SYSC 4906: Methodologies for Discrete Event Modelling and Simulation

Carleton University

March 3, 2025

Ryan Rizk 101149094

Airport Simulation Project

Introduction

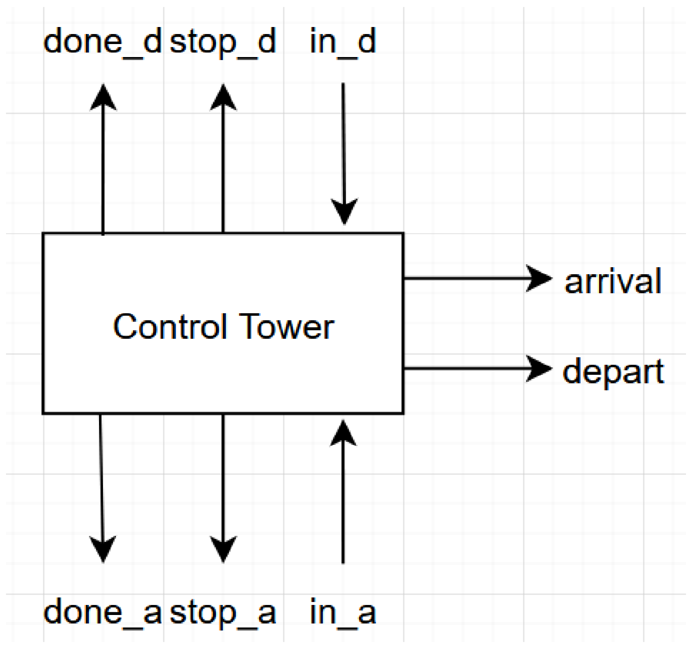
The purpose of this project is to show understanding of modeling discrete events using the DEVS formalism and converting a previous from the CD++ simulator to the Cadmium V2 simulator. This project contains a simulation of an airport that is capable of scheduling arriving and departing planes. Planes are designated to a specific location in the hangar based on their flight number. Control tower authorizes their takeoff and landing.

DEVS Models

Atomic Models

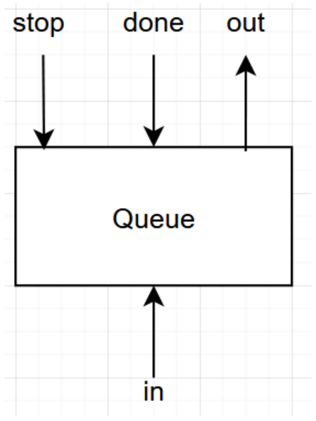
Control Tower: Determines which planes takeoff or land, giving priority to planes taking off rather than landing. The model uses two inputs and six outputs. The inputs in_a and in_d direct planes from the arrival and departure queues to the control tower. Then there are two outputs, arrival and depart, which send the plane to the runway. Finally, there are four control signal outputs that alert the queues when the runway is busy. When a plane arrives at the control tower, it takes one minute to service the plane.

The control tower begins in an idle state, when it receives a new plane from the arrival or depart queues, it enters an active state. After the necessary delay to clear the runway, the control tower is able to reschedule another plane and reenters the idle state.



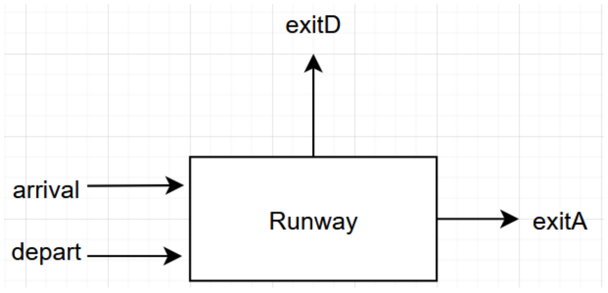
Queue: The queue has three inputs and one output. It receives planes through the in port and queues them for the control tower. Using a state variable, occupied, the queue is able to know if it should send the next plane or wait. This is done with the stop and done ports to alert when it is safe to send. Stop indicates to wait and done indicates it is safe.

The queues begin in an open state. When a plane comes to land or takeoff it is added to the queue of arrivals or departures. When the control tower gives the open signal, the queue is in the open state and closed for the occupied signal.



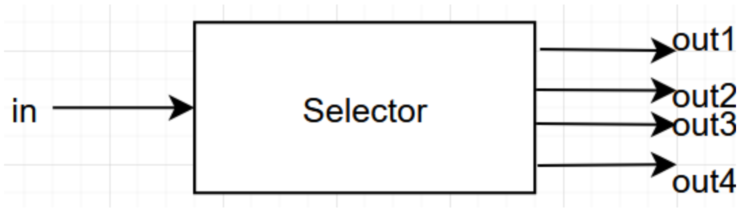
Runway: The runway is an essential part of the airport. It is responsible for directing planes when they are taking off or landing. This model has two input ports and two output ports. The input ports are incoming planes, arrival from landing and depart, coming from the control tower. The two output ports direct the plane to the runway to takeoff or to the hangar after landing.

The runway begins in an idle state. When it receives a new plane from the control tower it will determine if it is an arrival or departure based on the port. The runway enters an active state and outputs the plane to the appropriate port.



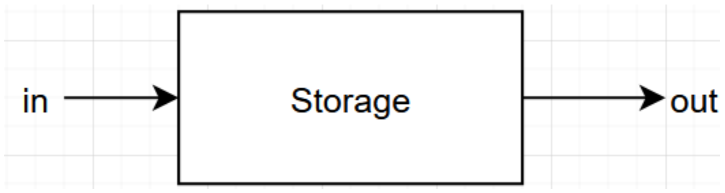
Selector: This model directs the arriving plane to the correct storage unit based on its plane number. It has one input port to receive a plane and four outputs, one to each storage unit.

The selector begins in an idle state. When it receives an input from the runway it enters an active state. The selector will then send the plane to the appropriate storage unit and reenter the idle state.

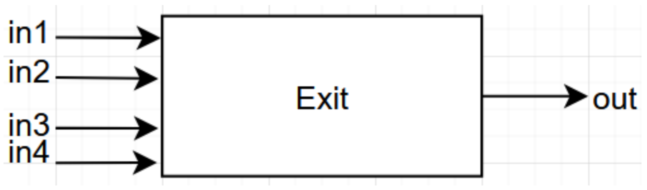


Storage: This model is responsible for queuing arrived planes until they are ready to depart. It has one input and one output to move the planes. It uses the plane number as the time delay until it is ready to depart.

The storage unit begins in an idle state. When it receives a plane to store, it enters an active state. The storage unit will be active for as long as there are planes in the unit. When the plane’s departure time arrives, the plane is sent to the exit. The storage unit becomes idle when empty.

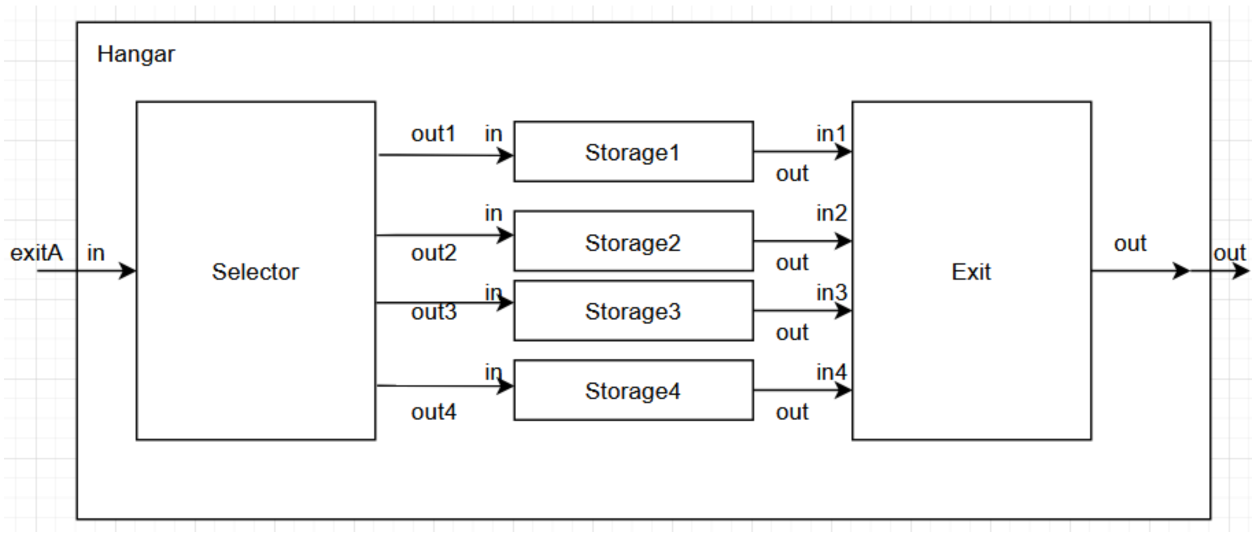


Exit: This model for the hangar exit is responsible for queue planes as they leave to ensure that there are no timing collisions if multiple planes attempt to leave storage at the same time.



Coupled Models

Hangar: The hangar coupled model contains one selector model, one exit model and four storage models. Planes are inputted from the runway and are sent to the selector. The selector determines which storage unit to use and when the plane is ready it is sent to the exit for takeoff.



DEVS Coupled Model Formalism:

Hangar = <X, Y, D, {Mi}, EIC, EOC, IC, Select>

X: {in: int}

Y: {out: int}

D: {Selector, Storage, Exit}

{Mi}: D(Selector) = M(Selector)

D(Storage) = M(Storage1)

D(Storage) = M(Storage2)

D(Storage) = M(Storage3)

D(Storage) = M(Storage4)

D(Exit) = M(Exit)

EIC: (Hangar->exitA, Selector->in)

EOC: (Exit->out, Hangar->out)

IC: (Selector->out1, Storage1->in)

(Selector->out2, Storage2->in)

(Selector->out3, Storage3->in)

(Selector->out4, Storage4->in)

(Storage1->out, Exit->in1)

(Storage2->out, Exit->in2)

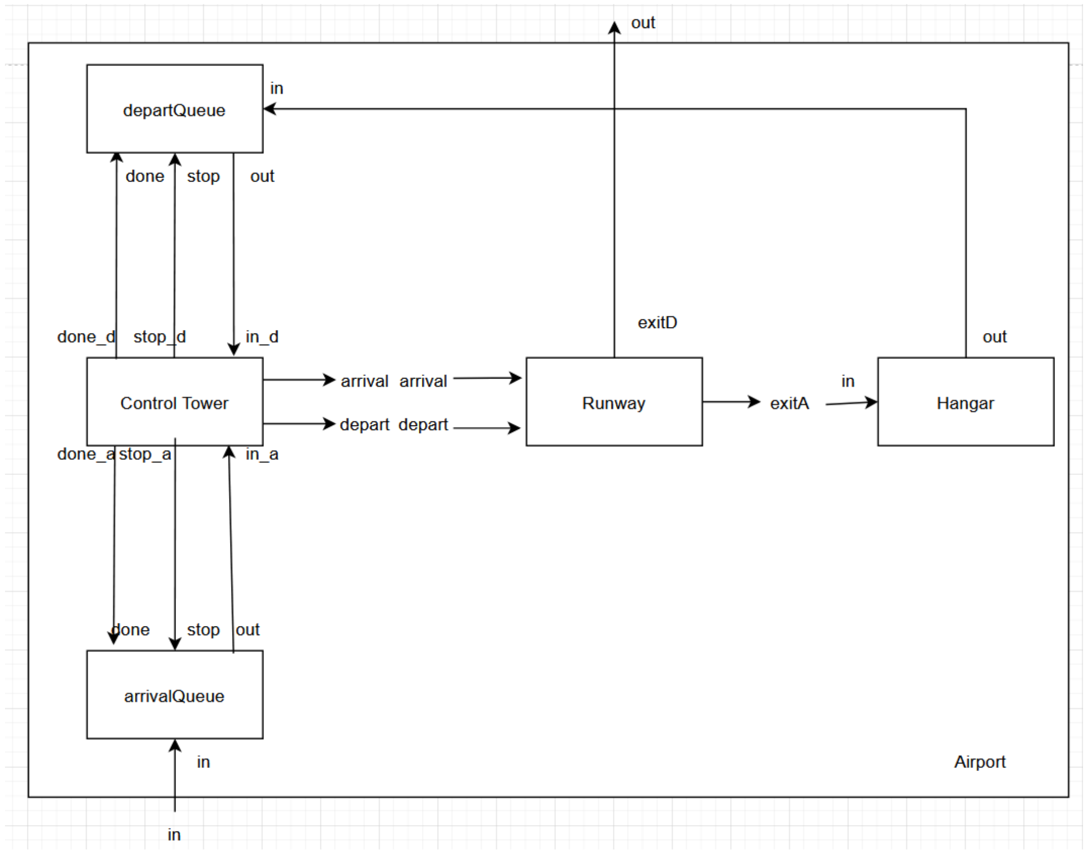
(Storage3->out, Exit->in3)

(Storage4->out, Exit->in4)

Select: ({Selector, Storage, Exit}) = Exit

({Selector, Storage}) = Storage

Airport: The airport coupled model describes the behaviour of the real system. It has one input for arriving planes and one output for departing planes. It contains the models: Control Tower, Queue, Runway and Hangar.



Airport = <X, Y, D, {Mi}, EIC, EOC, IC, Select>

X: {in: int}

Y: {out: int}

D: {Control Tower, Queue, Runway, Hangar}

{Mi}: D(Control Tower) = M(Control Tower)

D(Queue) = M(departQueue)

D(Queue) = M(arrivalQueue)

D(Runway) = M(Runway)

D(Hangar) = M(Hangar)

EIC: (Airport->in, arrivalQueue->in)

EOC: (Runway->exitD, Airport->out)

IC: (Control Tower->done_a, arrivalQueue->done)

(Control Tower->stop_a, arrivalQueue->stop)

(Control Tower->done_d, departQueue->done)

(Control Tower->stop_d, departQueue->stop)

(Control Tower->arrival, Runway->arrival)

(Control Tower->depart, Runway->depart)

(Runway->exitA, Hangar->in)

(Hangar->out, departQueue->in)

(departQueue->out, Control Tower->in_d)

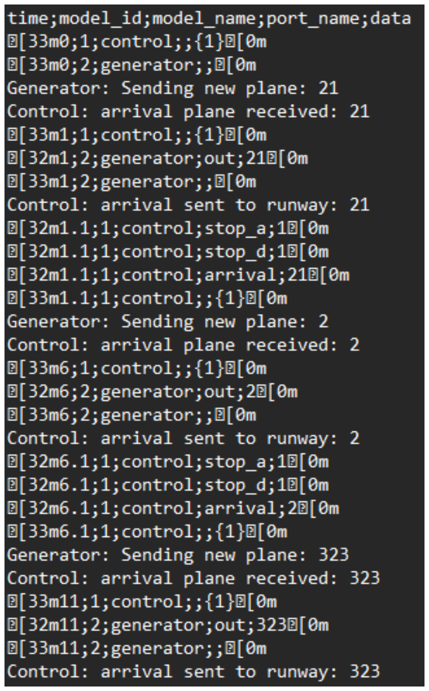
(arrivalQueue->out, Control Tower->in_a)

Select: ({arrivalQueue, departQueue}) = arrivalQueue

Simulation Results

Atomic Model Testing

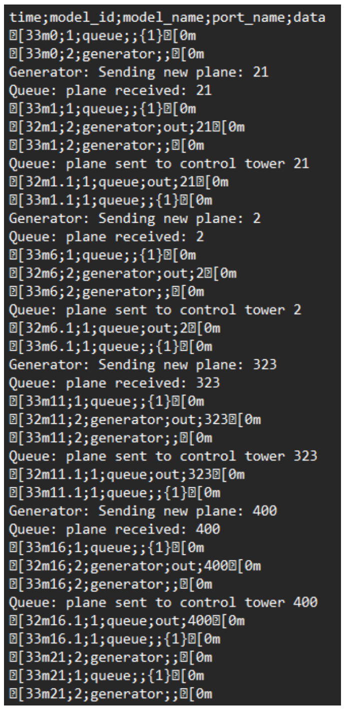
Control Tower: To verify the function of the control tower, a mimic input generator will be used to feed hard coded values with known behaviour. In this case, the Control Tower will receive two messages, each containing a plane number. The model should process the first plane and then wait till it is complete before sending the next one.



```
time;model_id;model_name;port_name;data
[33m0;1;control;;{1}[0m
[33m0;2;generator;;[0m
Generator: Sending new plane: 21
Control: arrival plane received: 21
[33m1;1;control;;{1}[0m
[32m1;2;generator;out;21[0m
[33m1;2;generator;;[0m
Control: arrival sent to runway: 21
[32m1.1;1;control;stop_a;1[0m
[32m1.1;1;control;stop_d;1[0m
[32m1.1;1;control;arrival;21[0m
[33m1.1;1;control;;{1}[0m
Generator: Sending new plane: 2
Control: arrival plane received: 2
[33m6;1;control;;{1}[0m
[32m6;2;generator;out;2[0m
[33m6;2;generator;;[0m
Control: arrival sent to runway: 2
[32m6.1;1;control;stop_a;1[0m
[32m6.1;1;control;stop_d;1[0m
[32m6.1;1;control;arrival;2[0m
[33m6.1;1;control;;{1}[0m
Generator: Sending new plane: 323
Control: arrival plane received: 323
[33m11;1;control;;{1}[0m
[32m11;2;generator;out;323[0m
[33m11;2;generator;;[0m
Control: arrival sent to runway: 323
```

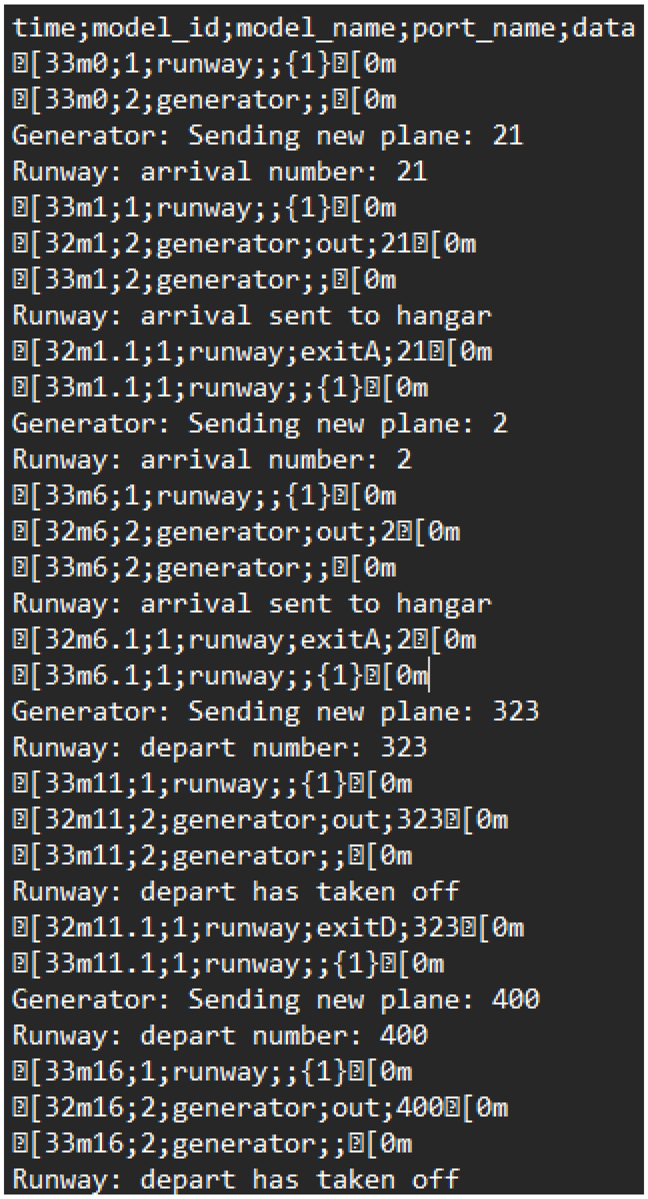
From the simulation, we can see that the Control Tower is working as described. It waits till the previous plane leaves, which takes 1 min (ta(6)).

Queue: Similar to the Control Tower, the queue should only send one plane at a time.



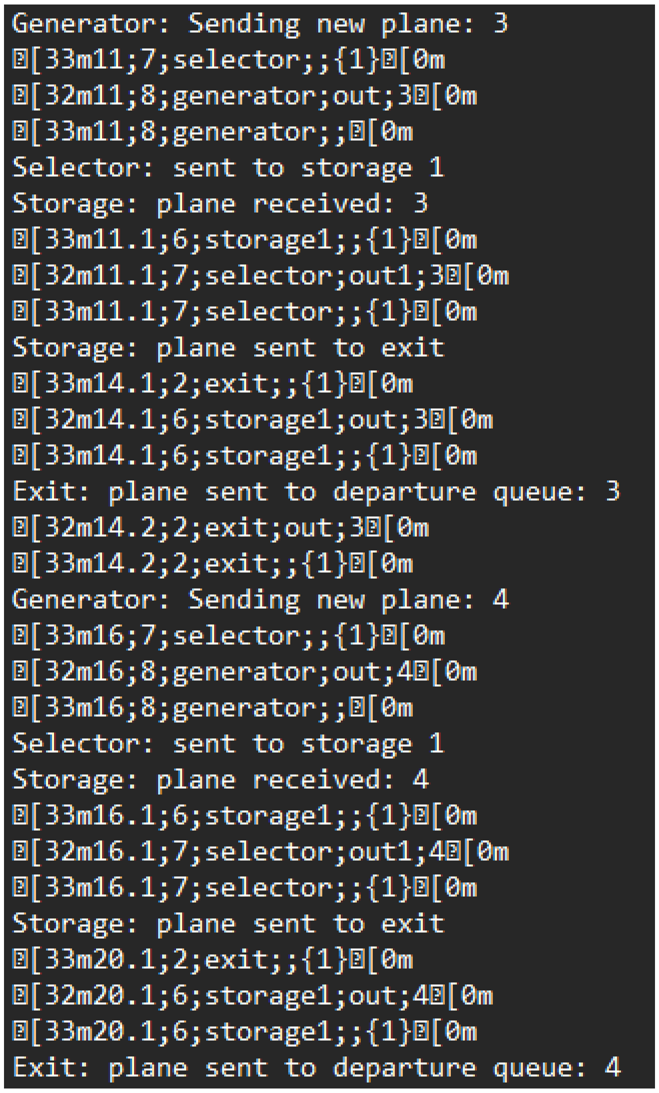
```
time;model_id;model_name;port_name;data
[33m0;1;queue;;{1}[0m
[33m0;2;generator;;[0m
Generator: Sending new plane: 21
Queue: plane received: 21
[33m1;1;queue;;{1}[0m
[32m1;2;generator;out;21[0m
[33m1;2;generator;;[0m
Queue: plane sent to control tower 21
[32m1.1;1;queue;out;21[0m
[33m1.1;1;queue;;{1}[0m
Generator: Sending new plane: 2
Queue: plane received: 2
[33m6;1;queue;;{1}[0m
[32m6;2;generator;out;2[0m
[33m6;2;generator;;[0m
Queue: plane sent to control tower 2
[32m6.1;1;queue;out;2[0m
[33m6.1;1;queue;;{1}[0m
Generator: Sending new plane: 323
Queue: plane received: 323
[33m11;1;queue;;{1}[0m
[32m11;2;generator;out;323[0m
[33m11;2;generator;;[0m
Queue: plane sent to control tower 323
[32m11.1;1;queue;out;323[0m
[33m11.1;1;queue;;{1}[0m
Generator: Sending new plane: 400
Queue: plane received: 400
[33m16;1;queue;;{1}[0m
[32m16;2;generator;out;400[0m
[33m16;2;generator;;[0m
Queue: plane sent to control tower 400
[32m16.1;1;queue;out;400[0m
[33m16.1;1;queue;;{1}[0m
[33m21;2;generator;;[0m
[33m21;1;queue;;{1}[0m
[33m21;2;generator;;[0m
```

Runway: The runway shall direct the planes accordingly. Arrived planes shall be sent to the hangar and departure planes shall takeoff. Simulation result shows correct behaviour.



```
time;model_id;model_name;port_name;data
[[33m0;1;runway;;{1}[[0m
[[33m0;2;generator;;[[0m
Generator: Sending new plane: 21
Runway: arrival number: 21
[[33m1;1;runway;;{1}[[0m
[[32m1;2;generator;out;21[[0m
[[33m1;2;generator;;[[0m
Runway: arrival sent to hangar
[[32m1.1;1;runway;exitA;21[[0m
[[33m1.1;1;runway;;{1}[[0m
Generator: Sending new plane: 2
Runway: arrival number: 2
[[33m6;1;runway;;{1}[[0m
[[32m6;2;generator;out;2[[0m
[[33m6;2;generator;;[[0m
Runway: arrival sent to hangar
[[32m6.1;1;runway;exitA;2[[0m
[[33m6.1;1;runway;;{1}[[0m
Generator: Sending new plane: 323
Runway: depart number: 323
[[33m11;1;runway;;{1}[[0m
[[32m11;2;generator;out;323[[0m
[[33m11;2;generator;;[[0m
Runway: depart has taken off
[[32m11.1;1;runway;exitD;323[[0m
[[33m11.1;1;runway;;{1}[[0m
Generator: Sending new plane: 400
Runway: depart number: 400
[[33m16;1;runway;;{1}[[0m
[[32m16;2;generator;out;400[[0m
[[33m16;2;generator;;[[0m
Runway: depart has taken off
```

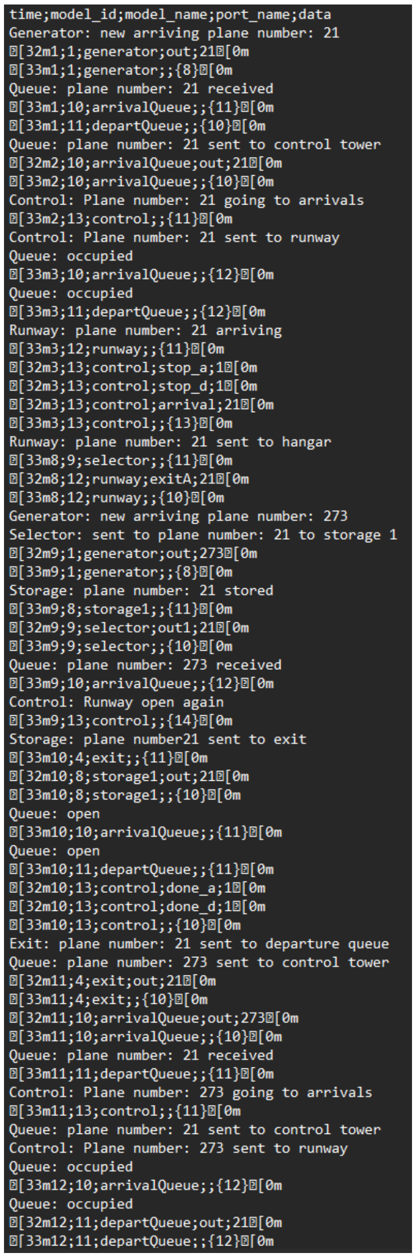
Hangar: The hangar shall store each arrival plane according to its plane number. The plane should then be ready to be sent to the departure queue at its departure time. For simulation, the plane number was also used as the time delay till the plane is able to depart again.



```
Generator: Sending new plane: 3
[33m11;7;selector;;{1}[0m
[32m11;8;generator;out;3[0m
[33m11;8;generator;;[0m
Selector: sent to storage 1
Storage: plane received: 3
[33m11.1;6;storage1;;{1}[0m
[32m11.1;7;selector;out1;3[0m
[33m11.1;7;selector;;{1}[0m
Storage: plane sent to exit
[33m14.1;2;exit;;{1}[0m
[32m14.1;6;storage1;out;3[0m
[33m14.1;6;storage1;;{1}[0m
Exit: plane sent to departure queue: 3
[32m14.2;2;exit;out;3[0m
[33m14.2;2;exit;;{1}[0m
Generator: Sending new plane: 4
[33m16;7;selector;;{1}[0m
[32m16;8;generator;out;4[0m
[33m16;8;generator;;[0m
Selector: sent to storage 1
Storage: plane received: 4
[33m16.1;6;storage1;;{1}[0m
[32m16.1;7;selector;out1;4[0m
[33m16.1;7;selector;;{1}[0m
Storage: plane sent to exit
[33m20.1;2;exit;;{1}[0m
[32m20.1;6;storage1;out;4[0m
[33m20.1;6;storage1;;{1}[0m
Exit: plane sent to departure queue: 4
```

System Testing

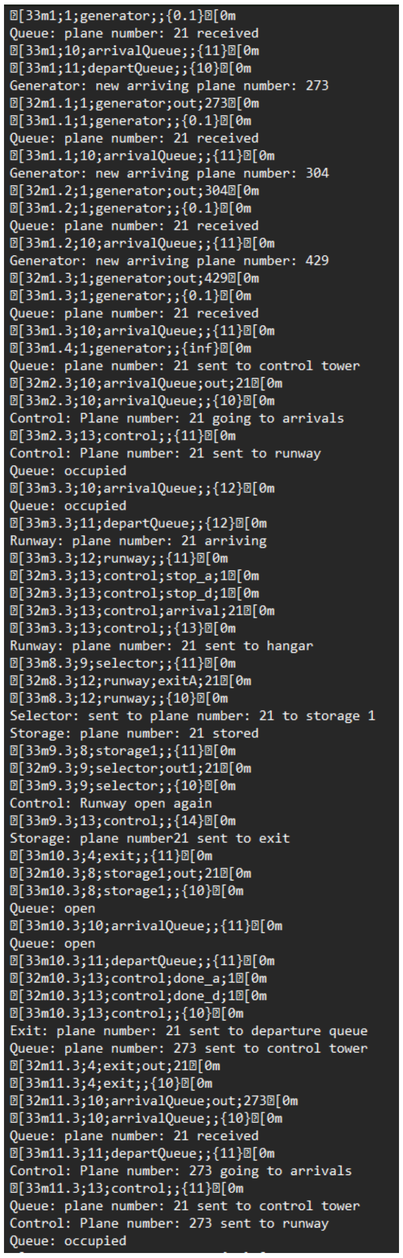
Coupling all of the models together for the complete simulation with scenario 1, many planes arriving intermittently:



```
time;model_id;model_name;port_name;data
Generator: new arriving plane number: 21
□[32m1;1;generator;out;21□[0m
□[33m1;1;generator;;{8}□[0m
Queue: plane number: 21 received
□[33m1;10;arrivalQueue;;{11}□[0m
□[33m1;11;departQueue;;{10}□[0m
Queue: plane number: 21 sent to control tower
□[32m2;10;arrivalQueue;out;21□[0m
□[33m2;10;arrivalQueue;;{10}□[0m
Control: Plane number: 21 going to arrivals
□[33m2;13;control;;{11}□[0m
Control: Plane number: 21 sent to runway
Queue: occupied
□[33m3;10;arrivalQueue;;{12}□[0m
Queue: occupied
□[33m3;11;departQueue;;{12}□[0m
Runway: plane number: 21 arriving
□[33m3;12;runway;;{11}□[0m
□[32m3;13;control;stop_a;1□[0m
□[32m3;13;control;stop_d;1□[0m
□[32m3;13;control;arrival;21□[0m
□[33m3;13;control;;{13}□[0m
Runway: plane number: 21 sent to hangar
□[33m8;9;selector;;{11}□[0m
□[32m8;12;runway;exitA;21□[0m
□[33m8;12;runway;;{10}□[0m
Generator: new arriving plane number: 273
Selector: sent to plane number: 21 to storage 1
□[32m9;1;generator;out;273□[0m
□[33m9;1;generator;;{8}□[0m
Storage: plane number: 21 stored
□[33m9;8;storage1;;{11}□[0m
□[32m9;9;selector;out1;21□[0m
□[33m9;9;selector;;{10}□[0m
Queue: plane number: 273 received
□[33m9;10;arrivalQueue;;{12}□[0m
Control: Runway open again
□[33m9;13;control;;{14}□[0m
Storage: plane number21 sent to exit
□[33m10;4;exit;;{11}□[0m
□[32m10;8;storage1;out;21□[0m
□[33m10;8;storage1;;{10}□[0m
Queue: open
□[33m10;10;arrivalQueue;;{11}□[0m
Queue: open
□[33m10;11;departQueue;;{11}□[0m
□[32m10;13;control;done_a;1□[0m
□[32m10;13;control;done_d;1□[0m
□[33m10;13;control;;{10}□[0m
Exit: plane number: 21 sent to departure queue
Queue: plane number: 273 sent to control tower
□[32m11;4;exit;out;21□[0m
□[33m11;4;exit;;{10}□[0m
□[32m11;10;arrivalQueue;out;273□[0m
□[33m11;10;arrivalQueue;;{10}□[0m
Queue: plane number: 21 received
□[33m11;11;departQueue;;{11}□[0m
Control: Plane number: 273 going to arrivals
□[33m11;13;control;;{11}□[0m
Queue: plane number: 21 sent to control tower
Control: Plane number: 273 sent to runway
Queue: occupied
□[33m12;10;arrivalQueue;;{12}□[0m
Queue: occupied
□[32m12;11;departQueue;out;21□[0m
□[33m12;11;departQueue;;{12}□[0m
```

From this simulation output we can see that each arriving plane arrives to the queue, is allowed to the runway by the control tower when open and then sent to the hangar for storage. Once the departure time of the stored planes reaches, the plane exits the hangar to enter the departures queue. The departing plane is then scheduled with the other incoming planes. All planes were serviced by the airport.

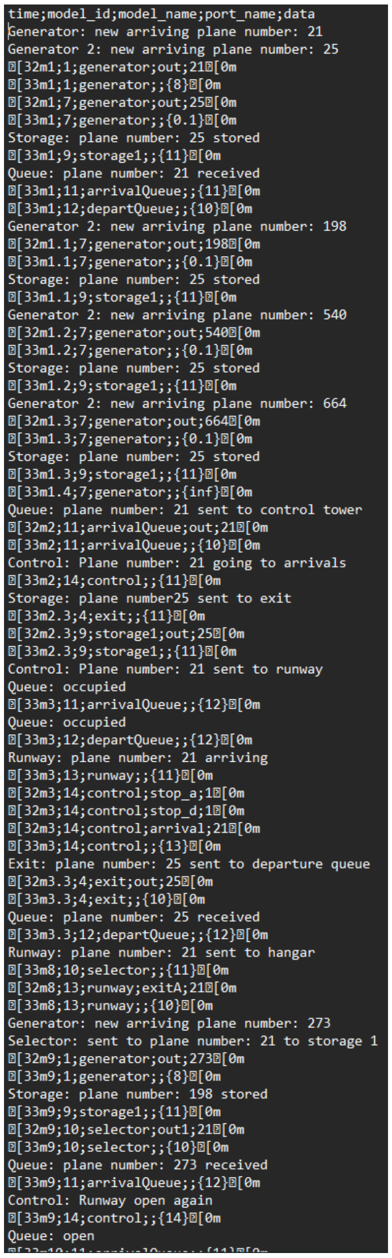
Scenario 2, many planes arriving to the airport at the same time:



```
[33m1;1;generator;;{0.1}[0m
Queue: plane number: 21 received
[33m1;10;arrivalQueue;;{11}[0m
[33m1;11;departQueue;;{10}[0m
Generator: new arriving plane number: 273
[32m1.1;1;generator;out;273[0m
[33m1.1;1;generator;;{0.1}[0m
Queue: plane number: 21 received
[33m1.1;10;arrivalQueue;;{11}[0m
Generator: new arriving plane number: 304
[32m1.2;1;generator;out;304[0m
[33m1.2;1;generator;;{0.1}[0m
Queue: plane number: 21 received
[33m1.2;10;arrivalQueue;;{11}[0m
Generator: new arriving plane number: 429
[32m1.3;1;generator;out;429[0m
[33m1.3;1;generator;;{0.1}[0m
Queue: plane number: 21 received
[33m1.3;10;arrivalQueue;;{11}[0m
[33m1.4;1;generator;;{inf}[0m
Queue: plane number: 21 sent to control tower
[32m2.3;10;arrivalQueue;out;21[0m
[33m2.3;10;arrivalQueue;;{10}[0m
Control: Plane number: 21 going to arrivals
[33m2.3;13;control;;{11}[0m
Control: Plane number: 21 sent to runway
Queue: occupied
[33m3.3;10;arrivalQueue;;{12}[0m
Queue: occupied
[33m3.3;11;departQueue;;{12}[0m
Runway: plane number: 21 arriving
[33m3.3;12;runway;;{11}[0m
[32m3.3;13;control;stop_a;1[0m
[32m3.3;13;control;stop_d;1[0m
[32m3.3;13;control;arrival;21[0m
[33m3.3;13;control;;{13}[0m
Runway: plane number: 21 sent to hangar
[33m8.3;9;selector;;{11}[0m
[32m8.3;12;runway;exitA;21[0m
[33m8.3;12;runway;;{10}[0m
Selector: sent to plane number: 21 to storage 1
Storage: plane number: 21 stored
[33m9.3;8;storage1;;{11}[0m
[32m9.3;9;selector;out1;21[0m
[33m9.3;9;selector;;{10}[0m
Control: Runway open again
[33m9.3;13;control;;{14}[0m
Storage: plane number21 sent to exit
[33m10.3;4;exit;;{11}[0m
[32m10.3;8;storage1;out;21[0m
[33m10.3;8;storage1;;{10}[0m
Queue: open
[33m10.3;10;arrivalQueue;;{11}[0m
Queue: open
[33m10.3;11;departQueue;;{11}[0m
[32m10.3;13;control;done_a;1[0m
[32m10.3;13;control;done_d;1[0m
[33m10.3;13;control;;{10}[0m
Exit: plane number: 21 sent to departure queue
Queue: plane number: 273 sent to control tower
[32m11.3;4;exit;out;21[0m
[33m11.3;4;exit;;{10}[0m
[32m11.3;10;arrivalQueue;out;273[0m
[33m11.3;10;arrivalQueue;;{10}[0m
Queue: plane number: 21 received
[33m11.3;11;departQueue;;{11}[0m
Control: Plane number: 273 going to arrivals
[33m11.3;13;control;;{11}[0m
Queue: plane number: 21 sent to control tower
Control: Plane number: 273 sent to runway
Queue: occupied
```

In scenario 2, we can see a similar output. Here we can see the queues working as described as only one plane may use the runway at a time. After the first plane is serviced and once the runway is cleared, the next plane is able to land and be sent to the hangar.

Scenario 3, some planes are already in storage:



```
time;model_id;model_name;port_name;data
Generator: new arriving plane number: 21
Generator 2: new arriving plane number: 25
[32m1;1;generator;out;21[0m
[33m1;1;generator;;{8}[0m
[32m1;7;generator;out;25[0m
[33m1;7;generator;;{0.1}[0m
Storage: plane number: 25 stored
[33m1;9;storage1;;{11}[0m
Queue: plane number: 21 received
[33m1;11;arrivalQueue;;{11}[0m
[33m1;12;departQueue;;{10}[0m
Generator 2: new arriving plane number: 198
[32m1.1;7;generator;out;198[0m
[33m1.1;7;generator;;{0.1}[0m
Storage: plane number: 25 stored
[33m1.1;9;storage1;;{11}[0m
Generator 2: new arriving plane number: 540
[32m1.2;7;generator;out;540[0m
[33m1.2;7;generator;;{0.1}[0m
Storage: plane number: 25 stored
[33m1.2;9;storage1;;{11}[0m
Generator 2: new arriving plane number: 664
[32m1.3;7;generator;out;664[0m
[33m1.3;7;generator;;{0.1}[0m
Storage: plane number: 25 stored
[33m1.3;9;storage1;;{11}[0m
[33m1.4;7;generator;;{inf}[0m
Queue: plane number: 21 sent to control tower
[32m2;11;arrivalQueue;out;21[0m
[33m2;11;arrivalQueue;;{10}[0m
Control: Plane number: 21 going to arrivals
[33m2;14;control;;{11}[0m
Storage: plane number25 sent to exit
[33m2.3;4;exit;;{11}[0m
[32m2.3;9;storage1;out;25[0m
[33m2.3;9;storage1;;{11}[0m
Control: Plane number: 21 sent to runway
Queue: occupied
[33m3;11;arrivalQueue;;{12}[0m
Queue: occupied
[33m3;12;departQueue;;{12}[0m
Runway: plane number: 21 arriving
[33m3;13;runway;;{11}[0m
[32m3;14;control;stop_a;1[0m
[32m3;14;control;stop_d;1[0m
[32m3;14;control;arrival;21[0m
[33m3;14;control;;{13}[0m
Exit: plane number: 25 sent to departure queue
[32m3.3;4;exit;out;25[0m
[33m3.3;4;exit;;{10}[0m
Queue: plane number: 25 received
[33m3.3;12;departQueue;;{12}[0m
Runway: plane number: 21 sent to hangar
[33m8;10;selector;;{11}[0m
[32m8;13;runway;exitA;21[0m
[33m8;13;runway;;{10}[0m
Generator: new arriving plane number: 273
Selector: sent to plane number: 21 to storage 1
[32m9;1;generator;out;273[0m
[33m9;1;generator;;{8}[0m
Storage: plane number: 198 stored
[33m9;9;storage1;;{11}[0m
[32m9;10;selector;out1;21[0m
[33m9;10;selector;;{10}[0m
Queue: plane number: 273 received
[33m9;11;arrivalQueue;;{12}[0m
Control: Runway open again
[33m9;14;control;;{14}[0m
Queue: open
[33m10;11;arrivalQueue;;{11}[0m
```

In scenario 3, many planes are already in storage from initialization. There are still planes arriving, but since the other planes are in the hangar, the runway is still open. From the simulation results we can see the model servicing plane numbers 25 and 21 at the same time since they are in separate parts of the airport.