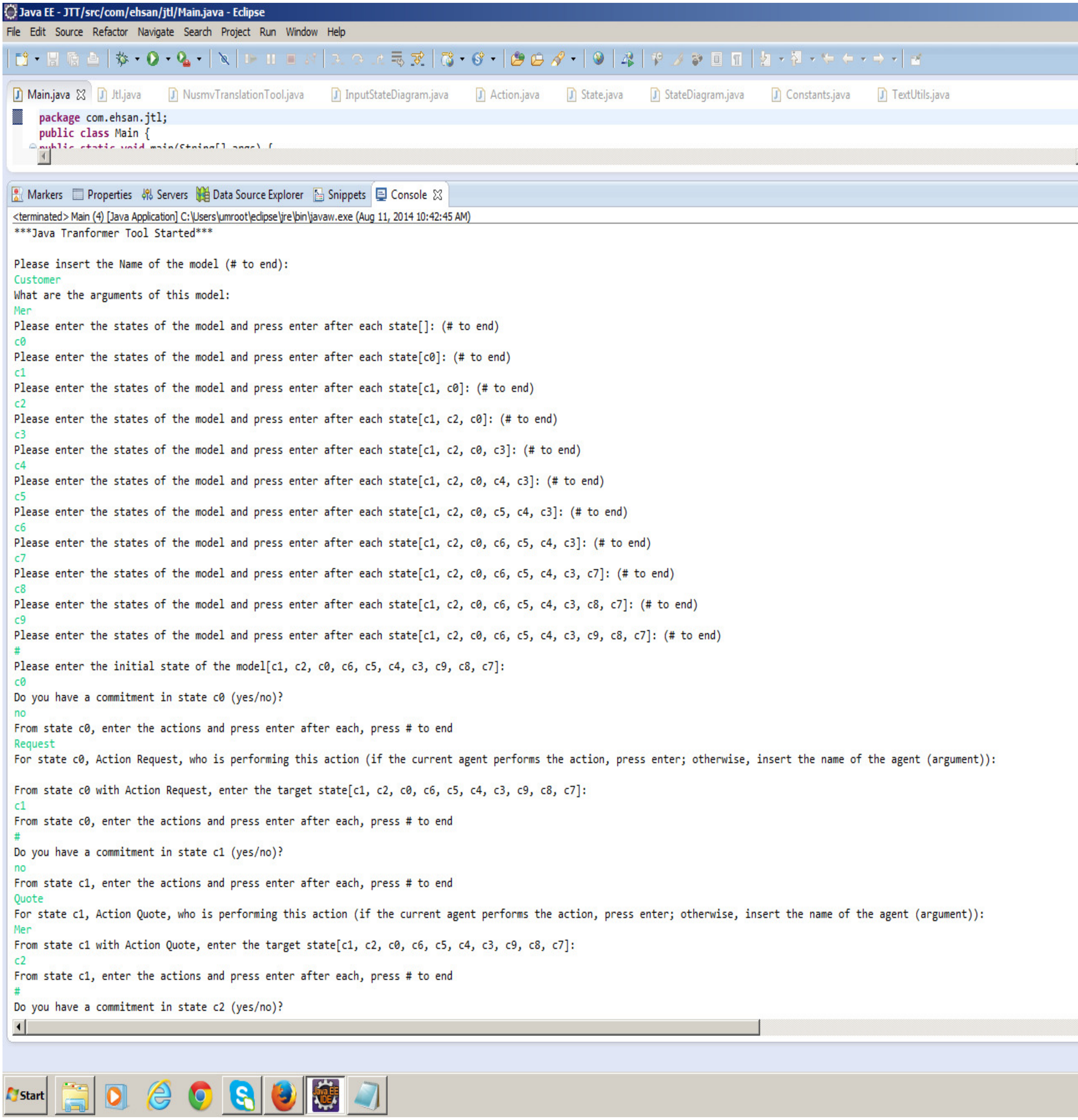
Example 1. The tool starts by asking the user to insert the name of the model, its arguments and states as in the figure below.

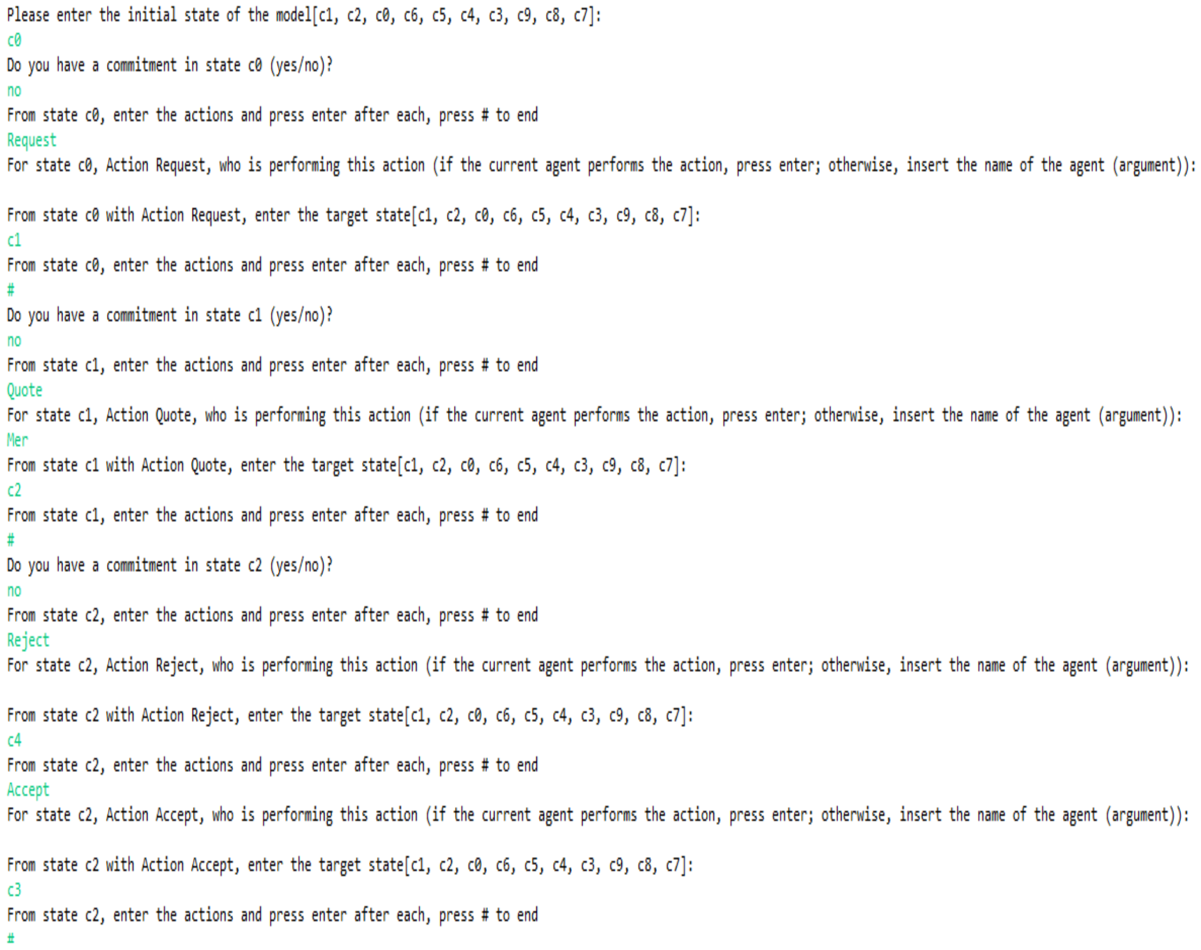


```java
package com.ehsan.jtl;
public class Main {
public static void main(String[] args) {
```

<terminated> Main (4) [Java Application] C:\Users\umroot\eclipse\jre\bin\javaw.exe (Aug 11, 2014 10:42:45 AM)

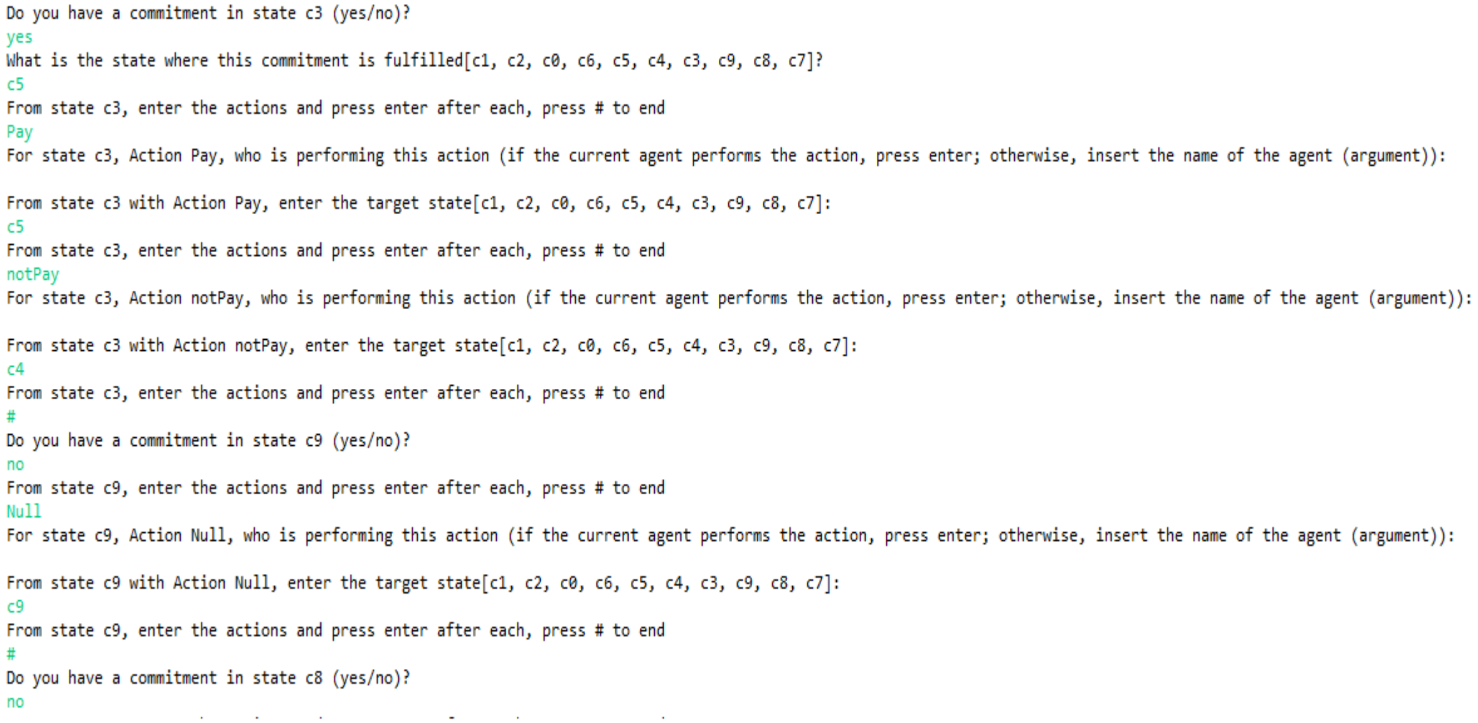
```
***Java Tranformer Tool Started***

Please insert the Name of the model (# to end):
Customer
What are the arguments of this model:
Mer
Please enter the states of the model and press enter after each state[]: (# to end)
c0
Please enter the states of the model and press enter after each state[c0]: (# to end)
c1
Please enter the states of the model and press enter after each state[c1, c0]: (# to end)
c2
Please enter the states of the model and press enter after each state[c1, c2, c0]: (# to end)
c3
Please enter the states of the model and press enter after each state[c1, c2, c0, c3]: (# to end)
c4
Please enter the states of the model and press enter after each state[c1, c2, c0, c4, c3]: (# to end)
c5
Please enter the states of the model and press enter after each state[c1, c2, c0, c5, c4, c3]: (# to end)
c6
Please enter the states of the model and press enter after each state[c1, c2, c0, c6, c5, c4, c3]: (# to end)
c7
Please enter the states of the model and press enter after each state[c1, c2, c0, c6, c5, c4, c3, c7]: (# to end)
c8
Please enter the states of the model and press enter after each state[c1, c2, c0, c6, c5, c4, c3, c8, c7]: (# to end)
c9
Please enter the states of the model and press enter after each state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]: (# to end)
#
Please enter the initial state of the model[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c0
Do you have a commitment in state c0 (yes/no)?
no
From state c0, enter the actions and press enter after each, press # to end
Request
For state c0, Action Request, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state c0 with Action Request, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c1
From state c0, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state c1 (yes/no)?
no
From state c1, enter the actions and press enter after each, press # to end
Quote
For state c1, Action Quote, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):
Mer
From state c1 with Action Quote, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c2
From state c1, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state c2 (yes/no)?
```

After that, the tool asks the user to insert the information of each state in the model as in the figure below.



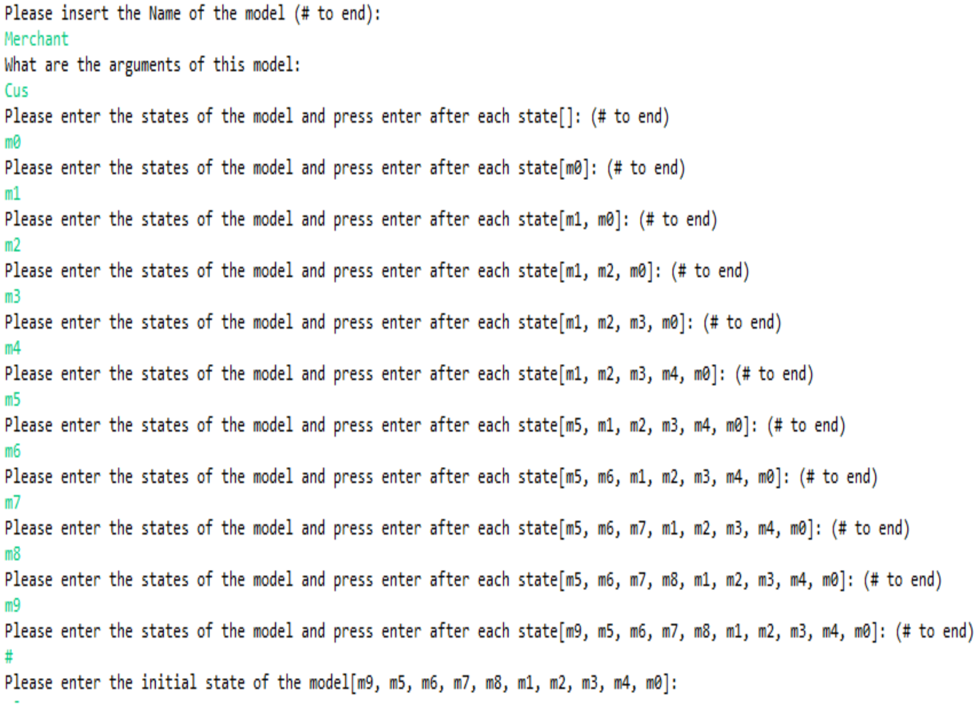
```
Please enter the initial state of the model[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c0
Do you have a commitment in state c0 (yes/no)?
no
From state c0, enter the actions and press enter after each, press # to end
Request
For state c0, Action Request, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state c0 with Action Request, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c1
From state c0, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state c1 (yes/no)?
no
From state c1, enter the actions and press enter after each, press # to end
Quote
For state c1, Action Quote, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):
Mer
From state c1 with Action Quote, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c2
From state c1, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state c2 (yes/no)?
no
From state c2, enter the actions and press enter after each, press # to end
Reject
For state c2, Action Reject, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state c2 with Action Reject, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c4
From state c2, enter the actions and press enter after each, press # to end
Accept
For state c2, Action Accept, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state c2 with Action Accept, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c3
From state c2, enter the actions and press enter after each, press # to end
#
```

If a state has commitments then the user should insert where this commitment is fulfilled as in the figure below.



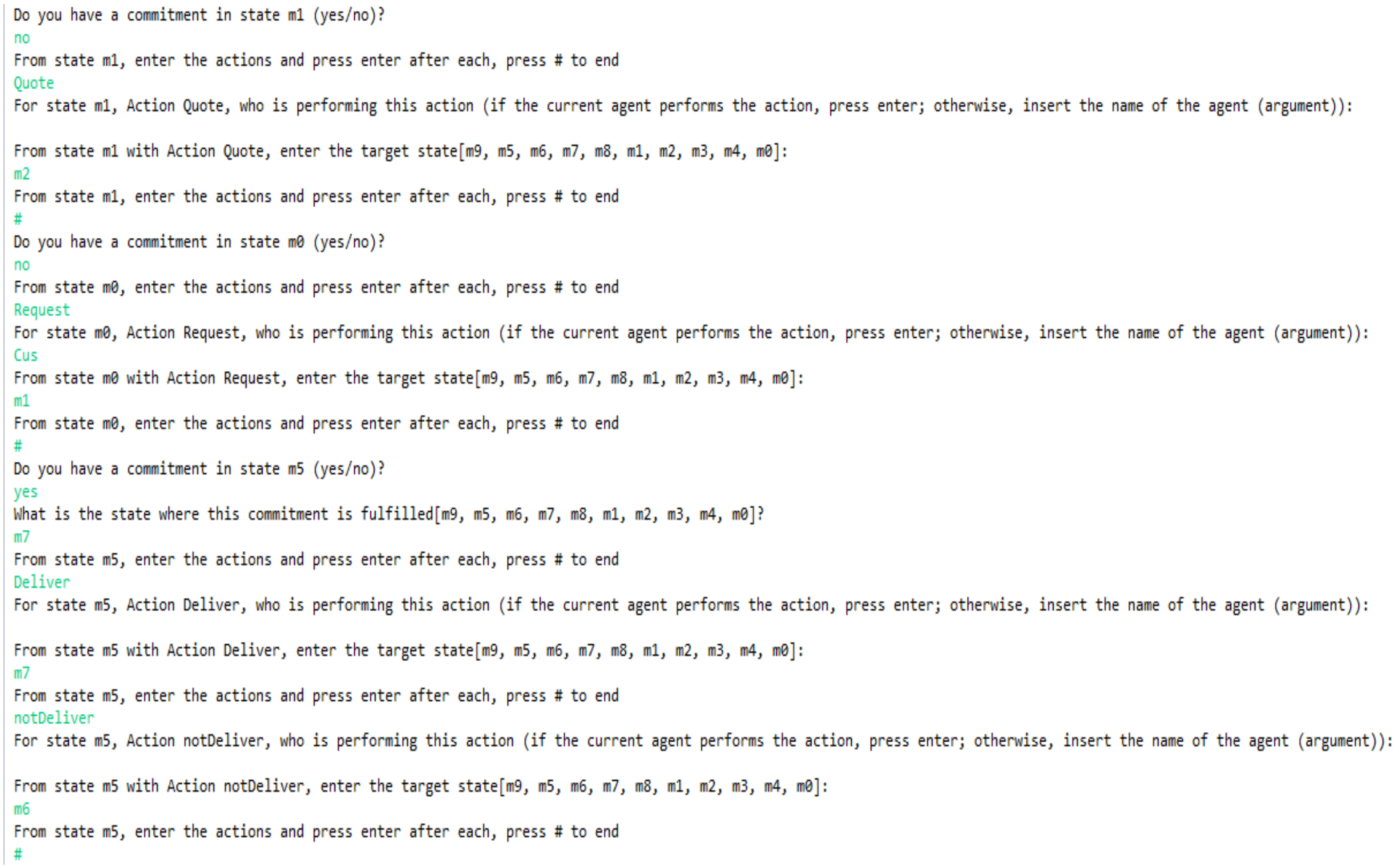
```
Do you have a commitment in state c3 (yes/no)?
yes
What is the state where this commitment is fulfilled[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]?
c5
From state c3, enter the actions and press enter after each, press # to end
Pay
For state c3, Action Pay, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state c3 with Action Pay, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c5
From state c3, enter the actions and press enter after each, press # to end
notPay
For state c3, Action notPay, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state c3 with Action notPay, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c4
From state c3, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state c9 (yes/no)?
no
From state c9, enter the actions and press enter after each, press # to end
Null
For state c9, Action Null, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state c9 with Action Null, enter the target state[c1, c2, c0, c6, c5, c4, c3, c9, c8, c7]:
c9
From state c9, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state c8 (yes/no)?
no
```

This process continues until inserting all information for all states in the model. After that, the user can start inserting the information of the second model as in the figure below.



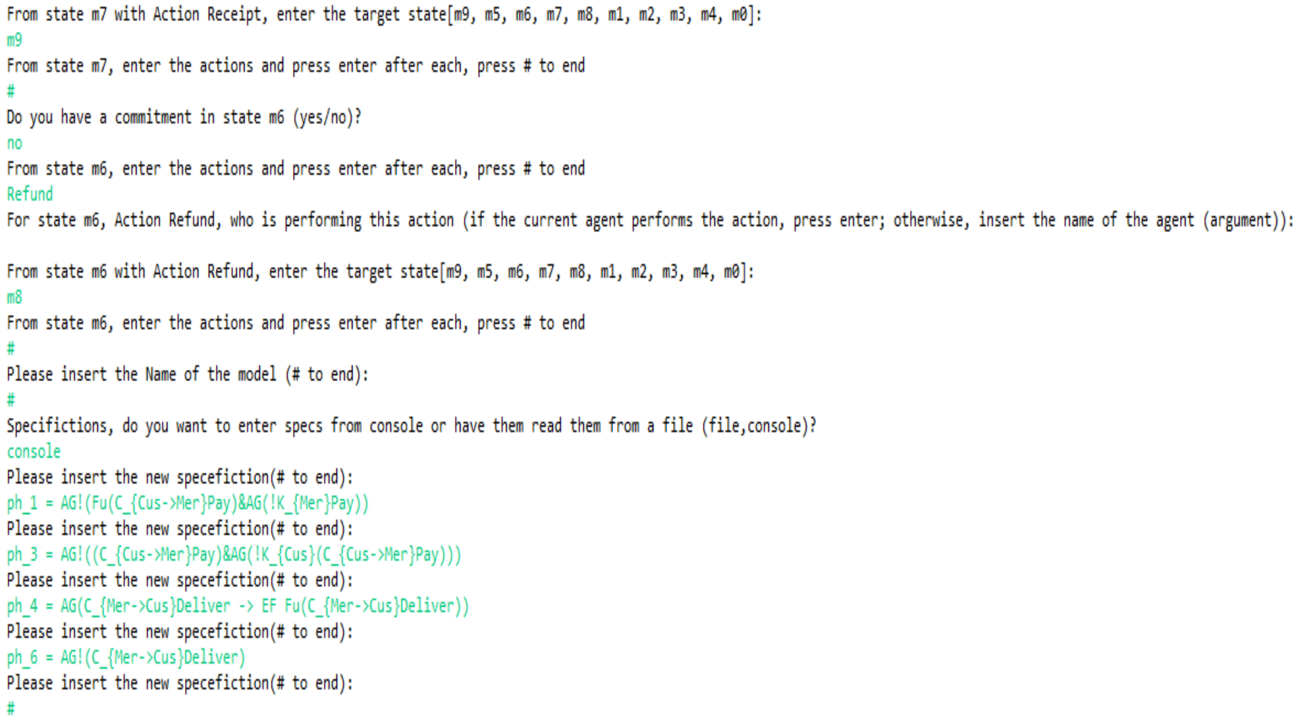
```
Please insert the Name of the model (# to end):
Merchant
What are the arguments of this model:
Cus
Please enter the states of the model and press enter after each state[]: (# to end)
m0
Please enter the states of the model and press enter after each state[m0]: (# to end)
m1
Please enter the states of the model and press enter after each state[m1, m0]: (# to end)
m2
Please enter the states of the model and press enter after each state[m1, m2, m0]: (# to end)
m3
Please enter the states of the model and press enter after each state[m1, m2, m3, m0]: (# to end)
m4
Please enter the states of the model and press enter after each state[m1, m2, m3, m4, m0]: (# to end)
m5
Please enter the states of the model and press enter after each state[m5, m1, m2, m3, m4, m0]: (# to end)
m6
Please enter the states of the model and press enter after each state[m5, m6, m1, m2, m3, m4, m0]: (# to end)
m7
Please enter the states of the model and press enter after each state[m5, m6, m7, m1, m2, m3, m4, m0]: (# to end)
m8
Please enter the states of the model and press enter after each state[m5, m6, m7, m8, m1, m2, m3, m4, m0]: (# to end)
m9
Please enter the states of the model and press enter after each state[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]: (# to end)
#
Please enter the initial state of the model[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]:
```

In a similar way to the previous model, the user has to insert the states of the model with their information as in the figure below.



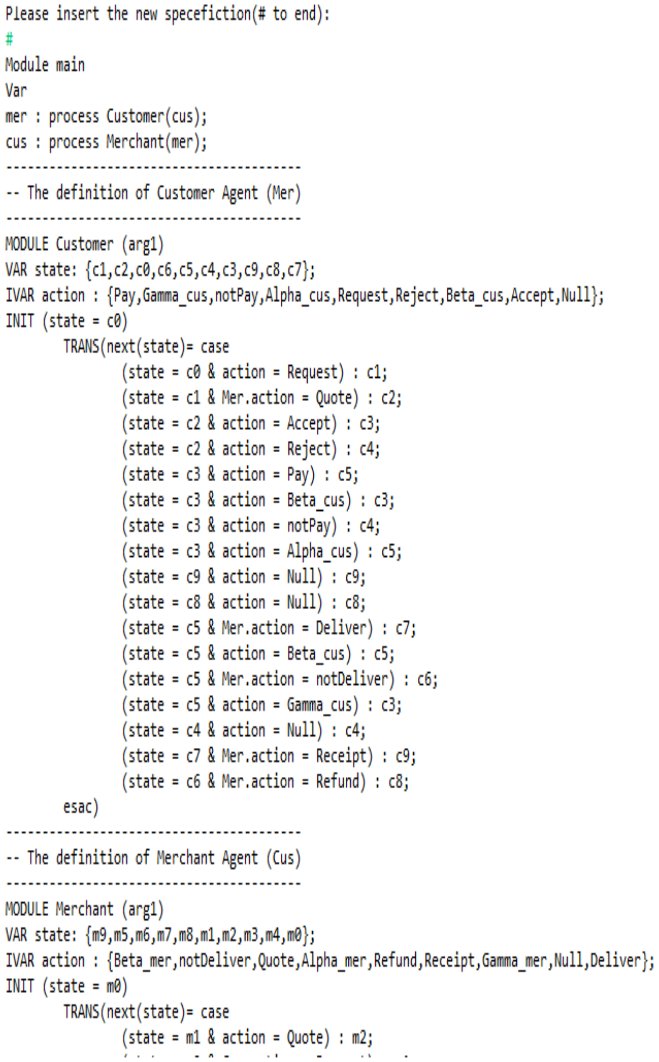
```
Do you have a commitment in state m1 (yes/no)?
no
From state m1, enter the actions and press enter after each, press # to end
Quote
For state m1, Action Quote, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state m1 with Action Quote, enter the target state[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]:
m2
From state m1, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state m0 (yes/no)?
no
From state m0, enter the actions and press enter after each, press # to end
Request
For state m0, Action Request, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):
Cus
From state m0 with Action Request, enter the target state[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]:
m1
From state m0, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state m5 (yes/no)?
yes
What is the state where this commitment is fulfilled[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]?
m7
From state m5, enter the actions and press enter after each, press # to end
Deliver
For state m5, Action Deliver, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state m5 with Action Deliver, enter the target state[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]:
m7
From state m5, enter the actions and press enter after each, press # to end
notDeliver
For state m5, Action notDeliver, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state m5 with Action notDeliver, enter the target state[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]:
m6
From state m5, enter the actions and press enter after each, press # to end
#
```

After that, the user has two choices to insert the specifications; either by using the console, or by using an input text file. In the figure below, the user inserts the specifications through the console.

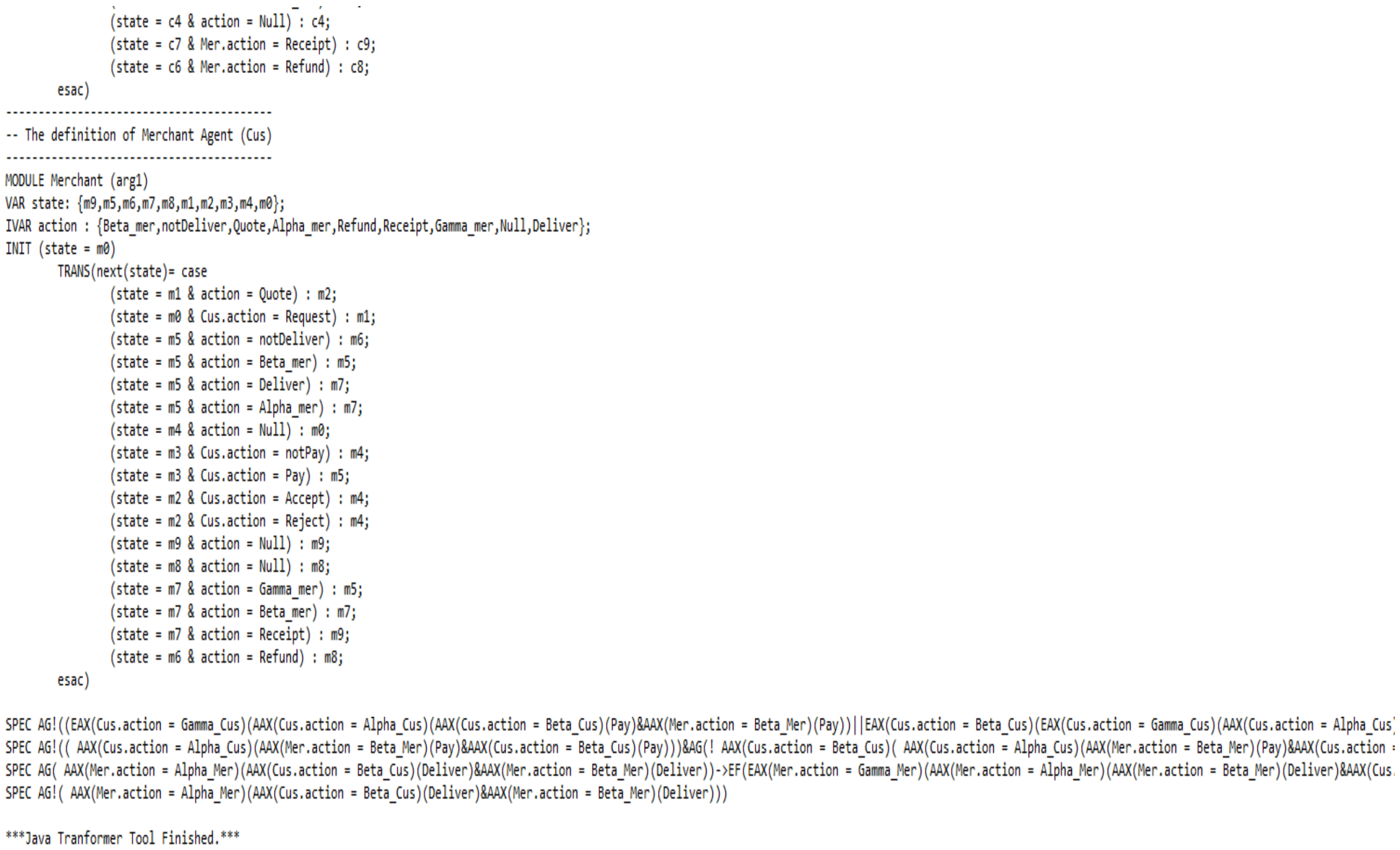


```
From state m7 with Action Receipt, enter the target state[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]:
m9
From state m7, enter the actions and press enter after each, press # to end
#
Do you have a commitment in state m6 (yes/no)?
no
From state m6, enter the actions and press enter after each, press # to end
Refund
For state m6, Action Refund, who is performing this action (if the current agent performs the action, press enter; otherwise, insert the name of the agent (argument)):

From state m6 with Action Refund, enter the target state[m9, m5, m6, m7, m8, m1, m2, m3, m4, m0]:
m8
From state m6, enter the actions and press enter after each, press # to end
#
Please insert the Name of the model (# to end):
#
Specifictions, do you want to enter specs from console or have them read them from a file (file,console)?
console
Please insert the new specefiction(# to end):
ph_1 = AG!(Fu(C_{Cus->Mer}Pay)&AG(!K_{Mer}Pay))
Please insert the new specefiction(# to end):
ph_3 = AG!((C_{Cus->Mer}Pay)&AG(!K_{Cus}(C_{Cus->Mer}Pay)))
Please insert the new specefiction(# to end):
ph_4 = AG(C_{Mer->Cus}Deliver -> EF Fu(C_{Mer->Cus}Deliver))
Please insert the new specefiction(# to end):
ph_6 = AG!(C_{Mer->Cus}Deliver)
Please insert the new specefiction(# to end):
#
```

When the user finishes inserting the specifications, the tool will generate the extended NuSMV code as in the following figures.



```
Please insert the new specefiction(# to end):
#
Module main
Var
mer : process Customer(cus);
cus : process Merchant(mer);
----------------------------------------
-- The definition of Customer Agent (Mer)
----------------------------------------
MODULE Customer (arg1)
VAR state: {c1,c2,c0,c6,c5,c4,c3,c9,c8,c7};
IVAR action : {Pay,Gamma_cus,notPay,Alpha_cus,Request,Reject,Beta_cus,Accept,Null};
INIT (state = c0)
        TRANS(next(state)= case
                (state = c0 & action = Request) : c1;
                (state = c1 & Mer.action = Quote) : c2;
                (state = c2 & action = Accept) : c3;
                (state = c2 & action = Reject) : c4;
                (state = c3 & action = Pay) : c5;
                (state = c3 & action = Beta_cus) : c3;
                (state = c3 & action = notPay) : c4;
                (state = c3 & action = Alpha_cus) : c5;
                (state = c9 & action = Null) : c9;
                (state = c8 & action = Null) : c8;
                (state = c5 & Mer.action = Deliver) : c7;
                (state = c5 & action = Beta_cus) : c5;
                (state = c5 & Mer.action = notDeliver) : c6;
                (state = c5 & action = Gamma_cus) : c3;
                (state = c4 & action = Null) : c4;
                (state = c7 & Mer.action = Receipt) : c9;
                (state = c6 & Mer.action = Refund) : c8;
        esac)
----------------------------------------
-- The definition of Merchant Agent (Cus)
----------------------------------------
MODULE Merchant (arg1)
VAR state: {m9,m5,m6,m7,m8,m1,m2,m3,m4,m0};
IVAR action : {Beta_mer,notDeliver,Quote,Alpha_mer,Refund,Receipt,Gamma_mer,Null,Deliver};
INIT (state = m0)
        TRANS(next(state)= case
                (state = m1 & action = Quote) : m2;
```



```
            (state = c4 & action = Null) : c4;
            (state = c7 & Mer.action = Receipt) : c9;
            (state = c6 & Mer.action = Refund) : c8;
        esac)
----------------------------------------
-- The definition of Merchant Agent (Cus)
----------------------------------------
MODULE Merchant (arg1)
VAR state: {m9,m5,m6,m7,m8,m1,m2,m3,m4,m0};
IVAR action : {Beta_mer,notDeliver,Quote,Alpha_mer,Refund,Receipt,Gamma_mer,Null,Deliver};
INIT (state = m0)
        TRANS(next(state)= case
            (state = m1 & action = Quote) : m2;
            (state = m0 & Cus.action = Request) : m1;
            (state = m5 & action = notDeliver) : m6;
            (state = m5 & action = Beta_mer) : m5;
            (state = m5 & action = Deliver) : m7;
            (state = m5 & action = Alpha_mer) : m7;
            (state = m4 & action = Null) : m0;
            (state = m3 & Cus.action = notPay) : m4;
            (state = m3 & Cus.action = Pay) : m5;
            (state = m2 & Cus.action = Accept) : m4;
            (state = m2 & Cus.action = Reject) : m4;
            (state = m9 & action = Null) : m9;
            (state = m8 & action = Null) : m8;
            (state = m7 & action = Gamma_mer) : m5;
            (state = m7 & action = Beta_mer) : m7;
            (state = m7 & action = Receipt) : m9;
            (state = m6 & action = Refund) : m8;
        esac)

SPEC AG!((EAX(Cus.action = Gamma_Cus)(AAX(Cus.action = Alpha_Cus)(AAX(Cus.action = Beta_Cus)(Pay)&AAX(Mer.action = Beta_Mer)(Pay))||EAX(Cus.action = Beta_Cus)(EAX(Cus.action = Gamma_Cus)(AAX(Cus.action = Alpha_Cus
SPEC AG!(( AAX(Cus.action = Alpha_Cus)(AAX(Mer.action = Beta_Mer)(Pay)&AAX(Cus.action = Beta_Cus)(Pay)))&AG(! AAX(Cus.action = Beta_Cus)( AAX(Cus.action = Alpha_Cus)(AAX(Mer.action = Beta_Mer)(Pay)&AAX(Cus.action =
SPEC AG( AAX(Mer.action = Alpha_Mer)(AAX(Cus.action = Beta_Cus)(Deliver)&AAX(Mer.action = Beta_Mer)(Deliver))->EF(EAX(Mer.action = Gamma_Mer)(AAX(Mer.action = Alpha_Mer)(AAX(Mer.action = Beta_Mer)(Deliver)&AAX(Cus.
SPEC AG!( AAX(Mer.action = Alpha_Mer)(AAX(Cus.action = Beta_Cus)(Deliver)&AAX(Mer.action = Beta_Mer)(Deliver)))

***Java Tranformer Tool Finished.***
```