Tutorial: Composite Structure of an Elevator Controller

Shuai Li, Jérémie Tatibouët, Patrick Tessier and François Terrier,
Asma Smaoui
{first-name}.{last-name}@cea.fr

In this tutorial, we will model the composite structure of the elevator controller system. You can use the draft Papyrus model in the folder "**043-ElevatorControllerModelComposite_Student**". Copy the model in your Papyrus project folder and refresh the project in Eclipse (F5) to open the model.

Parts and Ports

Objectives:

- Model the composite structure of the elevator controller system
- Add parts and ports
- Connect ports

Exercises:

In the pre-filled model, there is a new "SystemInternalStructure" composite structure diagram. This diagram contains the *requestPool* and *requestProcessor* parts of the **ElevatorControllerSystem**. Ports and connectors are already modeled for these two parts. As a reminder, **RequestProcessor** interacts with **RequestPool**, **MotorControl**, and **EntranceControl**. Requests from the environment are forwarded to the **RequestProcessor**. **MotorControl** and **FloorControl** control respectively the motor and floor hardware. Do the following exercises. You may do them on paper first or directly in the Papyrus model.

1. Model other parts of the system.
2. Similarly to how *requestPool* and *requestProcessor* are connected, add ports to the parts you modeled previously and connect the ports together.
3. The **ElevatorControllerSystem** owns some ports to interact with the hardware environment (e.g. *callPanel* port). Add ports to some parts and connect these ports to the ports of the **ElevatorControllerSystem**. Do not type the ports yet.
4. Transpose your model into Papyrus if it isn't done already.

Advices:

- In order to add a port to a part, you must first create the port for the class that types the part.

- You can create a port for a class, through the Model Explorer view: right click on the class > next child > port.
- To display the port on the part, drag-and-drop the port from the Model Explorer view to the part.
- Enlarge the size of the part if you find it hard to drag-and-drop the port to the part.

Interfaces and classes typing ports

Objectives:

- Model interfaces and classes that will type the ports
- Type the ports correctly

Exercises:

In the pre-filled model, there is a new "InterfaceImpl" class diagram. (This is not the "InterfaceDependencyRealization" diagram!) This diagram contains two classes that realize/use the **IRequestClient**, **IRequestSupplier**. Do the following exercises. You may do them on paper first or directly in the Papyrus model.

1. Model the classes that realize/use the **IFloorDoorRequest**, **IAcknowledgeable**, and **IMotorRequest** interfaces.
2. Model the classes that will require any of the interfaces **ICallPanel**, **IFloorDoor**, **IMotor**, and **IFloorPanel**, if there are any.
3. Type the ports modeled previously
4. Transpose your model into Papyrus if it isn't done already.

Advices:

- Remember that you can use the pattern of the two classes that realize/use **IRequestClient** and **IRequestSupplier**
- As a reminder, the interfaces are all in the **"ElevatorController::Interfaces"** package. You should place your interfaces and related classes in this package.
- Be sure to connect ports by considering how they are connected! (Rule of provided/required interfaces)