Elevator Environment

W.Pasman, 7 october 2009

Introduction

This document discusses how to use the Elevator environment. The elevator simulator is originally written by Chris Dailey and Neil McKellar and is available in its original form via http://sourceforge.net/projects/elevatorsim. We added an extra controller, the "GOAL-Controller", to control this simulator from GOAL. This modified simulator is provided as an example with the GOAL IDE installer.

Starting up

When the elevator simulator is run, you start with an empty window (Figure 1). Select the File/New menu to select the type of person behaviour you want (Figure 2). For example, Random rider insertion places people at random floors and lets them pick a random target floor. Evening Traffic places people at all floors and lets them pick the first floor at random times. Random Rider is discussed in more detail below; the other simulations are modifications of the same concept.

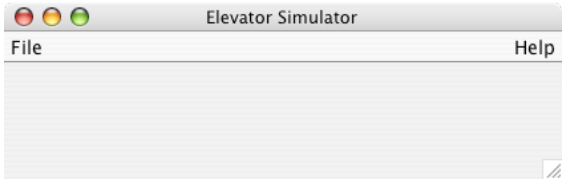


Figure 1. Initial window of the elevator simulator.

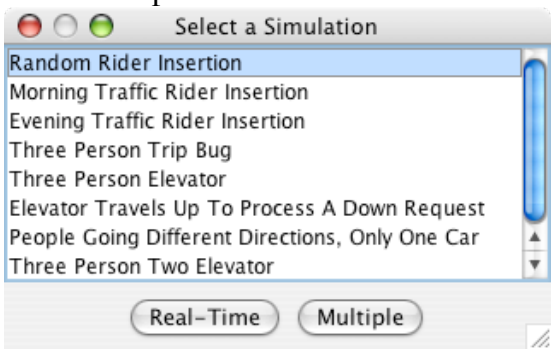


Figure 2. Person behaviour selection window

When you press select "Random Rider Insertion" and then "Real-Time", you get to the set-up panel for the single-run simulator (Figure 3). With Multiple you can run multiple simulations, but this mode is not supported with GOAL.

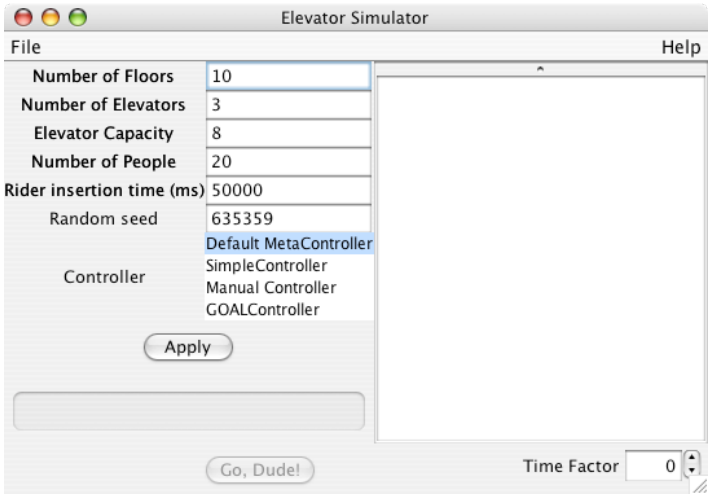


Figure 3. The set-up panel for the single-run random-rider insertion.

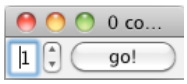


Figure 4. Manual Controller

The Random Rider Real Time Setup

The random rider is a person controller that places people at random floors, and lets them pick a floor and press a button at a random time.
The random rider real-time setup panel (Figure 3) allows you to select the number of floors, The elevator capacity is the maximum number of people that an elevator can hold.

- The number of people is the total number of people in the building. These people will be distributed randomly over the floors.
- The rider insertion time determines the total running time of the simulator. The simulator will stop after that time, *even when the GOAL MAS has not even started* (which you have to do by un-pausing the MAS).
- The random seed is used to generate the pseudo-random numbers used to control the simulator. Using the same seed will cause identical 'random' distribution of persons over the floors and the same 'random' times that these persons press buttons.
- The Controller is the mechanism that determines the behaviour of the elevators. The Default MetaController and SimpleController are elevator controllers that are built-in in the simulator. The Manual Controller (Figure 4) will let you 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. The manual controller is simplistic and always turns on the “up” direction at the floor. The GOALController is the only controller that GOAL can connect to. So if you want to use the elevator environment together with GOAL you should pick the GOALController.

Adjust the setup according to your plan and press "Apply". This activates the GOALController and sets up the environment ready to run (Figure 5). From this point, goal agents can succesfully connect to the simulator. The elevator environment is not yet active though and no percepts will be generated for the GOAL agents.

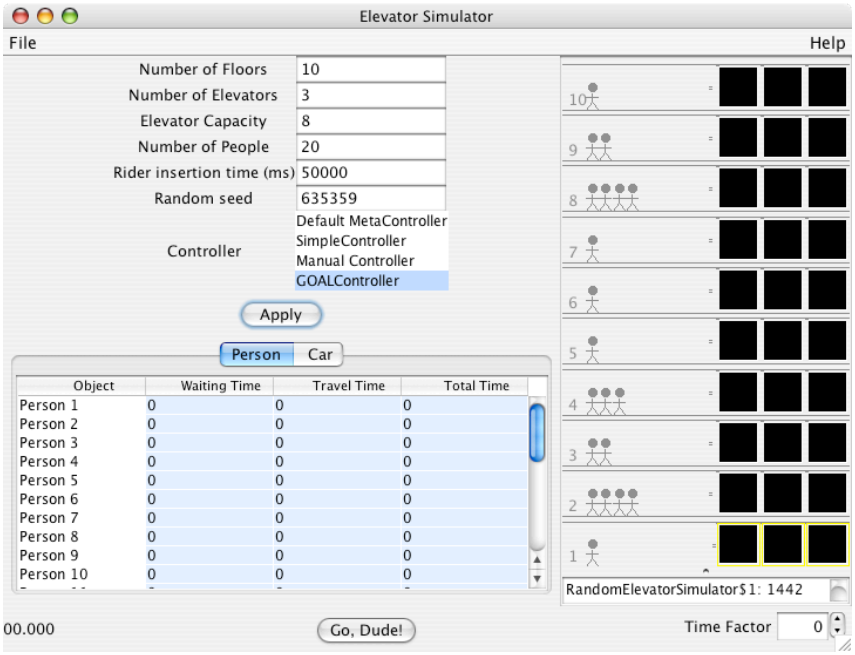


Figure 5. Elevator simulator ready to run, after pressing "Apply".

The elevator simulator is started at the moment you press "Go, Dude!". The elevator's clock starts at that moment and corresponding percepts will be passed to the GOAL agents (Figure 6).

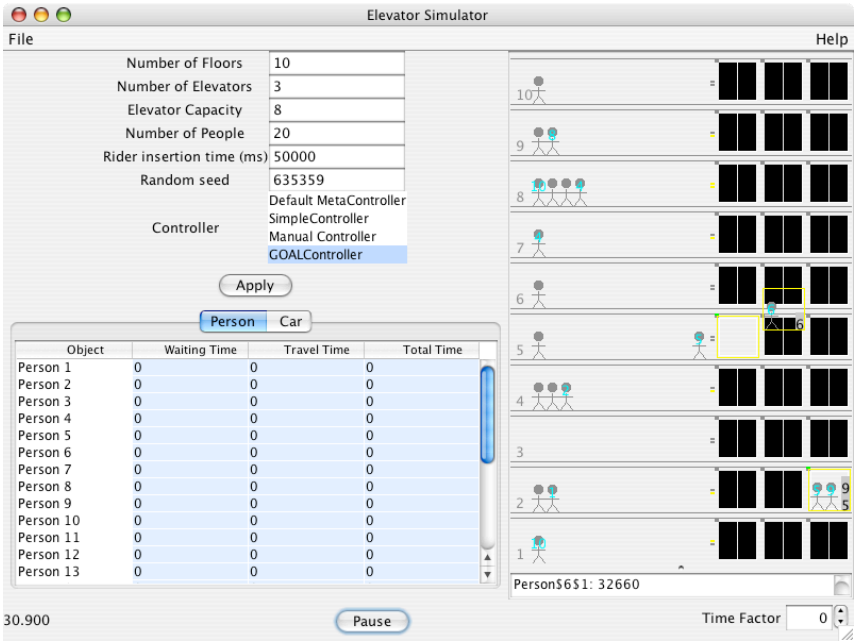


Figure 6. Elevator simulator while running.

Real-Time Graphics

The elevator simulator has a real-time rendering of the state of the world. It shows many things (see Figure 7):

1. The current time
2. The actual positions of elevators
3. The actual positions of persons
4. The target floor that a person is trying to reach,
5. The state of the elevator call buttons,
6. The state of the elevator doors,
7. Which buttons have been pressed in the elevators,
8. The elevator direction lights that indicate which direction the elevator will go after closing the doors
9. All events that were generated so far
10. The time factor. See the section Time Factor below.
11. A statistics panel showing car and person statistics.

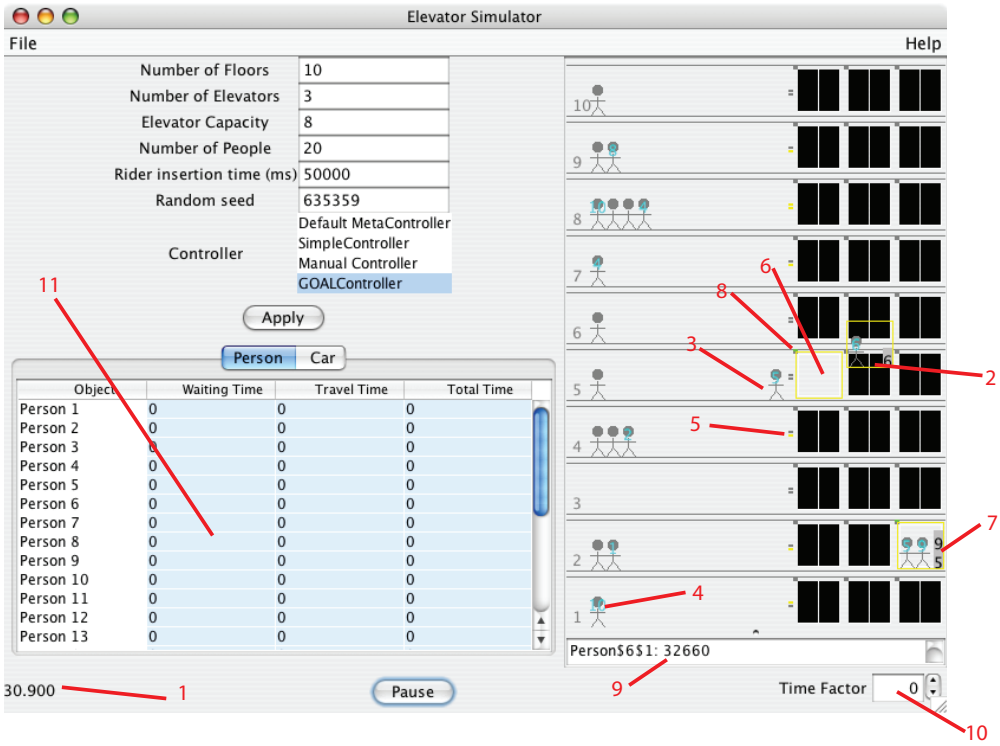


Figure 7. The items being visualized during a simulation. See text.

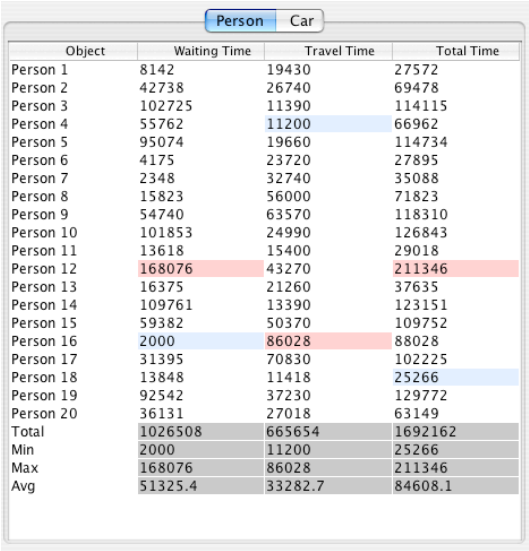
A person only carries a target floor number between the moment he decides to go to a floor (this happens at a random time before the Rider Insertion Time given at start-up) and the moment he arrives there. So persons not carrying a label may not have picked a target floor yet or already be at their target floor. Occasionally it happens that a person carries a target floor number but does not enter into an available elevator. This seems a bug of the simulator, or maybe this is intended behaviour.
When everyone has reached his destination, the elevator simulator will stop. This can be seen from the stopped clock and from the absence of persons labeled with a target floor number.

Time Factor

The time factor determines the speed of the simulated time. The higher this number n is, the faster the simulated time goes compared to the real time. Negative values result in slowdown. The simulated time goes as 2^n relative to the real time.
You should set this value only after the GOALController has been applied. The Time Factor is set to the preferred value the moment "Apply" is pressed.
GOAL runs independently, in parallel to the elevator environment. Therefore, changing the time factor can change the behaviour of your GOAL program if your GOAL code is sensitive to absolute time.

Statistics

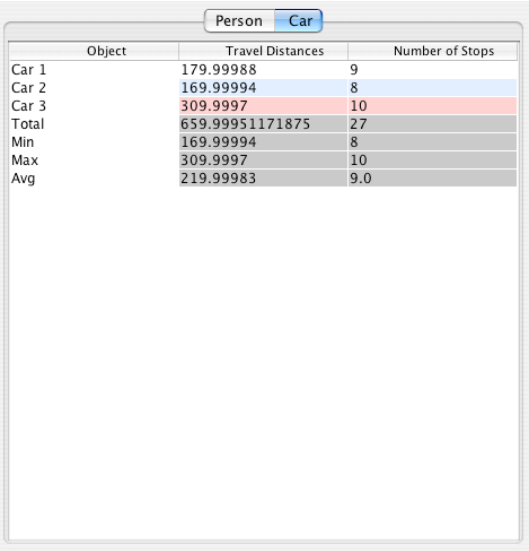
In order to gather and show the statistics, you first have to press the Pause button. After that, the statistical information about the persons (Figure 8) and about the cars (Figure 9) can be inspected.



Person | Car

Object	Waiting Time	Travel Time	Total Time
Person 1	8142	19430	27572
Person 2	42738	26740	69478
Person 3	102725	11390	114115
Person 4	55762	11200	66962
Person 5	95074	19660	114734
Person 6	4175	23720	27895
Person 7	2348	32740	35088
Person 8	15823	56000	71823
Person 9	54740	63570	118310
Person 10	101853	24990	126843
Person 11	13618	15400	29018
Person 12	168076	43270	211346
Person 13	16375	21260	37635
Person 14	109761	13390	123151
Person 15	59382	50370	109752
Person 16	2000	86028	88028
Person 17	31395	70830	102225
Person 18	13848	11418	25266
Person 19	92542	37230	129772
Person 20	36131	27018	63149
Total	1026508	665654	1692162
Min	2000	11200	25266
Max	168076	86028	211346
Avg	51325.4	33282.7	84608.1

Figure 8. Statistics for the persons



Person | Car

Object	Travel Distances	Number of Stops
Car 1	179.99988	9
Car 2	169.99994	8
Car 3	309.9997	10
Total	659.99951171875	27
Min	169.99994	8
Max	309.9997	10
Avg	219.99983	9.0

Figure 9. Statistics for the cars.

The GOALController Interface

The GOALController allows GOAL to interface with the elevator simulator.
The GOALController allows each agent to control one elevator. The GOAL Controller keeps track of the agent names that are being used, and it assigns each new agent that it encounters to the next available elevator. This means that you should provide just the same number of agents as you have elevators. We refer to the elevator associated with the agent as **the elevator**.
The GOALController refers to the floors as numbered in the graphical representation. So the **floor number** is an integer number between 1 (ground floor) and the topfloor (default is 10 but depends on your setting for "Number of floors" in the setup panel Figure 3).

There is one action:

`goto(Level, Dir)`	This command instructs the elevator to go to given Level, and to light up the given elevator direction light when the elevator arrives. If the elevator is traveling, the previous travel will be replaced with the one given. The elevator will handle all other actions itself: opening and closing doors, the (un)loading of persons, etc. `Level` is the target floor number. `Dir` is either `up` or `down`.

The elevator environment also provides GOAL agents with percepts. The following percepts are provided:

`percept(atFloor(L))`	Indicates that the car is at floor number L and that no goto() action has yet been received by the elevator. After this percept, one goto() action can be taken. At this moment, the car call light for the direction given with the associated goto action is turned off, but it can be pressed on again immediately afterwards by other people on the floor.
`percept(fButtonOn(L,Dir))`	Indicates that at level L the direction light D is on (because someone pressed the floor button). The

	light will stay on and you will receive the percept until an car heading in direction D arrives at floor L.
`percept(eButtonOn(L))`	Indicates that some person in the car pressed the button to go to floor L. The button stays on and this percept will be given until the car arrives at floor L.
`percept(doorState(S))`	Indicates that the car is atFloor() and the doors of the car are in state S. S can be opened, opening, closed or closing. This percept is sent only when the door state has changed.
`percept(people(N))`	Indicates that there are N people in the car now. This percept is sent only when N changes.
`percept(capacity(N))`	Indicates that the car can hold at most N people. This percept is sent only 1 time at start.
`percept(floorCount(N))`	Indicates that there are N floors.
`percept(carPosition(L))`	Indicates that the car is at position L. This percept is given also while the car is traveling. Here L is a floating point number where the fractional part indicates the position relative to the nearest floors.
`percept(timefactor(T))`	Indicates that the simulator currently uses time factor T. 2^T is the ratio of simulated time/real time.

EIS interface

After you have set up all settings and press "apply", the environment is created and cars are added. The new entities are named 'carX' with X the number of the car, starting at 0.

Initialization Keys

Table 1 shows the available initialization keys. These key-value pairs are passed to the environment using the ManageEnvironment command. The command is INIT and the parameters are the key-value pairs. Each key-value pair is an eis Function object, with the name of the function the Key, and the single argument of the function the value.
GOAL users can use the init[] option for the MAS file to set these parameters. GOAL will determine the Type of the value from the type of the object in the MAS file. See the GOAL Guide for more details.
When the Init command provides the SIMULATOR, the simulator GUI will not pop up. When the Init command provides all the values required by the selected SIMULATOR (see Table 1), the given values will be directly applied to the simulator and the user will not have the opportunity to change the values.

Table 1. Initialization key-value pairs for the ManageEnvironment command. Type INT refers to a Numeral holding either an Integer or Long. Similarly DBL refers to a Numeral holding either Double or Float.

Key	Value(s)	Type of Value	Required by SIM
`Simulation`	1. `Random Rider Insertion` 2. `Morning Traffic Rider Insertion` 3. `Evening Traffic Rider Insertion` 4. `Three Person Trip Bug` 5. `Three Person Elevator` 6. `Elevator Travels Up To Process A Down Request` 7. `People Going Different Directions, Only One Car` 8. `Three Person Two Elevator`	Identifier	-
`Controller`	Controller. One of 1. `Default MetaController` 2. `SimpleController` 3. `Manual Controller` 4. `GOAL Controller`	Identifier	always

`Floors`	Number of floors	INT	1235678
`Cars`	Number of cars (elevators)	INT	1235678
`Capacity`	Capacity of each car	INT	1
`People`	Number of people in total	INT	1
`NPeoplePerFloor`	Number of people per floor	INT	23
`InsertionTime`	Insertion time in milliseconds	INT	1
`InsertionTimeHr`	Insertion time in hours	DBL	23
`UpDestination`	Up destination	INT	7
`DownDestination`	Down destination	INT	7
`StandardDev`	Standard deviation	INT	23
`RandomSeed`	Random seed	INT	123
`Insert2ndReqAt`	Insert second request at	INT	4
`TimeFactor`	Time Factor. Allowed value [-20..20]	INT	

Elevator Timing Details

When fine tuning the elevator performance, it may be useful to know the time characteristics of the cars. Table 2 gives a number of characteristics.

Table 2. Timing characteristics

Characteristic	time
Open doors	2s
Close doors	2s
Wait after last passenger enters car	3s
Passenger board car	2s
Passenger leave car	2s
Car travel speed	4.03s/floor

Known Issues

It was noticed that sometimes people take the "wrong" elevator, e.g. they take an elevator going down while they want to go up. It is not clear whether this is a bug or planned behaviour of the elevator simulator. Your agent has to be robust for these cases.