

Elevator Controller

The system being considered is a small office building which has an elevator providing service for the five floors in the building. You are to model the following FSP processes: **ELEVATOR**, **DISPLAY**, and **PASSENGER** along with parallel compositions **BUILDING** and **SAFE_BUILDING**.

The statements below provide information about each process. They do **not** provide a complete set of requirements for each process, in that there may be other actions that you will need to define to completely model the system. While solving this problem, be sure to consider the physical system. If something can not physically happen in the real system your model should not allow it either except for instances where magic is specified to happen.

ELEVATOR

- The elevator shall hold one passenger.
- The elevator executes the actions, **openDoor** and **closeDoor**, to start opening and closing the elevator door. It takes one **tick** for the elevator door to fully open or close.
- After a passenger has boarded, the elevator shall wait for the passenger to select the desired floor and then start closing the door. The elevator shall begin moving to the desired floor after the door is fully closed.
- After a passenger has exited, the elevator shall start closing the door. After the door is fully closed, the elevator shall begin returning to the first floor. When the elevator reaches the first floor it shall wait with the door open for a passenger to board.
- The elevator shall execute the actions, **next[f:1..5]**, when it begins leaving a floor to indicate the next floor it is going to.
- The elevator shall execute the actions, **floor[f:1..5]**, to indicate that it has arrived at that particular floor.
- It takes two **ticks** for the elevator to travel from one floor to the next going up or down.
- The management wants to move passengers through the building as quickly as possible. The elevator shall not "kill time" instead of boarding or exiting passengers.
- The elevator shall start operation on the first floor with its door open.

DISPLAY

- The display of elevator position shall be on the first floor. The display shall have five lights indicating the five floors in the building. When the elevator is stopped, the display shall illuminate the light for the floor where it is stopped. When the elevator starts moving, the display will turn off the light for the floor the elevator is leaving and illuminate the light for the floor the elevator is moving toward.
- The display shall always allow the actions **on[f:1..5]** to execute whenever a floor light is lit. The display shall execute the actions **off[f:1..5]** once to turn a floor light off.

PASSENGER

- There is no queuing of the passengers on the first floor. Each passenger is equally likely to be the next passenger to board the elevator.
- Passengers execute the actions, **board** and **exit**, to begin boarding and exiting the elevator,

respectively. A passenger shall begin boarding or exiting the elevator only when the doors have fully opened.

- It takes one **tick** for the passenger to board or exit the elevator.
- After boarding the elevator, the passenger shall execute one of the actions, **call[f:2..5]**, to indicate the desired floor. Each passenger shall select the four upper floors with equal likelihood.
- The passengers are very busy people. Except when waiting for an elevator on the first floor, a passenger shall not "kill time" if there are other productive actions to execute.
- Because the elevator is so small and the office workers are on a health kick, they only ride the elevator going up. After exiting the elevator at the desired floor, the passenger magically reappears back on the first floor to wait for the elevator again.

BUILDING and SAFE_BUILDING

- Model a system with 2 passengers.
- Define appropriate progress properties for the system.
- Define appropriate safety properties for the system. Compose your safety properties with your **BUILDING** FSP to create the **SAFE_BUILDING** composite FSP.
- Using hiding, relabeling, and interfacing, as appropriate, in your **BUILDING** FSP, make **openDoor**, **closeDoor**, **on[f:1..5]**, **off[f:1..5]**, **call[f:2..5]**, **board**, **exit**, and **tick** the visible actions in **SAFE_BUILDING**. Actions executed by the passengers may be prefaced with process labels.
- Provide a **Passenger** menu with only the passenger actions **board**, **exit**, and **call[f:2..5]**.

Other Notes and Project Requirements

- Create a structure diagram for the system that includes the safety properties.
- Make appropriate use of constants, ranges, sets, and process parameters, particularly to make sure that changing a timing parameter only requires a change in a single place.
- Consider the specification of ticks to be the number of **tick** actions that must be executed and not the number of intervals between **tick** actions that must pass.
- Within the model file be sure to include full documentation per the [specification for documentation of FSP models](#). Provide a description of what your progress and safety properties are capturing.

Submission Instructions

- Put your fully documented FSP in a file named, **elevator.lts**. Place your structure diagram in a Word, Visio, PowerPoint or PDF file named, **elevator.eee**, where **eee** is the appropriate extension for the type of file you submit.
- Place these two files in a zip named, **elevator.zip**, and deposit it before the due date and time, in the *Unit 3 Exercise* dropbox in the myCourses site for your section.