SYSC 5104/HCIN 5405: METHODOLOGIES FOR DISCRETE EVENT MODELLING AND SIMULATION

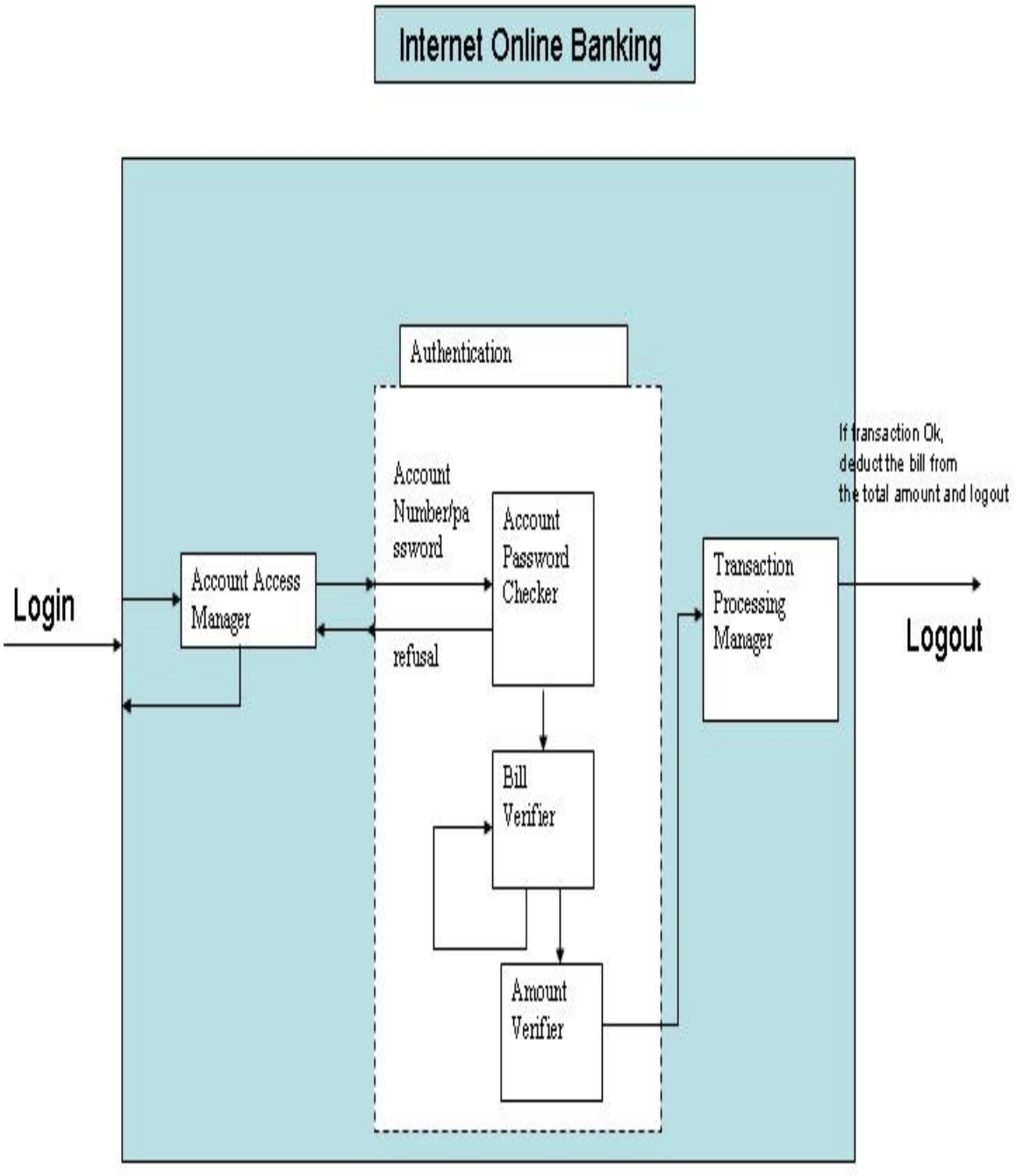
Assignment 1: Internet Online Banking System

Rishabh Sudhir Jiresal 101167141
Carleton University

Table of Content

Conceptual Model Description (Part 1):

Following is what initialized planned and submitted as a proposal:



Problem/Solution:

An internet online banking system is modeled. This is a simplified version of a real commercial online banking system. Modeling is done on the account number and password verification.

Conceptual model:

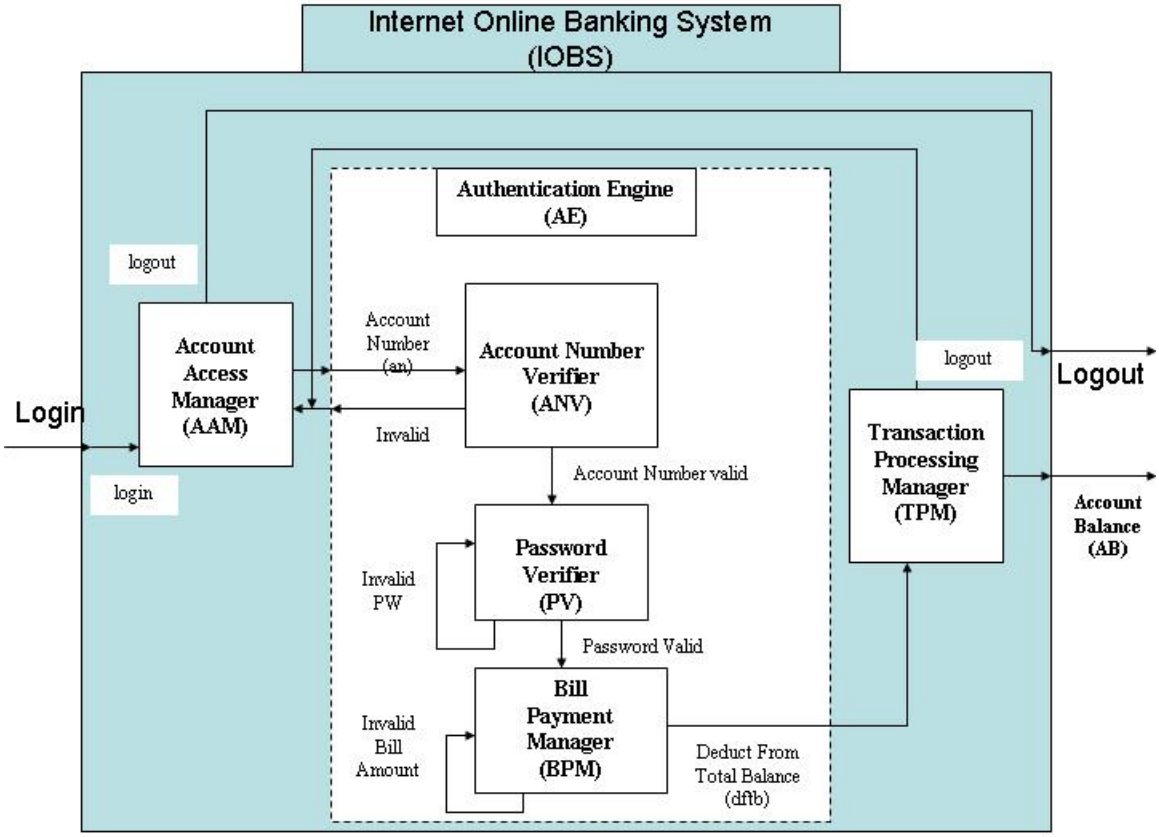
This model conceives three sub models; one coupled model has three atomic models Following components are involved in this modeling.

- **Account Access Manager**: is to manage taking the input of online banking account and password, (which is passed to the Authentication Manager), and take the result from Authentication Manger to make further decision. If it fails, it will keep asking for input of account number and password
- **Authentication Manager**: get the input from Account Access Manager and do the verification and it has the following atomic models
    - **Account number/PIN verifier Manager** : for simplicity, we are not going to predefine a database which store the matching account number and PIN number. A random function will return Yes or No, to simulate the PIN is acceptable or not
    - **Bill Verifier Manager**: checks the bill (the associated institute) has already listed in the account. If not, will ask to pay another bill. (for simplicity, adding to bill payment is not implemented), and random function is implemented to simulate the bill is existing
    - **Amount Verifier Manager**: check the amount in your amount with the bill amount about to pay, if enough to cover, function will return TRUE.
- Transaction Processing Manager: In real life, the input of this block is to ask user to confirm and click “Do you really want to pay this bill”. Here, I am just use the output of Bill Verifer and Amount Verifier. If both validation are passed, then it will perform the transaction and output to logout the session

State Variables:

Account Access Manager: sigma, account number, phase.
Amount Verifier Manager: sigma, amount, Number_of_trials.
Bill Verifier: sigma, bill number, Number_of_trials
PIN Verifier: sigma, PIN, Number_of_trials, accountNumber
Transaction Processing:

And after a fine tuning up, a slight modification has been made, and the updated model looks like the following.



Model Specification(Part 2):

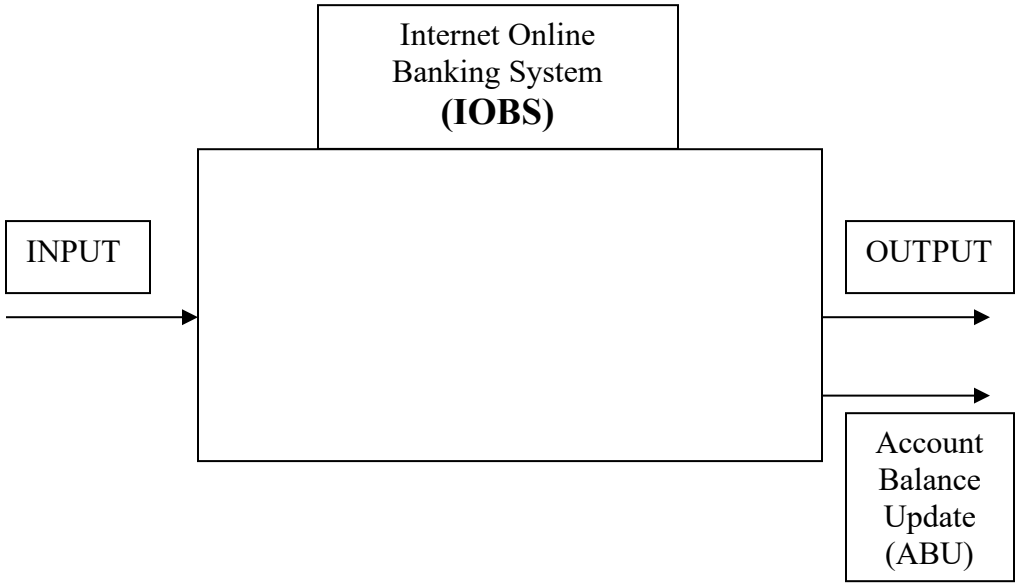
Terminology Acronym:

IOBS: Internet Online Banking System
AAM: Account Access Manager
AE: Authentication Engine
ANV: Account Number Verifier
PV: Password Verifier
BPM: Bill Payment Manager
TPM: Transaction Processing Manager

Coupled Model(s):

Top level Coupled Model is the "Internet Online Banking":

It models the online web bank account access, and bill payment transaction operation in a simpler version vs. real-life's.



Port definitions:

IN PORTS: LOGIN
OUT PORTS: LOGOUT

IOBS = <X,Y,D,EIC,EOC,IC,SELECT>

Where:
X = {LOGIN}

Y = {LOGOUT, ABU}

D = {AAM, TPM, AE}

EIC = {IOBS.login, AAM.login}

EOC = {(AAM.logout, IOBS.logout),
(TPM.logout, IOBS.logout)}

IC = { (AAM.an_out, AE.an_in),
(AE.ian, AAM.ian_in),
(AE.dftb_out, TPM.in) }

SELECT: { AAM, TPM} = TPM
{AE, TPM} = TPM

To test this top level modeling, input data can be feed into the system, and output data will be generated by Cadmium, therefore data can be viewed and interpreted.
Here, input passes two parameters which are {login, invalid_account_number} in the form a custom data type (Message_t) which has two variables {valid, invalid}.

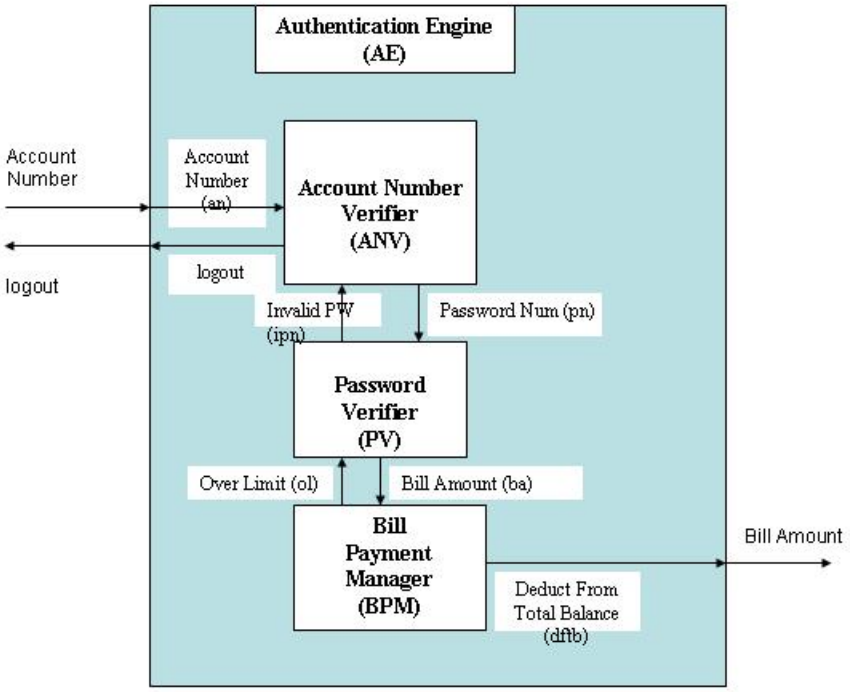
Input can be: (where time is specified and in put "1" means login event happens)

00:00:10 1 0
00:04:20 1 0
00:10:30 1 0 //Here at time = 10 mins and 30 seconds, there is a login attempt given by 1 and there is no invalid account number which is given by 0.

Output:
As mentioned, this is the generated data, and output should be "1", which indicates the online banking session has been closed.
If the bill payment transaction is processed successful, output will also update the account balance. (The output will be in the form of Message_t and will have two values such as {664, 1} in which 664 will be the updated balance and 1 will be the logout status which shows that the system has been logged out)

Authentication Engine (AE):

This is a sub-model of Internet Online Banking System(IOBS). It gets input from the Account Access Manager (AAM), the input is an account number which is randomly generated by the AAM,. The account number then pass to Account Number Verifier (ANV), for simplicity, ANV has a random generator, generates a value of TRUE or FALSE. If TRUE, meaning the inputted Account number is a valid account number, and ANV will use another random generator to generates a Password word. If it is FALSE, then it will send a logout signal back to AAM to end the login session. The Password Verifier gets the Password Number (pn) as input, and it has a random generator to generates TRUE or FALSE. TRUE means password matches the valid account number that has just been inputted, and then it will random generates a value representing the Bill payment amount. If FALSE, it will send an Invalid PW (ipn) signal back to ANV to generate another password number. The Bill payment Manager (BPM) gets an numeric bill amount number as an input, and compares the current amount balance (Initially, we assume the account has a balance of $3000), it verifies the bill payment amount with the balance, if it is smaller or equal, then it send the bill amount to TPM as an output. If is bigger than the balance, it send an Over Limit (ol) signal back to PV to generate another bill amount.



Input Port: Amount Number (an)
Output Ports: Bill Amount(ba)

Formal specifications:
AE = <X,Y,D,EIC,EOC,IC,SELECT>
Where:
X = {AccountNumber}
Y = { BillAmount }
D = { ANV, PV, BPM}
EIC = { (AE.in, ANV.an) }
EOC = { (BPM.ba, AE.ba)}
IC = { (PV.pn_in, ANV.pn_out),
(BPM.ba_in, PV.ba_out),
(ANV.ipn_in, PV.ipn_out),
(PV.ol_in, BPM.ol_out) }

SELECT: Priority order (Descending) : ANV, PV, BPM

To test this model, we can apply the input (account number), and let Cadmium generate it corresponding output and have the data result interpreted.

Input:
00:00:50 132123
00:01:20 142232

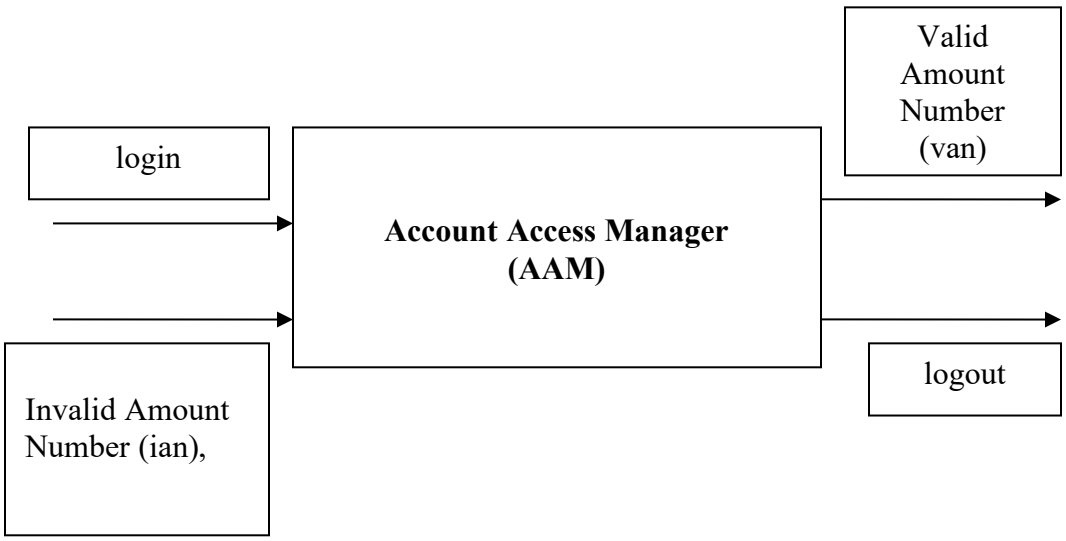
Output:
Logout signal will be sent out if the account number is not valid. (This will be sent by AccountNumberVerifier if the account number is not valid)
Else, a valid bill payment amount will be sent out

Atomic Models:

A random processing time with its corresponding distribution function specified in *.MA file is associated with each of the atomic models list as following.

Account Access Manager (AAM)

This model gets a user login signal as its input; it will then generate a random online bank Account Number, and pass it to Authentication Engine (AE). This model can also receive the Invalid Account Number (ian) as an input, and then will send an output signal out.



Definition of Ports:
Input Ports: login, ian
Output Ports: logout, van

Formal specification:
AAM = <X, Y, S, δ_{int}, δ_{ext}, λ, ta>
Where:

X = {
login | default is value of 0 or 1. 0 means that user has not logged in, a 1 means user just enters to the system ,

ian | default value is a value of 0 or 1. 0 means it is either, user enters the valid account number, or system is expecting to be logged in; value of 1 means the user entered an invalid account number, and AAM is expected to logout. }

Y = { logout, van}

S = { Phase, sigma, model_active, randAccountNumber, login_status, ian_status,}

δ_{ext} (S,e,X) {

```
        Case port
          login:
                Case Phase
                        Passive:
                          Sigma = generate random Processing Time;
                          Phase = busy;
                          randAccountNumber= Generate random number of a six
                        digital integer number
                          login_status = 1;
                          ian_status = 0;

                        busy:
                           // Ignore the request;

          ian:
                Case Phase
                        busy:
                          Sigma = generate random Processing Time;
                          login_status = 0;
                          ian_status = 1;

                        Passive:
                           // Ignore the request;
```

}

λ (S) {

```
        case phase
                        busy:
                                If   login_status ( is equal to 1)
                                     // send randAccountNumber to out
                                      Send (van_port,
                                                AAM. randAccountNumber);
                                Else (ian_status = = 1)
                                    // a logout request
                                    Send (logout_port, logout);

                        Passive: // Should not happen.
```

}

δ_{int} (S) {

```
        case phase
            busy:
                sigma = infinity;
                If  ian_status = = 1 and no login_status
                    phase = passive;
            passive: //never happens
    }
```

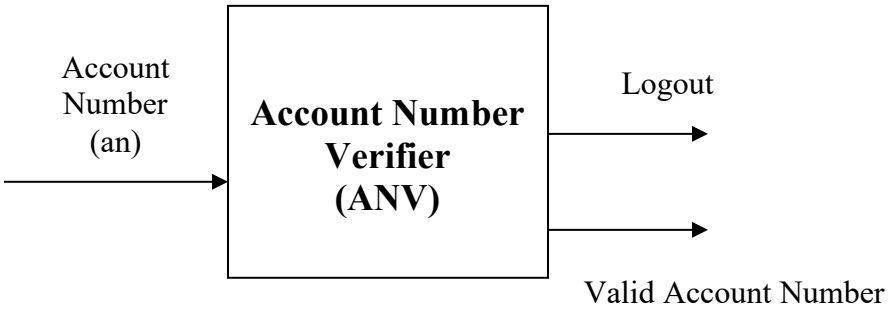
The logic and functionality of this model block can be tested by inputting event of login and Invalid_Account_Number(ian), output can be generated and used to interpreted. The input is given in the form of Message_t which has two parameters named {valid, invalid} which here will be {login, ian}.

Input:
00:00:40 1 0 //login and valid account number will produce an account number
00:01:50 1 1 //input with invalid account number will logout
00:02:40 0 1 //invalid input will not output anything and it will be ignored
00:03:50 1 0

Output: (respective to input)
An account number of any six digit integer number is sent out to Account Number output port
Value of 1 at logout output port

Account Number Verifier (ANV)

This model get the input (Account Number), and then triggers a random generated to randomly verifies it is a valid number or not , if it is invalid Account Number, it sends the logout_request to logout_output port. Else if it is a valid Account Number, it then sends the ValidAccountNumber signal to the ValidAccountNumber output port.



Definition of Ports:
Input Ports: AccountNumber_IN
Output Ports: { ValidAccountNumber_OUT , logout_OUT} i.e. {output_OUT}
(Message_t data typed)

Formal specification:
ANV= <X, Y, S, δ_{int}, δ_{ext}, λ, ta>
Where:

X = {
AccountNumber | any six digit integer number }

Y = { output_OUT }

S = { Phase, sigma, randAccountNumberValid, model_active }

δ_{ext} (S,e,X) {

```
        Case port
          AccountNumber:
              Case Phase
                    Passive:
                      Sigma = generate random Processing Time;
                      Phase = busy;
                      randAccountNumberValid = Generate random number
                                                between 0 and 1;

                    busy:
                      // Ignore the request;
}
```

```
λ (S) {
                    case phase
                        busy:
                            If   randAccountNumberValid
                                //send out to its port
                                Send (ValidAccountNumber_OUT port, 1);
                            Else
                                // logout signal
                                Send (logout_OUT, logout);

                        Passive: // Should not happen.
    }

δint (S) {
                    case phase
                        busy:
                            phase = passive;
                            sigma = infinity;
                        passive: //never happens
}
```

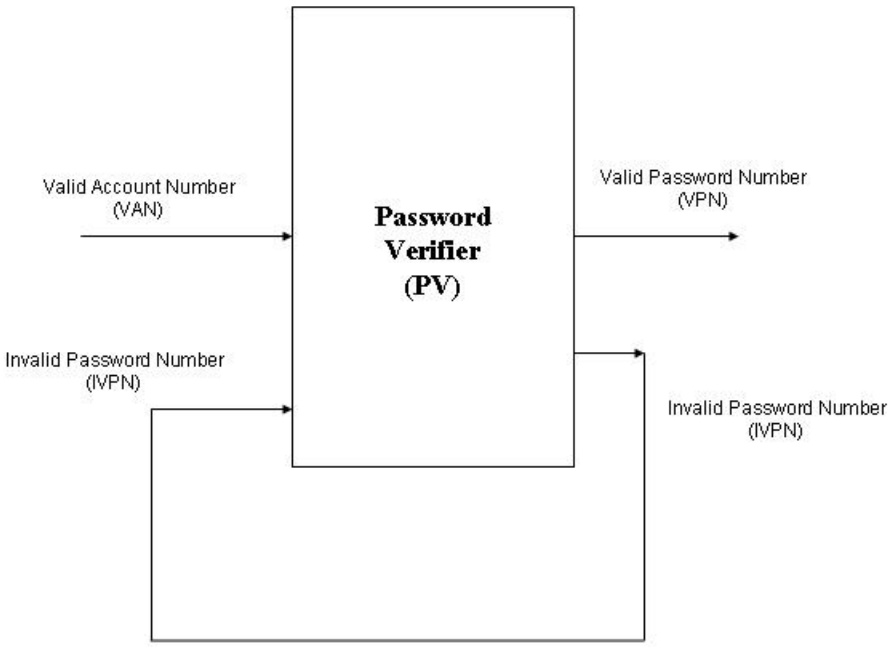
The logic and functionality of this model block can be tested by inputting event of Login and Account Number(the randomly generated six digit number), output is generated by Cadmium builder and used to interpreted.

Input:
00:00:50 132123 //random account number given as input to the model
00:01:20 142232

Output:
Value of {1,0} at output_OUT //This shows that the account number is valid
Value of {0,1} at output_OUT //This shows that the account number is invalid

Password Verifier (PV)

The model has two inputs (VAN_IN and IVPN_IN), In the initial state, both input signals are "0".
When receiving a VAN_IN, the model has a random generator, simulating user's password has been accepted or not. A randomly generated number of "1" or "0" where "1" means, user password number is valid and a signal VPN_OUT is sent out to output port, and "0" means invalid password number, and a IVPN_OUT signal is sent to IVPN_output port.
When IVPN output port is sent out, it also enable the IVPN_input port. When receiving IVPN_IN signal, this model will regenerate another value to indicate password number is acceptable. The model will loop till an VPN signal is sent out.



Definition of Ports:
Input Ports: VAN_IN, IVPN_IN //Message_t format {valid, invalid} i.e. **Input_IN**
Output Ports: VPN_OUT, IVPN_OUT //Message_t format {valid, invalid} i.e. **Output_OUT**

Formal specification:
PV= <X, Y, S, δ_{int}, δ_{ext}, λ, ta>
Where:

X = { Input_IN in the form of

(VAN_IN | a value of either “0” or “1”, where 0 means that no Valid account number has been enter, and “1” means account number has been validated. And “1” will trigger the generator to generate a Password validation result number.
IVPN_IN | a value of either “0” or “1”, where 1 will trigger the generator to regenerate a number) }

Y = { Output_OUT in the form of (VPN_OUT , IVPN_OUT)}

S = { Phase, sigma, randPasswordNumber, valid}

δ_{ext} (S,e,X) {

```
        Case port
          Input_IN(1,1)|| Input_IN(1,0):
                Case Phase
                      Passive:
                        Sigma = generate random Processing Time;
                        Phase = busy;
                        randPasswordNumber = Generate random number (0
                              or 1);
                      if randPasswordNumber is 1
                      valid = true;
                      else if  randPasswordNumber is 0
                      valid =false;
```

}

λ (S) {

```
                case phase
                      busy:
                            If   valid (is equal to true)
                                  //send out to its port
                                  Send (output_OUT,(1 , 2));
                            Else
                                  //send out to its port
                                  Send (output_OUT (1,1));



                      Passive: // Should not happen.
    }
```

δ_{int} (S) {

```
                case phase
                      busy:
                            phase = passive;
                            sigma = infinity;
```

passive: //never happens
}

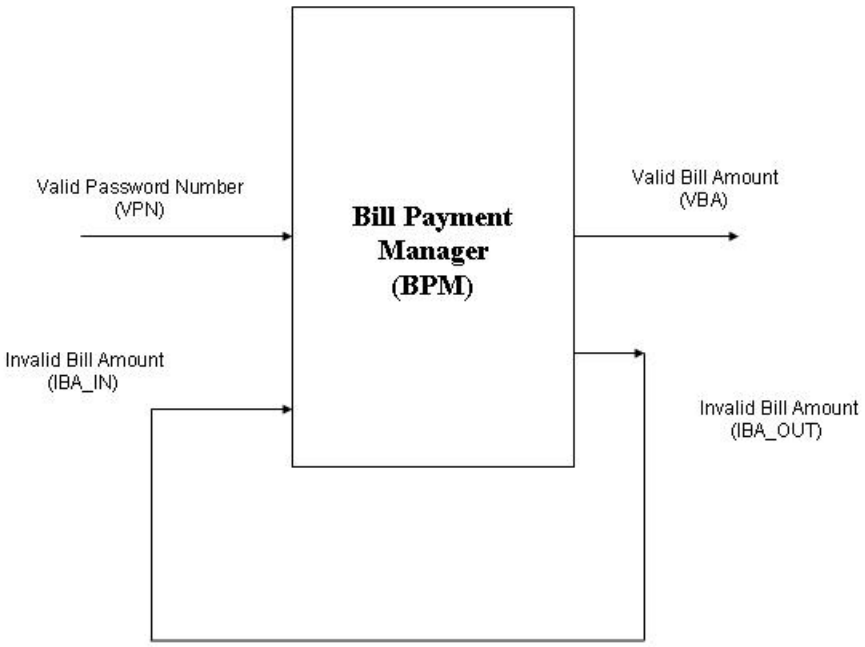
The logic and functionality of this model block can be tested by inputting event of VAN (0 or 1), output is generated by CD++ builder and used to interpreted. **In Cadmium, the output will be given in the format of Message_t as well as for the input Message_t will be used. During the first input, the second parameter of the Message_t will be ignored since it is looped and until valid password is achieved.**

Input:
00:01:50 1 1
00:02:50 1 1
00:03:50 1 1
00:04:50 1 1 //The first input 1 is the valid account number which is output by the ANV and the second input is 1 which will be ignored for the first time and will be internally randomly generated and the model will be looped until we get a valid output

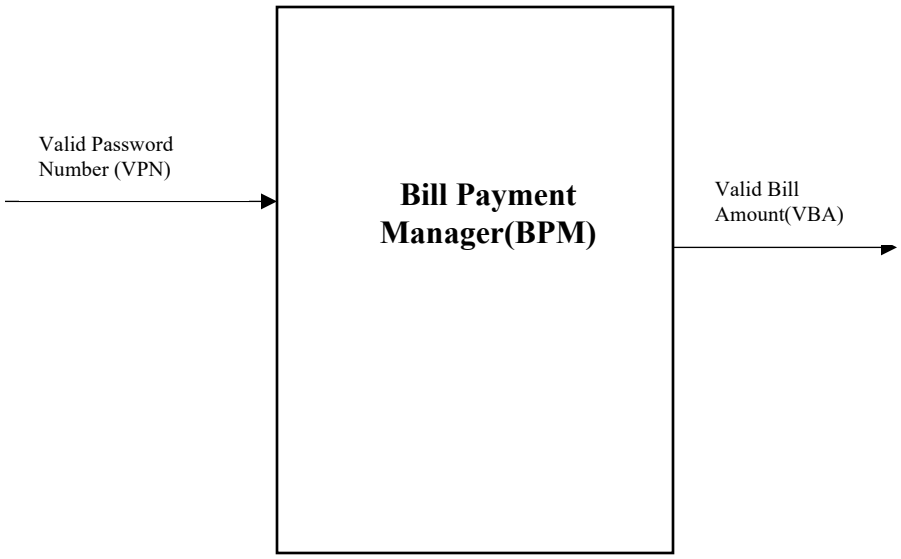
Output:
Value of {1, 2} at output port if valid password
Value of {1, 1} at output port if invalid password and the model will loop until they get a valid password

Bill Payment Manager (BPM)

Similar to the previous model, this model has two inputs, In the initial state, both input signals are initialized to be “0”.
When receiving a Valid Password Number(VPN_IN) signal from the input port, it triggers the model’s internal random generator, sends out a randomly generated number between “1” to “4000” (initially the account balance is $3000). It then compared the account balance, if it is equal or smaller than a signal Valid Bill Amount (VBA_OUT) to VBA_OUT port, and if the compared result is larger, then it sends a IBA_out signal is sent to IBA_out port.
When IBA_Out is sent out, it also enable the IBA_IN port. When receiving IBA_IN signal, this model will regenerate another bill amount value to check if it is valid amount or not. The model will loop till a VBA_OUT signal is sent out.



But due to the increasing complexity of the model, let us consider that the amount generator will not generate an invalid amount and even if it is generated, it will be discarded, and a valid amount will be generated automatically. This is done to remove the looping of the model which was responsible to increase the complexity.
So now, the model will have one input of data type Message_t and one output of data type integer which will have the valid bill amount.



Definition of Ports:
Input Ports: Input_IN
Output Ports: VBA_OUT

Formal specification:
BPM = <X, Y, S, δ_{int}, δ_{ext}, λ, ta>
Where:

X = { Input_IN
(VPN_IN | a value of either "0" or "1", where 0 means that no Valid Password number has been enter, and "1" means Password number has been validated. And "1" will trigger the generator to generate a numeric bill payment amount, it and compares it with current account balance.
The second value in the input will be ignored)
}

Y = { VBA_OUT}

S = { Phase, sigma, randBilAmount, current_account_balance}

δ_{ext} (S,e,X) {
Case port
Input_IN(valid = 1 && invalid != 1):
Case Phase
Passive:
Sigma = generate random Processing Time;
Phase = busy;
randBilAmount = Generate random number (1 to 4000);

```
                    if the randBillAmount is not appropriate(i.e. greater than
                    3000)
                    fix the value
                    busy:
                      // Ignore the request;


}

λ (S) {
                case phase
                    busy:
                                //send out to its port
                                Send (VBA_OUT, randBilAmount);



                    Passive: // Should not happen.
        }

δint (S) {
                case phase
                    busy:
                            phase = passive;
                            sigma = infinity;
                    passive: //never happens
}
```

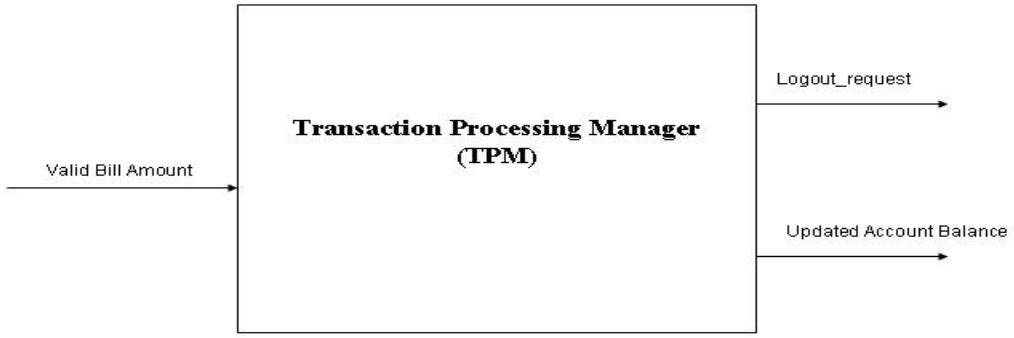
The logic and functionality of this model block can be tested by inputting event of VPN_IN (0 or 1), output is generated by CD++ builder and used to interpreted.

Input:
00:00:10 1 0 //Valid Password input and second parameter is ignored
00:00:40 1 0
00:01:20 1 1
00:02:00 1 0

Output:
Value of “Valid_Bill_Amount” at VBA output port

Transaction Processing Manager (TPM)

This models detects the signal from the input port of Valid Bill Amount(VBA_IN), and deduct the amount from the current and amount balance, and sent the updated balance to the Updated Account Balance (UAB_OUT) port, and it also sends the logout signal to the LOGOUT port.



Definition of Ports:
Input Ports: VBA_IN
Output Ports: Output_OUT(UAB_OUT, LOGOUT) :- Message_t custom datatype used

Formal specification:
TPM = <X, Y, S, δ_{int}, δ_{ext}, λ, ta>
Where:
X = { VBA_IN | VBA is a value smaller to the current balance 0- 3000 }
Y = { UAB_OUT | the new amount cash balance (0-3000),
LOGOUT | a signal to logout}
S = { Phase, sigma, current_account_balance, billAmount }

δ_{ext} (S,e,X) {
Case port
VBA_IN:

Case Phase
Passive:
Sigma = generate random Processing Time;
Phase = busy;

```
                    current_account_balance = current_account_balance –
                        billAmount;

                busy:
                  // Ignore the request;

        }


        λ (S) {
                case phase
                        busy:
                                        // print the account balance
                                        Send (UAB_OUT,
current_account_balance)
                                        Send (LOGOUT_Port, LOGOUT)

                        Passive: // Should not happen.
        }


        δint (S) {
                case phase
                        busy:
                                phase = passive;
                                sigma = infinity;
                        passive: //never happens
                }
```

Testing can be done by inputting one or more events at the input port and observing the output:

Input:
00:00:40 200

Output:
2800 at UAB_OUT port
1 at LOGOUT port

Model Testing and Verification(Part 3):

Testing has been done on each of the following five Atomic Models as well as the two coupled Models:

Atomic Models:

Account Access Manager (AAM):

AccountAccessManager Inputs:

Input:
00:00:40 1 0 //login and valid account number will produce an account number
00:01:50 1 1 //input with invalid account number will logout
00:02:40 0 1 //invalid input will not output anything and it will be ignored
00:03:50 1 0

Account Access Manager Output Messages:

00:00:40:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 0**}] generated by model input_reader_login
00:00:50:000
[AccountAccessManager_defs::van: {**840187**}, AccountAccessManager_defs::logout: {}] generated by model AAM1
00:01:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 1**}] generated by model input_reader_login
00:02:00:000
[AccountAccessManager_defs::van: {}, AccountAccessManager_defs::logout: {**1**}] generated by model AAM1
00:02:40:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**0 1**}] generated by model input_reader_login
00:03:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 0**}] generated by model input_reader_login
00:04:00:000
[AccountAccessManager_defs::van: {**783099**}, AccountAccessManager_defs::logout: {}] generated by model AAM1

Generated Output States:

00:00:00:000
State for model input_reader_login is next time: 00:00:00:000
State for model AAM1 is login_status: 0 & ian_status 0 Account number: 0
00:00:00:000
State for model input_reader_login is next time: 00:00:40:000
State for model AAM1 is login_status: 0 & ian_status 0 Account number: 0

00:00:40:000
State for model input_reader_login is next time: 00:01:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 840188
00:00:50:000
State for model input_reader_login is next time: 00:01:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 840188
00:01:50:000
State for model input_reader_login is next time: 00:00:50:000
State for model AAM1 is login_status: 1 & ian_status 1 Account number: 394383
00:02:00:000
State for model input_reader_login is next time: 00:00:50:000
State for model AAM1 is login_status: 1 & ian_status 1 Account number: 394383
00:02:40:000
State for model input_reader_login is next time: 00:01:10:000
State for model AAM1 is login_status: 1 & ian_status 1 Account number: 394383 ***//This account number will not be given to the output since the ian_status is 1 and the model will be logged out***
00:03:50:000
State for model input_reader_login is next time: inf
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099
00:04:00:000
State for model input_reader_login is next time: inf
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099

Account Number Verifier (ANV):

AccountNumberVerifier Inputs:

Input:
00:00:50 132123 //random account number given as input to the model
00:01:20 142232

Account Number Verifier Output Messages:

Value of {1,0} at output_OUT //This shows that the account number is valid
Value of {0,1} at output_OUT //This shows that the account number is invalid

00:00:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {**132123**}] generated by model input_reader
00:01:00:000
[AccountNumberVerifier_defs::output_OUT: {**1 0**}] generated by model ANV1***//{1,0} status is the valid account number status hence the system transfers the number status further***
00:01:20:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {**142232**}] generated by model input_reader
00:01:30:000
[AccountNumberVerifier_defs::output_OUT: {**0 1**}] generated by model ANV1 ***//{0,1} status is the logout status hence the system is halted***

Generated output states:

00:00:00:000
State for model input_reader is next time: 00:00:00:000
State for model ANV1 is account number valid: 0
00:00:00:000
State for model input_reader is next time: 00:00:50:000
State for model ANV1 is account number valid: 0
00:00:50:000
State for model input_reader is next time: 00:00:30:000
State for model ANV1 is account number valid: 1
00:01:00:000
State for model input_reader is next time: 00:00:30:000
State for model ANV1 is account number valid: 1
00:01:20:000
State for model input_reader is next time: inf
State for model ANV1 is account number valid: 0
00:01:30:000
State for model input_reader is next time: inf
State for model ANV1 is account number valid: 0

Password Verifier (PV):

PasswordVerifier Inputs:

00:01:50 1 1
00:02:50 1 1
00:03:50 1 1
00:04:50 1 1 //The first input 1 is the valid account number which is output by the ANV and the second input is 1 which will be ignored for the first time and will be internally randomly generated and the model will be looped until we get a valid output

Password Verifier Output Messages:
Value of {1, 2} at output port if valid password
Value of {1, 1} at output port if invalid password and the model will loop until they get a valid password

00:01:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 1**}] generated by model input_reader
00:02:00:000
[PasswordVerifier_defs::Output_OUT: {**1 2}**] generated by model PV1
00:02:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 1**}] generated by model input_reader
00:03:00:000
[PasswordVerifier_defs::Output_OUT: {**1 1**}] generated by model PV1 ***//This Happened because the model looped due to invalid password***
00:03:10:000
[PasswordVerifier_defs::Output_OUT: {**1 2**}] generated by model PV1 ***//Here the model created valid password and gave this output***
00:03:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 1**}] generated by model input_reader
00:04:00:000
[PasswordVerifier_defs::Output_OUT: {**1 2**}] generated by model PV1
00:04:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 1**}] generated by model input_reader
00:05:00:000
[PasswordVerifier_defs::Output_OUT: {**1 2**}] generated by model PV1

Generated output states:
00:00:00:000
State for model input_reader is next time: 00:00:00:000
State for model PV1 is password number: 0
00:00:00:000
State for model input_reader is next time: 00:01:50:000

State for model PV1 is password number: 0
00:01:50:000
State for model input_reader is next time: 00:01:00:000
State for model PV1 is password number: 1
00:02:00:000
State for model input_reader is next time: 00:01:00:000
State for model PV1 is password number: 1
00:02:50:000
State for model input_reader is next time: 00:01:00:000
State for model PV1 is password number: 0
00:03:00:000
State for model input_reader is next time: 00:01:00:000
State for model PV1 is password number: 1
00:03:10:000
State for model input_reader is next time: 00:01:00:000
State for model PV1 is password number: 1
00:03:50:000
State for model input_reader is next time: 00:01:00:000
State for model PV1 is password number: 1
00:04:00:000
State for model input_reader is next time: 00:01:00:000
State for model PV1 is password number: 1
00:04:50:000
State for model input_reader is next time: inf
State for model PV1 is password number: 1
00:05:00:000
State for model input_reader is next time: inf
State for model PV1 is password number: 1

Bill Payment Manager (BPM):

BillPaymentManager Inputs:

00:00:10 1 0
00:00:40 1 0
00:01:20 1 1
00:02:00 1 0

BillPaymentManager output messages:

00:00:00:000
00:00:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {}] generated by model input_reader
00:00:10:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {1 0}] generated by model input_reader
00:00:20:000
[BillPaymentManager_defs::VBA_OUT: {**1384**}] generated by model BPM1
00:00:40:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {1 0}] generated by model input_reader
00:00:50:000
[BillPaymentManager_defs::VBA_OUT: {**2887**}] generated by model BPM1
00:01:20:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {1 1}] generated by model input_reader
00:02:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {1 0}] generated by model input_reader
00:02:10:000
[BillPaymentManager_defs::VBA_OUT: {**778**}] generated by model BPM1

Generated output states:

00:00:00:000
State for model input_reader is next time: 00:00:00:000
State for model BPM1 is bill amount: 0Invalid bill? 0
00:00:00:000
State for model input_reader is next time: 00:00:10:000
State for model BPM1 is bill amount: 0Invalid bill? 0
00:00:10:000
State for model input_reader is next time: 00:00:30:000
State for model BPM1 is bill amount: 1384Invalid bill? 0
00:00:20:000
State for model input_reader is next time: 00:00:30:000
State for model BPM1 is bill amount: 1384Invalid bill? 0
00:00:40:000

State for model input_reader is next time: 00:00:40:000
State for model BPM1 is bill amount: 2887Invalid bill? 0
00:00:50:000
State for model input_reader is next time: 00:00:40:000
State for model BPM1 is bill amount: 2887Invalid bill? 0
00:01:20:000
State for model input_reader is next time: 00:00:40:000
State for model BPM1 is bill amount: 2887Invalid bill? 0
00:02:00:000
State for model input_reader is next time: inf
State for model BPM1 is bill amount: 778Invalid bill? 0
00:02:10:000
State for model input_reader is next time: inf
State for model BPM1 is bill amount: 778Invalid bill? 0

Transaction Processing Manager (TPM):

TransactionProcessManager Inputs:

00:00:40 200
00:01:10 30
00:01:50 45

TransactionProcessManager output messages:

00:00:00:000
00:00:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {}] generated by model input_reader
00:00:40:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {**200**}] generated by model input_reader
00:00:50:000
[TransactionProcessManager_defs::Output_OUT: {**2800 1**}] generated by model TPM1
00:01:10:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {**30**}] generated by model input_reader
00:01:20:000
[TransactionProcessManager_defs::Output_OUT: {**2970 1**}] generated by model TPM1
00:01:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {**45**}] generated by model input_reader
00:02:00:000
[TransactionProcessManager_defs::Output_OUT: {**2955 1**}] generated by model TPM1

Generated output states:

00:00:00:000
State for model input_reader is next time: 00:00:00:000
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:00:00:000
State for model input_reader is next time: 00:00:40:000
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:00:40:000
State for model input_reader is next time: 00:00:30:000
State for model TPM1 is bill amount: 200Current Account Balance: 2800
00:00:50:000
State for model input_reader is next time: 00:00:30:000
State for model TPM1 is bill amount: 200Current Account Balance: 2800
00:01:10:000
State for model input_reader is next time: 00:00:40:000
State for model TPM1 is bill amount: 30Current Account Balance: 2970
00:01:20:000
State for model input_reader is next time: 00:00:40:000
State for model TPM1 is bill amount: 30Current Account Balance: 2970

00:01:50:000
State for model input_reader is next time: inf
State for model TPM1 is bill amount: 45Current Account Balance: 2955
00:02:00:000
State for model input_reader is next time: inf
State for model TPM1 is bill amount: 45Current Account Balance: 2955

Coupled Models:

Top level Coupled Model is the “Internet Online Banking System“

IOBS inputs:
00:00:10 1 0
00:04:20 1 0
00:10:30 1 0

Linkings:
dynamic::translate::make_IC<AccountAccessManager_defs::van, inp_accountnumber**>("AAM1","ae"**),

dynamic::translate::make_IC<outp_aebillamount, TransactionProcessManager_defs::VBA_IN>("**ae","TPM1**")

Online Banking Output Messages:
00:00:00:000
00:00:00:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {}] generated by model input_reader1
00:00:10:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 0**}] generated by model input_reader1
00:00:20:000
[AccountAccessManager_defs::van: {**840187**}, AccountAccessManager_defs::logout: {}] generated by model AAM1
00:00:30:000
[AccountNumberVerifier_defs::output_OUT: {**0 1**}] generated by model ANV1
00:04:20:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 0**}] generated by model input_reader1
00:04:30:000
[AccountAccessManager_defs::van: {**783099**}, AccountAccessManager_defs::logout: {}] generated by model AAM1

00:04:40:000
[AccountNumberVerifier_defs::output_OUT: {**1 0**}] generated by model ANV1
00:04:50:000
[PasswordVerifier_defs::Output_OUT: {**1 2**}] generated by model PV1
00:05:00:000
[BillPaymentManager_defs::VBA_OUT: {**2336**}] generated by model BPM1
00:05:10:000
[TransactionProcessManager_defs::Output_OUT: {**664 1**}] generated by model TPM1
00:10:30:000
[cadmium::basic_models::pdevs::iestream_input_defs<Message_t>::out: {**1 0**}] generated by model input_reader1
00:10:40:000
[AccountAccessManager_defs::van: {**335222**}, AccountAccessManager_defs::logout: {}] generated by model AAM1
00:10:50:000
[AccountNumberVerifier_defs::output_OUT: {**0 1**}] generated by model ANV1

Online Banking Output States:
00:00:00:000
State for model input_reader1 is next time: 00:00:00:000
State for model AAM1 is login_status: 0 & ian_status 0 Account number: 0
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:00:00:000
State for model input_reader1 is next time: 00:00:10:000
State for model AAM1 is login_status: 0 & ian_status 0 Account number: 0
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:00:10:000
State for model input_reader1 is next time: 00:04:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 840188
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:00:20:000
State for model input_reader1 is next time: 00:04:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 840188
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0

State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:00:30:000
State for model input_reader1 is next time: 00:04:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 840188
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:04:20:000
State for model input_reader1 is next time: 00:06:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:04:30:000
State for model input_reader1 is next time: 00:06:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:04:40:000
State for model input_reader1 is next time: 00:06:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 0Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:04:50:000
State for model input_reader1 is next time: 00:06:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 2336Invalid bill? 0
State for model TPM1 is bill amount: 0Current Account Balance: 3000
00:05:00:000
State for model input_reader1 is next time: 00:06:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 2336Invalid bill? 0
State for model TPM1 is bill amount: 2336Current Account Balance: 664
00:05:10:000
State for model input_reader1 is next time: 00:06:10:000
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 783099

State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 2336Invalid bill? 0
State for model TPM1 is bill amount: 2336Current Account Balance: 664
00:10:30:000
State for model input_reader1 is next time: inf
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 335223
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 2336Invalid bill? 0
State for model TPM1 is bill amount: 2336Current Account Balance: 664
00:10:40:000
State for model input_reader1 is next time: inf
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 335223
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 2336Invalid bill? 0
State for model TPM1 is bill amount: 2336Current Account Balance: 664
00:10:50:000
State for model input_reader1 is next time: inf
State for model AAM1 is login_status: 1 & ian_status 0 Account number: 335223
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 2336Invalid bill? 0
State for model TPM1 is bill amount: 2336Current Account Balance: 664

Authentication Engine (AE)

AuthenticationEngine Inputs:
00:00:50 132123
00:02:20 142232

Linkings:
dynamic::translate::make_IC<AccountNumberVerifier_defs::output_OUT, PasswordVerifier_defs::Input_IN>("**ANV1**","**PV1**"),

dynamic::translate::make_IC<PasswordVerifier_defs::Output_OUT,PasswordVerifier_defs::Input_IN>("**PV1**","**PV1**"),

dynamic::translate::make_IC<PasswordVerifier_defs::Output_OUT, BillPaymentManager_defs::Input_IN>("**PV1**","**BPM1**")

Authentication Engine Output Messages:
00:00:00:000
00:00:00:000

[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {}] generated by model input_reader1
00:00:50:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {**132123**}] generated by model input_reader1
00:01:00:000
[AccountNumberVerifier_defs::output_OUT: {**1 0**}] generated by model ANV1
00:01:10:000
[PasswordVerifier_defs::Output_OUT: {**1 1**}] generated by model PV1
00:01:20:000
[PasswordVerifier_defs::Output_OUT: {**1 2**}] generated by model PV1
00:01:30:000
[BillPaymentManager_defs::VBA_OUT: {**916**}] generated by model BPM1
00:02:20:000
[cadmium::basic_models::pdevs::iestream_input_defs<int>::out: {**142232**}] generated by model input_reader1
00:02:30:000
[AccountNumberVerifier_defs::output_OUT: {**1 0**}] generated by model ANV1
00:02:40:000
[PasswordVerifier_defs::Output_OUT: {**1 2**}] generated by model PV1
00:02:50:000
[BillPaymentManager_defs::VBA_OUT: {**1387**}] generated by model BPM1

Authentication Engine Output States:
00:00:00:000
State for model input_reader1 is next time: 00:00:00:000
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
00:00:00:000
State for model input_reader1 is next time: 00:00:50:000
State for model ANV1 is account number valid: 0
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
00:00:50:000
State for model input_reader1 is next time: 00:01:30:000
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
00:01:00:000
State for model input_reader1 is next time: 00:01:30:000
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 0
State for model BPM1 is bill amount: 0Invalid bill? 0
00:01:10:000

State for model input_reader1 is next time: 00:01:30:000
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 0Invalid bill? 0
00:01:20:000
State for model input_reader1 is next time: 00:01:30:000
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 916Invalid bill? 0
00:01:30:000
State for model input_reader1 is next time: 00:01:30:000
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 916Invalid bill? 0
00:02:20:000
State for model input_reader1 is next time: inf
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 916Invalid bill? 0
00:02:30:000
State for model input_reader1 is next time: inf
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 916Invalid bill? 0
00:02:40:000
State for model input_reader1 is next time: inf
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 1387Invalid bill? 0
00:02:50:000
State for model input_reader1 is next time: inf
State for model ANV1 is account number valid: 1
State for model PV1 is password number: 1
State for model BPM1 is bill amount: 1387Invalid bill? 0