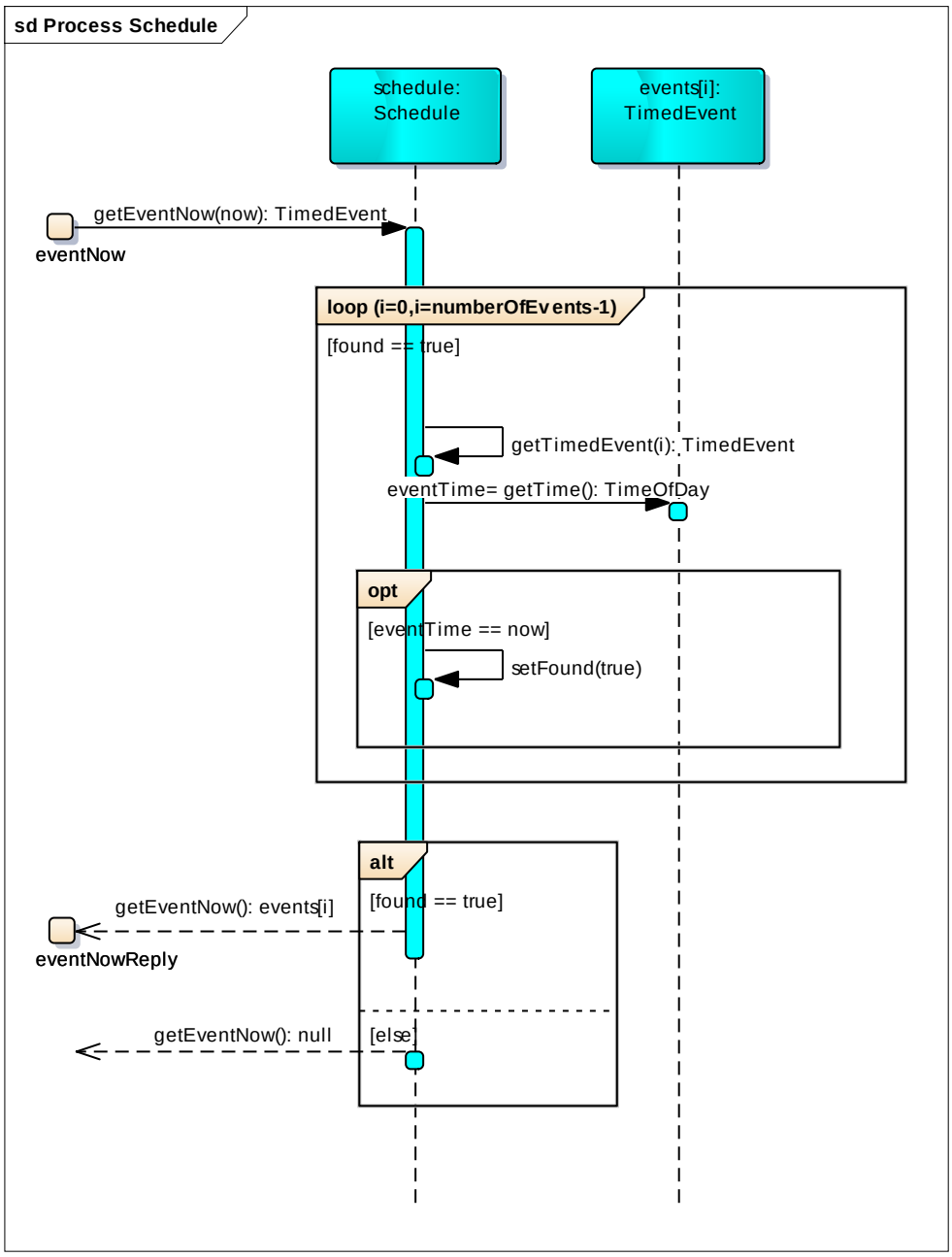
Learning UML

O'Reilly

Exercises – Combined Fragments

If you have followed along, you should have a sequence diagram called *Process Schedule* like this.



If you haven't got it, then you can add it.

Create a new package in the Interaction View called *Find Timed Events* with no diagram, then create an Interaction in that package with the same name.

Add lifelines for *:EventResponder*, *eventQueue*, *scheduleCollection*, *schedules[i]:Schedule*. and *events[i]:TimedEvent*.

Add a loop combined fragment with *(j=0,j=scheduleCollection.size()-1)* in the name field.

Within the loop draw a message from *:EventResponder* to *scheduleCollection* and call it *getSchedule* with the argument *j* and the return type of *Schedule*.

Find the diagram *Process Schedule* in the project browser and drag it into the diagram as an interaction occurrence so that it covers the *schedules[j]* and *event[i]* lifelines.

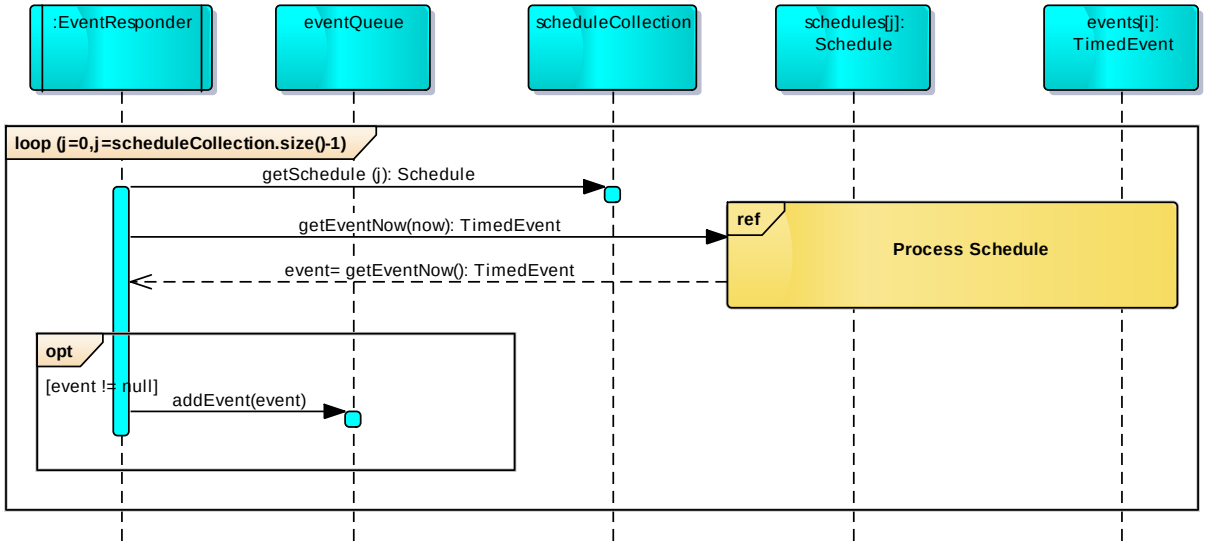
Within the loop draw a message from *:EventResponder* to the interaction *Process Schedule*. Name it *getEventNow*, with the return type *TimedEvent*, a *TimeOfDay* parameter, and the argument *now*. This should match the message from the gate in the *Process Schedule* diagram.

Also within the loop draw a message from the interaction *Process Schedule* to the *:EventResponder* lifeline and make it a reply message. Name it *getEventNow*, with the result assigned to *event*.

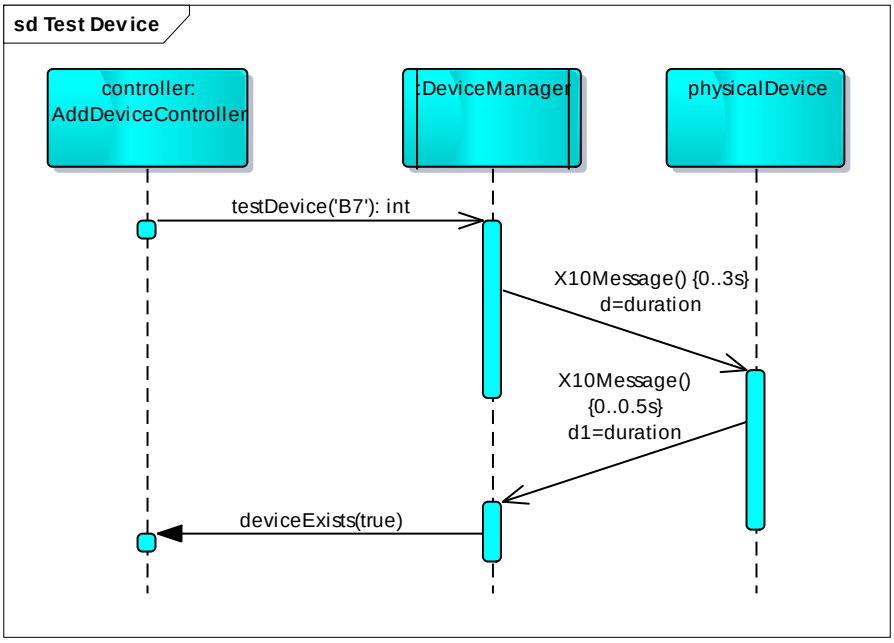
Now within the loop add an opt combined fragment that covers the *:EventResponder* and *eventQueue* lifelines. Set the condition for the interaction operand to be *event != null*.

Draw a message from *:EventResponder* to the *eventQueue* and name it *addEvent*, and pass it a parameter of type *TimedEvent*, with the argument *event*. (Note that this won't work in practice, as the DeviceManager needs to know which device the event applies to and the address of that device, so we need a different kind of event at this point, but this will do to illustrate the concept of an opt combined fragment.)

Your diagram should look something like this.



To practice creating an alt combined fragment, take the diagram *Test Device*, which should look like this.



Add the additional lifeline, combined fragment and messages so that it looks like this.