SYSC 5104

Assignment 1

ATM Model Converted from CD++

Owen Petersen (101233850)

10 February 2025

Original model by Hesham Saadawi

The most significant change I have made from the original CD++ model of the ATM system is the removal of the UI interface atomic model which was replaced with input ports at the top level. Additionally, I created a RequestHandler atomic model to help manage these new inputs and coordinate better between the other components of the Validation coupled model. They style of the previous CD++ tests was also slightly adapted but their purpose remained mostly the same.

**Coupled Models:**

Top Model:

Description: The top model encapsulates the entire ATM system.

Formal Specification:

Tests:

Previous tests:

These tests are insufficient since the top model now has additional input ports

New Tests

1. Goal: Successful sequence of inputs to withdraw money and received cash output.

00:00:01:000 Card\_in 1

00:00:04:000 Pin 1234

00:00:06:000 Amount 150

Expected Results: Port Cash will give an output of 150 and the Card will also give an output of 1 around the same time signifying that the cash is dispensed and the card is returned to the user.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;amount\_in file;;6

0;2;pin\_in file;;4

0;3;card\_in file;;1

0;5;cash\_manager;;Next State = 0

0;7;account\_balance;;Next State = 0

0;8;request\_handler;;Next State = 0

0;9;pin\_verifier;;Next State = 0

0;10;card\_reader;;Next State = 0

1;3;card\_in file;out;1

1;3;card\_in file;;inf

1;10;card\_reader;;Next State = 1

2;9;pin\_verifier;;Next State = 1

2;10;card\_reader;card\_number\_out;111

2;10;card\_reader;;Next State = 2

3;10;card\_reader;;Next State = 2

4;2;pin\_in file;out;1234

4;2;pin\_in file;;inf

4;9;pin\_verifier;;Next State = 2

4;10;card\_reader;;Next State = 2

4.5;8;request\_handler;;Next State = 1

4.5;9;pin\_verifier;pin\_valid\_out;1

4.5;9;pin\_verifier;;Next State = 0

5;10;card\_reader;;Next State = 2

6;1;amount\_in file;out;150

6;1;amount\_in file;;inf

6;8;request\_handler;;Next State = 2

6;10;card\_reader;;Next State = 2

6;7;account\_balance;;Next State = 1

6;8;request\_handler;request\_out;32766

6;8;request\_handler;;Next State = 3

6.1;7;account\_balance;approved\_out;0

6.1;7;account\_balance;;Next State = 0

6.1;8;request\_handler;;Next State = 5

6.1;8;request\_handler;eject\_out;1

6.1;8;request\_handler;;Next State = 0

6.1;10;card\_reader;;Next State = 3

7.1;10;card\_reader;card\_out;1

7.1;10;card\_reader;;Next State = 0

7.1;1;amount\_in file;;inf

7.1;2;pin\_in file;;inf

7.1;3;card\_in file;;inf

7.1;5;cash\_manager;;Next State = 0

7.1;7;account\_balance;;Next State = 0

7.1;8;request\_handler;;Next State = 0

7.1;9;pin\_verifier;;Next State = 0

7.1;10;card\_reader;;Next State = 0

1. Goal: Invalid sequence of inputs causing the card to be ejected early and no cash dispensed  
   00:00:03:000 Pin 1234

00:00:05:000 Card\_in 1

00:00:07:000 Amount 150

Expected Results: The Pin input will be ignored since the model is not in the correct state to receive it. The Card input will be received correctly but the Amount input will be received in the wrong state and cause the card to be ejected  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;amount\_in file;;7

0;2;pin\_in file;;3

0;3;card\_in file;;5

0;5;cash\_manager;;Next State = 0

0;7;account\_balance;;Next State = 0

0;8;request\_handler;;Next State = 0

0;9;pin\_verifier;;Next State = 0

0;10;card\_reader;;Next State = 0

1;10;card\_reader;;Next State = 0

2;10;card\_reader;;Next State = 0

3;2;pin\_in file;out;1234

3;2;pin\_in file;;inf

3;9;pin\_verifier;;Next State = 0

3;10;card\_reader;;Next State = 0

4;10;card\_reader;;Next State = 0

5;3;card\_in file;out;1

5;3;card\_in file;;inf

5;10;card\_reader;;Next State = 1

6;9;pin\_verifier;;Next State = 1

6;10;card\_reader;card\_number\_out;111

6;10;card\_reader;;Next State = 2

7;1;amount\_in file;out;150

7;1;amount\_in file;;inf

7;8;request\_handler;;Next State = 5

7;10;card\_reader;;Next State = 2

0;8;request\_handler;eject\_out;1

0;8;request\_handler;;Next State = 0

0;10;card\_reader;;Next State = 3

1;10;card\_reader;card\_out;1

1;10;card\_reader;;Next State = 0

2;10;card\_reader;;Next State = 0

3;10;card\_reader;;Next State = 0

4;10;card\_reader;;Next State = 0

5;10;card\_reader;;Next State = 0

6;10;card\_reader;;Next State = 0

7;10;card\_reader;;Next State = 0

7;1;amount\_in file;;inf

7;2;pin\_in file;;inf

7;3;card\_in file;;inf

7;5;cash\_manager;;Next State = 0

7;7;account\_balance;;Next State = 0

7;8;request\_handler;;Next State = 0

7;9;pin\_verifier;;Next State = 1

7;10;card\_reader;;Next State = 0

Validation:

Description: The validation coupled model is a collection of components responsible for validating a request and determining how it should be handled (i.e. what outputs).

Formal Specification:

Tests:

Previous tests:

1. Goal: Entering the wrong pin 3 times will eject the card.

00:00:01:000 CardNumber 111

00:00:03:000 Pin 0000

00:00:05:000 Pin 0001

00:00:07:000 Pin 0002

Expected Results: After the third incorrect input from port Pin there should be an output of 1 in the Eject port signalling that there have been too many failed attempts and the card should be ejected.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;card\_number\_in file;;1

0;2;pin\_in file;;3

0;4;account\_balance;;Next State = 0

0;5;request\_handler;;Next State = 0

0;6;pin\_verifier;;Next State = 0

1;1;card\_number\_in file;out;111

1;1;card\_number\_in file;;inf

1;6;pin\_verifier;;Next State = 1

3;2;pin\_in file;out;0

3;2;pin\_in file;;2

3;6;pin\_verifier;;Next State = 2

3.5;5;request\_handler;;Next State = 1

3.5;6;pin\_verifier;pin\_valid\_out;0

3.5;6;pin\_verifier;;Next State = 1

5;2;pin\_in file;out;1

5;2;pin\_in file;;2

5;6;pin\_verifier;;Next State = 2

5.5;5;request\_handler;;Next State = 1

5.5;6;pin\_verifier;pin\_valid\_out;0

5.5;6;pin\_verifier;;Next State = 1

7;2;pin\_in file;out;2

7;2;pin\_in file;;inf

7;6;pin\_verifier;;Next State = 2

7.5;5;request\_handler;;Next State = 1

7.5;6;pin\_verifier;pin\_valid\_out;0

7.5;6;pin\_verifier;;Next State = 1

7.5;1;card\_number\_in file;;inf

7.5;2;pin\_in file;;inf

7.5;4;account\_balance;;Next State = 0

7.5;5;request\_handler;;Next State = 1

7.5;6;pin\_verifier;;Next State = 1

New Tests:

1. Goal: Successful sequence of inputs producing a valid transaction and outputting the amount to dispense and a signal to eject the card.

00:00:01:000 CardNumber 111

00:00:03:000 Pin 1234

00:00:05:000 Amount 150

Expected Results: Port Dispense will give an output of 150 and the Eject will also give an output of 1 around the same time.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;amount\_in file;;5

0;2;card\_number\_in file;;1

0;3;pin\_in file;;3

0;5;account\_balance;;Next State = 0

0;6;request\_handler;;Next State = 0

0;7;pin\_verifier;;Next State = 0

1;2;card\_number\_in file;out;111

1;2;card\_number\_in file;;inf

1;7;pin\_verifier;;Next State = 1

3;3;pin\_in file;out;1234

3;3;pin\_in file;;inf

3;7;pin\_verifier;;Next State = 2

3.5;6;request\_handler;;Next State = 1

3.5;7;pin\_verifier;pin\_valid\_out;1

3.5;7;pin\_verifier;;Next State = 0

5;1;amount\_in file;out;150

5;1;amount\_in file;;inf

5;6;request\_handler;;Next State = 2

5;5;account\_balance;;Next State = 1

5;6;request\_handler;request\_out;32765

5;6;request\_handler;;Next State = 3

5.1;5;account\_balance;approved\_out;0

5.1;5;account\_balance;;Next State = 0

5.1;6;request\_handler;;Next State = 5

5.1;6;request\_handler;eject\_out;1

5.1;6;request\_handler;;Next State = 0

5.1;1;amount\_in file;;inf

5.1;2;card\_number\_in file;;inf

5.1;3;pin\_in file;;inf

5.1;5;account\_balance;;Next State = 0

5.1;6;request\_handler;;Next State = 0

5.1;7;pin\_verifier;;Next State = 0

1. Goal: Invalid sequence of inputs causing the card to be ejected early.  
   00:00:01:000 Amount 150

00:00:03:000 CardNumber 111

00:00:05:000 Amount 150

Expected Results: Each time the Amount input is received there should be a corresponding Eject output since the pin has not been verified correctly.   
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;amount\_in file;;1

0;2;card\_number\_in file;;3

0;4;account\_balance;;Next State = 0

0;5;request\_handler;;Next State = 0

0;6;pin\_verifier;;Next State = 0

1;1;amount\_in file;out;150

1;1;amount\_in file;;4

1;5;request\_handler;;Next State = 5

0;5;request\_handler;eject\_out;1

0;5;request\_handler;;Next State = 0

3;2;card\_number\_in file;out;111

3;2;card\_number\_in file;;inf

3;6;pin\_verifier;;Next State = 1

5;1;amount\_in file;out;150

5;1;amount\_in file;;inf

5;5;request\_handler;;Next State = 5

0;5;request\_handler;eject\_out;1

0;5;request\_handler;;Next State = 0

0;1;amount\_in file;;inf

0;2;card\_number\_in file;;inf

0;4;account\_balance;;Next State = 0

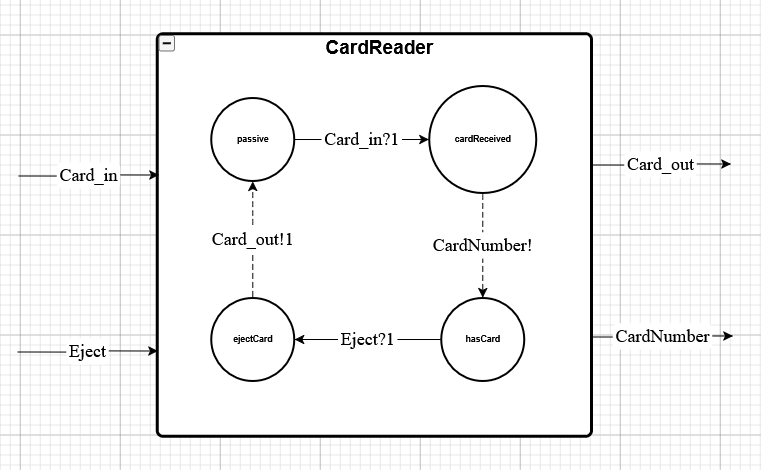
0;5;request\_handler;;Next State = 0

0;6;pin\_verifier;;Next State = 1

**Atomic Models:**

Card Reader:

Description: The card reader model takes a card as input and generates a card number. The card can be ejected at any time if an eject input is received, but only if a card is still present. This means the model will only output a card if an eject is received after a card input is received and no previous eject has occurred since the card input.



Formal Specification

Tests:

Previous Tests:

1. Goal: Successful card inserted followed by card ejected

00:00:01:000 Card\_in 1

00:00:03:000 Eject 1

Expected Results: After the Card\_in input there should be an CardNumber output generated. Then the Eject input is sent which should be quickly followed by a Card\_out output.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;eject\_in file;;3

0;2;card\_in file;;1

0;3;counter model;;Next State = 0

1;2;card\_in file;out;1

1;2;card\_in file;;inf

1;3;counter model;;Next State = 1

2;3;counter model;card\_number\_out;111

2;3;counter model;;Next State = 2

3;1;eject\_in file;out;1

3;1;eject\_in file;;inf

3;3;counter model;;Next State = 3

3;1;eject\_in file;;inf

3;2;card\_in file;;inf

3;3;counter model;;Next State = 3

New Tests:

1. Goal: Ejecting without card present

00:00:01:000 Eject 1

Expected Results: The model should ignore the input since there is no card to eject.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;eject\_in file;;1

0;2;counter model;;Next State = 0

1;1;eject\_in file;out;1

1;1;eject\_in file;;inf

1;2;counter model;;Next State = 0

2;2;counter model;;Next State = 0

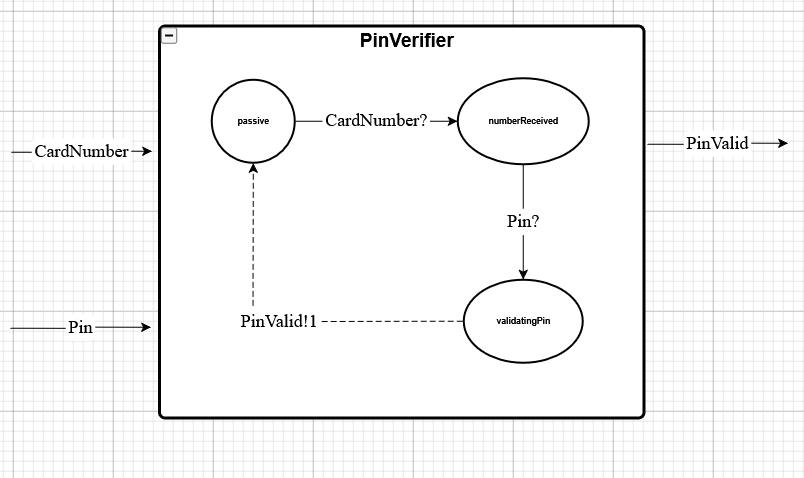
3;2;counter model;;Next State = 0

3;1;eject\_in file;;inf

3;2;counter model;;Next State = 0

Pin Verifier:

Description: The model received a pin input and compares it against the expected pin value. The output is produced depending on if the received value matches the expected value so the model acts as a simple comparator. The pin is only checked if a card number has already been received (otherwise there’s no pin to check against). As a simplification of the model, only one expected pin is used but in theory this model could be expanded to retrieve a specific pin based on the card number input.



Formal Specification:

If (Pin == expected\_pin) {

State = passive;

} else {

State = numberReceived;

}

validatingPin {

If (Pin == expected\_pin) {

PinValid->true;

} else {

PinValid->false;

}

Tests:

Previous Tests:

1. Goal: Successful pin received  
   00:00:01:000 CardNumber 111

00:00:03:000 Pin 1234

Expected Results: An output of 1 is produced in the PinValid port  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;pin\_in file;;3

0;2;card\_number\_in file;;1

0;3;pin\_verifier model;;Next State = 0

1;2;card\_number\_in file;out;111

1;2;card\_number\_in file;;inf

1;3;pin\_verifier model;;Next State = 1

3;1;pin\_in file;out;1234

3;1;pin\_in file;;inf

3;3;pin\_verifier model;;Next State = 2

3.5;3;pin\_verifier model;pin\_valid\_out;1

3.5;3;pin\_verifier model;;Next State = 0

3.5;1;pin\_in file;;inf

3.5;2;card\_number\_in file;;inf

3.5;3;pin\_verifier model;;Next State = 0

New Tests:

1. Goal: Pin received before card inserted

00:00:01:000 Pin 1234

Expected Results: No output is produced, input is ignored.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;pin\_in file;;3

0;2;pin\_verifier model;;Next State = 0

0;1;pin\_in file;;3

0;2;pin\_verifier model;;Next State = 0

1. Goal: Wrong pin received three times

00:00:01:000 Card\_in 1

00:00:03:000 Pin 0000

00:00:05:000 Pin 0001

00:00:07:000 Pin 0002

Expected Results: Each wrong pin gives an output of 0 in PinValid port and after the third wrong Pin input an output of 1 is produced in Eject port.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;pin\_in file;;3

0;2;card\_number\_in file;;1

0;3;pin\_verifier model;;Next State = 0

1;2;card\_number\_in file;out;111

1;2;card\_number\_in file;;inf

1;3;pin\_verifier model;;Next State = 1

3;1;pin\_in file;out;0

3;1;pin\_in file;;2

3;3;pin\_verifier model;;Next State = 2

3.5;3;pin\_verifier model;pin\_valid\_out;0

3.5;3;pin\_verifier model;;Next State = 1

5;1;pin\_in file;out;1

5;1;pin\_in file;;2

5;3;pin\_verifier model;;Next State = 2

5.5;3;pin\_verifier model;pin\_valid\_out;0

5.5;3;pin\_verifier model;;Next State = 1

7;1;pin\_in file;out;2

7;1;pin\_in file;;inf

7;3;pin\_verifier model;;Next State = 2

7.5;3;pin\_verifier model;pin\_valid\_out;0

7.5;3;pin\_verifier model;;Next State = 1

7.5;1;pin\_in file;;inf

7.5;2;card\_number\_in file;;inf

7.5;3;pin\_verifier model;;Next State = 1

Account Balance:

Description: The model received a request for a withdrawal and decides whether or not there is sufficient funds in the account for the transaction, then outputs a corresponding approval. The model also decrements the balance meaning the successive requests will deplete the account balance. The balance must be set to an initial value since the model does not currently support deposit transactions.

A diagram of a account balance

AI-generated content may be incorrect.

Formal Specification:

{

If (account\_balance >= Request) {

Account\_balance -= Request;

}

}

Tests:

Previous Tests:

1. Goal: Successful request approved (default balance is 1000)

00:00:01:000 Request 150

00:00:03:000 Request 320

Expected Results: Both request should result in an output of 1 produced in port Approved  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;request\_in file;;1

0;2;account\_balance model;;Next State = 0

1;1;request\_in file;out;150

1;1;request\_in file;;2

1;2;account\_balance model;;Next State = 1

1.1;2;account\_balance model;approved\_out;1

1.1;2;account\_balance model;;Next State = 0

3;1;request\_in file;out;320

3;1;request\_in file;;inf

3;2;account\_balance model;;Next State = 1

3.1;2;account\_balance model;approved\_out;1

3.1;2;account\_balance model;;Next State = 0

3.1;1;request\_in file;;inf

3.1;2;account\_balance model;;Next State = 0

New Tests:

1. Goal: Request amount too high

00:00:01:000 Request 1500

00:00:03:000 Request 800

00:00:05:000 Request 250

Expected Results: The first input is rejected for being too high of an amount and produced an output of 0 on port Approved. The second request is valid and produces an output of 1 on port Approved. The balance is now decreased by 800 and there is no longer sufficient funds for the third request so it produced an output of 0 in port Approved.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;request\_in file;;1

0;2;account\_balance model;;Next State = 0

1;1;request\_in file;out;1500

1;1;request\_in file;;2

1;2;account\_balance model;;Next State = 1

1.1;2;account\_balance model;approved\_out;1

1.1;2;account\_balance model;;Next State = 0

3;1;request\_in file;out;800

3;1;request\_in file;;2

3;2;account\_balance model;;Next State = 1

3.1;2;account\_balance model;approved\_out;0

3.1;2;account\_balance model;;Next State = 0

5;1;request\_in file;out;250

5;1;request\_in file;;inf

5;2;account\_balance model;;Next State = 1

5.1;2;account\_balance model;approved\_out;0

5.1;2;account\_balance model;;Next State = 0

5.1;1;request\_in file;;inf

5.1;2;account\_balance model;;Next State = 0

Request Handler:

Description: This model is responsible for coordinating where inputs should be forwarded to (e.g. an amount input is sent as a request output and later sent as a dispense output after received an approved input). The model also handles the state machine for ensuring inputs are received in the correct order (i.e. pin validated, request received, then request approved). If a request is denied or if a pin is invalid 3 times the model will output an eject signal.

A diagram of a system

AI-generated content may be incorrect.

Formal Specification:

Tests:

Previous Tests:

N/A (model did not previously exist)

New Tests:

1. Goal: Successful sequence of events

00:00:01:000 PinValid 1

00:00:04:000 Amount 150

00:00:05:000 Approved 1

Expected Results: After the Amount input is received an output of 150 in produced in port Request. 1 is received in port Approved which causes an output of 150 in port Dispense.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;pin\_valid\_in file;;1

0;2;approved\_in file;;5

0;3;amount\_in file;;4

0;4;request\_handler model;;Next State = 0

1;1;pin\_valid\_in file;out;1

1;1;pin\_valid\_in file;;inf

1;4;request\_handler model;;Next State = 1

4;3;amount\_in file;out;150

4;3;amount\_in file;;inf

4;4;request\_handler model;;Next State = 2

4;4;request\_handler model;request\_out;32767

4;4;request\_handler model;;Next State = 3

5;2;approved\_in file;out;1

5;2;approved\_in file;;inf

5;4;request\_handler model;;Next State = 4

5;4;request\_handler model;dispense\_out;32767

5;4;request\_handler model;;Next State = 0

5;1;pin\_valid\_in file;;inf

5;2;approved\_in file;;inf

5;3;amount\_in file;;inf

5;4;request\_handler model;;Next State = 0

1. Goal: Request denied

00:00:01:000 PinValid 1

00:00:04:000 Amount 1500

00:00:05:000 Approved 0

Expected Results: After the Amount input is received an output of 1500 in produced in port Request. 0 is received in port Approved which causes an output of 1 in port Eject.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;pin\_valid\_in file;;1

0;2;approved\_in file;;5

0;3;amount\_in file;;4

0;4;request\_handler model;;Next State = 0

1;1;pin\_valid\_in file;out;1

1;1;pin\_valid\_in file;;inf

1;4;request\_handler model;;Next State = 1

4;3;amount\_in file;out;1500

4;3;amount\_in file;;inf

4;4;request\_handler model;;Next State = 2

4;4;request\_handler model;request\_out;32767

4;4;request\_handler model;;Next State = 3

5;2;approved\_in file;out;0

5;2;approved\_in file;;inf

5;4;request\_handler model;;Next State = 5

5;4;request\_handler model;eject\_out;1

5;4;request\_handler model;;Next State = 0

5;1;pin\_valid\_in file;;inf

5;2;approved\_in file;;inf

5;3;amount\_in file;;inf

5;4;request\_handler model;;Next State = 0

Cash Manager:

Description: This model is responsible for taking an approved withdrawal request (in the form of a dispense input) and outputting the correct amount of cash. In a real life system this module would interact with the physical cash dispenser to distribute the correct number of bills corresponding the dispense amount.

A diagram of a cash manager

AI-generated content may be incorrect.

Formal Specification:

Tests:

Previous Tests:

1. Goal: Successful dispensing of cash

00:00:01:000 Dispense 150

Expected Results: An output is produced at port Cash with the same value that was received in the input port.  
Actual Results:

sep=;

time;model\_id;model\_name;port\_name;data

0;1;dispense\_in file;;1

0;2;cash\_manager model;;Next State = 0

1;1;dispense\_in file;out;150

1;1;dispense\_in file;;inf

1;2;cash\_manager model;;Next State = 1

2;2;cash\_manager model;cash\_out;150

2;2;cash\_manager model;;Next State = 0

2;1;dispense\_in file;;inf

2;2;cash\_manager model;;Next State = 0

New Tests:

The model is simple enough that no new tests are added.