WORKSHOP 4.1.1

Creating a system architecture based on templates and interfaces

Contents

Introduction....1
Creating the packages....1
Creating the interfaces....1
Creating the template....2
Creating the subsystems....2
Creating a configuration....6
Extra work....7

Introduction

In this workshop, a configurable vehicle architecture will be built.

Creating the packages

1. Create a new package **W1_Reconfigurable** inside the course package you created in one of the previous lectures.
2. Inside this package, create **Interfaces** and **Templates** sub-packages.

Creating the interfaces

1. In *W1_Reconfigurable.Interfaces*, create a partial model called **Engine** (create a model first, then open the code layer, and add the keyword partial).
    a. Drag in a ***Modelica.Blocks.Interfaces.RealInput***, position it to the left, and call it **throttle.**
    b. Drag in a ***Modelica.Mechanics.Rotational.Interfaces.Flange_a***, position it to the right and call it **transmissionFlange**.
2. Repeat this for *Transmission*, *Driveline*, and *Chassis* (make sure all are partial):
    a. Create a model **Transmission** and add two flanges, called **engineFlange1** and **drivelineFlange**, to the left and right, respectively.
    b. Create a **Driveline** model with **flange_ a,** and **flange_b**.
    c. Create a **Chassis** model with **drivelineFlange** and an output connector named **speed**. Add: `parameter Modelica.SIunits.Velocity v_start "Chassis initial speed";`

Now, you should have a library looking like the Figure 1.

Copyright © 2022 Modelon

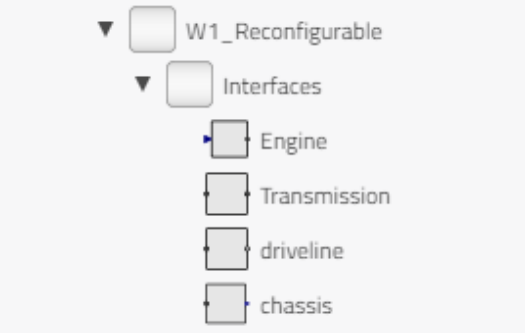

Figure 1 Package Structure

Creating the template

1. In *W1_Reconfigurable.Templates*, create a partial model called **StandardCar**. (Create a model first, then open the code layer, and add the keyword partial):

```
partial model StandardCar
```

2. Drag an *Interfaces.Engine* component into the model, and repeat for the other subsystems and make sure to call them **engine**, **transmission**, **driveline**, and **chassis**. Make sure that the naming of your subsystems is right, these names are not so easy to change later, and align them in that order (see Figure 2).
3. Connect them. Now you should have a model looking like the Figure 2.

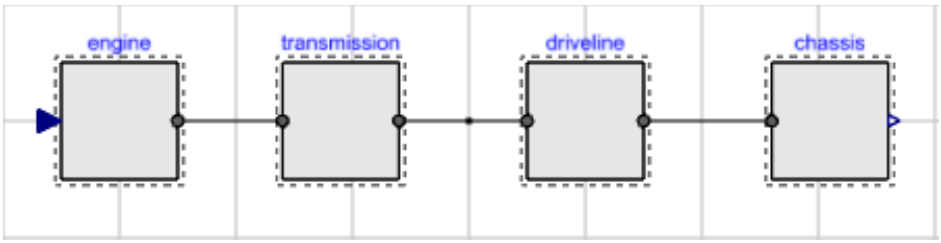

Figure 2 StandardCar Template

Creating the subsystems

4. In *W1_Reconfigurable*, create a sub-package called **SubSystems**
5. Right-click on *Interfaces.Engine* and select **Extend**.

Copyright © 2022 Modelon

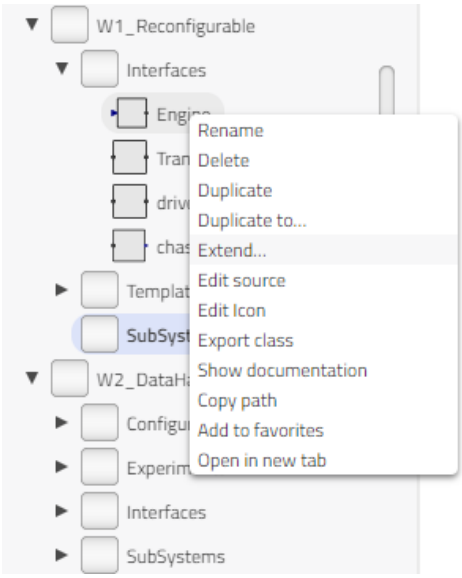

Figure 3 Extending an interface

6. Create a *BasicEngine* model in the *SubSystems* package, see Figure 4.

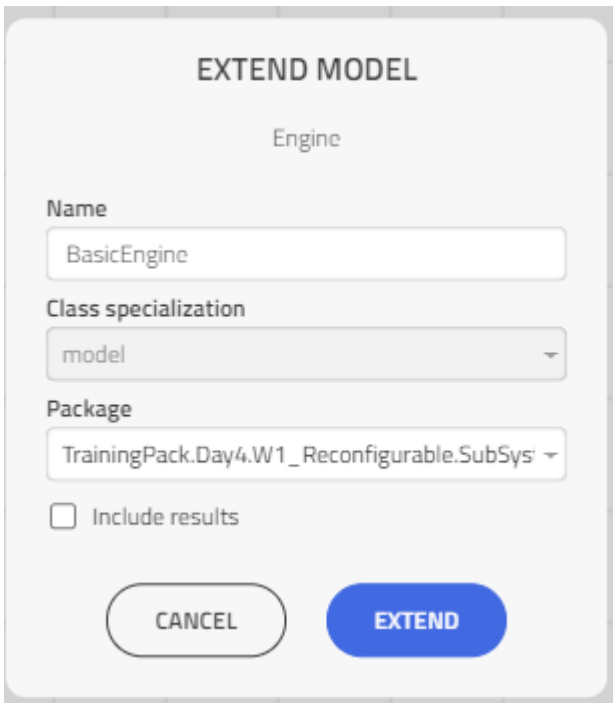

Figure 4 Creating new basic engine

7. Build the model using ***Modelica.Blocks.Nonlinear.Limiter***, ***Modelica.Blocks.Math.Gain***, and ***Modelica.Mechanics.Rotational.Sources.Torque***. Set *uMax*=1, *uMin*=0 in the limiter block and propagate the gain as **max_torque** with a default of 300 Nm. The model should look like Figure 5. Use the code block reference shown below.

Copyright © 2022 Modelon

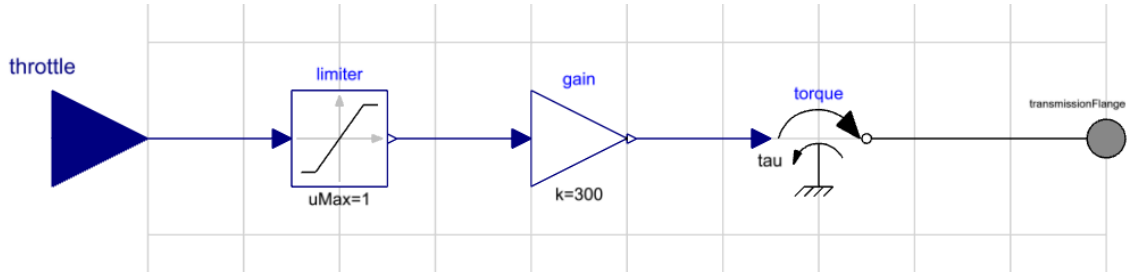

Figure 5 BasicEngine model

```
model BasicEngine "Engine with ideal throttle characteristics"
    extends .TrainingPack.Day4.W1_Reconfigurable.Interfaces.Engine;
    .Modelica.Blocks.Math.Gain gain(k = max_torque) annotation(•••);
    .Modelica.Mechanics.Rotational.Sources.Torque torque annotation(•••);
    parameter .Modelica.Units.SI.Torque max_torque = 300 "Torque for full throttle";
    .Modelica.Blocks.Nonlinear.Limiter limiter(uMax = 1,uMin = 0) annotation(•••);
```

8. Create models for the other sub-systems:
    a. **FixedTransmission** with a ***Modelica.Mechanics.Rotational.Components.IdealGear*** and propagated parameter *ratio*=1.0 as shown in Figure 6.

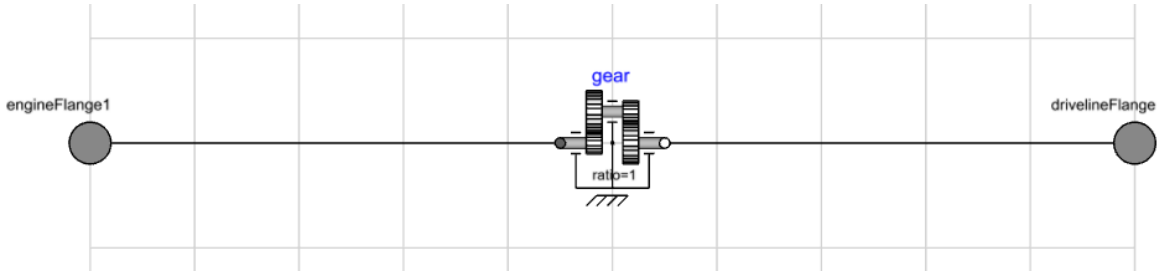

Figure 6 FixedTransmission model

    b. **RigidDriveline** with an *Inertia* and *inertia* parameter *J*=0.1 propagated as shown in Figure 7.

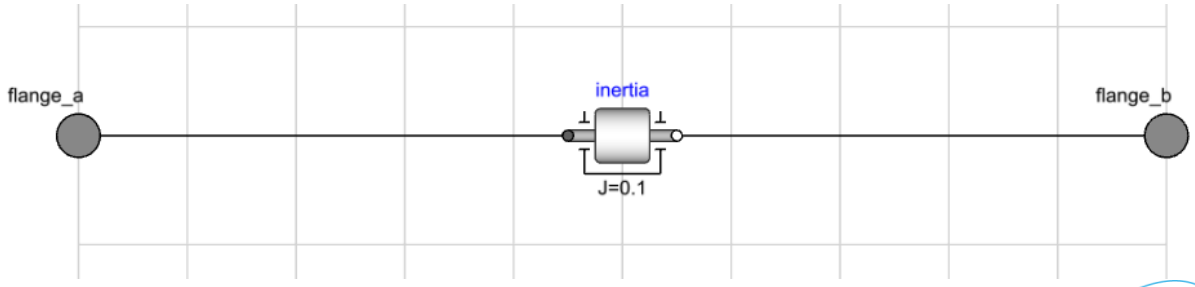

Figure 7 RigidDriveline model

    c. **IdealChassis** using components from *Rotational* and *Translational* libraries as shown in Figure 8.

Copyright © 2022 Modelon

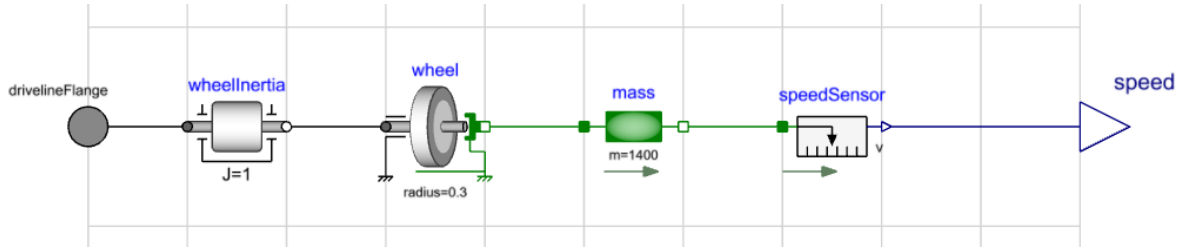

Figure 8 IdealChassis model

d. Propagate parameters and set default values for *J_wheel*=1, *R_wheel*=0.3, *m_chassis*=1400 and *v_start* (note especially *v_start* default value was set in the template, see the Figure 9).

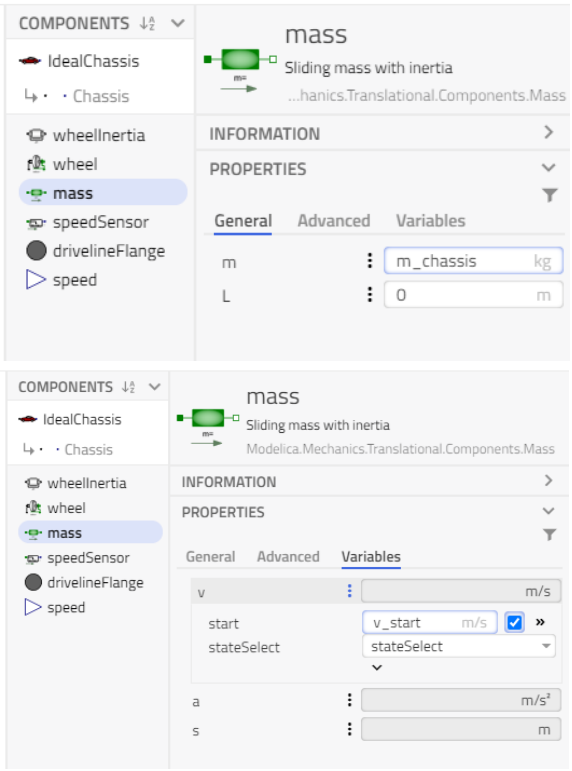

Figure 9 Propagating Chassis parameter

e. Now, the code for chassis should look like Figure 10.

```
model IdealChassis "Ideal chassis with no losses"
    extends Interfaces.Chassis;
    .Modelica.Mechanics.Rotational.Components.Inertia wheelInertia(J = J_wheel) annotation(...);
    .Modelica.Mechanics.Rotational.Components.IdealRollingWheel wheel(radius = R_wheel) annotation(...);
    .Modelica.Mechanics.Translational.Components.Mass mass(m = m_chassis,v(fixed = true,start = v_start)) annotation(...);
    .Modelica.Mechanics.Translational.Sensors.SpeedSensor speedSensor annotation(...);
    parameter .Modelica.Units.SI.Inertia J_wheel = 1 "Wheel inertia";
    parameter .Modelica.Units.SI.Distance R_wheel = 0.3 "Wheel radius";
    parameter .Modelica.Units.SI.Mass m_chassis = 1400 "mass of the chassis";
equation
    connect(wheel.flangeR,wheelInertia.flange_b) annotation(...);
    connect(wheelInertia.flange_a,drivelineFlange) annotation(...);
    connect(wheel.flangeT,mass.flange_a) annotation(...);
    connect(mass.flange_b,speedSensor.flange) annotation(...);
    connect(speedSensor.v,speed) annotation(...);
    annotation(...);
end IdealChassis;
```

Figure 10 IdealChassis code layer

f. Create nice icons for your subsystems.

Copyright © 2022 Modelon

Creating a configuration

1. In *W1_Reconfigurable*, create a sub-package called **Configurations**
2. Extend the *Templates.StandardCar* and create **AccelerationTest** in the package **Configurations.**
3. Open the Properties section, and use the dropdown menu on the engine-model and select BasicEngine. It should be like Figure 11.

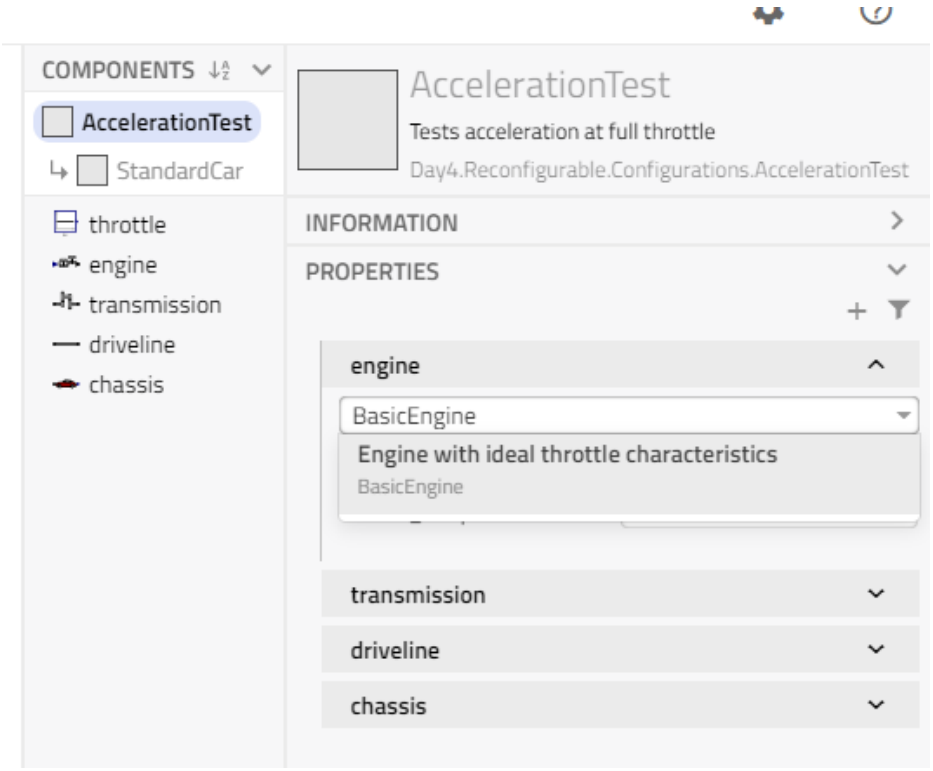

Figure 11 Changing class

4. Repeat for all the other sub-models as well, add a constant input for full *throttle* (=1.0). You should now have something like Figure 12.

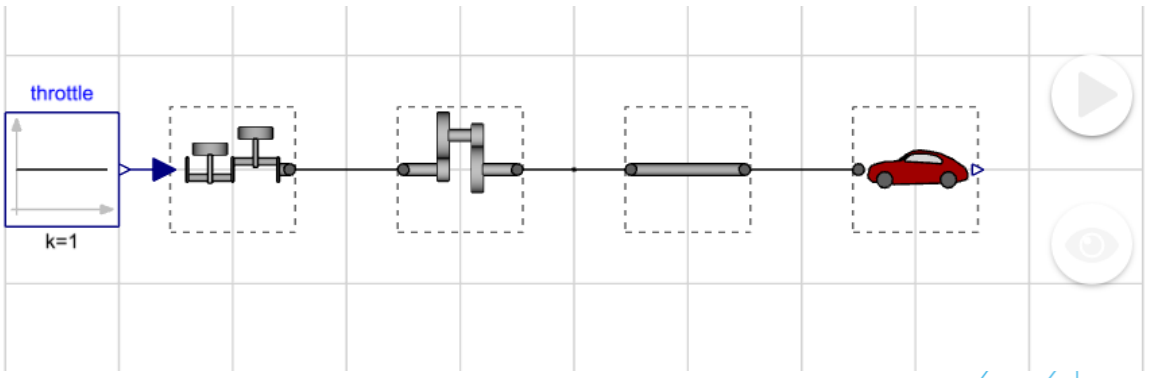

Figure 12 Changing class for all components and adding input block

5. Complete the model with a constant input for full *throttle* (=1.0), set *v_start* =25 m/s and simulate for 10 s.

Copyright © 2022 Modelon

Extra work

1. Add losses to the chassis model. Use for example ***Modelica.Mechanics.Rotational.Sources.LinearSpeedDependentTorque*** and ***Modelica.Mechanics.Rotational.Sources.QuadraticSpeedDependentTorque*** to model rolling resistance and aerodynamic drag.
2. Currently, only *v_start* is a parameter of the subsystem interfaces. Move all parameters that make sense to the interface levels.
3. Add a brake system with pedal force as input.
4. Create a driver model that follows the input speed by accelerating and braking.

Copyright © 2022 Modelon