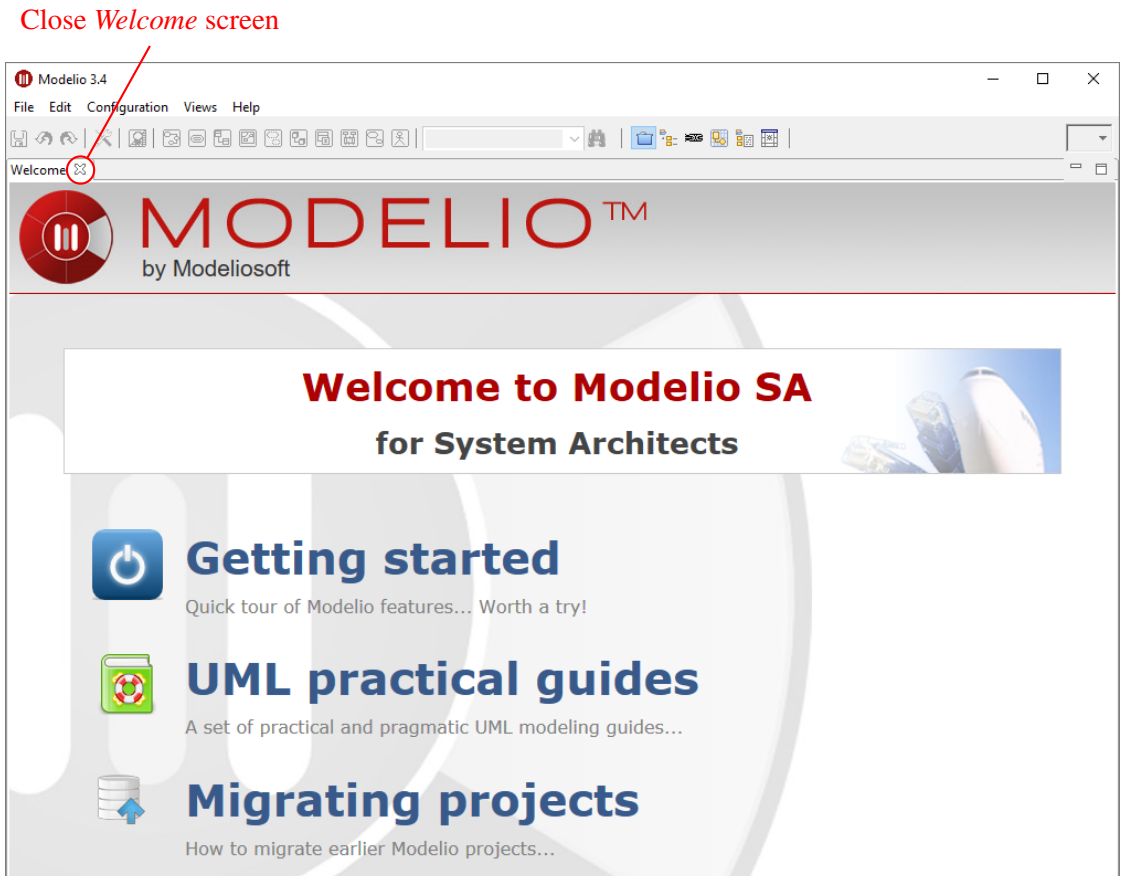
Multi-models Used to configure co-simulations, including which FMUs are used and other co-simulation settings.

SysML Architectural models that are used to create model and multi-model descriptions.

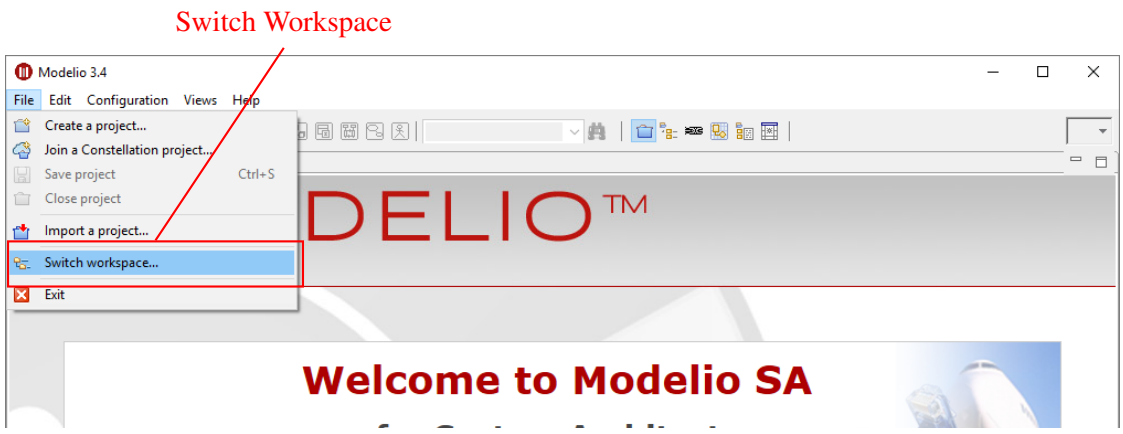


2 Edit Architecture

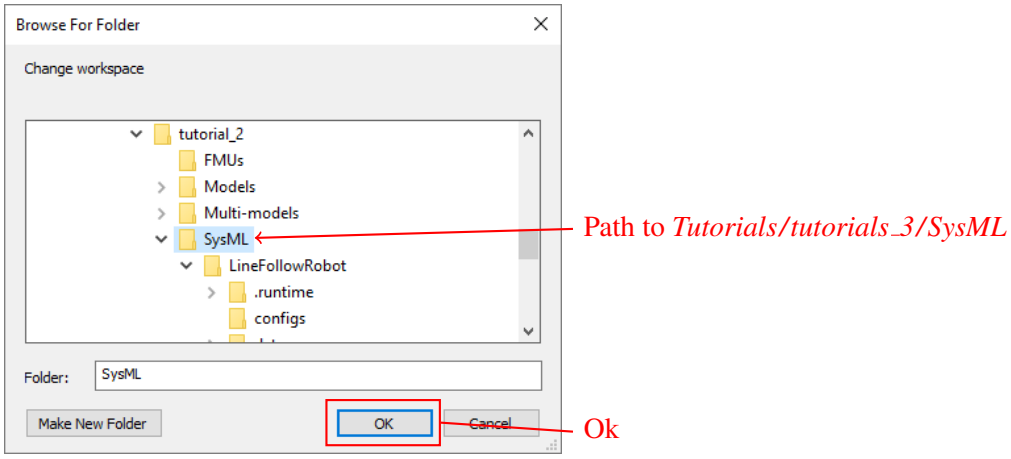
Step 6. Launch *Modelio*. On first loading, you may have to close the *Welcome* screen (you can bring it back with *Help* > *Welcome* if you need)



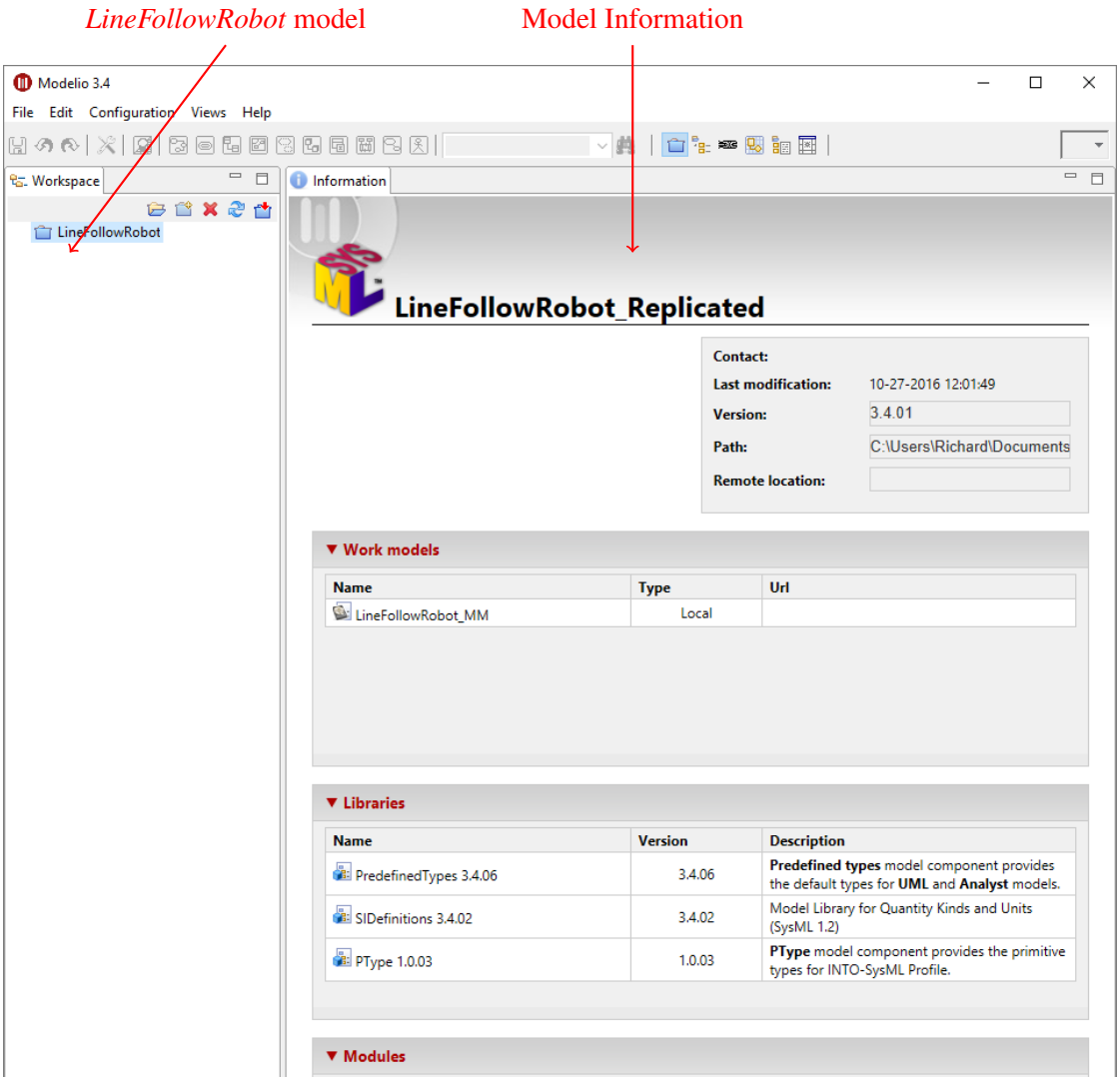
Step 7. A workspace must be chosen, select *File* > *Switch Workspace*.



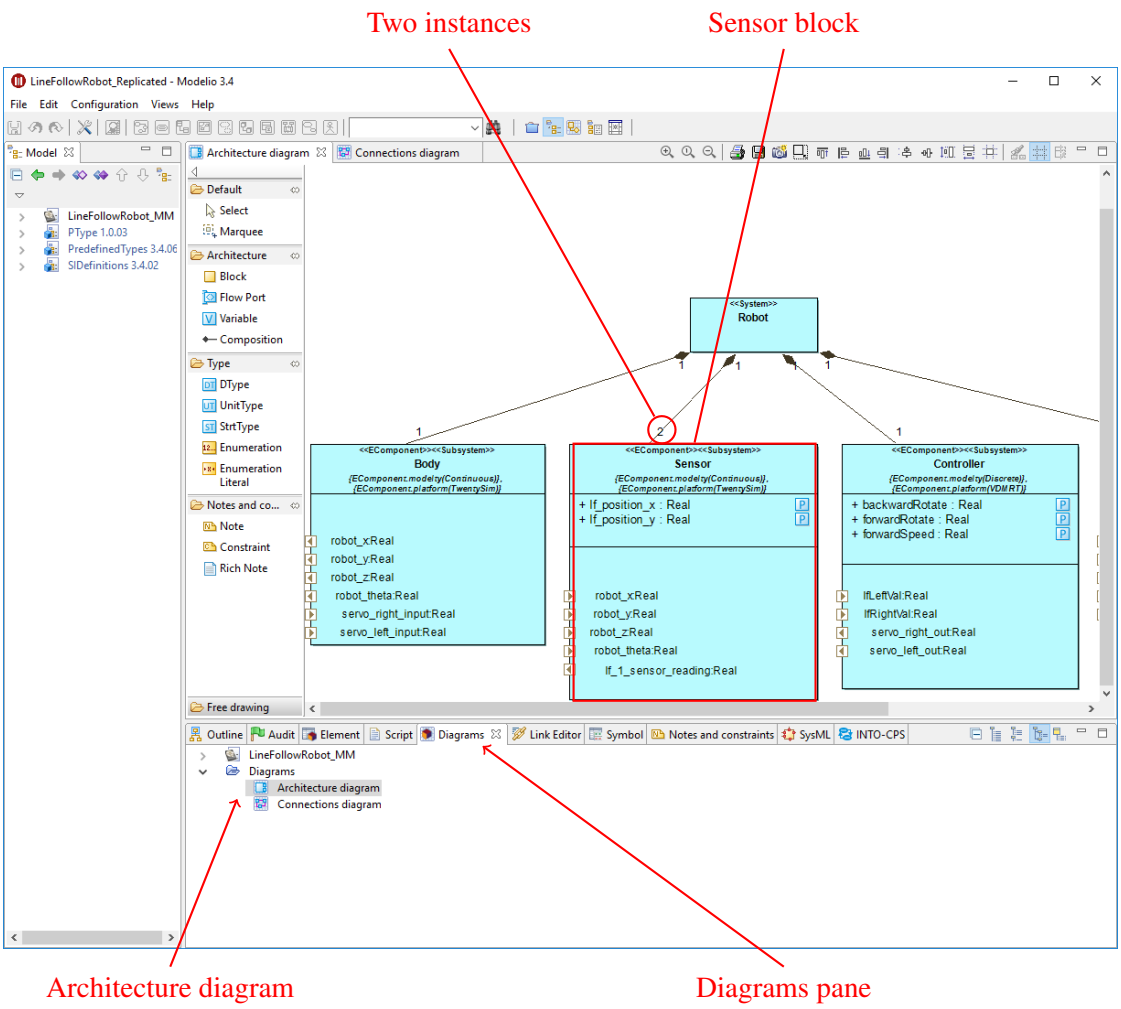
Step 8. Set the *Workspace* to the location of *Tutorials/tutorials_3/SysML* and click *Ok*.



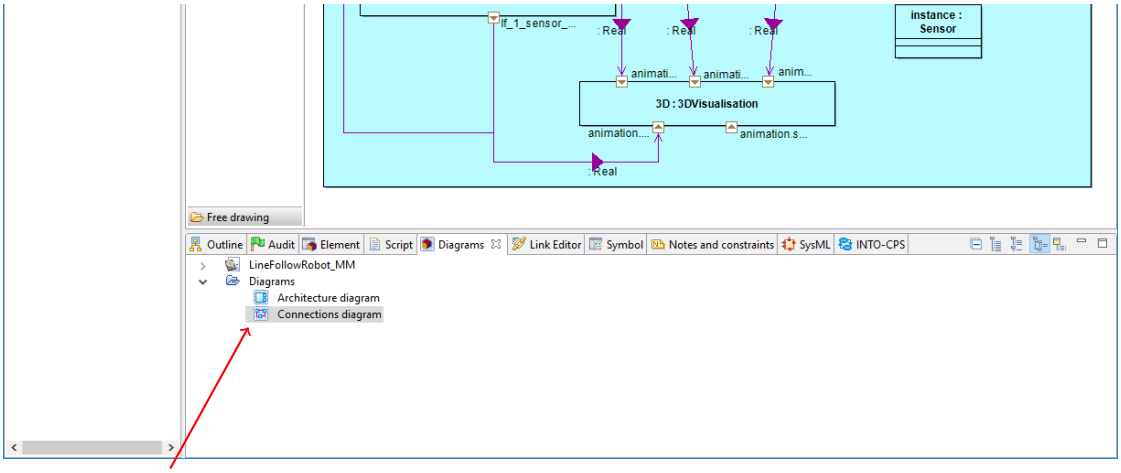
Step 9. Left-click on the *LineFollowRobot* model once on the left to see details of the model. Double-click the *LineFollowRobot* model to open the model.



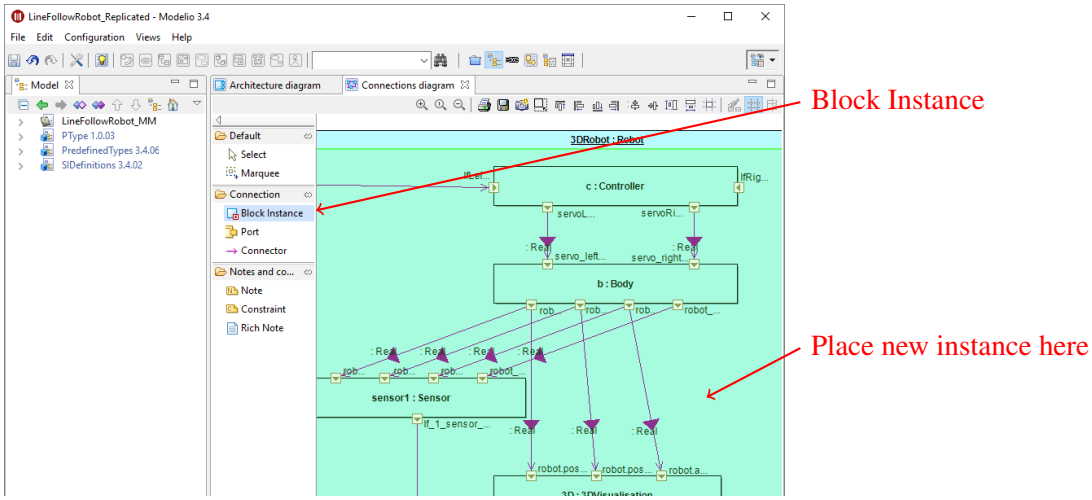
Step 10. In the *Diagrams* pane, expand the *Diagrams* folder and double click the *Architecture Diagram*. The diagram below will open. Notice there are two instances of the Sensor model.



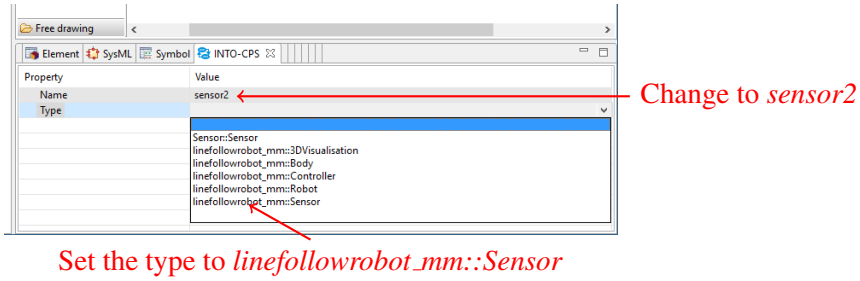
Step 11. Double click the *Connections Diagram*



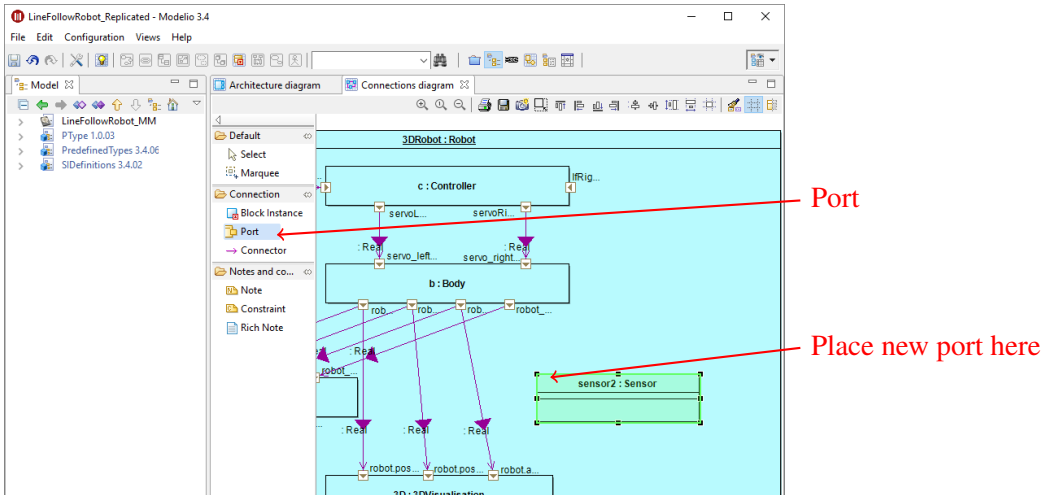
Step 12. To add a new Sensor, select *Block Instance* from the palette menu and add the new instance to the 3DRobot – simply click inside the 3DRobot, as indicated below.



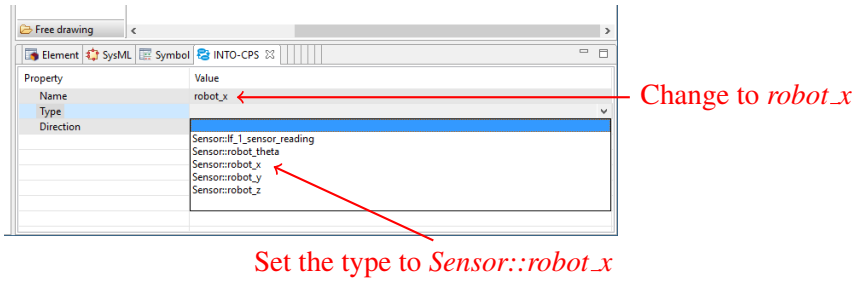
Step 13. In the *INTO-CPS* panel, change the name of the new instance to 'sensor2' and set the type to be 'linefollowrobot_mm::Sensor'.



Step 14. The next step is to add ports to the sensor instance. Select *Port* from the palette menu and add the new port to the sensor2.



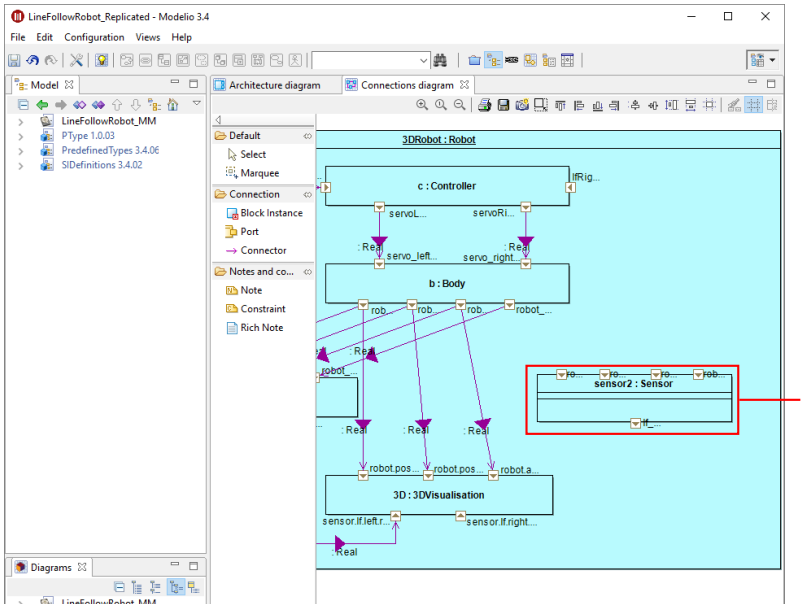
Step 15. Select the new port and in the *INTO-CPS* panel change the name to 'robot_x' and type to be 'Sensor::robot_x'.



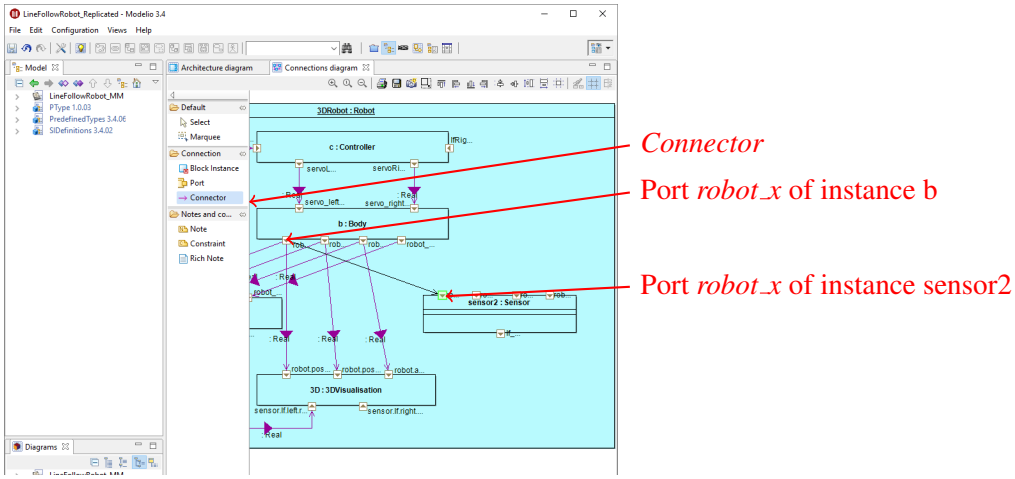
Step 16. Repeat steps 14 and 15 to add four more ports:

- Name: 'robot_y'; Type: 'Sensor::robot_y'.
- Name: 'robot_z'; Type: 'Sensor::robot_z'.
- Name: 'robot_theta'; Type: 'Sensor::robot_theta'.
- Name: 'lf_1_sensor_reading'; Type: 'Sensor::lf_1_sensor_reading'.

The connections diagram should look like that below:



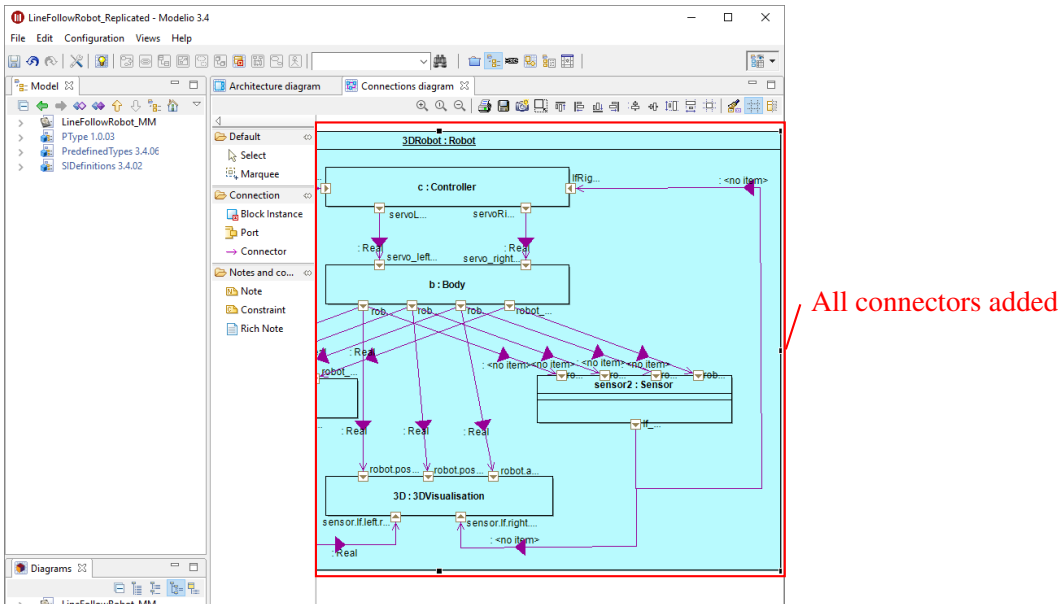
Step 17. The next step is to add connections between the different models of the robot. Select *Connector* from the palette menu and connect the *robot_x* port of the *body* component to the *robot_x* port of the new *sensor2* component



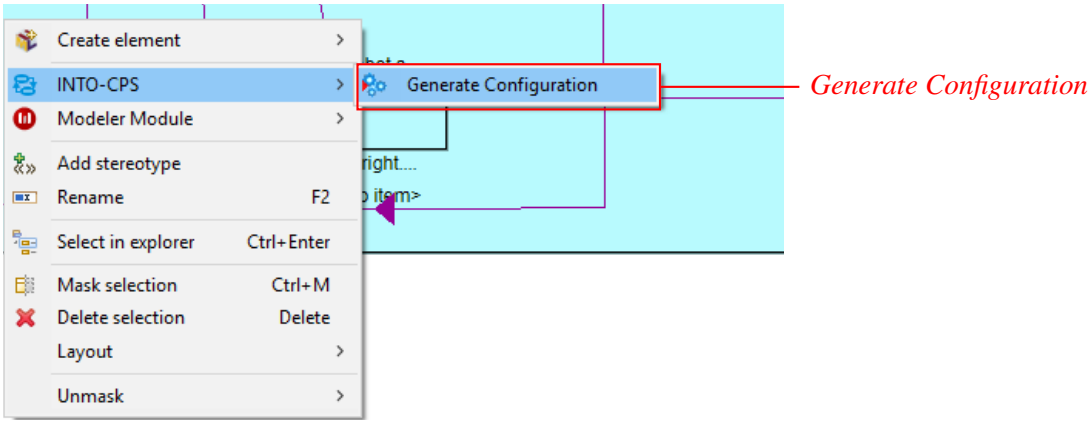
Step 18. Repeat step 17 to add five more connectors:

- 'body.robot_y' to 'sensor2.robot_y'.
- 'body.robot_z' to 'sensor2.robot_z'.
- 'body.robot_theta' to 'sensor2.robot_theta'.
- 'sensor2.lf_1_sensor_reading' to 'controller.lfRightVal'.
- 'sensor2.lf_1_sensor_reading' to '3D.animation.sensor.lf.right'.

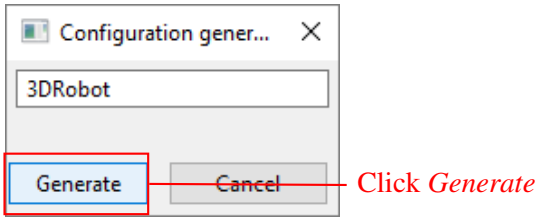
The connections diagram should look like that below:



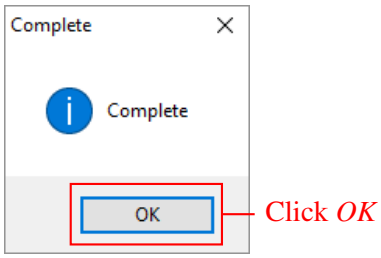
Step 19. To export this new configuration, right click on the 3DRobot instance and select *INTO-CPS > Generate Configuration*. *If nothing happens, closing and re-opening Modelio often helps.*



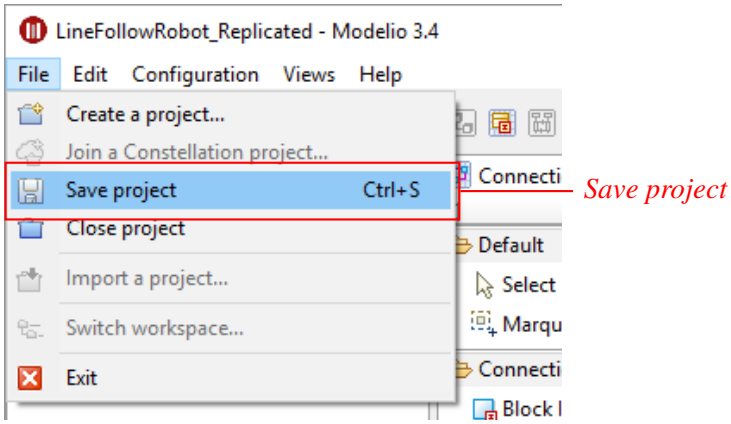
Step 20. Click *Generate*. If this seems to be unresponsive, then click *Cancel*, save the model, close and reopen *Modelio* and try again.



Step 21. Click *OK*.

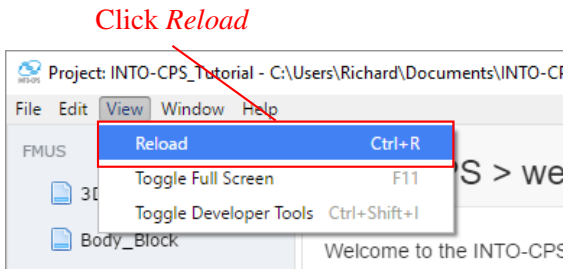


Step 22. Finally, save the SysML model.

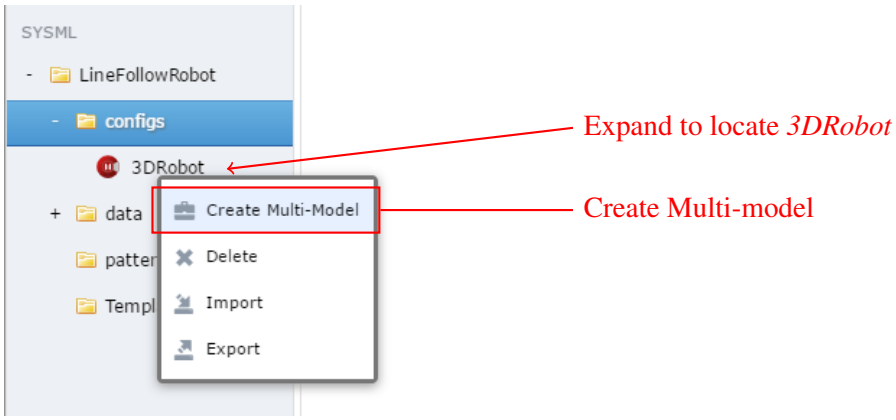


3 Configuring a Multi-model

Step 23. Return to the *INTO-CPS Application* and reload the view by selecting *View* > *Reload*.



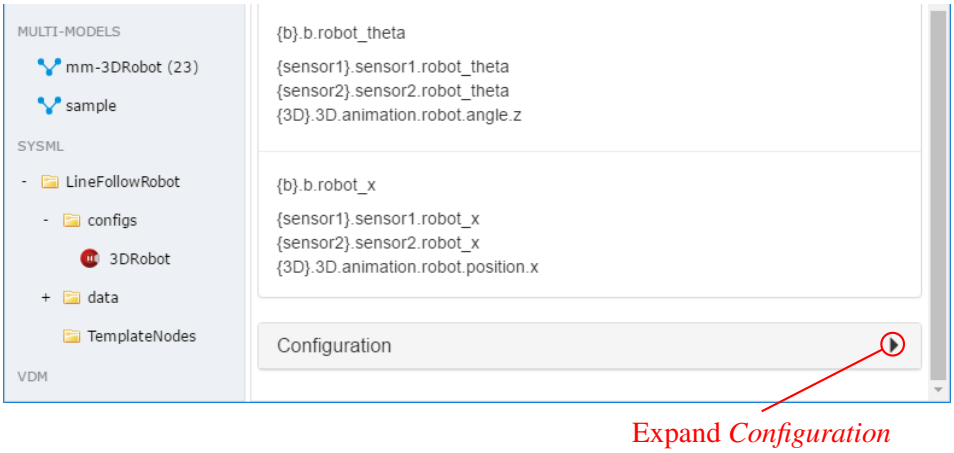
Step 24. In the SysML entry of the project browser, expand the *LineFollowRobot* and then *config* folders. There should be a *3DRobot* icon (as in the Figure below). Right click on *3DRobot*, select *Create Multi-Model*. You can just accept the default name in the prompt that appears.



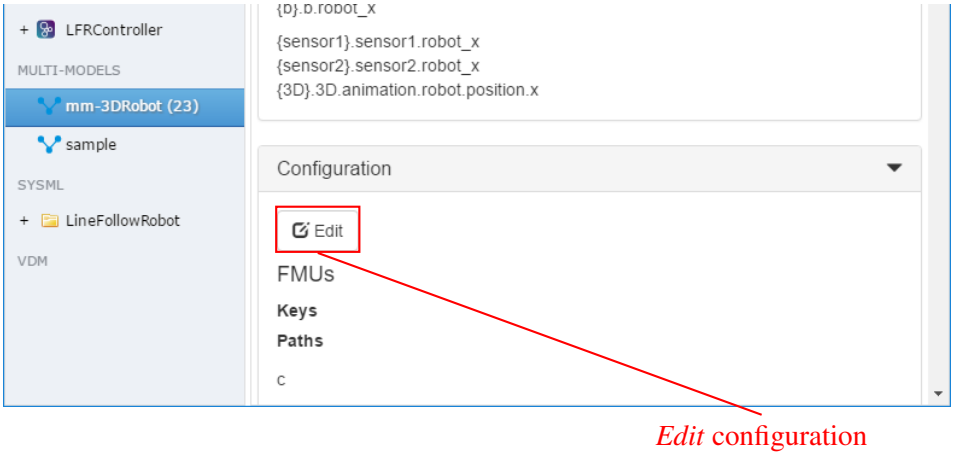
Step 25. A new multi-model configuration has been created and is shown in the multi-model entry of the project browser. Double-click on the new multi-model to open it.



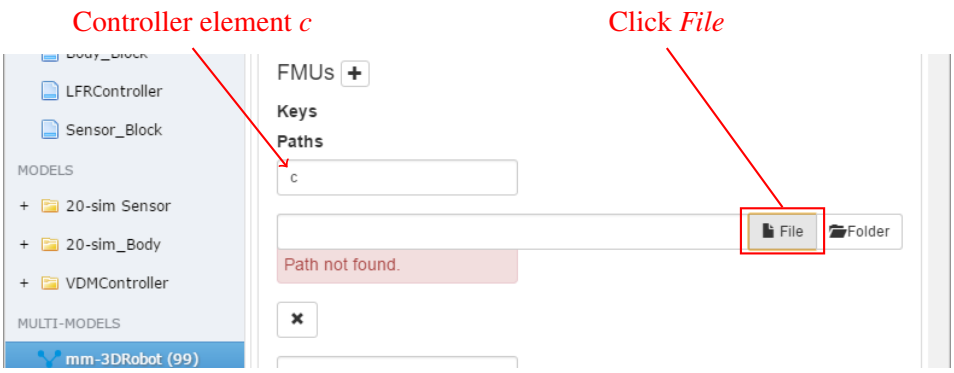
Step 26. We need to associate FMUs with this multi-model and set its parameters. Expand the *Configuration* section of the multi-model by clicking on the triangle.



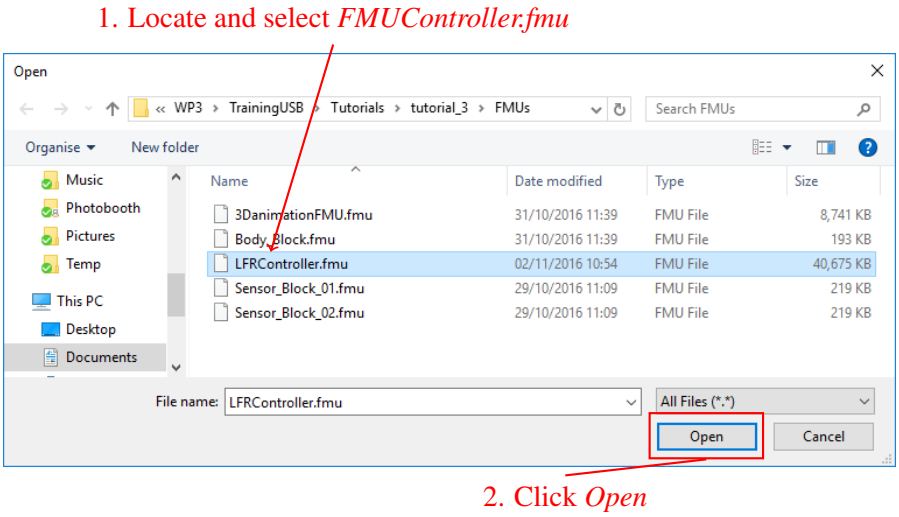
Step 27. Scroll down and click *Edit*.



Step 28. In the *FMUs* section, next to the *Controller* element *c*, click the *File* button.

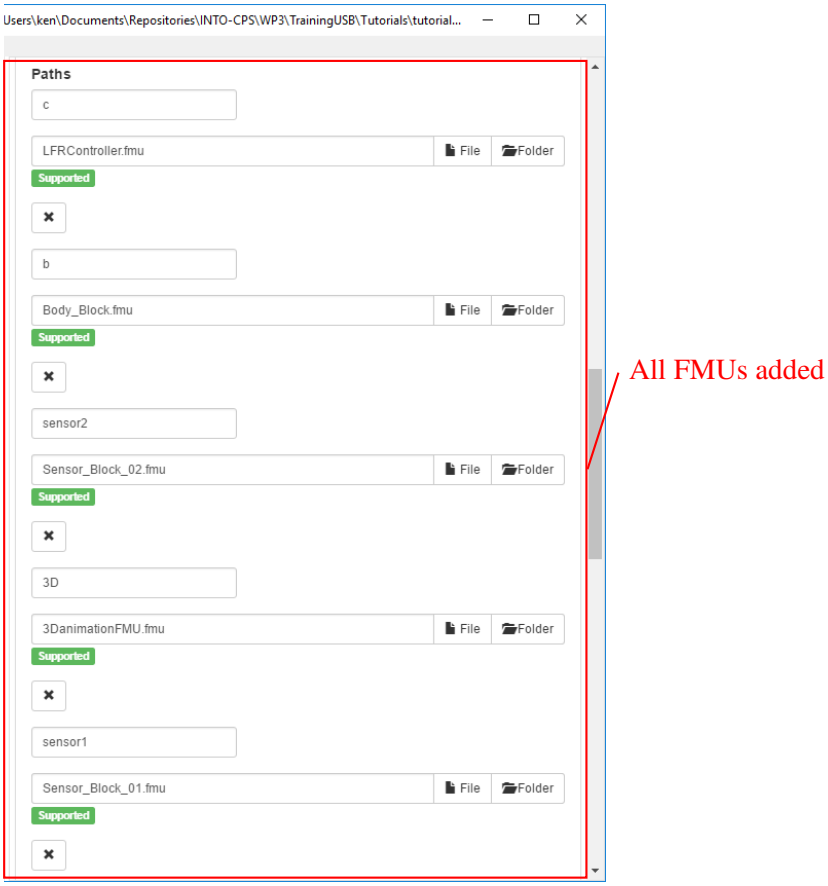


Step 29. A file browser window will open and show five FMUs (if the file browser does not show the FMUs, navigate to *tutorials_3/FMUs*). Select *FMUController.fmu* and click *Open*.



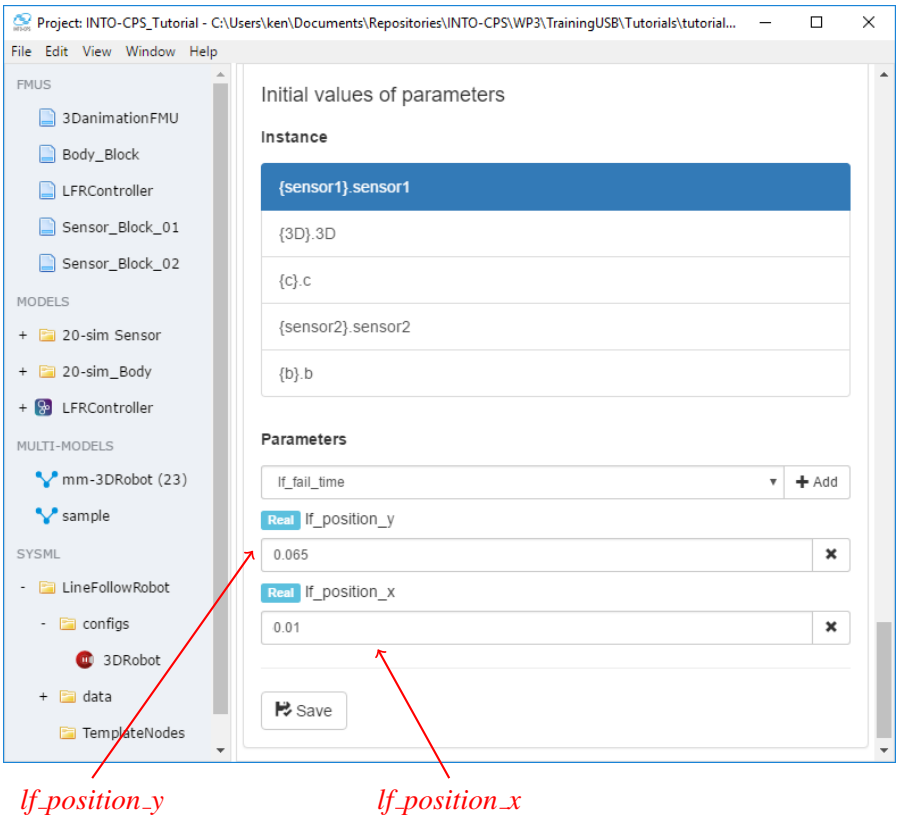
Step 30. The LFRController has been added. Repeat this for the remaining elements:

- *b* : *Body_Block.fmu*
- *3D* : *3DanimationFMU.fmu*
- *sensor1* : *Sensor_Block_01.fmu*
- *sensor2* : *Sensor_Block_02.fmu*



Step 31. Next the sensor positions must be defined. Scroll down to the *Initial values of parameters section*, and click {*sensor1*}.*sensor1*. In the *Parameters* section, enter the following values:

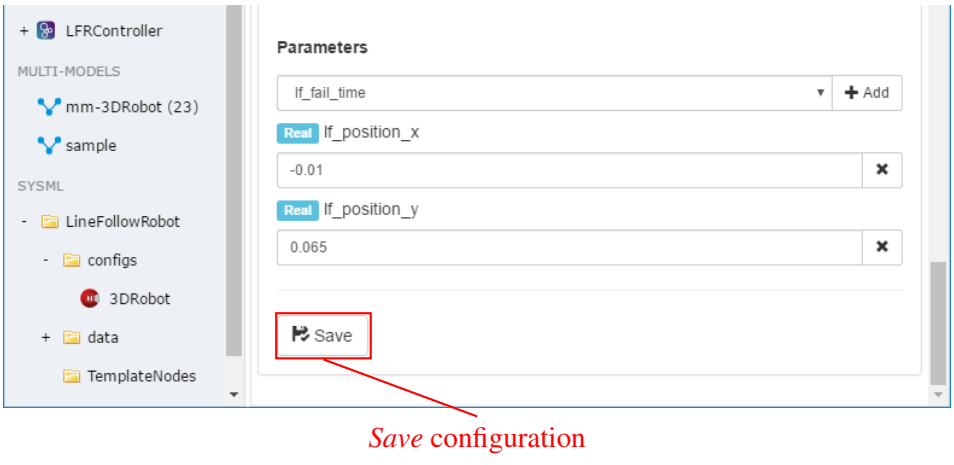
- *lf_position_y* = 0.065
- *lf_position_x* = 0.01



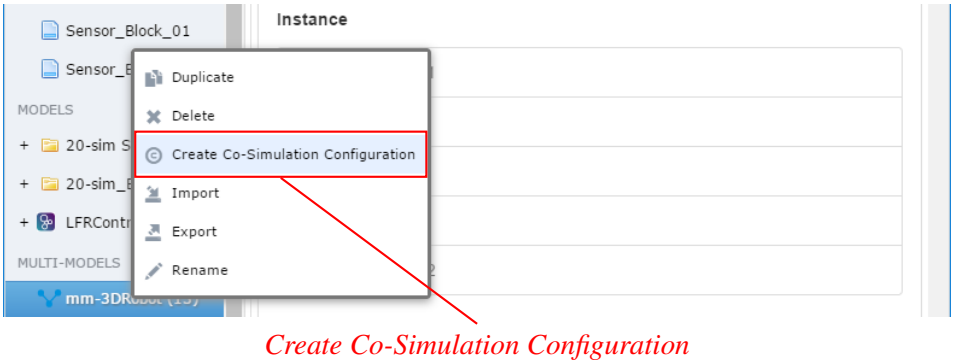
Step 32. Repeat the previous step for the second sensor – {*sensor2*}.*sensor2* with the following values:

- *lf_position_x* = -0.01
- *lf_position_y* = 0.065

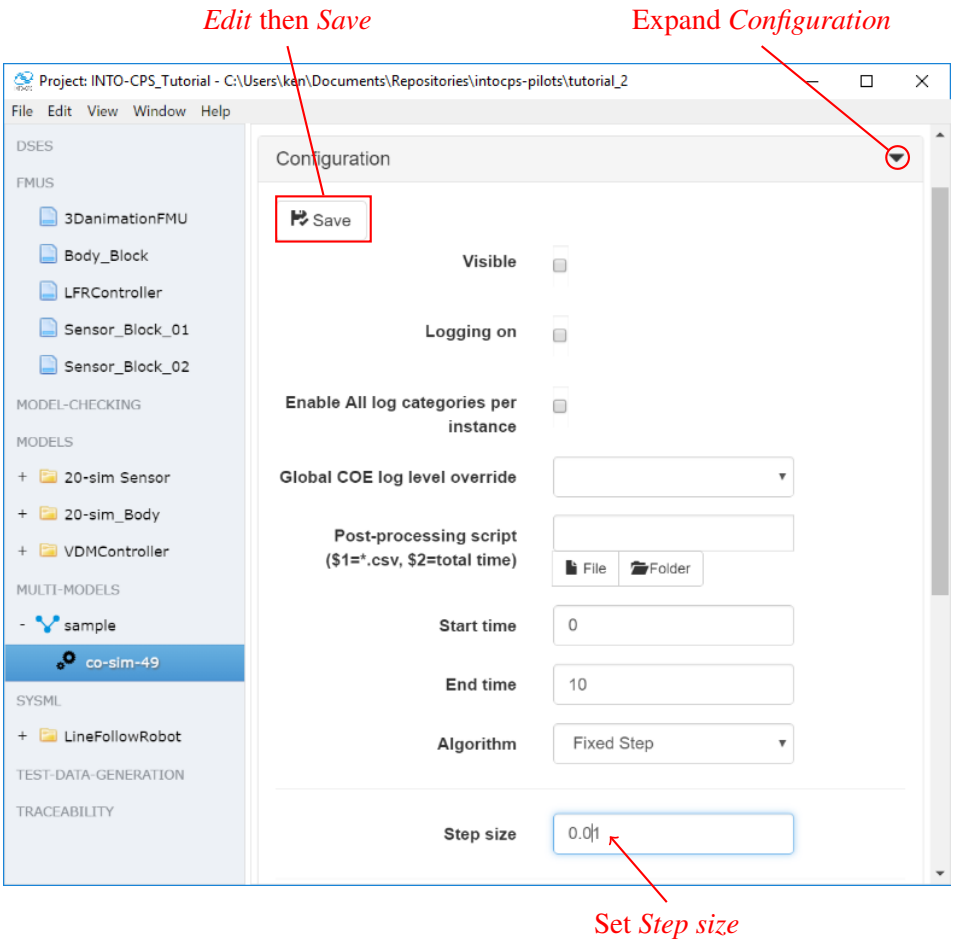
Step 33. *Save* the *Configuration*.



Step 34. The multi-model configuration is complete. Right-click on the multi-model configuration and select *Create Co-simulation Configuration*.



Step 35. Set the *Step size* to 0.01. Don't forget to press *Edit* then *Save*.

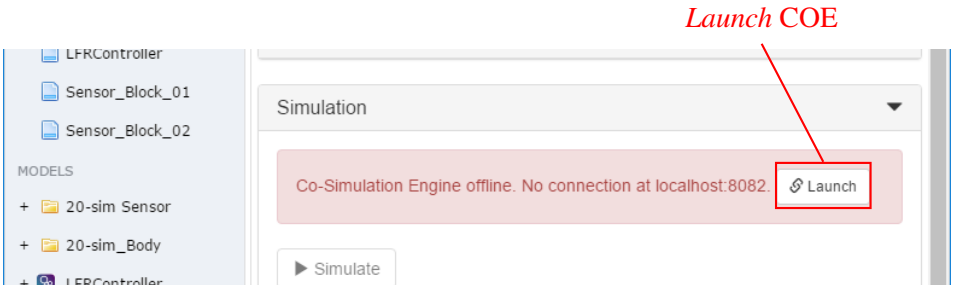


Step 36. Check *lf_1_sensor_reading* from {*sensor1*}.*sensor1* and {*sensor2*}.*sensor2* to see the sensor values appear in the *Livestream Configuration*.

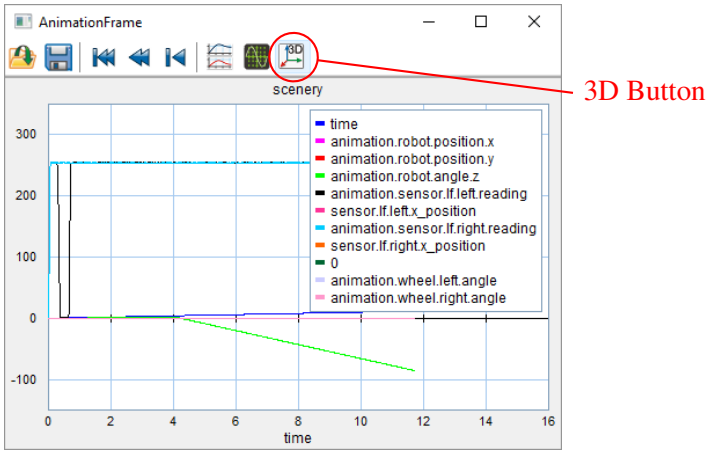


4 Running a Co-simulation

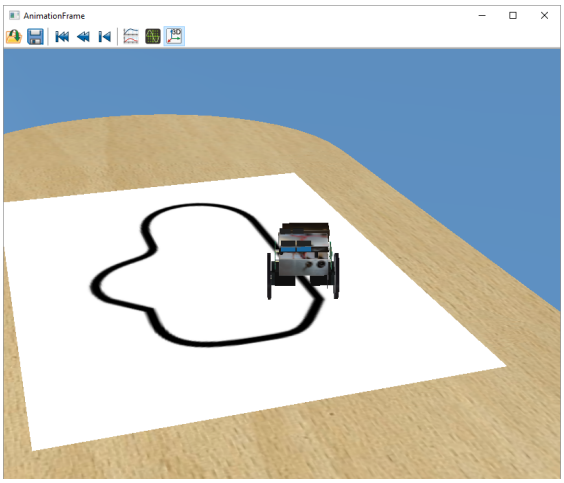
Step 37. Launch the COE if necessary (see *Tutorial 1 — First Co-simulation* for a reminder if needed).



Step 38. When the COE is running (see *Tutorial 1* for more details if you need a reminder), click the *Simulate* button. After a few seconds, a Java window called *Animation Frame* will appear like the one below. It shows a plot of variables from the co-simulation. You can click the *3D* button to see the 3D visualisation of the robot.



Step 39. A 3D model of the line following robot will appear. This view may be changed by clicking and dragging the mouse (note this is currently quite sensitive, so dont make quick movements). When the simulation has finished, this window will close. If everything went well, the robot should follow the line.



5 Additional Exercises

When this tutorial is complete, either move onto Tutorial 3, or try the following additional exercises:

1. Experiment with different sensor positions. Repeat steps 13 and 14 to change the position of the left and right sensors relative the the robot body. How do the different values effect the simulation?

2. Experiment with different robot speeds. Repeat steps 13 and 14 to change the different speed values for the *Controller*. How do the different values effect the simulation?