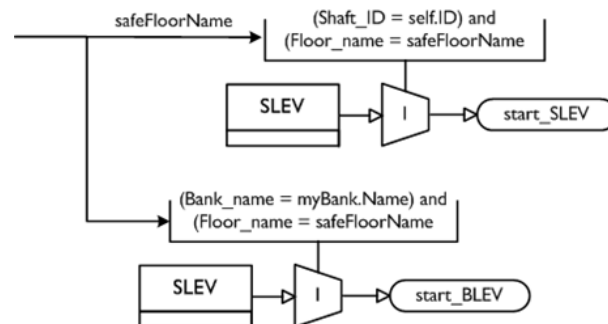
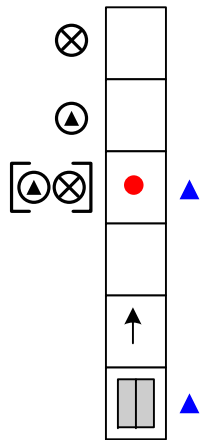


Scrall

Elevator Example Actions

Leon Starr



August 5, 2003

Version 1.0
mint.scrall.tn.2



About this note

This note demonstrates the Scroll symbols described in mint.scroll.tn.1 on some OAL (object action language) snippets taken from the Elevator Case Study.

Notes (in purple) have been added to some of the examples to clarify and elaborate on some of the Scroll rules. In some cases, these notes point out syntax issues yet to be resolved.

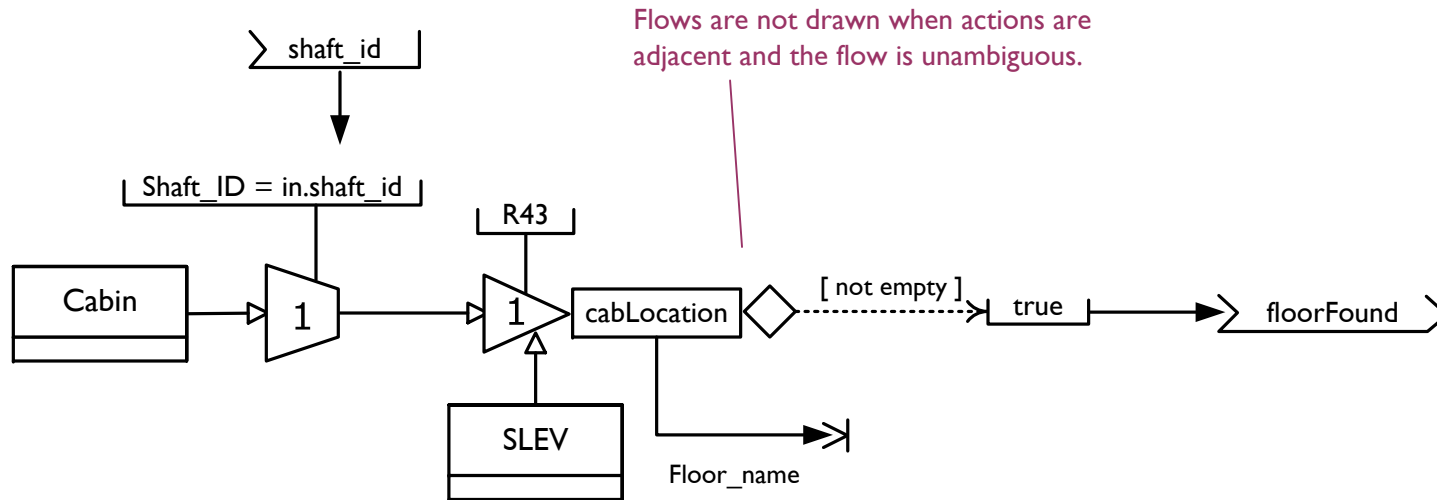


Elevator Example Actions

select any theCabin from instances of CAB where
 (selected.Shaft_ID == param.shaft_id);

select one cabLocation related by theCabin->SLEV[R43];
 param.floorFound = true;

return cabLocation.Floor_name;



Elevator Example Actions

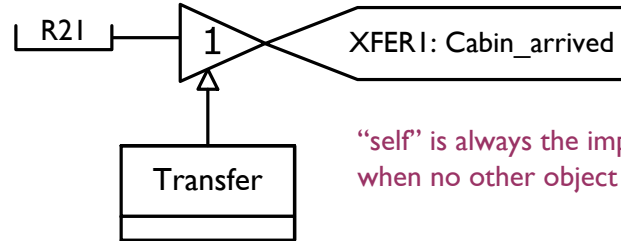
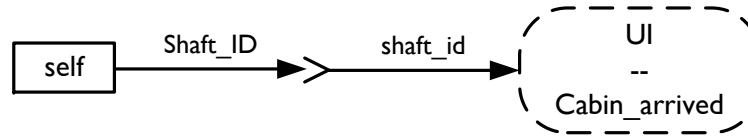


```

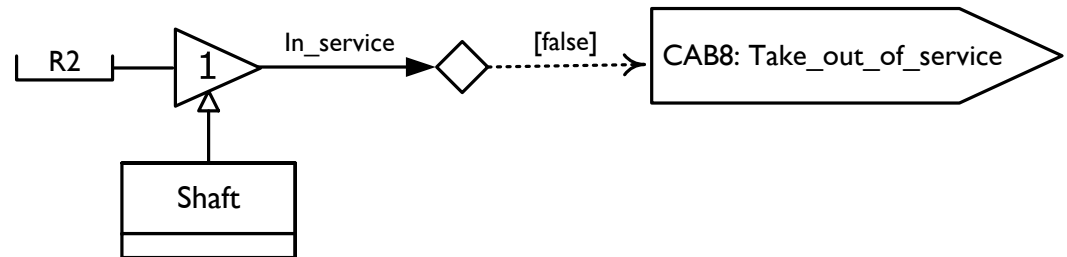
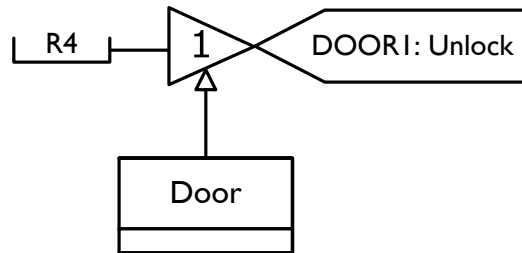
UI::Cabin_arrived(shaft_id:self.Shaft_ID);
select one myXfer related by self->XFER[R21];
Generate XFER1: Cabin_arrived () to myXfer;

select one myDoor related by self->DOOR[R4];
generate DOOR1: Unlock () to myDoor;

select one myShaft related by self->SHAFT[R2];
if (not myShaft.In_service)
    generate CAB8:Take_out_of_service () to self;
    
```



“self” is always the implied starting point when no other object is input into a finder.



Elevator Example Actions



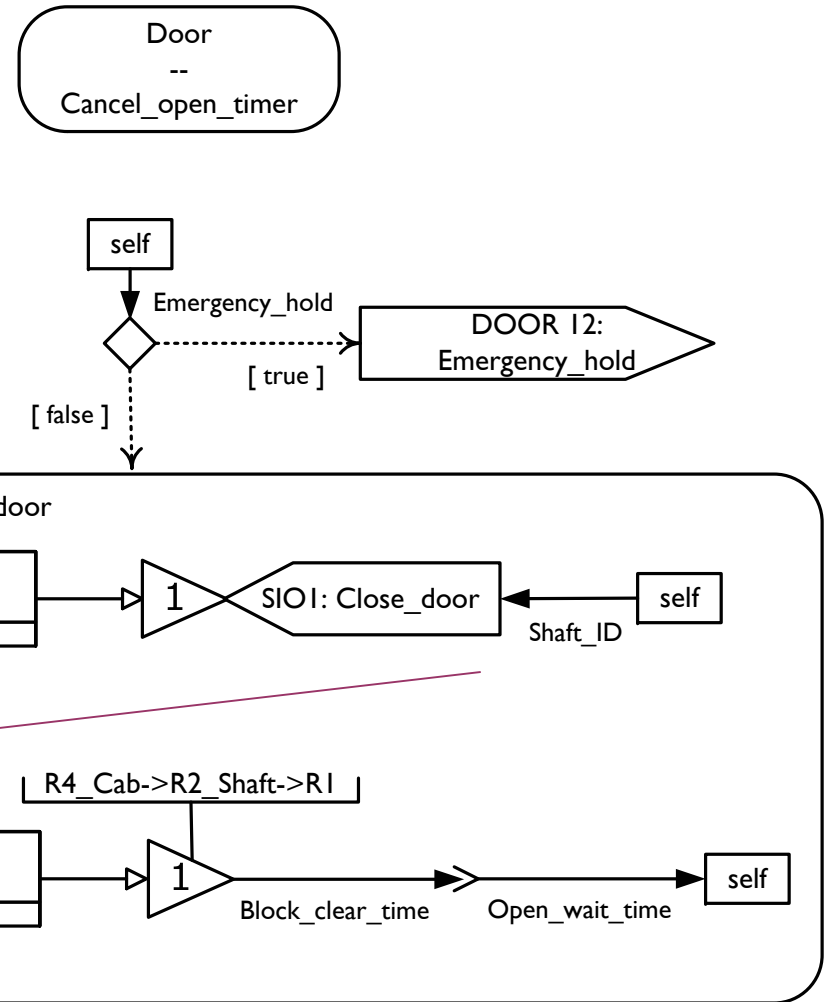
```

// State: Door.Closing
self.Cancel_open_timer();

if (self.Emergency_hold)
    generate DOOR12:Emergency_hold () to self;
else
    // Start closing the door
    // Stubbed for now SIO::close_door (shaft_id:self.Shaft_ID);
    select one mySIO_stub related by self->SIO[R44];
    generate SIO1:Close_door(shaft_id:self.Shaft_ID) to mySIO_stub;

    // If the close fails, we will reopen the doors for a potentially shorter
    // duration to wait for the obstruction to clear
    select one myBank related by self->CAB[R4]->SHAFT[R2]->BANK[R1];
    self.Open_wait_time = myBank.Block_clear_time;
end if;

```

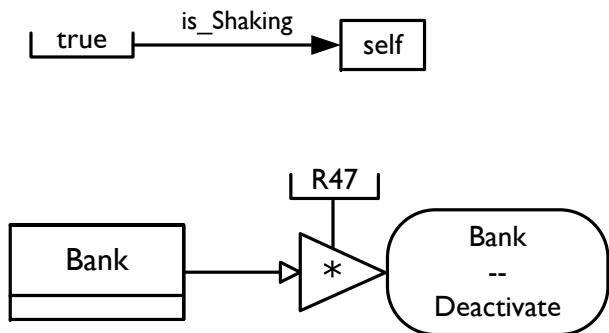


Socket not necessary since there is only one parameter specified for the SIO1 event.

Elevator Example Actions



```
// State: Building.Shaking
self.is_Shaking = true;
select many all_my_Banks related by self->BANK[R47];
for each this_Bank in all_my_Banks
  this_Bank.Deactivate();
end for;
```



In this case adjacency eliminates the need for an object flow to a multi-object store to a control flow to the class operation. The abbreviation works because there is no other possible interpretation. Either an object flow or a data flow may be output by a finder. But a data flow would have to be drawn and labeled, so the output must be an object flow. Since an object flowing into a class operation is considered an invocation we can assume that each selected bank instance is invoking its Deactivate operation.

Elevator Example Actions

Leon Starr
mint.scrall.tn.2

Page 6 of 10



```

// State: Xfer.Creating
newXfer = false;
swapXfer = false;

select any existingXfer from instances of XFER where (selected.Shaft_ID == self.ID);

if (empty existingXfer)
    newXfer = true; // nothing there - just create a new xfer
elif (existingXfer.Destination_floor != rcvd_evt.Dest_floor)
    swapXfer = true; // xfer exists with different floor - swap it with the new xfer
end if;

// If it's a swap, first delete the existing Xfer
if (swapXfer)
    // all this just to delete a stupid association
    select one theSLEV related by existingXfer->SLEV[R21];
    select one theCabin related by existingXfer->CAB[R21];
    unrelate theSLEV from theCabin across R21 using existingXfer;
    delete object instance existingXfer;
end if;

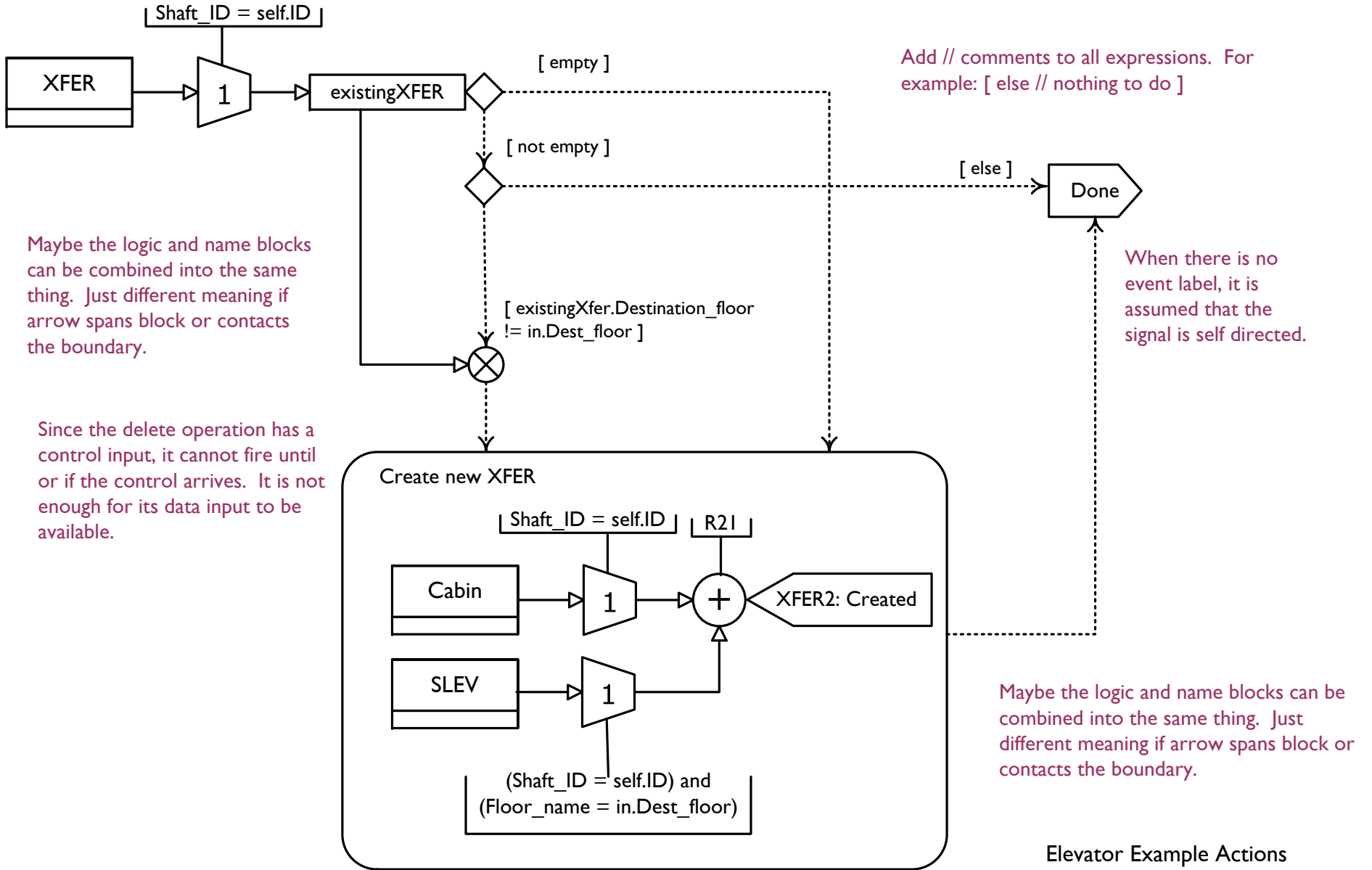
// if either is true, we create a new transfer
if (newXfer OR swapXfer)
    select any theCabin from instances of CAB where (selected.Shaft_ID == self.ID);
    select any theSLEV from instances of SLEV where
        (selected.Shaft_ID == self.ID AND
         selected.Floor_name == rcvd_evt.Dest_floor);
    create object instance newXfer_instance of XFER;
    relate theSLEV to theCabin across R21 using newXfer_instance;
    generate XFER2: Created () to newXfer_instance;
end if;

generate SHAFT3: Done to self;

```

Scroll on next page.





Elevator Example Actions



```

// Operation Tran_Stub::Get_safe_stop (floorFound:boolean)

select one myCabin related by self->CAB[R45];
select one cabLocation related by myCabin->SLEV[R43];
dirMotion = self.dir_scale; // 0 = stationary, 1 = up, -1 = down
param.floorFound = false; // default assumption

if (dirMotion != 0) // cabin is moving
    // Return the next servicable floor in the current direction
    // of cabin motion if one can be found.

    // get the cabin's last reported BLEV location
    select one myShaft related by myCabin->SHAFT[R2];
    select any searchBLEV from instances of BLEV where
    (
        (selected.Floor_name == cabLocation.Floor_name) and
        (selected.Bank_name == myShaft.Bank_name)
    );
    // Get the next higher/lower BLEV in the current direction of motion
    if (dirMotion == 1) // going up
        select one searchBLEV related by
            searchBLEV->BLEV[R31.'is accessed below'];
    elif (dirMotion == -1) // going down
        select one searchBLEV related by
            searchBLEV->BLEV[R31.'is accessed above'];
    end if;

    if (not_empty searchBLEV)
        param.floorFound = true;
        return searchBLEV.Floor_name;
    end if;
else // cabin is stationary
    // return the current cabin location
    param.floorFound = true;
    return cabLocation.Floor_name;
end if;

```

Scroll on next page.

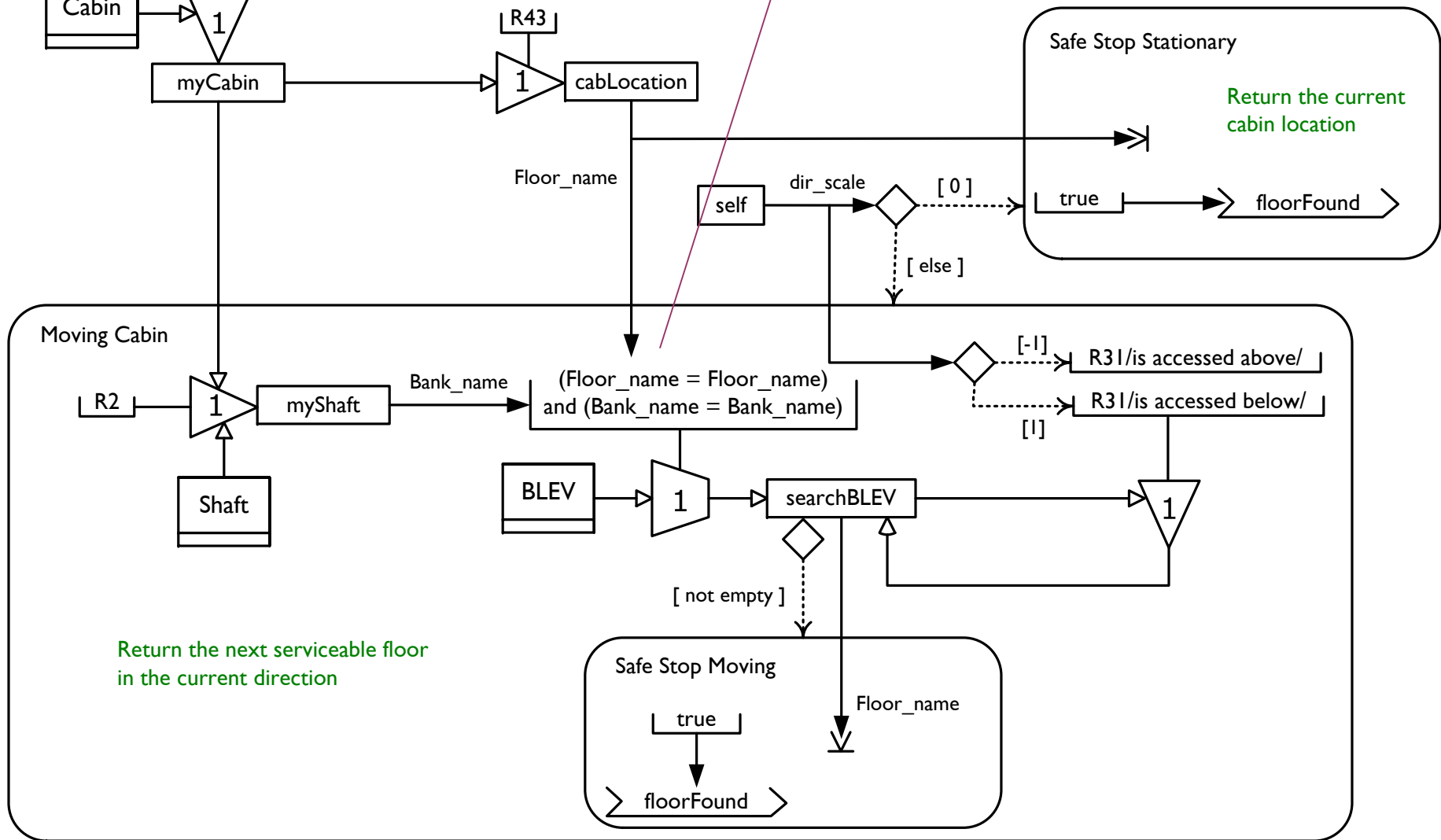


Elevator Example Actions

Error condition is the initial assumption



Since we know the lhs is an attribute of the input class and we know that data flow references may only appear on the rhs, there is no ambiguity in shortening `cabLocation.Floor_name` and `myShaft.Bank_name` to `Floor_name` and `Bank_name` respectively. And to keep the diagram simple, the abbreviations are preferable. Nonetheless, the full specification would still be legal.



Elevator Example Actions

