Model-based System Engineering

Mini-Project 2

By

Stefanie Scholz, Stefan Schuler, Anton Komarov

Tasks

- Build a code generator

  -> using Xtend

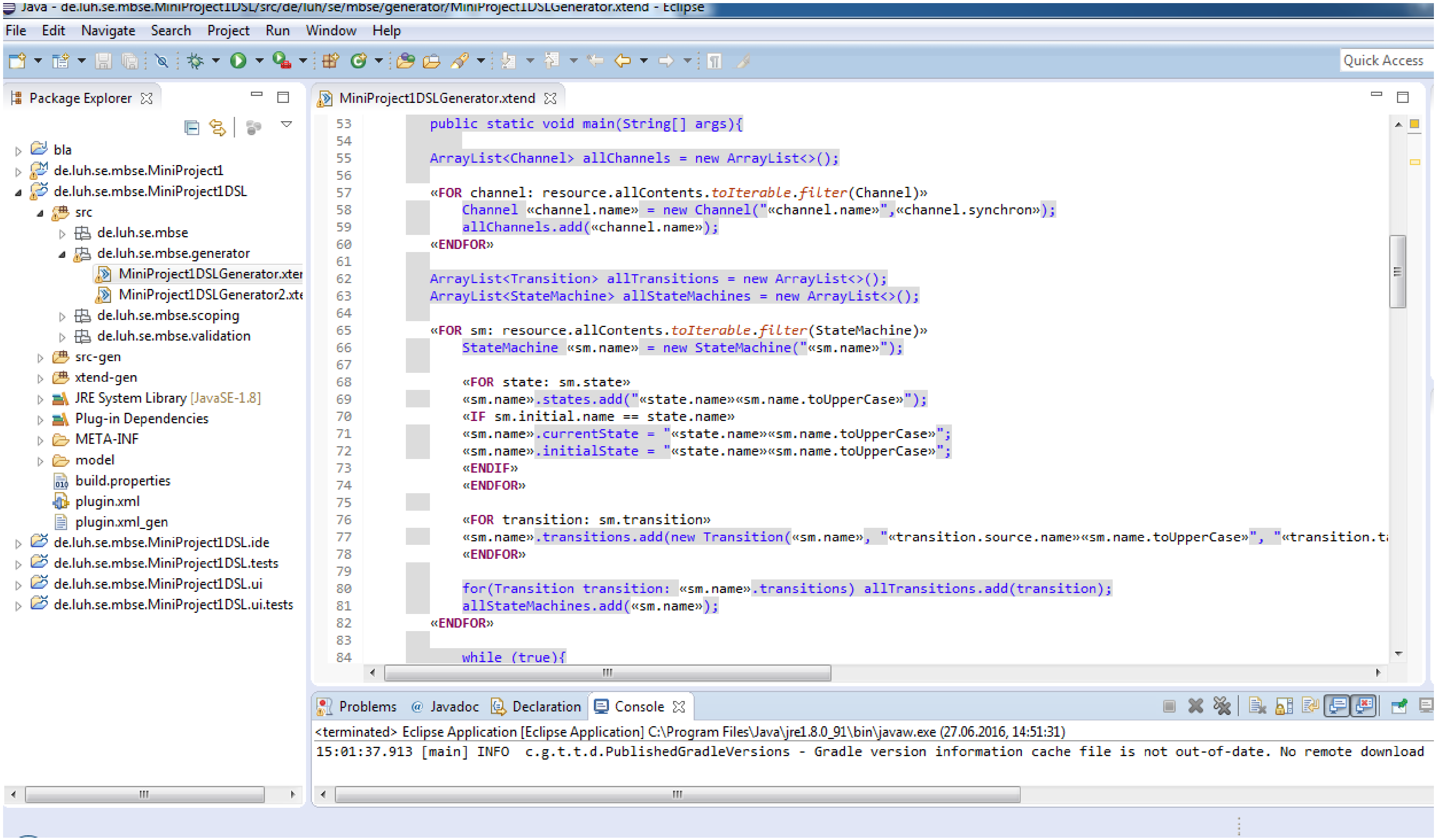
- Build an interpreter

  -> extend the ecore-model

- Model the interpreter logic

  -> using Henshin

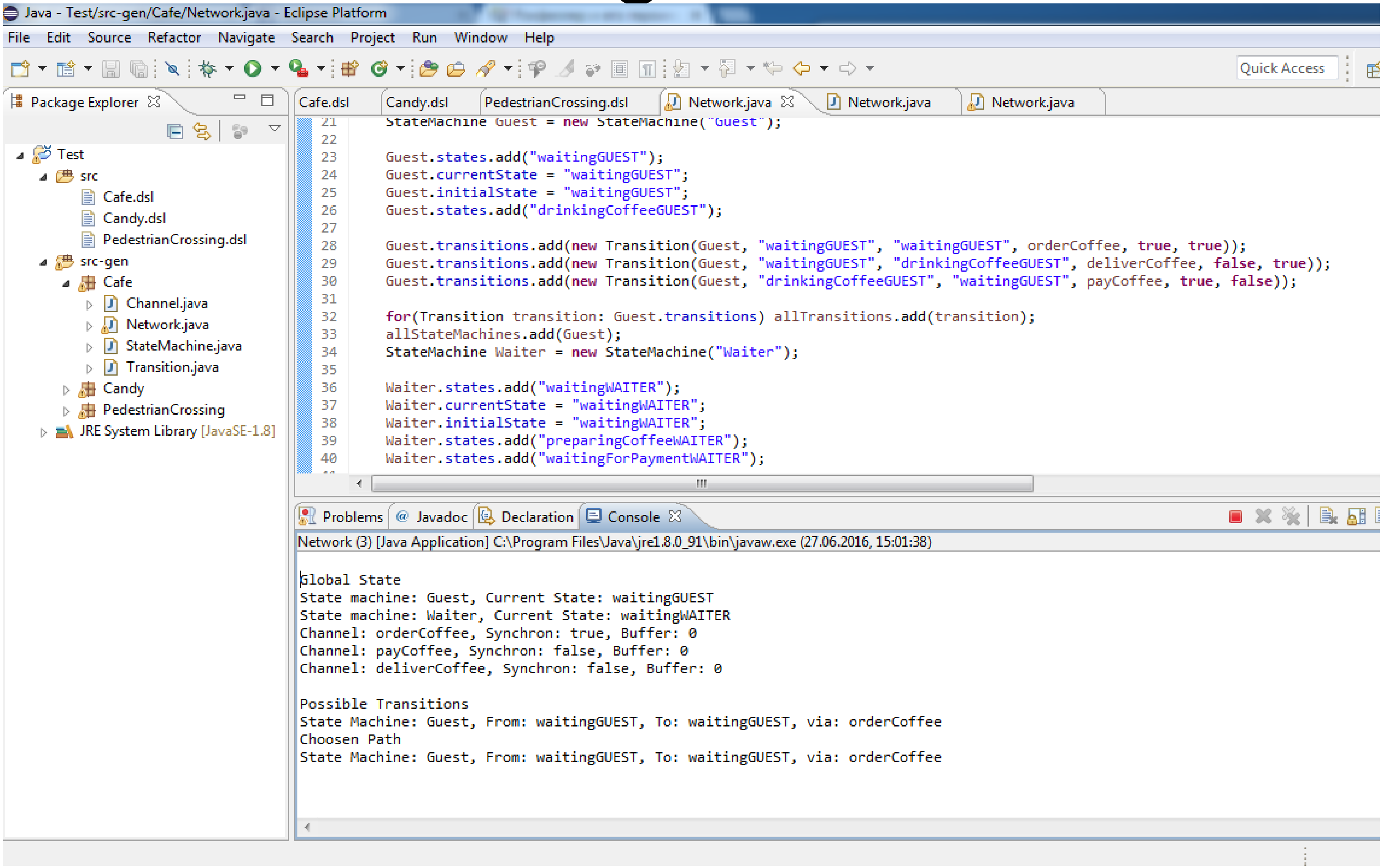
Code generator


Java - de.luh.se.mbse.MiniProject1DSL/src/de/luh/se/mbse/generator/MiniProject1DSLGenerator.xtend - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Package Explorer
bla
de.luh.se.mbse.MiniProject1
de.luh.se.mbse.MiniProject1DSL
src
de.luh.se.mbse
de.luh.se.mbse.generator
MiniProject1DSLGenerator.xter
MiniProject1DSLGenerator2.xte
de.luh.se.mbse.scoping
de.luh.se.mbse.validation
src-gen
xtend-gen
JRE System Library [JavaSE-1.8]
Plug-in Dependencies
META-INF
model
build.properties
plugin.xml
plugin.xml_gen
de.luh.se.mbse.MiniProject1DSL.ide
de.luh.se.mbse.MiniProject1DSL.tests
de.luh.se.mbse.MiniProject1DSL.ui
de.luh.se.mbse.MiniProject1DSL.ui.tests
MiniProject1DSLGenerator.xtend
Problems
Javadoc
Declaration
Console

```
public static void main(String[] args){

ArrayList<Channel> allChannels = new ArrayList<>();

«FOR channel: resource.allContents.toIterable.filter(Channel)»
    Channel «channel.name» = new Channel("«channel.name»",«channel.synchron»);
    allChannels.add(«channel.name»);
«ENDFOR»

ArrayList<Transition> allTransitions = new ArrayList<>();
ArrayList<StateMachine> allStateMachines = new ArrayList<>();

«FOR sm: resource.allContents.toIterable.filter(StateMachine)»
    StateMachine «sm.name» = new StateMachine("«sm.name»");

    «FOR state: sm.state»
    «sm.name».states.add("«state.name»«sm.name.toUpperCase»");
    «IF sm.initial.name == state.name»
    «sm.name».currentState = "«state.name»«sm.name.toUpperCase»";
    «sm.name».initialState = "«state.name»«sm.name.toUpperCase»";
    «ENDIF»
    «ENDFOR»

    «FOR transition: sm.transition»
    «sm.name».transitions.add(new Transition(«sm.name», "«transition.source.name»«sm.name.toUpperCase»", "«transition.t
    «ENDFOR»

    for(Transition transition: «sm.name».transitions) allTransitions.add(transition);
    allStateMachines.add(«sm.name»);
«ENDFOR»

    while (true){
```

```
<terminated> Eclipse Application [Eclipse Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (27.06.2016, 14:51:31)
15:01:37.913 [main] INFO  c.g.t.t.d.PublishedGradleVersions - Gradle version information cache file is not out-of-date. No remote download
```

Code generator



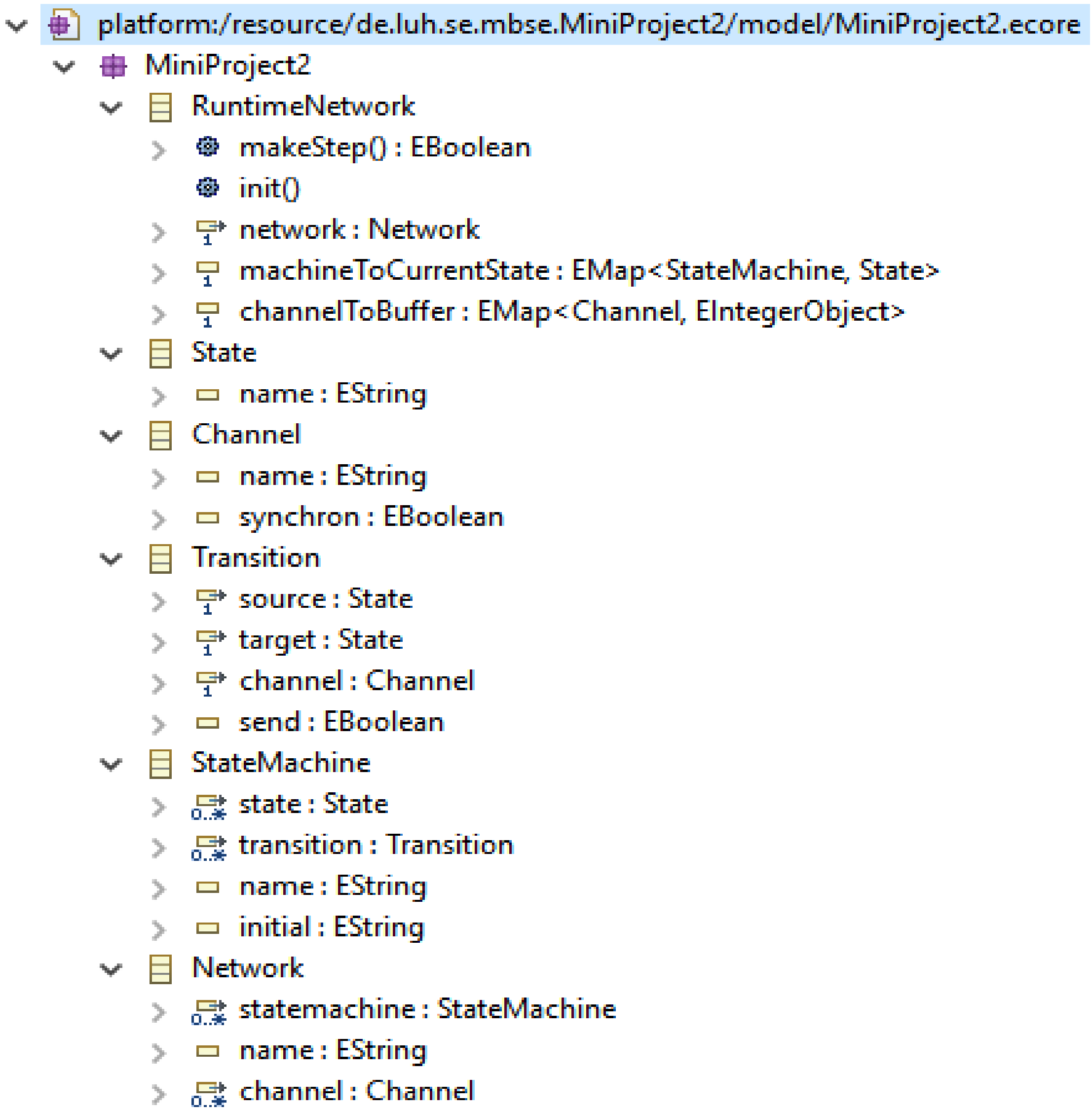
```java
    StateMachine Guest = new StateMachine("Guest");

    Guest.states.add("waitingGUEST");
    Guest.currentState = "waitingGUEST";
    Guest.initialState = "waitingGUEST";
    Guest.states.add("drinkingCoffeeGUEST");

    Guest.transitions.add(new Transition(Guest, "waitingGUEST", "waitingGUEST", orderCoffee, true, true));
    Guest.transitions.add(new Transition(Guest, "waitingGUEST", "drinkingCoffeeGUEST", deliverCoffee, false, true));
    Guest.transitions.add(new Transition(Guest, "drinkingCoffeeGUEST", "waitingGUEST", payCoffee, true, false));

    for(Transition transition: Guest.transitions) allTransitions.add(transition);
    allStateMachines.add(Guest);
    StateMachine Waiter = new StateMachine("Waiter");

    Waiter.states.add("waitingWAITER");
    Waiter.currentState = "waitingWAITER";
    Waiter.initialState = "waitingWAITER";
    Waiter.states.add("preparingCoffeeWAITER");
    Waiter.states.add("waitingForPaymentWAITER");
```

Network (3) [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (27.06.2016, 15:01:38)

```
Global State
State machine: Guest, Current State: waitingGUEST
State machine: Waiter, Current State: waitingWAITER
Channel: orderCoffee, Synchron: true, Buffer: 0
Channel: payCoffee, Synchron: false, Buffer: 0
Channel: deliverCoffee, Synchron: false, Buffer: 0

Possible Transitions
State Machine: Guest, From: waitingGUEST, To: waitingGUEST, via: orderCoffee
Choosen Path
State Machine: Guest, From: waitingGUEST, To: waitingGUEST, via: orderCoffee
```

Interpreter

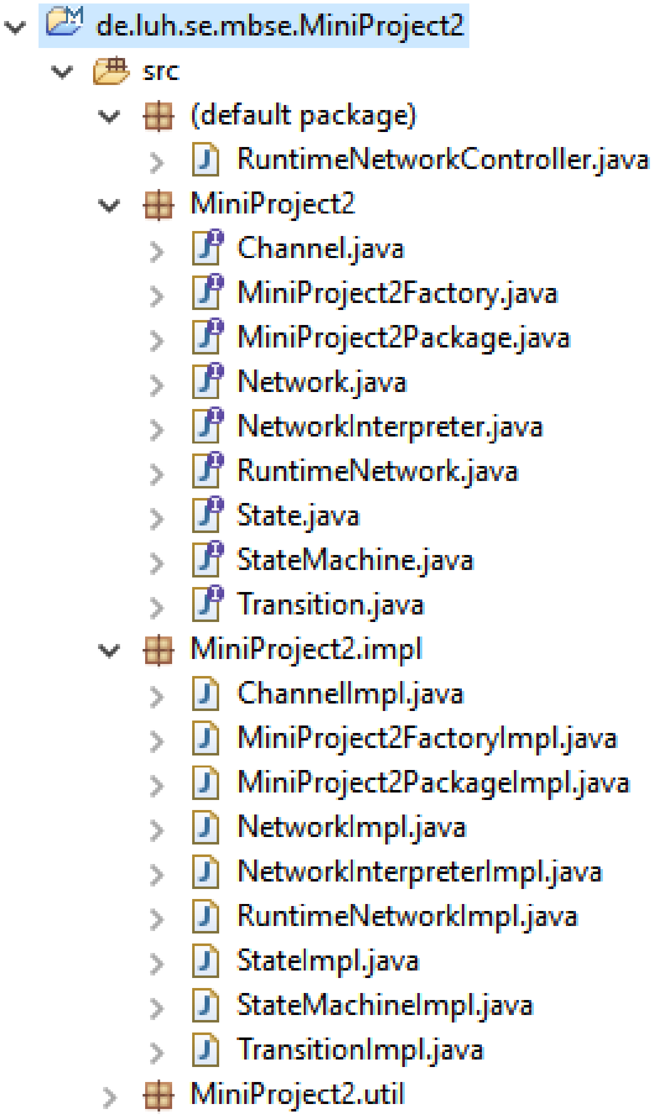
- Extend ecore model



```
platform:/resource/de.luh.se.mbse.MiniProject2/model/MiniProject2.ecore
  MiniProject2
    RuntimeNetwork
      makeStep() : EBoolean
      init()
      network : Network
      machineToCurrentState : EMap<StateMachine, State>
      channelToBuffer : EMap<Channel, EIntegerObject>
    State
      name : EString
    Channel
      name : EString
      synchron : EBoolean
    Transition
      source : State
      target : State
      channel : Channel
      send : EBoolean
    StateMachine
      state : State
      transition : Transition
      name : EString
      initial : EString
    Network
      statemachine : StateMachine
      name : EString
      channel : Channel
```

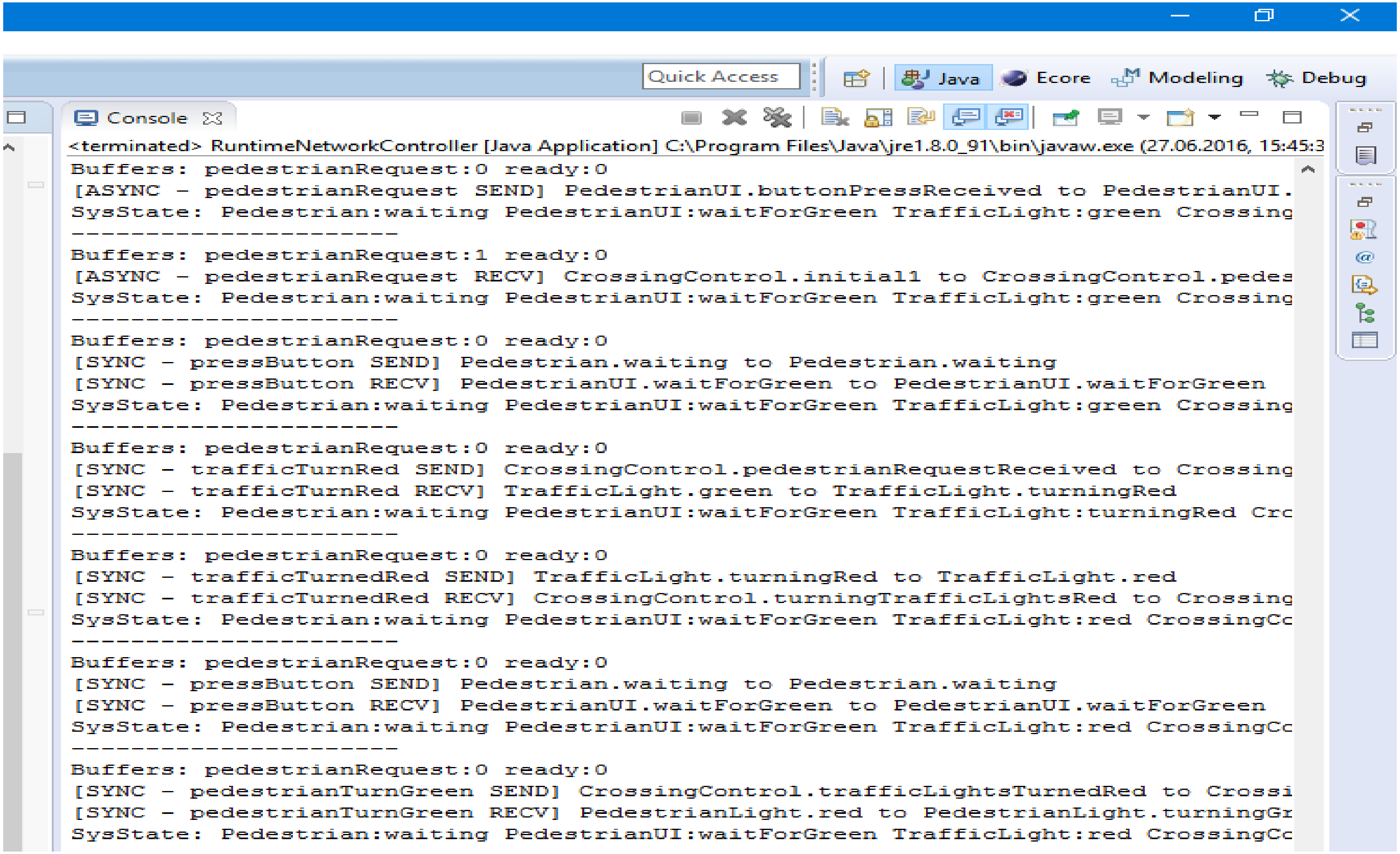
Interpreter

- Create java Files



```
de.luh.se.mbse.MiniProject2
  src
    (default package)
      RuntimeNetworkController.java
    MiniProject2
      Channel.java
      MiniProject2Factory.java
      MiniProject2Package.java
      Network.java
      NetworkInterpreter.java
      RuntimeNetwork.java
      State.java
      StateMachine.java
      Transition.java
    MiniProject2.impl
      ChannelImpl.java
      MiniProject2FactoryImpl.java
      MiniProject2PackageImpl.java
      NetworkImpl.java
      NetworkInterpreterImpl.java
      RuntimeNetworkImpl.java
      StateImpl.java
      StateMachineImpl.java
      TransitionImpl.java
    MiniProject2.util
```

Interpreter

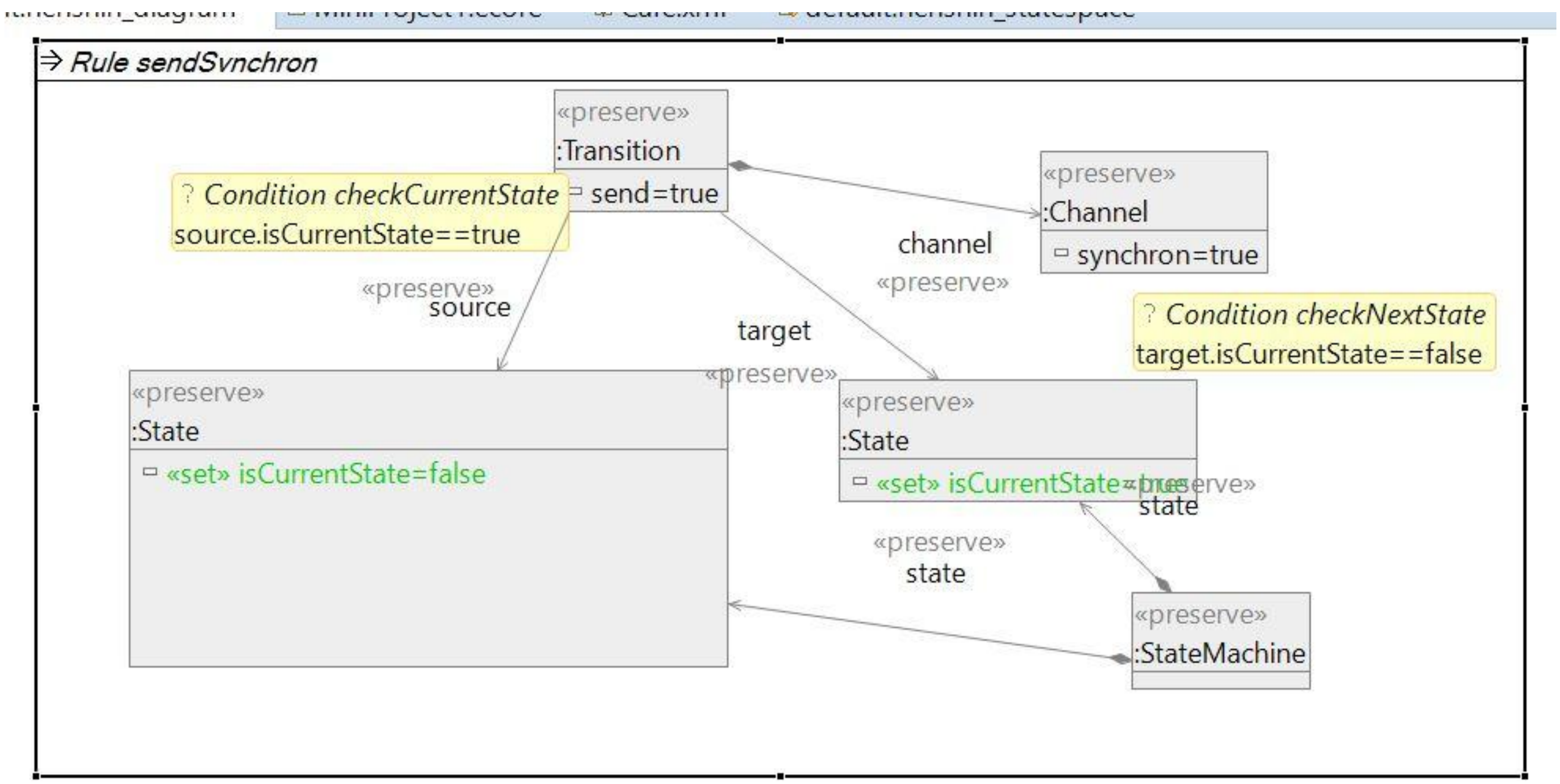


```
<terminated> RuntimeNetworkController [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (27.06.2016, 15:45:3
Buffers: pedestrianRequest:0 ready:0
 [ASYNC - pedestrianRequest SEND] PedestrianUI.buttonPressReceived to PedestrianUI.
SysState: Pedestrian:waiting PedestrianUI:waitForGreen TrafficLight:green Crossing
-----------------------
Buffers: pedestrianRequest:1 ready:0
 [ASYNC - pedestrianRequest RECV] CrossingControl.initial1 to CrossingControl.pedes
SysState: Pedestrian:waiting PedestrianUI:waitForGreen TrafficLight:green Crossing
-----------------------
Buffers: pedestrianRequest:0 ready:0
 [SYNC - pressButton SEND] Pedestrian.waiting to Pedestrian.waiting
 [SYNC - pressButton RECV] PedestrianUI.waitForGreen to PedestrianUI.waitForGreen
SysState: Pedestrian:waiting PedestrianUI:waitForGreen TrafficLight:green Crossing
-----------------------
Buffers: pedestrianRequest:0 ready:0
 [SYNC - trafficTurnRed SEND] CrossingControl.pedestrianRequestReceived to Crossing
 [SYNC - trafficTurnRed RECV] TrafficLight.green to TrafficLight.turningRed
SysState: Pedestrian:waiting PedestrianUI:waitForGreen TrafficLight:turningRed Cro
-----------------------
Buffers: pedestrianRequest:0 ready:0
 [SYNC - trafficTurnedRed SEND] TrafficLight.turningRed to TrafficLight.red
 [SYNC - trafficTurnedRed RECV] CrossingControl.turningTrafficLightsRed to Crossing
SysState: Pedestrian:waiting PedestrianUI:waitForGreen TrafficLight:red CrossingCo
-----------------------
Buffers: pedestrianRequest:0 ready:0
 [SYNC - pressButton SEND] Pedestrian.waiting to Pedestrian.waiting
 [SYNC - pressButton RECV] PedestrianUI.waitForGreen to PedestrianUI.waitForGreen
SysState: Pedestrian:waiting PedestrianUI:waitForGreen TrafficLight:red CrossingCo
-----------------------
Buffers: pedestrianRequest:0 ready:0
 [SYNC - pedestrianTurnGreen SEND] CrossingControl.trafficLightsTurnedRed to Crossi
 [SYNC - pedestrianTurnGreen RECV] PedestrianLight.red to PedestrianLight.turningGr
SysState: Pedestrian:waiting PedestrianUI:waitForGreen TrafficLight:red CrossingCo
```

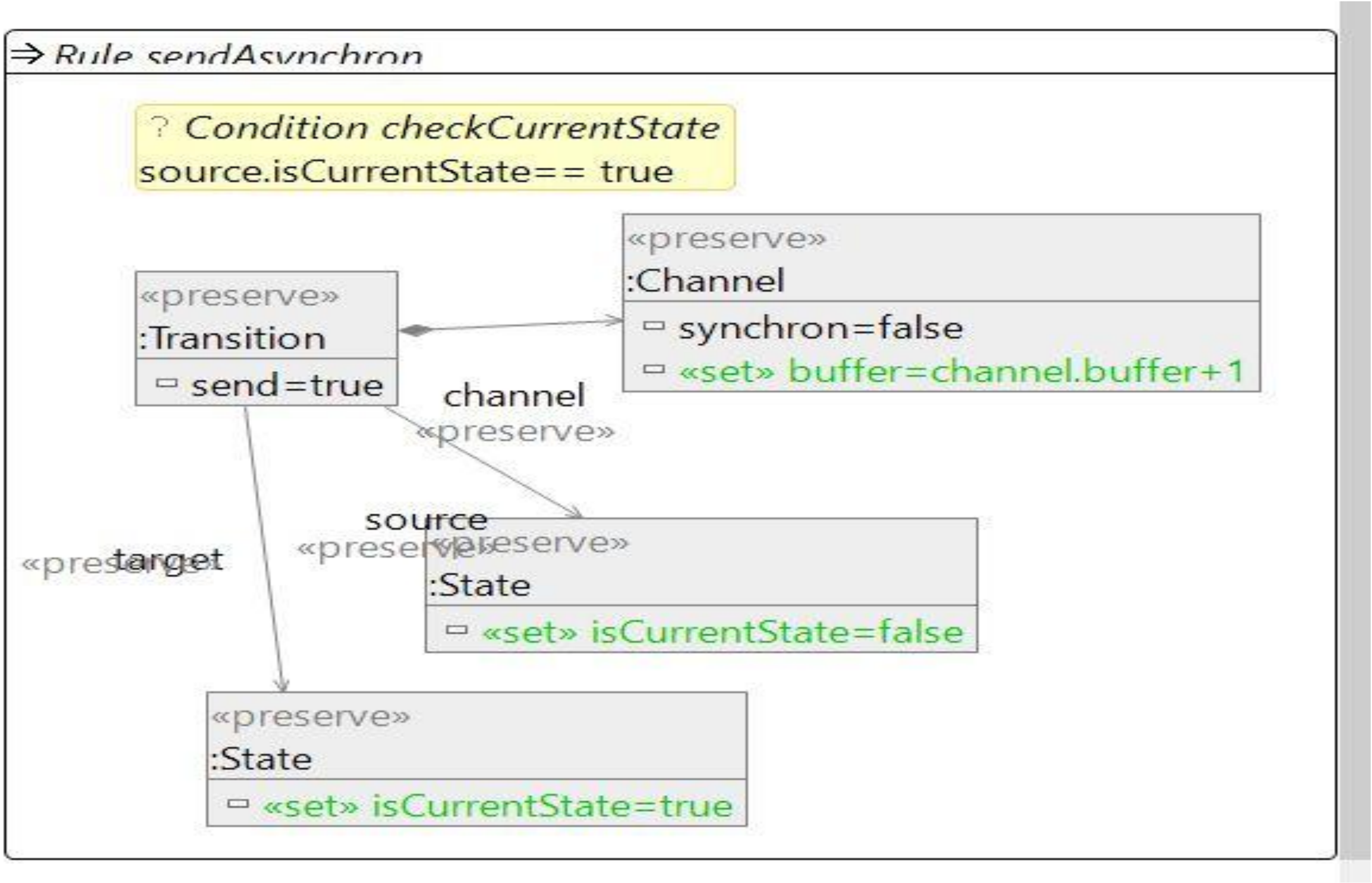
Henshin model

- Extend ecore-model with buffers for Channels and a boolean flag „isCurrentState“ for the State
- Creating a Henshin-model
- Define some rules

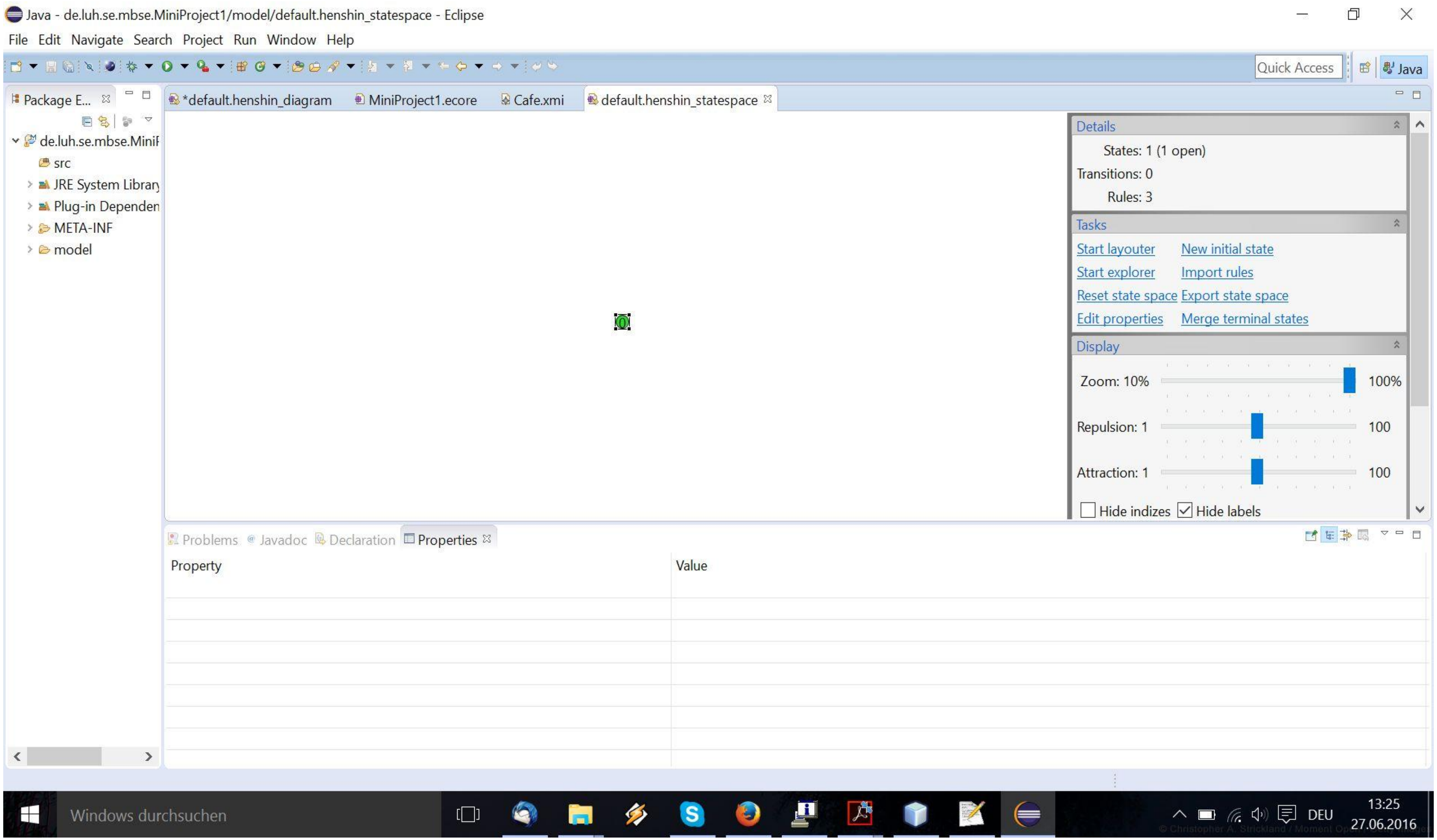
Henshin model



Henshin model



Henshin model



Java - de.luh.se.mbse.MiniProject1/model/default.henshin_statespace - Eclipse

Details

States: 1 (1 open)

Transitions: 0

Rules: 3

Display

Zoom: 10% 100%

Repulsion: 1 100

Attraction: 1 100

☐ Hide indizes ☑ Hide labels

Property	Value