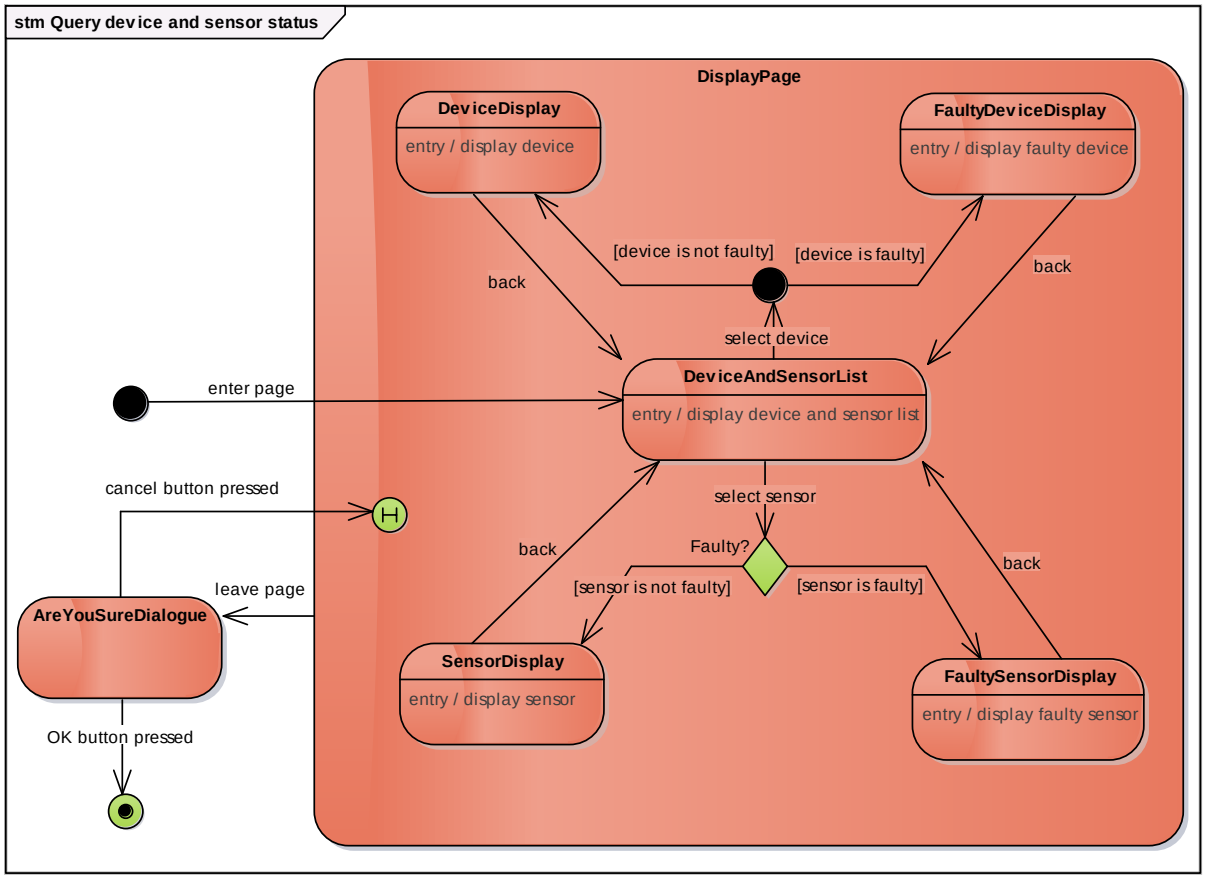
Learning UML

O'Reilly

Exercises – Nested State Machines

If you have followed along, you should have a statemachine diagram called *Query device and sensor status*. Your diagram should look something like this.



Make some space at the bottom of the diagram and enlarge the state *FaultySensorDisplay*.

Add an initial pseudostate and two states called *Active* and *Removed*.

In the *Class View* find the *Sensor* class and add a new attribute *removed* of type *boolean*. Add operations *setRemoved(removed: boolean)* and *getRemoved() : boolean*.

Add a transition from the initial state to Active with the guard *sensor.getRemoved()==false*, and a transition from the initial state to *Removed* with the guard *sensor.getRemoved()==true*. (Note that you can type behaviours into the textbox if you don't select Effect is a Behaviour, and this allows you to set values as arguments.)

Add transitions from *Active* to *Removed* and *Removed* to *Active* with the triggers *remove* and *activate* respectively. Set the effect behaviour of these to be *setRemoved(true)* for the *remove* trigger and *setRemoved(false)* for the *activate* trigger, where both of these are operations of the *Sensor* class.

Create a new nested state machine diagram for the state *FaultyDeviceDisplay* in a composite structure diagram.

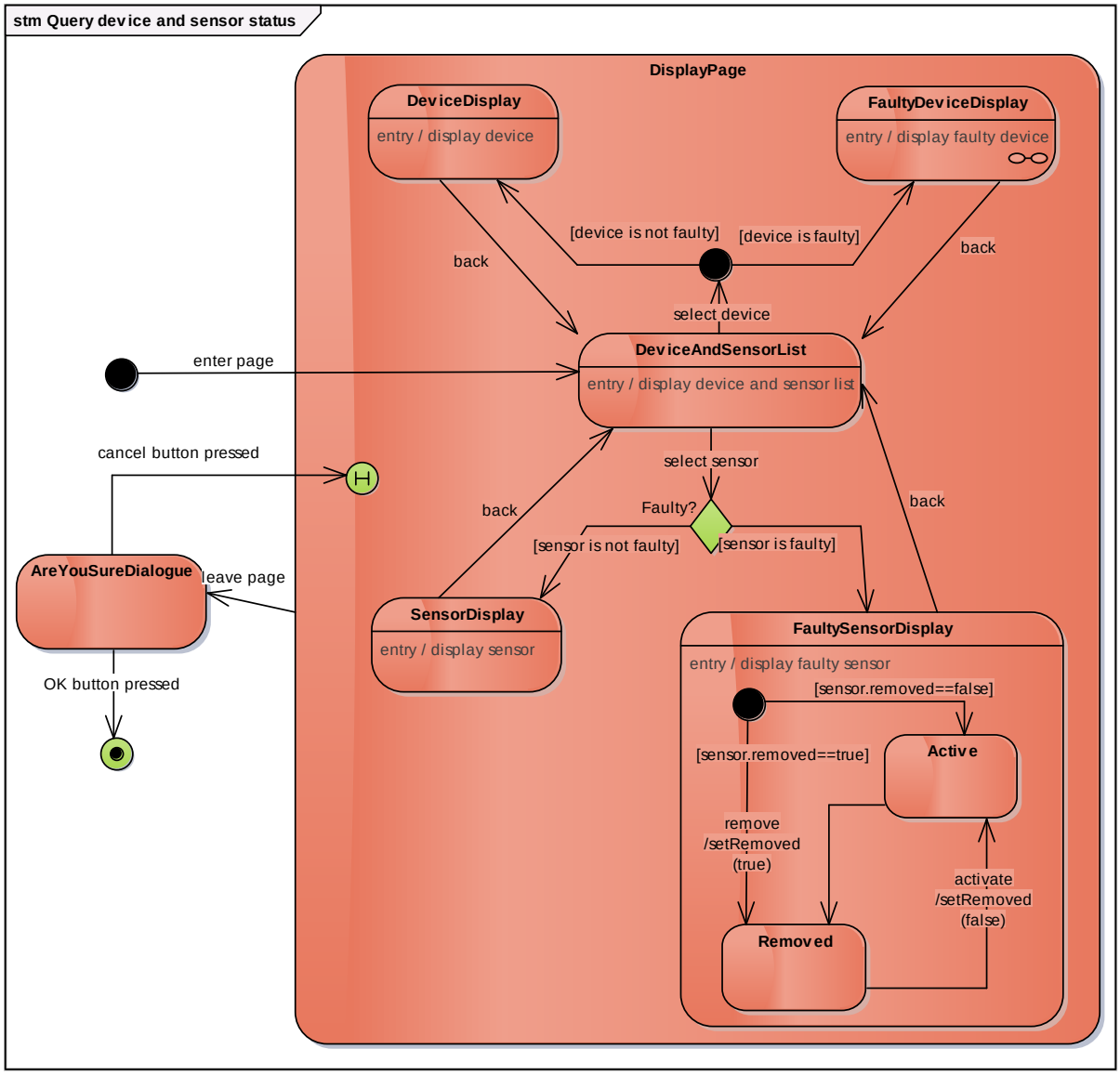
Add an initial pseudostate and two states called *Active* and *Removed*.

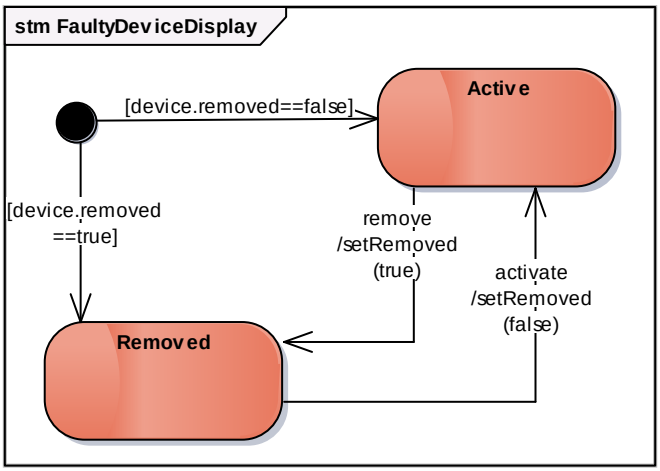
In the *Class View* find the *Device* class and add a new attribute *removed* of type *boolean*. Add operations *setRemoved(removed: boolean)* and *getRemoved() : boolean*.

Add a transition from the initial state to Active with the guard *device.getRemoved()==false*, and a transition from the initial state to *Removed* with the guard *device.getRemoved()==true*.

Add transitions from *Active* to *Removed* and *Removed* to *Active* with the triggers *remove* and *activate* respectively. Set the effect behaviour of these to be *setRemoved(true)* for the *remove* trigger and *setRemoved(false)* for the *activate* trigger, where both of these are operations of the *Device* class.

Your diagrams should look something like this.




stm FaultyDeviceDisplay
Active
[device.removed==false]
[device.removed
==true]
remove
/setRemoved
(true)
activate
/setRemoved
(false)
Removed