

# State Machine and Code Generation

A [state machine](#) consists of a number of states and the transition between states. After creating a controller class, create a sub-state machine diagram from the controller class. Finally, you can [generate source code](#) based on the sub-state machine diagram. In this tutorial, you will model a controller class and state machine. After all, you will generate state machine code and play with the sample application. Finally, you will export [SCXML](#) from your state machine.

August 16, 2010

User Rating: / 10

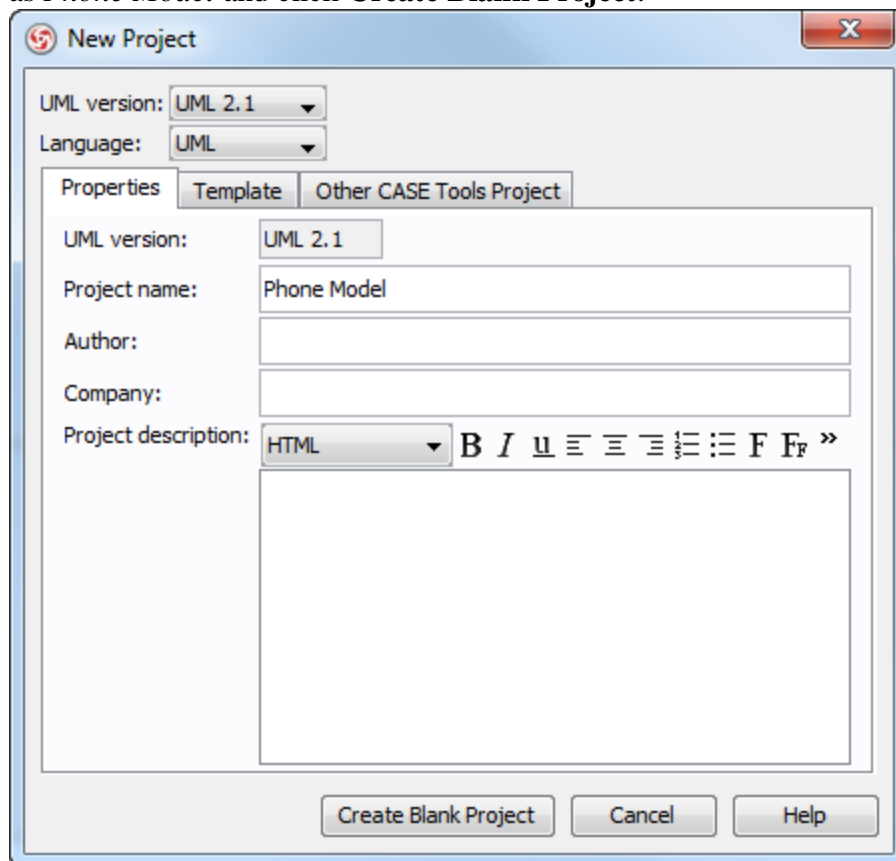
Views: 7,517

[PDF Link](#)

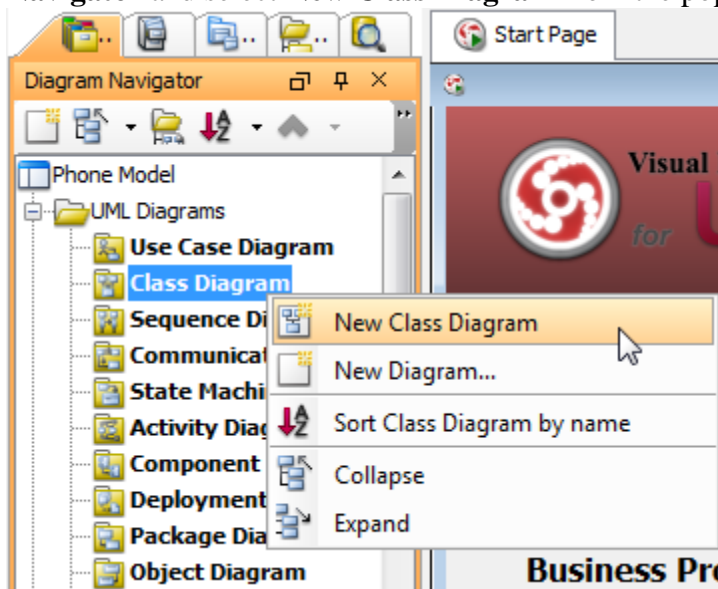
[Add comments](#)

Edition: [Modeler or above](#) ([Edition comparison](#))

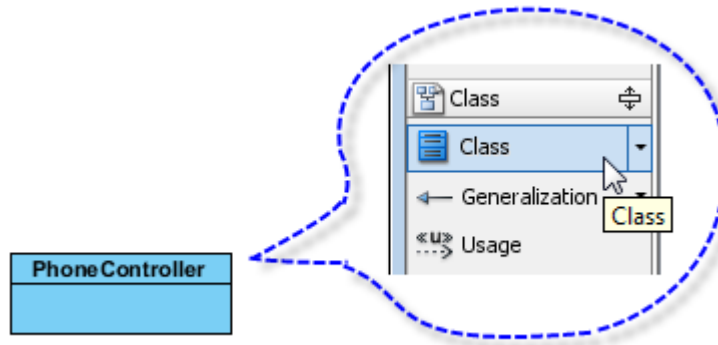
1. Create a new project with VP-UML. Select **File > New Project** from the main menu. Name the project as *Phone Model* and click **Create Blank Project**.



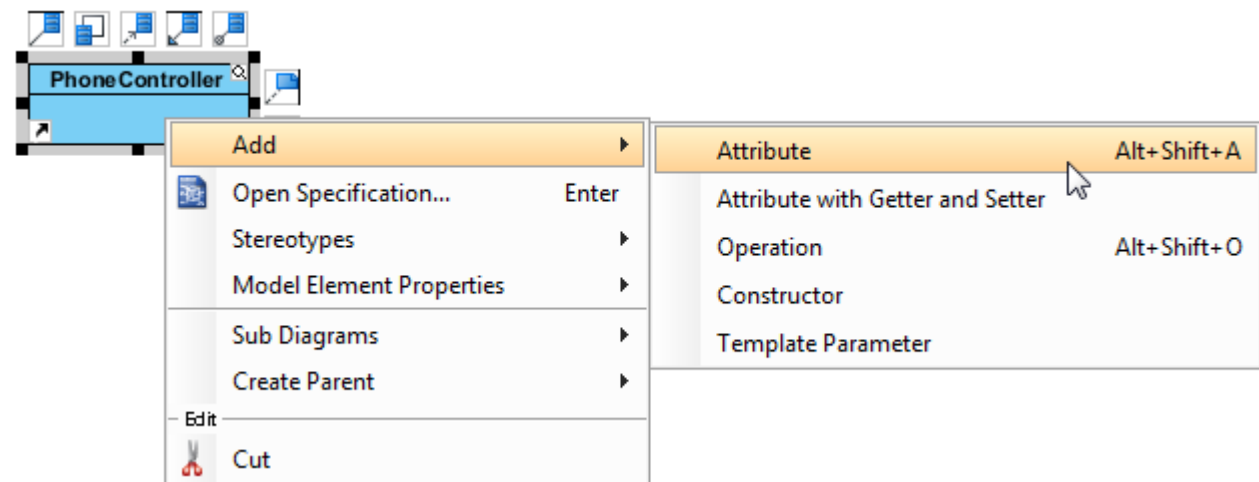
2. Create a class diagram form **Diagram Navigator**. Right click on **Class Diagram** in **Diagram Navigator** and select **New Class Diagram** from the pop-up menu.



3. Create a controller class. Select **Class** from diagram toolbar, click on the diagram to create a class and name it as *PhoneController*.



4. Next, add a few attributes to the controller class. Right click on the class and select **Add > Attribute** from the pop-up menu.



5. Enter *\_dialingNumber : String* and press **Enter** to create the next attribute. Continue to add attributes as shown below:

Name	Type
------	------

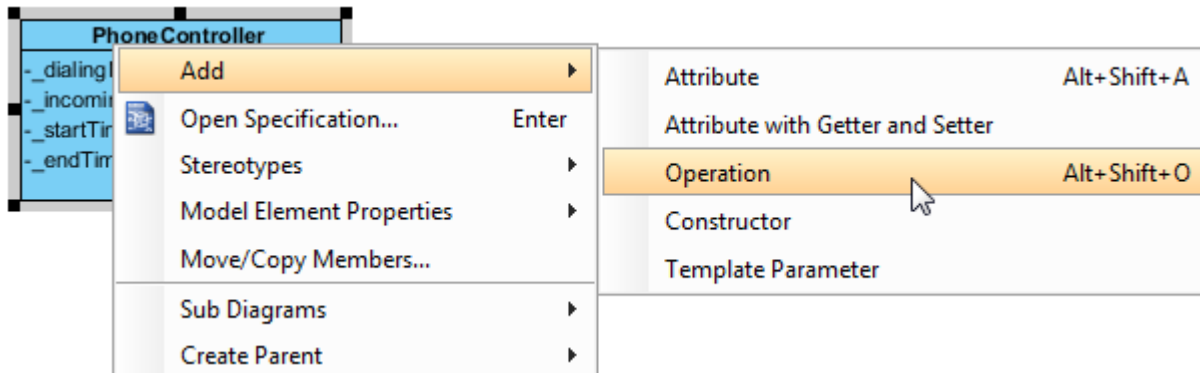
_incomingCallNumber	String
_startTime	long
_endTime	long

6.

The result of *PhoneController* class is shown as below:

PhoneController
-_dialingNumber : String
-_incomingCallNumber : String
-_startTime : long
-_endTime : long

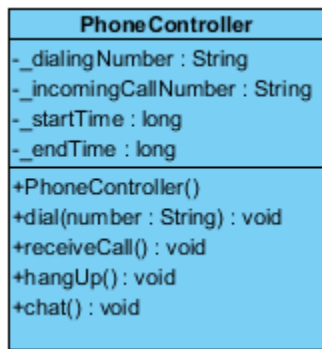
7. Add operations to the controller class. Those operations will cause the state to change. Right click on the class and select **Add > Operation** from the pop-up menu.



8. Enter *PhoneController* to add the constructor and press **Enter** to create the next operation. Continue to add operations as shown below:

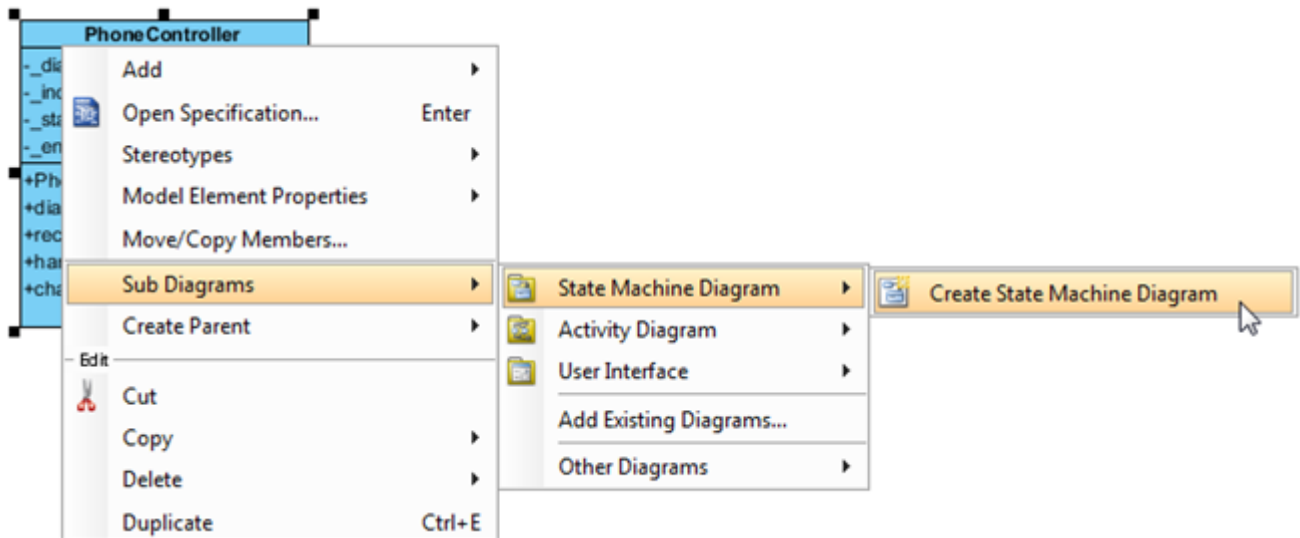
Name	Return Type
dial(number : String) : void	void
receiveCall	void
hangUp	void
chat	void

9.

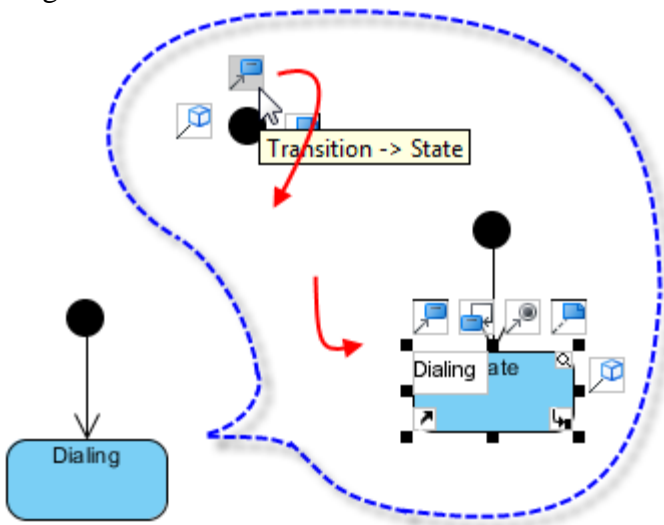


Controller class is shown as below:

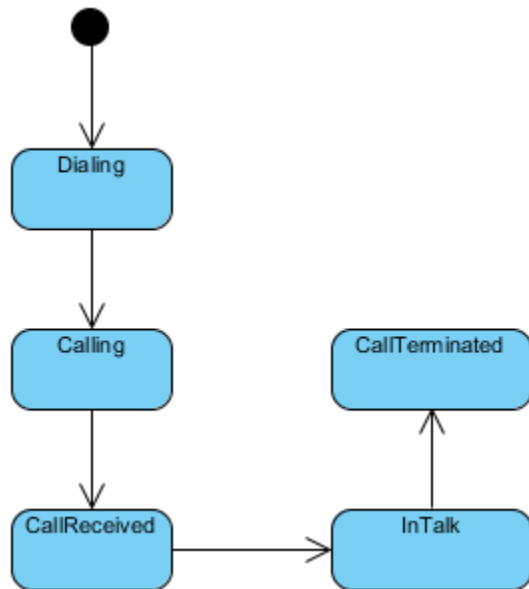
10. Now, we are going to draw the state machine for *PhoneController* class. Right click on *PhoneController* and select **Sub Diagram** > **State Machine Diagram** > **Create State Machine Diagram** from the popup menu.



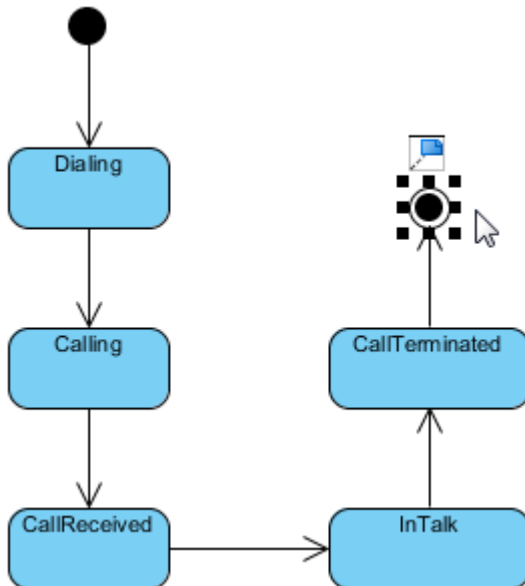
11. A state machine diagram is created with an initial node appears. Draw the *Dialing* state through resource icon. Move the mouse pointer over the initial node, press on its resource icon **Transition -> State** and drag it out. Release the mouse button until finish creating. Name the state as *Dialing*.



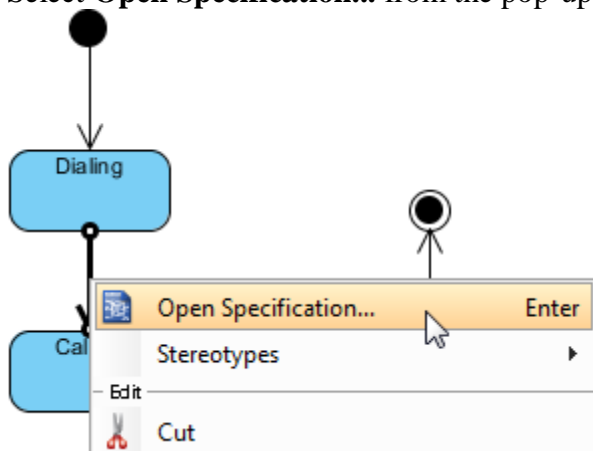
12. Follow the image below to create a few more states: *Calling*, *CallReceived*, *InTalk* and *CallTerminated* and then connect them with transition.



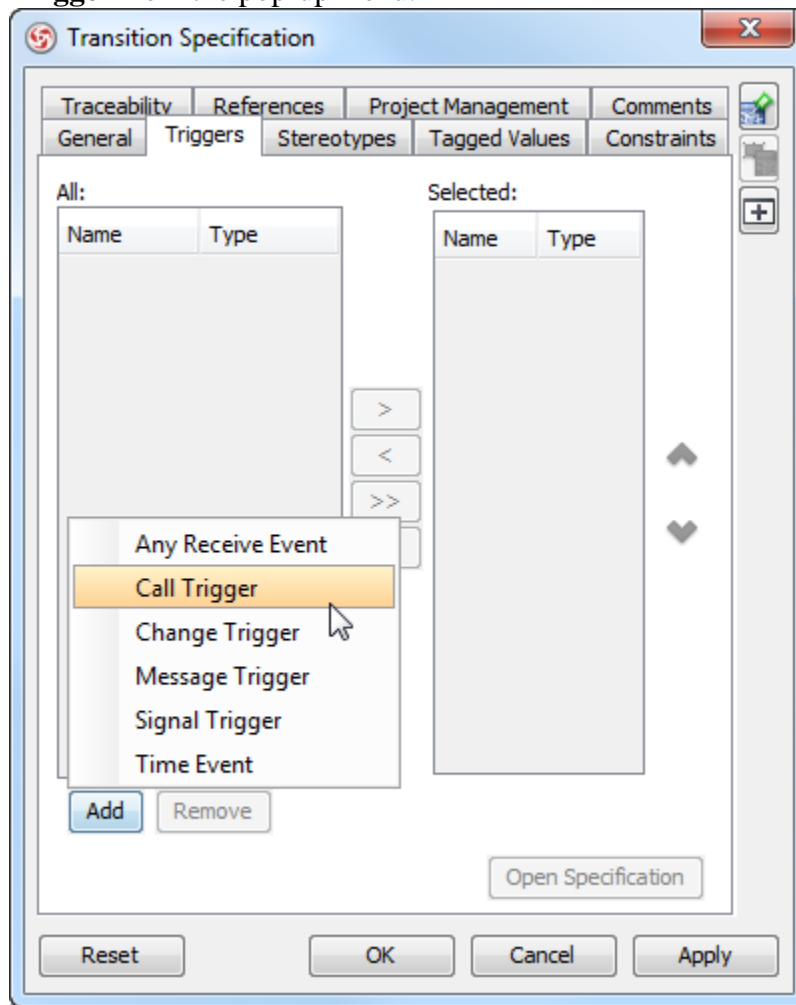
13. Create the final state from *CallTerminated* state through resource centric interface.



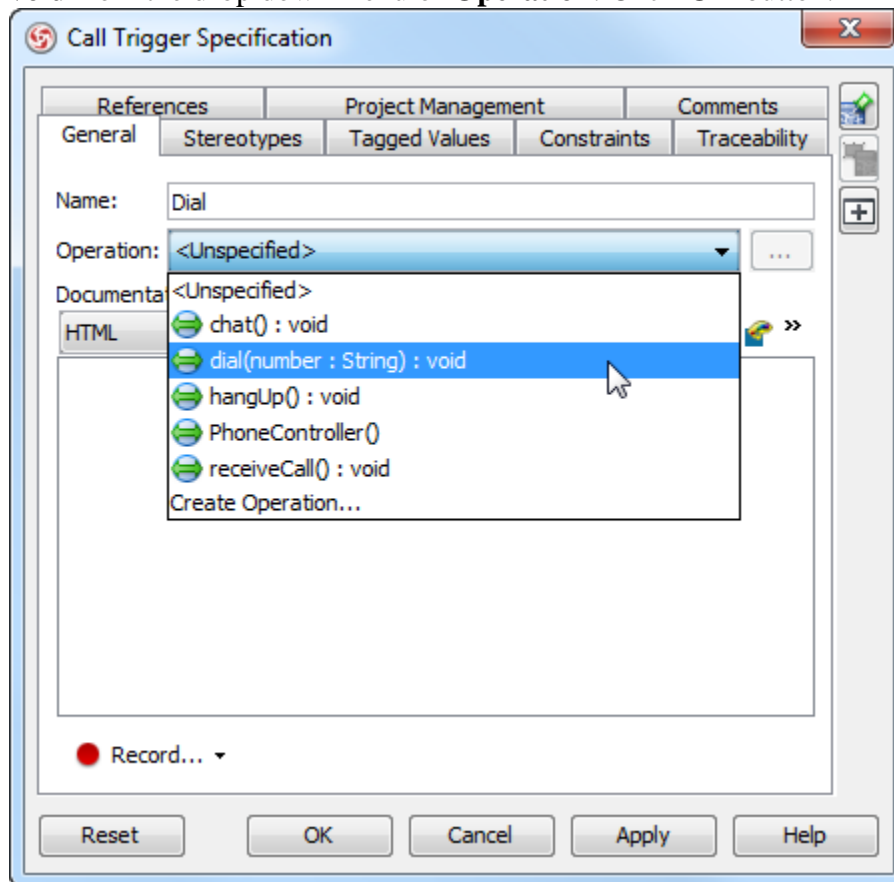
14. Now, add a trigger to transitions. Right click on the transition between *Dialing* state and *Calling* state. Select **Open Specification...** from the pop-up menu.



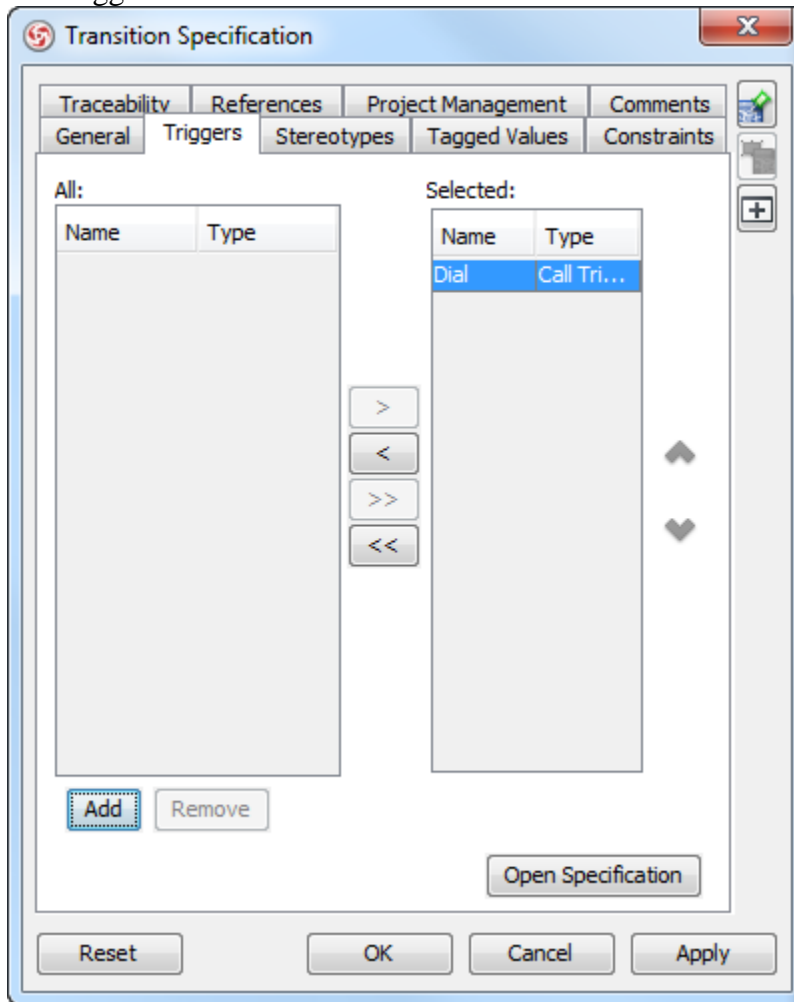
15. In the **Transition Specification** dialog box, open **Triggers** tab. Click **Add** button and select **Call Trigger** from the pop-up menu.



16. In the **Call Trigger Specification** dialog box, enter *Dial* as trigger name. Select **dial(number : String) : void** from the drop down menu of **Operation**. Click **OK** button.



17. The trigger is selected for the transition. Click **OK** button.



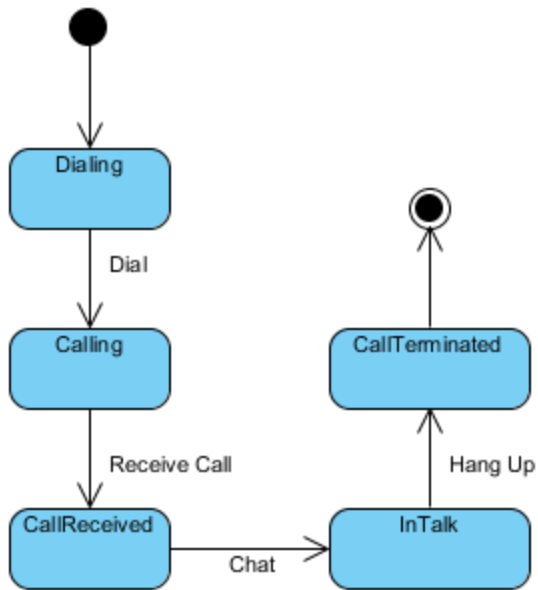
18. Continue to add a few more triggers following the table below:

Transition	Trigger name
Calling -> Call Received	Receive Call
CallReceived -> InTalk	Chat
InTalk -> CallTerminated	Hang up

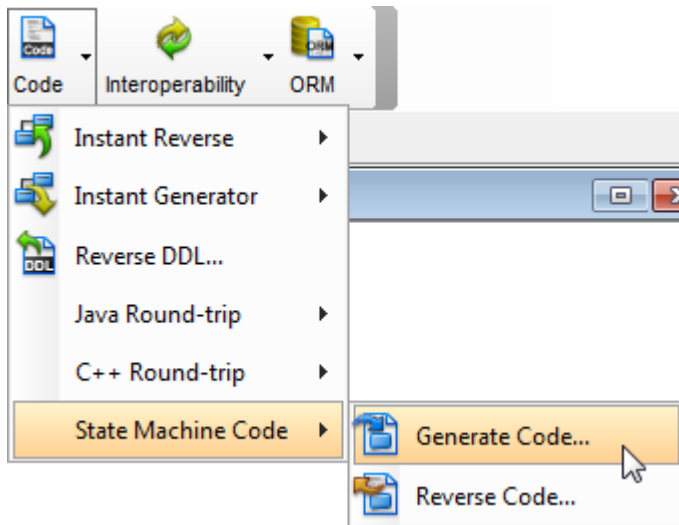


19.

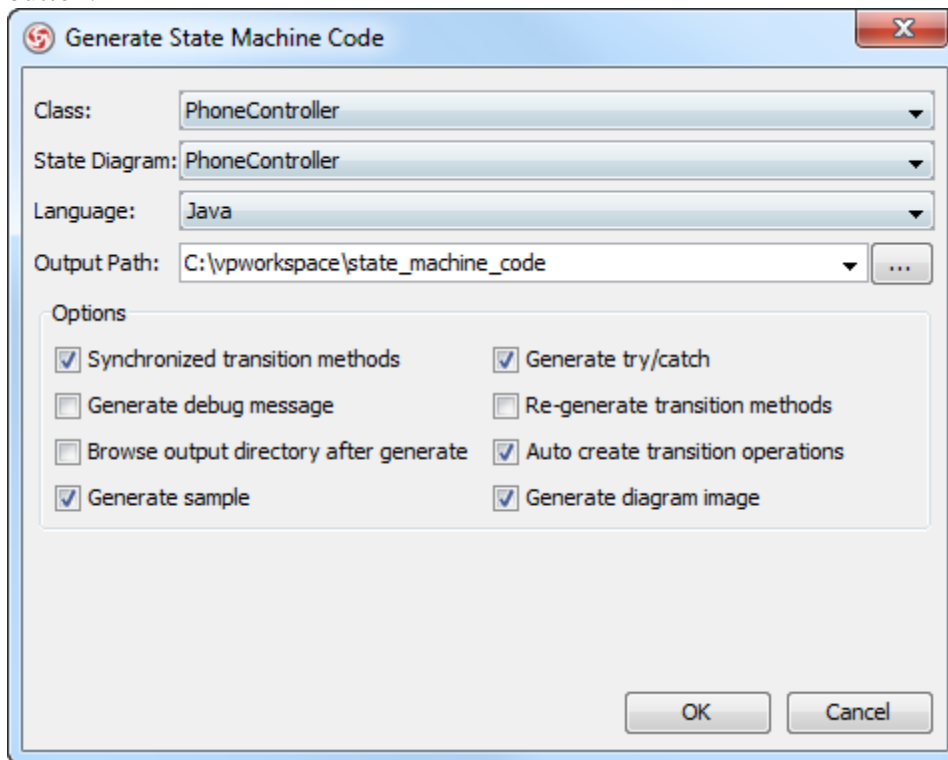
shown as follows:



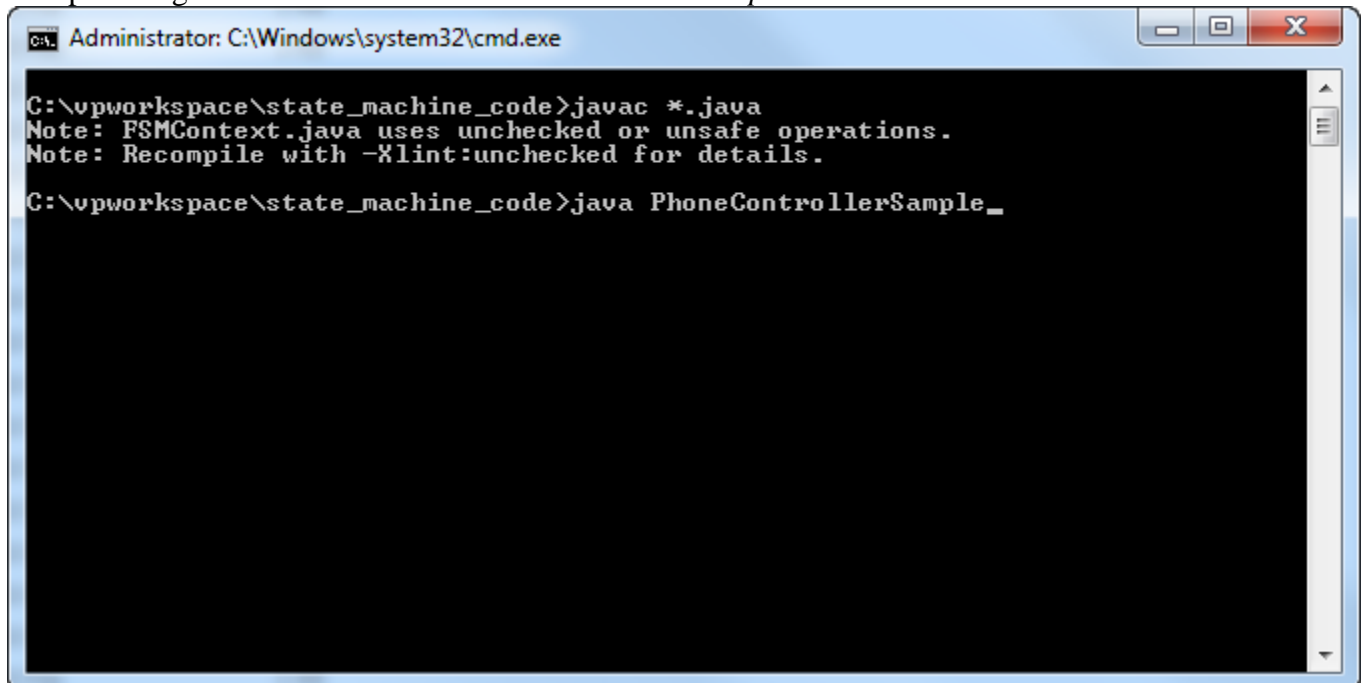
20. Now, let's do code generation. Select **Code > State Machine Code > Generate Code...** from the toolbar.



21. In the **Generate State Machine Code**, specify the directory to store the generated code and click **OK** button.



22. Compile the generated code and run *PhoneControllerSample* class.



Run the sample application by clicking **Proceed** button repeatedly and observe the change of states.

