GO MODEL RAILWAY PROJECT
A SIMPLE DSL FOR THE MODEL RAILWAY DOMAIN
QUICK GUIDE[1]

Created by Benedek Horváth, Raimund-Andreas Konnerth and Tamás Nádudvari as a homework for the "Model Driven Software Development" course at the Budapest University of Technology and Economics (http://www.inf.mit.bme.hu/en)

[1] Document style based on PineappleDataflowNet usage guide.

TECHNOLOGIES USED (HAS TO BE INSTALLED IN ECLIPSE)

- Eclipse Modeling Tools 4.4.2
- EMF IncQuery 0.9.1
- Xtext Complete IDE 2.7.3
- Xtend IDE 2.7.3
- Sirius – 2.0
- e(fx)clipse IDE – 1.2

HOW TO USE IT

LOAD METAMODEL AND SAMPLE MODEL (PAGE 2)

OPEN THE SAMPLE STATEMACHINE MODEL (PAGE 5)

SIMULATE A SAMPLE RAILWAY MODEL IN DESMO-J (PAGE 6)

GENERATE SIMULATION CODE FROM THE STATEMACHINE MODEL (PAGE 7)

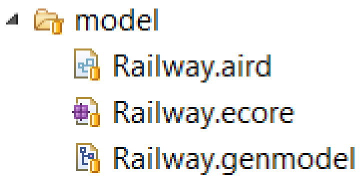
LOAD METAMODEL AND SAMPLE MODEL

Clone the project and load the following projects into the workspace (the grey projects are generated, but they can be loaded as well):

- hu.bme.mit.inf.gomrp.railway
- hu.bme.mit.inf.gomrp.railway.dsl.design
- hu.bme.mit.inf.gomrp.railway.dsl.text
- hu.bme.mit.inf.gomrp.railway.dsl.text.sdk
- hu.bme.mit.inf.gomrp.railway.dsl.text.ui
- hu.bme.mit.inf.gomrp.railway.edit
- hu.bme.mit.inf.gomrp.railway.editor
- hu.bme.mit.inf.gomrp.railway.incquery
- hu.bme.mit.inf.gomrp.railway.incquery.validation
- hu.bme.mit.inf.gomrp.simulation.auto
- hu.bme.mit.inf.gomrp.simulation.integration
- hu.bme.mit.inf.gomrp.simulation.manual
- hu.bme.mit.inf.gomrp.simulation.trace
- hu.bme.mit.inf.gomrp.simulation.trace.edit
- hu.bme.mit.inf.gomrp.simulation.trace.editor
- hu.bme.mit.inf.gomrp.simulation.trace.incquery
- hu.bme.mit.inf.gomrp.simulation.trace.incquery.validation
- hu.bme.mit.inf.gomrp.simulation.trace.visualizer
- hu.bme.mit.inf.gomrp.simulation.traceability
- hu.bme.mit.inf.gomrp.simulation.traceability.edit
- hu.bme.mit.inf.gomrp.simulation.traceability.editor
- hu.bme.mit.inf.gomrp.simulation.traceability.incquery
- hu.bme.mit.inf.gomrp.simulation.traceability.incquery.validation
- hu.bme.mit.inf.gomrp.statemachine
- hu.bme.mit.inf.gomrp.statemachine.dsl.text
- hu.bme.mit.inf.gomrp.statemachine.dsl.text.sdk
- hu.bme.mit.inf.gomrp.statemachine.dsl.text.ui

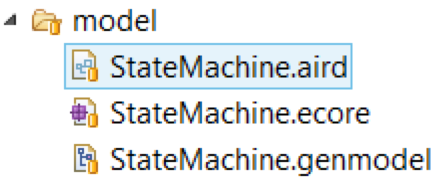
- hu.bme.mit.inf.gomrp.statemachine.edit
- hu.bme.mit.inf.gomrp.statemachine.editor
- hu.bme.mit.inf.gomrp.statemachine.incquery
- hu.bme.mit.inf.gomrp.statemachine.incquery.validation

The RailwayDomainModel metamodel can be found in the hu.bme.mit.inf.gomrp.railway project.



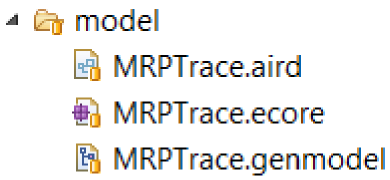
model
Railway.aird
Railway.ecore
Railway.genmodel

The StateMachine metamodel can be found in the hu.bme.mit.inf.gomrp.statemachine project.



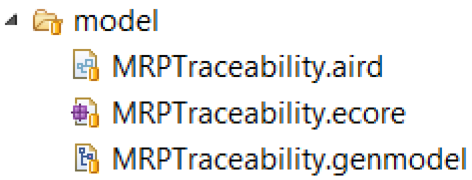
model
StateMachine.aird
StateMachine.ecore
StateMachine.genmodel

The Trace metamodel can be found in the hu.bme.mit.inf.gomrp.trace project.



model
MRPTrace.aird
MRPTrace.ecore
MRPTrace.genmodel

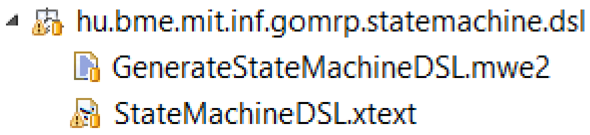
The Traceability metamodel can be found in the hu.bme.mit.inf.gomrp.traceability project.



model
MRPTraceability.aird
MRPTraceability.ecore
MRPTraceability.genmodel

You can generate the metamodels using the genmodel files. However the latest codes generated from the metamodels can be cloned from the repository.

The textual language for the StateMachine can be found in the hu.bme.mit.inf.gomrp.statemachine.dsl.text project.



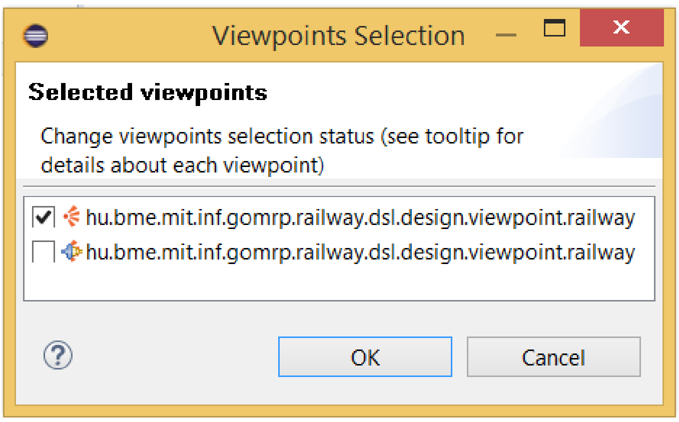
hu.bme.mit.inf.gomrp.statemachine.dsl
GenerateStateMachineDSL.mwe2
StateMachineDSL.xtext

Right click on the file GenerateStateMachineDSL.mwe2 and select Run as → MWE2 Workflow. However the latest codes generated from the DSL can be cloned from the repository.

Start a runtime Eclipse application and load the following projects:

- hu.bme.mit.inf.gomrp.railway.dsl.design
- hu.bme.mit.inf.gomrp.railway.instance.complex

The sample instance model can be found in the file ftsrg.rdm in the project hu.bme.mit.inf.gomrp.railway.instance.complex. In order to open it in Sirius, please open the Modeling Perspective (Window -> Open Perspective -> Other -> Modeling), and set the topmost viewpoint for the project (right click on the hu.bme.mit.inf.gomrp.railway.instance.complex project -> Viewpoints Selection).



Viewpoints Selection

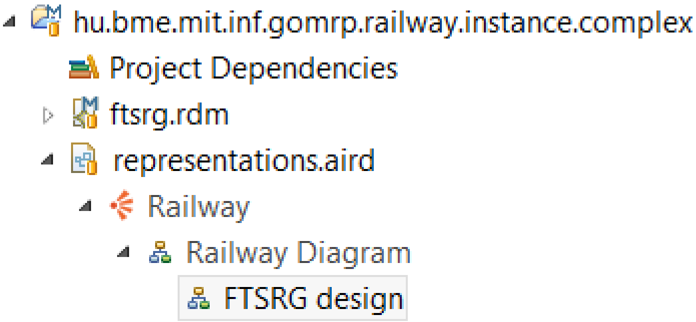
Selected viewpoints

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

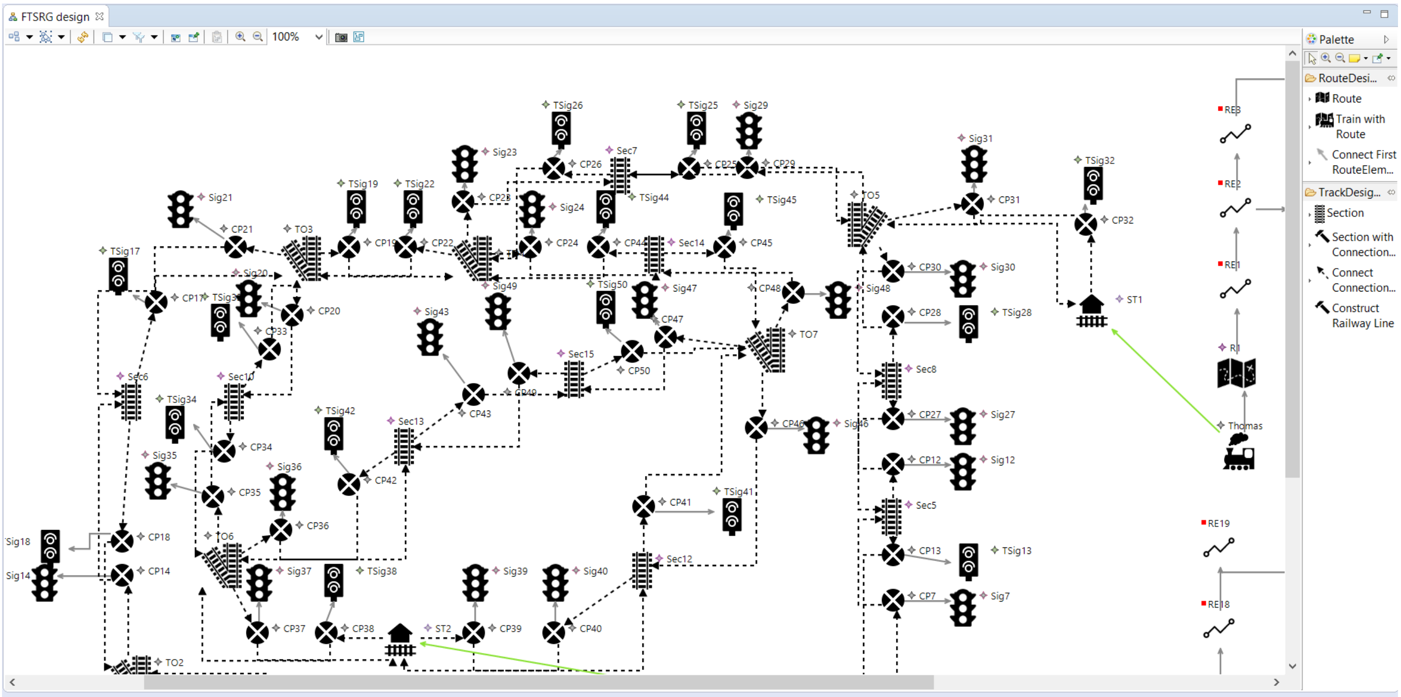
☑ hu.bme.mit.inf.gomrp.railway.dsl.design.viewpoint.railway

□ hu.bme.mit.inf.gomrp.railway.dsl.design.viewpoint.railway

After that, double-click on the representations.aird, and open each section after one another. Finally, double-click on the FTSRG design, and it opens in a new Sirius Editor.



- hu.bme.mit.inf.gomrp.railway.instance.complex
  - Project Dependencies
  - ftsrg.rdm
  - representations.aird
    - Railway
      - Railway Diagram
        - FTSRG design

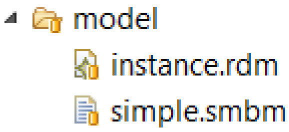


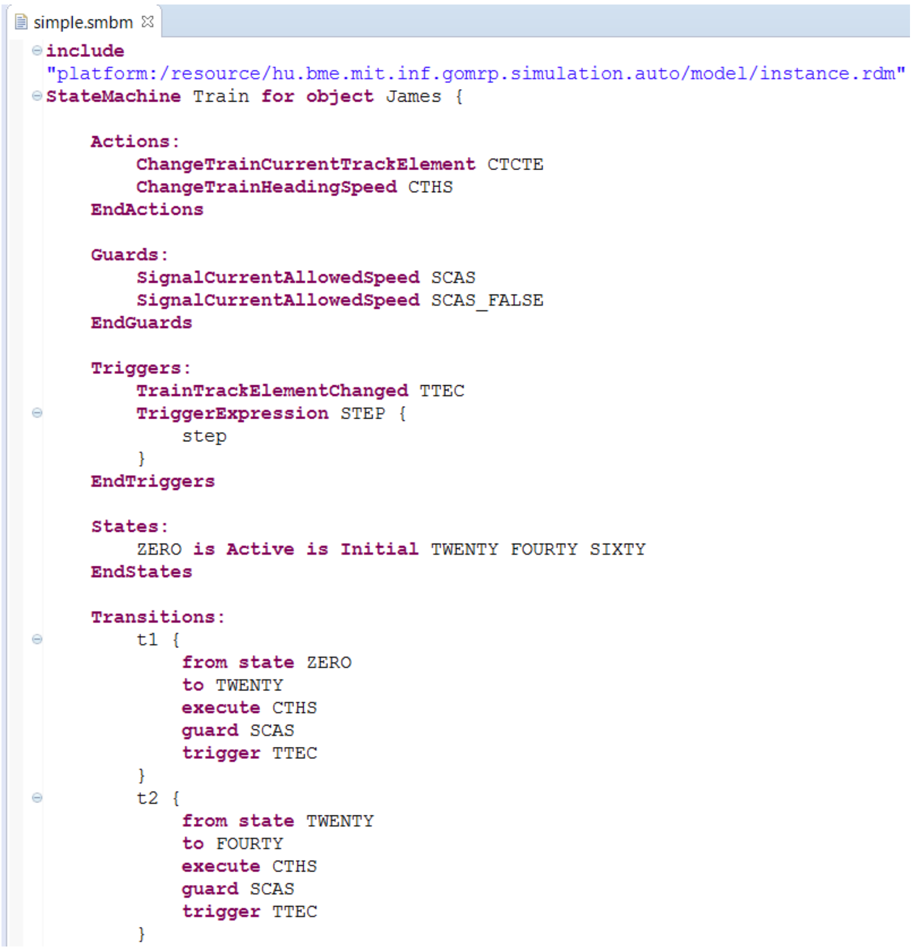
OPEN THE SAMPLE STATEMACHINE MODEL

Load the following project in the runtime Eclipse:

- hu.bme.mit.inf.gomrp.simulation.auto

Double-click on the model/simple.smbm file. The Xtext editor for the StateMachine opens.





```
include
"platform:/resource/hu.bme.mit.inf.gomrp.simulation.auto/model/instance.rdm"
StateMachine Train for object James {

    Actions:
        ChangeTrainCurrentTrackElement CTCTE
        ChangeTrainHeadingSpeed CTHS
    EndActions

    Guards:
        SignalCurrentAllowedSpeed SCAS
        SignalCurrentAllowedSpeed SCAS_FALSE
    EndGuards

    Triggers:
        TrainTrackElementChanged TTEC
        TriggerExpression STEP {
            step
        }
    EndTriggers

    States:
        ZERO is Active is Initial TWENTY FOURTY SIXTY
    EndStates

    Transitions:
        t1 {
            from state ZERO
            to TWENTY
            execute CTHS
            guard SCAS
            trigger TTEC
        }
        t2 {
            from state TWENTY
            to FOURTY
            execute CTHS
            guard SCAS
            trigger TTEC
        }
```

The include line is required for importing the RailwayDomainModel, that contains the model railway track and the trains.

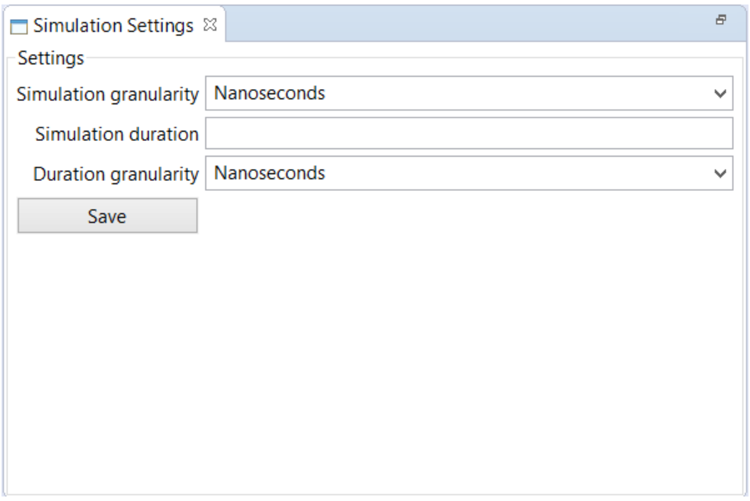
Simulate a sample railway model in DESMO-J

We used a discrete event simulation framework (DESMO-J), for simulating the railway model.

Load the following project in the runtime Eclipse:

- hu.bme.mit.inf.gomrp.simulation.manual

A sample model can be found in the hu.bme.mit.inf.gomrp.simulation.manual project, model/My.rdm.



Simulation Settings

Settings

Simulation granularity: Nanoseconds

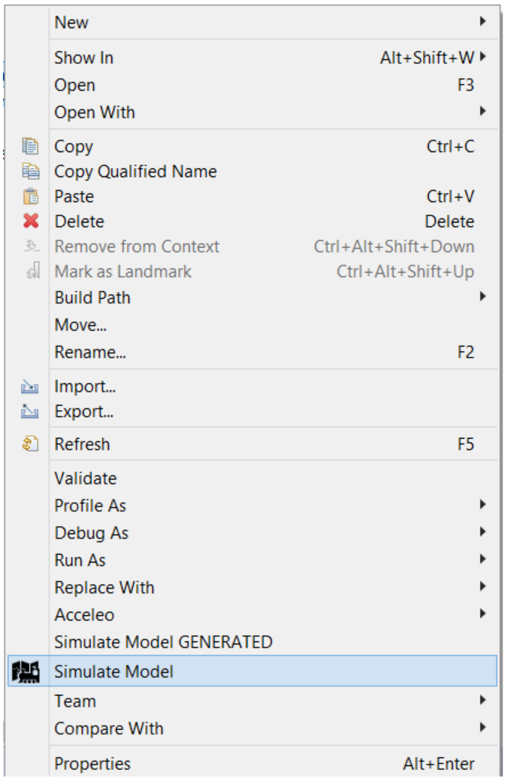
Simulation duration:

Duration granularity: Nanoseconds

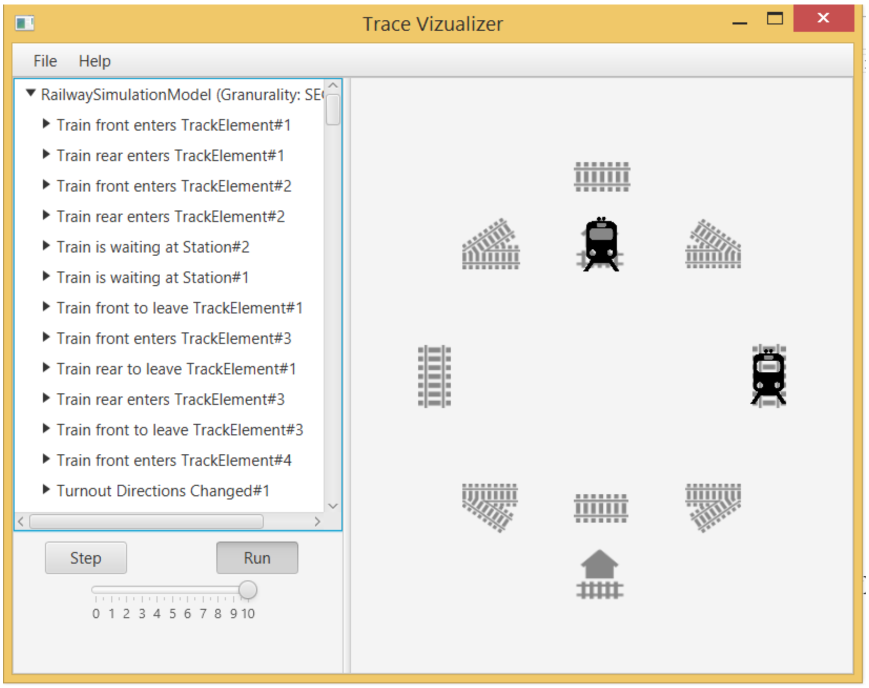
First of all set the settings for the Simulation: Window -> Show View -> Other -> MRP Simulation -> Simulation Settings. Set the Simulation granularity, the Simulation duration and the duration's granularity. The difference between the Simulation granularity and the Duration Granularity is the simulation granularity means the unit for the simulation, while the duration granularity means the unit of the simulation's duration.

Recommended values: Simulation granularity Seconds, Simulation duration 8, Simulation granularity Minutes. Click the Save button, to save the settings.

After that, open the Project Explorer, and right-click on the hu.bme.mit.inf.gomrp.simulation.manual model/My.rdm and select Simulate Model.



After the simulation finished, you can view the Trace in the TraceVisualizer (click Yes in the pop-up window).



```
▼ RailwaySimulationModel (Granurality: SE
  ▸ Train front enters TrackElement#1
  ▸ Train rear enters TrackElement#1
  ▸ Train front enters TrackElement#2
  ▸ Train rear enters TrackElement#2
  ▸ Train is waiting at Station#2
  ▸ Train is waiting at Station#1
  ▸ Train front to leave TrackElement#1
  ▸ Train front enters TrackElement#3
  ▸ Train rear to leave TrackElement#1
  ▸ Train rear enters TrackElement#3
  ▸ Train front to leave TrackElement#3
  ▸ Train front enters TrackElement#4
  ▸ Turnout Directions Changed#1
```

Click Open in the menu, and find the hu.bme.mit.inf.gomrp.simulation.manual project in the filesystem, and open the My_trace.mrptrace file. Adjust the simulation pace (0-10) and click Run. The trace will be replayed in the Trace Visualizer.

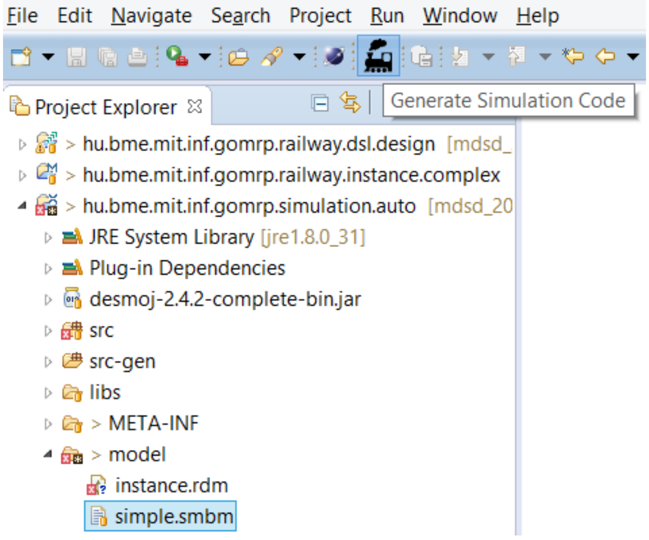
Besides, a new model file is created that stores the model's state after the simulation (My_simulated.rdm).

GENERATE SIMULATION CODE FROM THE STATEMACHINE MODEL

Load the following project in the runtime Eclipse (just like at the Open the sample StateMachine model section):

- hu.bme.mit.inf.gomrp.simulation.auto

Click on the model/simple.smbm file, and click on the locomotive icon in the top toolbar (under the menu bar) in the Eclipse.



```
▷ > hu.bme.mit.inf.gomrp.railway.dsl.design [mdsd_
▷ > hu.bme.mit.inf.gomrp.railway.instance.complex
◢ > hu.bme.mit.inf.gomrp.simulation.auto [mdsd_20
    ▷ JRE System Library [jre1.8.0_31]
    ▷ Plug-in Dependencies
    ▷ desmoj-2.4.2-complete-bin.jar
    ▷ src
    ▷ src-gen
    ▷ libs
    ▷ > META-INF
    ◢ > model
          instance.rdm
          simple.smbm
```

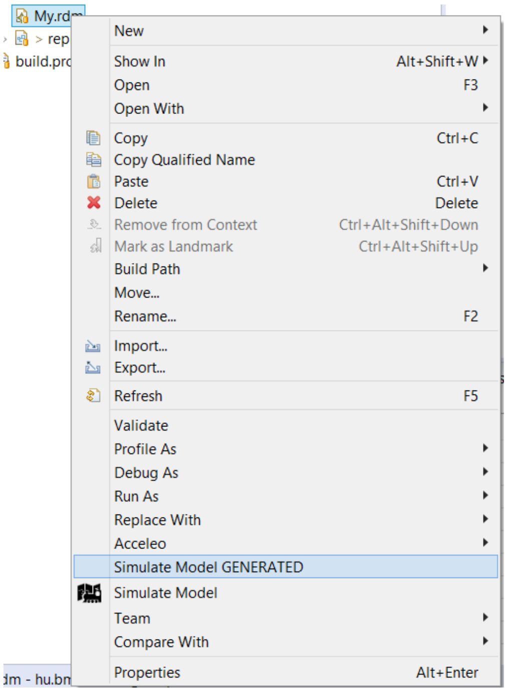
The code has been generated in the src-gen/ folder. You can open them, there are plenty TODO stubs, which have to be implemented in order to get a working simulation.

Move the files (and packages) from the src-gen/ folder into the src/ folder. Please make sure, that the packages are moved as well!

There are prepared (originally generated from the simple.smbm, after added implementation to the TODO methods) java files in the /ready folder. Copy them from the /ready folder, and replace each Java file in the src/ folder, with respect to the last segment of the package name.

Start a new Runtime Eclipse (from the Runtime Eclipse), and load the hu.bme.mit.inf.gomrp.simulation.manual project in the Runtime's Runtime Eclipse.

Set the Simulation Settings (see SIMULATE A SAMPLE RAILWAY MODEL IN DESMO-J section), and right-click on the model/My.rdm and select Simulate Model GENERATED.



Now the simulation's trace cannot be seen in the TraceVisualizer, but the trace file (model/My_trace.mrptrace) and the simulation's result model file (model/My_simulated.rdm) can be opened in Tree Editors.