The Elevator Simulator

15 September 2014

Introduction

This document briefly discusses how to use the elevator simulator. The simulator provides a controller called *EIS Controller* that supports connecting it to an agent framework and allows agents to control elevators.

Launching the Simulator

When used to connect to an agent platform, the elevator simulator should be launched from that platform (e.g., from within the IDE used to develop code for the agent platform). Check if the agent platform that you are using distributes an example project for the elevator simulator that you can use to launch the simulator.

The "Go Dude!" button only starts the simulator, but not the agents in your agent platform. Therefore we recommend to start your system from the agent system and not use this button.

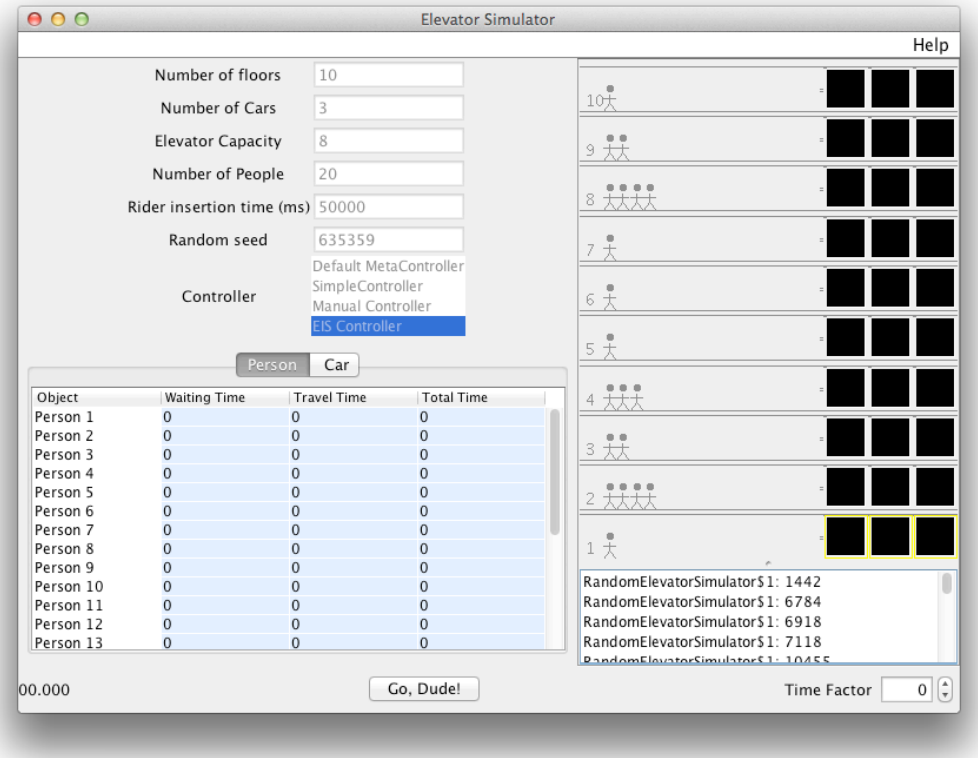

Figure 1 Elevator simulator GUI

The values shown in the top left of Figure 1 are the simulation settings. These values are shown here only for reference. You can change the settings using the environment init parameters (see the section Initialization Keys). The exact values available here depends on the chosen Simulation.

The options shown here are for the Random Rider Simulation. It places people at random floors and selects random target floors for people, but the randomness is controlled by the random seed. Alternative Simulations include the *evening traffic* option which places people at all floors and lets them pick the first floor at random times. The *random rider insertion* is discussed in more detail below; the other simulations are modifications of the same concept.

The Random Rider Simulation Option

The *random rider* option places people at random floors, and has them pick a floor and press a button at a random time. The setup panel (the left upper window in Figure 1) allows you to set various parameters used during a simulation run, including:

- the *number of floors* in the building,
- the *number of cars*, i.e., the number of elevator cars in the building,
- the *elevator capacity* of an elevator car, i.e., the maximum number of people that an elevator car can enter,
- the *number of people*, i.e., the total number of people in the building; in the random rider insertion variant of the simulator these people will be distributed randomly over the floors,
- the *rider insertion time* which determines the total running time of the simulator; the simulator will stop after that time, *even when a MAS has not yet been started*,
- the *random seed* which is used to generate pseudo-random numbers for controlling the simulator; using the same seed will result in the same distribution of persons over floors pressing buttons at the same times *if the agent controlling the elevator behaves exactly the same as well*.
- the *controller* is the mechanism that determines the behavior of the elevators. The so-called *EIS Controller* is the only controller that can be used in combination with an agent platform. Other controllers can be used when running the simulator "standalone". The *Manual Controller*, for example, will allow you to control the elevators manually: a small GUI will pop up for each car, allowing you to select a new target floor for that car at any time.

Real-Time Graphics

The elevator simulator window provides a real-time rendering of the state of the simulator. We briefly explain the areas indicated and numbered in Figure 2. The top left area was already discussed in the section The Random Rider Simulation Option. Area (1) is the *clock* and displays the current time. The clock is stopped when all people have reached their destination. Area (2) displays the *current position of elevators*. Area (3) displays the *positions of persons* outside elevators. Within that area, area (4) indicates the *target floor* that a person is trying to reach. A person only carries a target floor number when that person has decided to go to a floor (which is done at some random time before the *rider insertion time* option) and the moment the person arrives at that floor. Persons without a target floor label thus either did not yet pick a target floor or are already at their target floor.[1] When there are no persons any more labeled with a target floor number and the clock has been stopped, the simulation has ended. Area (5) displays the *state of the elevator call buttons*. Area (6) displays the *state of the elevator doors* (open, opening, closed, closing). Area (7) within the elevators highlights the *buttons that have been pressed inside the elevators*. Area (8) displays the *elevator direction lights* which indicate the direction that the elevator will move towards when the doors have been closed. Area (9) displays all *events* that were generated so far. Area (10) displays the *time factor* that is used. The time factor determines the "speed" of simulated time. The higher the faster simulated time progresses relative to real time; negative values slow the simulator down. A time factor *n* speeds up simulated time exponentially; i.e., speed is doubled when 1 is added to the time factor. Note that agents may run independently from the simulator (in separate processes or threads) and changing the time factor may not only change speed but also the behavior of your agents that control the elevators.

[1] Occasionally, it happens that a person that has a target floor number does not enter into an available elevator.

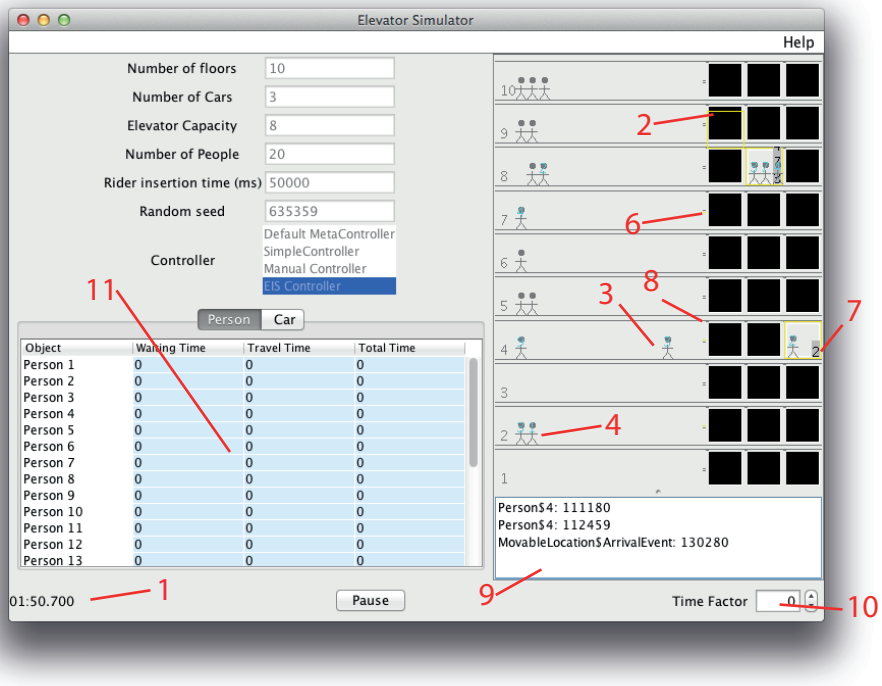

Figure 2 Explanation of Items in Simulator GUI

Area (11) displays *statistics* about cars and persons. In order to gather and show the statistics, you need to press the *Pause* button. Only after pressing that button the statistical information about persons (Figure 3) and cars (Figure 4) can be inspected.

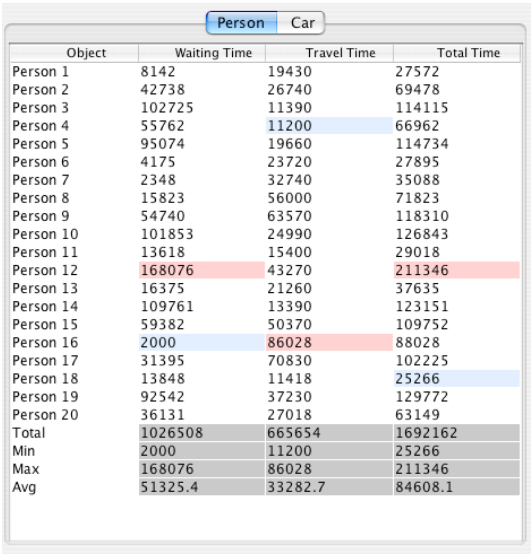

Figure 3. Statistics for persons

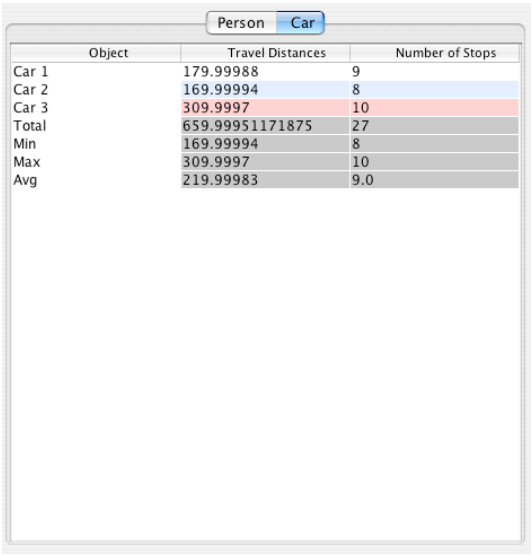

Figure 4. Statistics for cars

Actions and Percepts

To each elevator that is available in the simulator an agent can be connected that controls that elevator. The elevators are named *car<Nr>* where *<Nr>* is the number of the elevator car, counting from 0. An agent can perform only one action of going to a floor in the simulator to control an elevator and can perceive multiple percepts that inform it about the state of the elevator in the simulator. Floors are numbered (see Figure 1), with 1 referring to the ground floor and the default top floor is 10 (of course, if you changed the *Number of floors* parameter, the number for this floor will be changed accordingly). Below, we list the single action and the percepts that the simulator supports. We use `L` to refer to the floor

*l*evel (an integer > 0), `D` to refer to a *d*irection (either `up` or `down`), `S` to refer to a door *s*tate, `N` to refer to a positive integer number, and `T` to refer to the time factor of the simulation.

The action that an agent can perform to control an elevator is a go to action:

`goto(L, D)`	Instructs the elevator to go to a given floor `L` (a positive integer > 0) and, when the elevator arrives, to turn on the direction light that signals the direction `D` (where `D` is either `up` or `down`).

The opening and closing of doors upon arrival and leaving as well as the (un)loading of persons is handled by the simulator and cannot be controlled by the agent.

The elevator simulator provides each agent with the following percepts:

`atFloor(L)`	The elevator car controlled by the agent is at floor number L. Receiving this percept also means that no *goto* action has been issued to the elevator by the agent; moreover, it implies that the elevator call button on the floor is turned off, but note that it can be pressed on immediately thereafter again by people on the floor.
`fButtonOn(L,D)`	Someone pressed the floor button at level `L` and wants to travel in direction `D`. The light will stay on and the agent will receive the percept until an elevator car heading in direction `D` arrives at floor `L`.
`eButtonOn(L)`	Someone in the elevator car pressed the button to go to floor `L`. The button stays on and the agent will receive the percept until the car arrives at floor `L`.
`doorState(S)`	The elevator car is at a floor and the doors of the car are in state `S` (either `opened`, `opening`, `closed` or `closing`). This percept is *only sent when the door state has just changed*.
`people(N)`	There are `N` people in the elevator car. This percept is *only sent when the number `N` has just changed*.
`capacity(N)`	The car can hold at most `N` people. This percept is *only sent once* at the start of a simulation.
`floorCount(N)`	There are `N` floors. The percept is *only sent once* at the start of a simulation.
`carPosition(L)`	The car is at position `L`. The percept is also sent when the car is traveling. `L` is a floating point number where the fractional part indicates the position relative to the nearest floor.
`timefactor(T)`	The simulator uses time factor `T`. 2^T is the ratio of simulated and real time.

Known Issues

- Sometimes people take the "wrong" elevator, e.g., they take an elevator going down while they want to go up. Your agent has to be robust for these cases.
- Occasionally people do not press the elevator call button.

Initialization Keys

The simulator environment provides several initialization keys. Table 1 provides an overview of the available keys. If the initialization command that is sent to the simulator when it is launched includes the type of simulator (e.g., *random rider insertion*), the GUI that allows to select a simulator will not be shown. If the initialization command provides all the values required by the selected simulator type (see Table 1), the given values will be directly applied to the simulator and the user will not be given the opportunity to change the values.

Table 1. Initialization keys with their associated values.

Key	Value(s)	Type of Value	Required by SIM
Simulation	Random Rider Insertion Morning Traffic Rider Insertion Evening Traffic Rider Insertion Three Person Trip Bug Three Person Elevator Elevator Travels Up To Process A Down Request People Going Different Directions, Only One Car Three Person Two Elevator	Identifier	-
Controller	Default MetaController SimpleController Manual Controller EIS Controller	Identifier	always
Floors	Number of floors	Integer	1235678
Cars	Number of cars (elevators)	Integer	1235678
Capacity	Capacity of each car	Integer	1
People	Number of people in total	Integer	1
NPeoplePerFloor	Number of people per floor	Integer	23
InsertionTime	Insertion time in milliseconds	Integer	1
InsertionTimeHr	Insertion time in hours	Double/float	23
UpDestination	Up destination	Integer	7
DownDestination	Down destination	Integer	7
StandardDev	Standard deviation	Integer	23
RandomSeed	Random seed	Integer	123
Insert2ndReqAt	Insert second request at	Integer	4
TimeFactor	Time Factor; allowed values [-20..20]	Integer	

Acknowledgement

The source code for the elevator simulator has been written by Chris Dailey and Neil McKellar and is available at http://sourceforge.net/projects/elevatorsim.