Modelling Reactive Systems

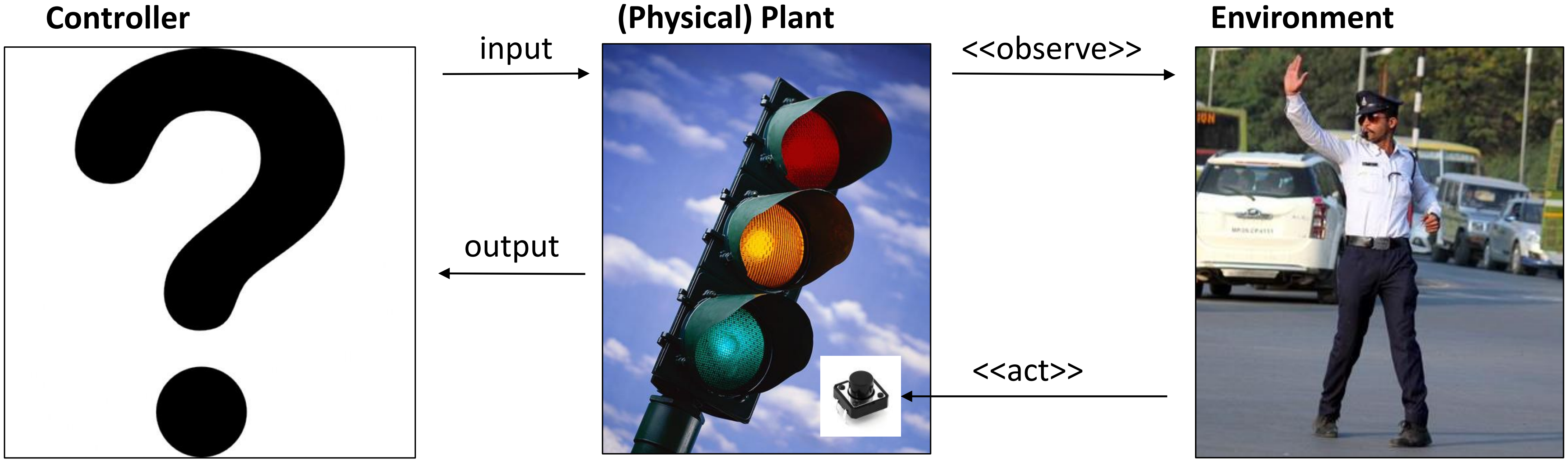
- Interaction with the environment: reactive to *events*.
- Autonomous behaviour: *timeouts*.
- Describe system with *modes* (hierarchical) and *concurrent* units.
- Use programming language + threads and timeouts (OS)?

“Nontrivial software written with threads, semaphores, and mutexes are incomprehensible to humans”[1]

Programming language (and OS) is too low-level
-> most appropriate formalism: “what” vs. “how”

[1] E. A. Lee, "The problem with threads," in *Computer*, vol. 39, no. 5, pp. 33-42, May 2006.

Running Example



Controller

(Physical) Plant

Environment

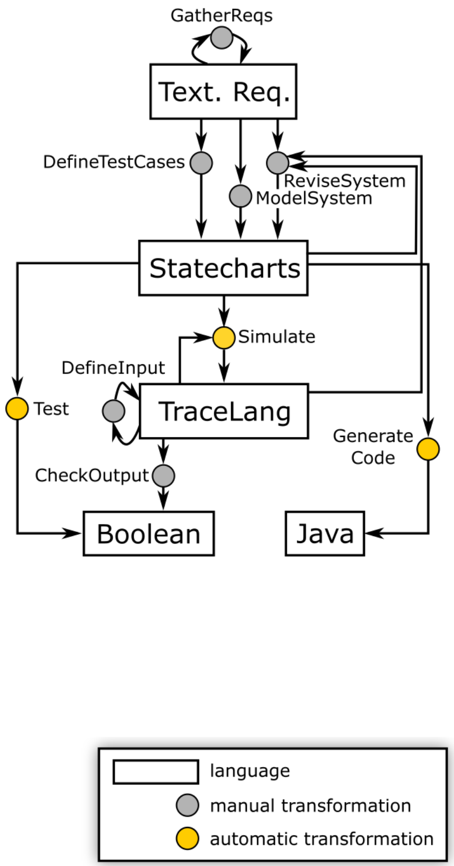
input

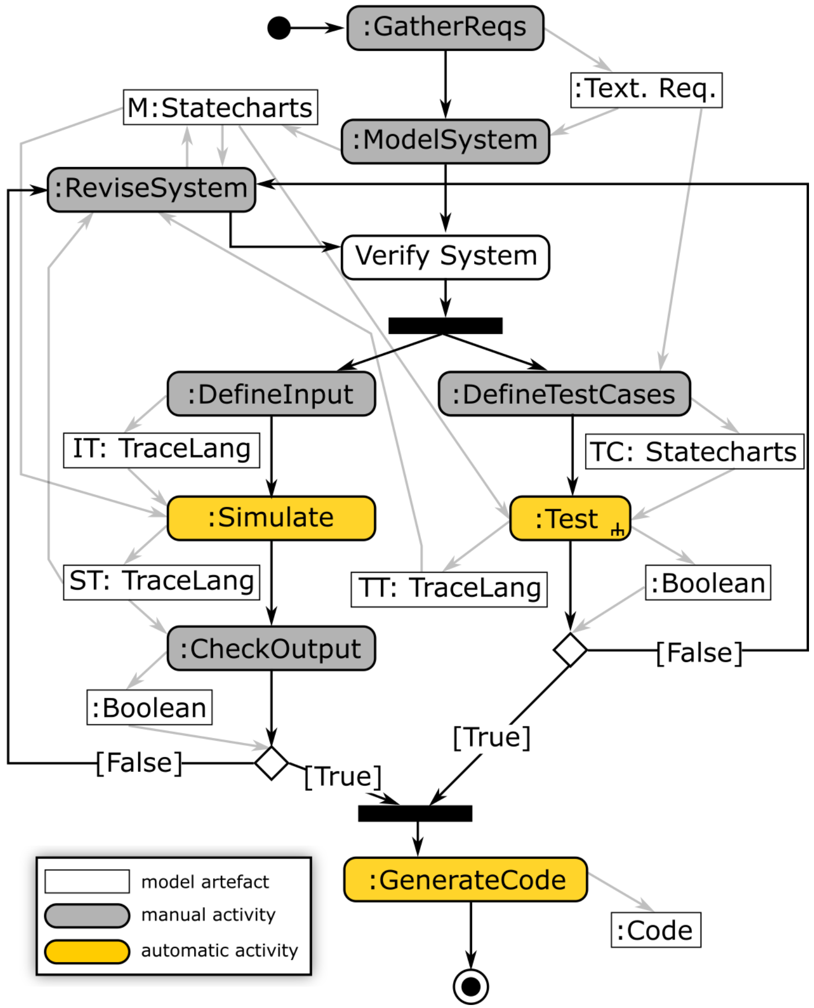
output

<<observe>>

<<act>>

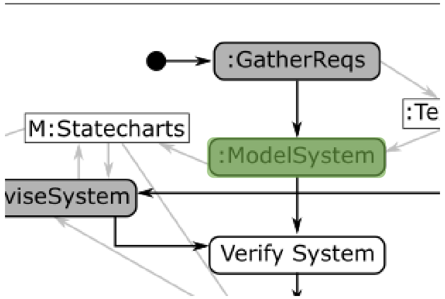
Workflow





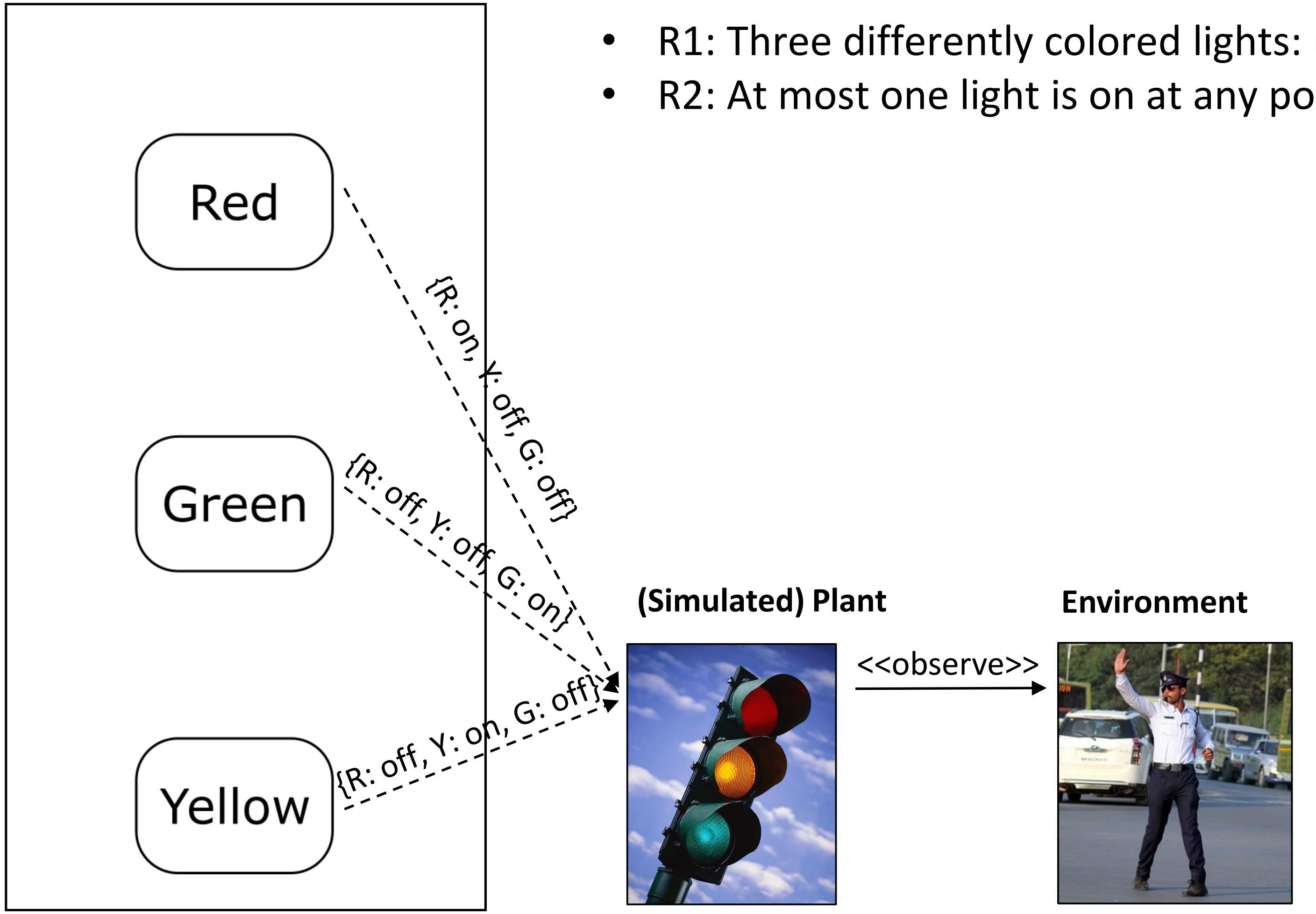
[3] Hans Vangheluwe and Ghislain C. Vansteenkiste. A multi-paradigm modeling and simulation methodology: Formalisms and languages. In European Simulation Symposium (ESS), pages 168-172. Society for Computer Simulation International (SCS), October 1996. Genoa, Italy.

[4] FTG+PM: An Integrated Framework for Investigating Model Transformation Chains, Levi Lúcio, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, Maris Jukss. System Design Languages Forum (SDL) 2013, Montreal, Quebec. LNCS Volume 7916, pp 182-202, 2013.

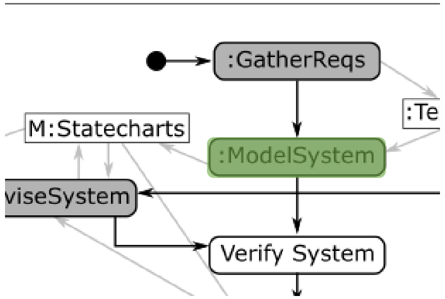


States

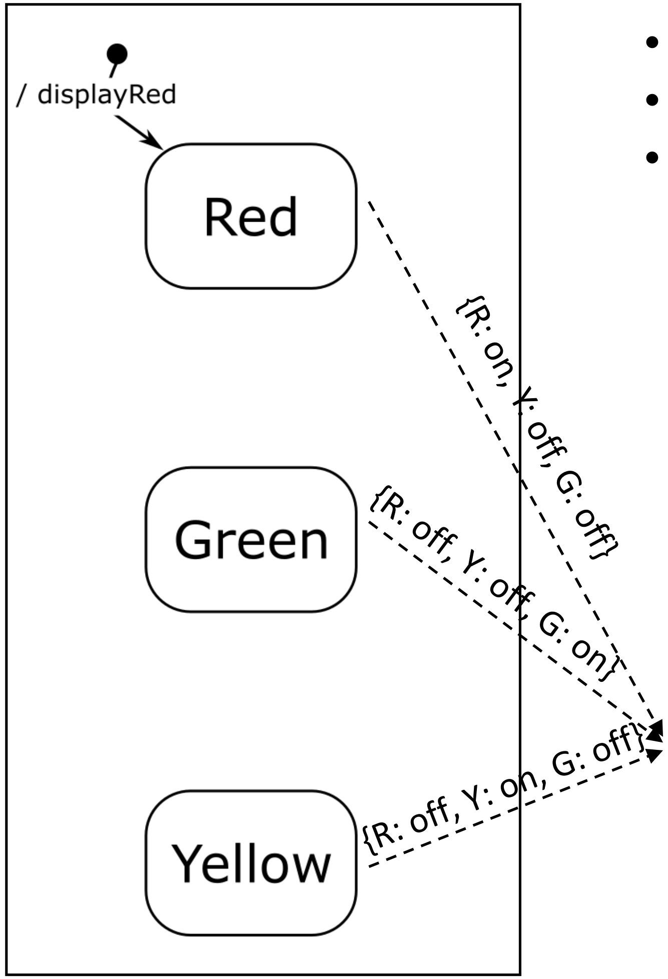
- R1: Three differently colored lights: red, green, yellow.
- R2: At most one light is on at any point in time.

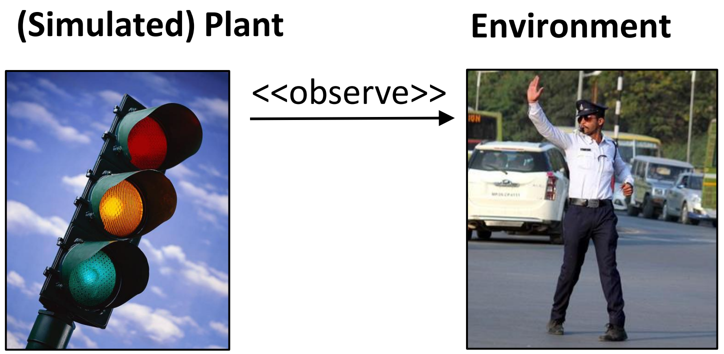


Default State

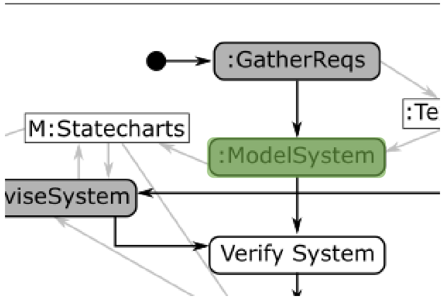


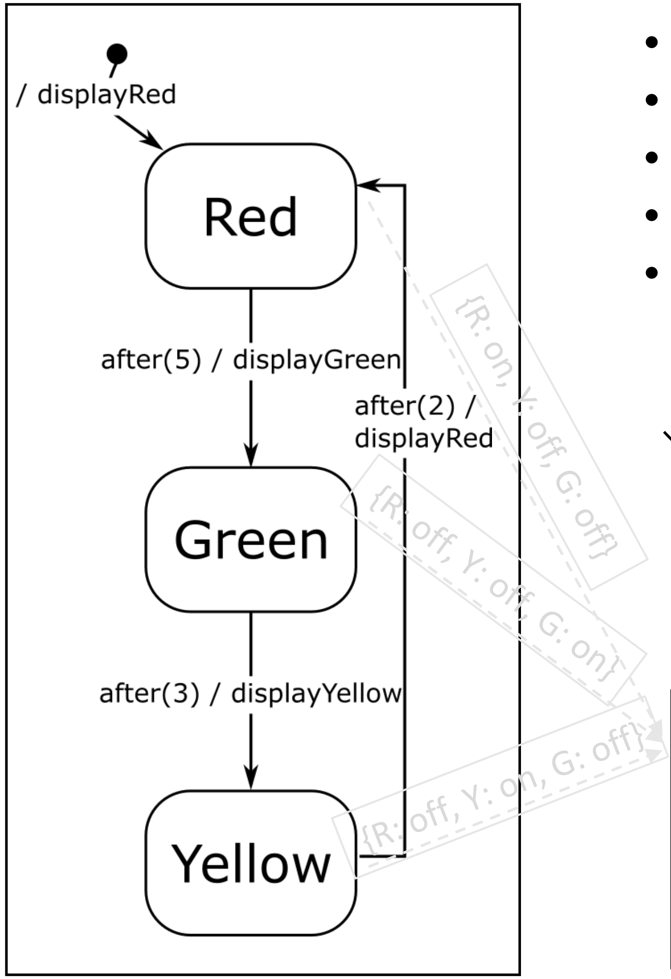
- R1: Three differently colored lights: red, green, yellow.
- R2: At most one light is on at any point in time.
- R3: Initially, the red light is on.





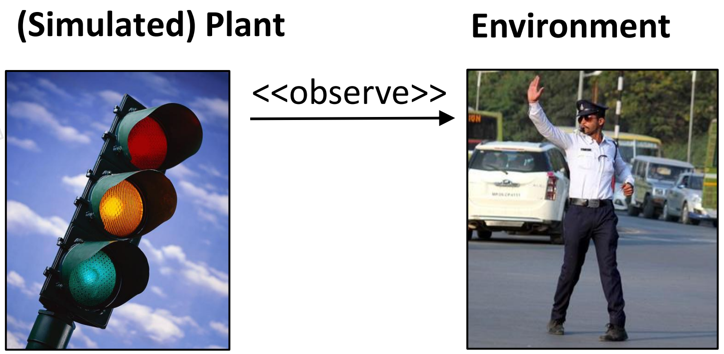
Transitions

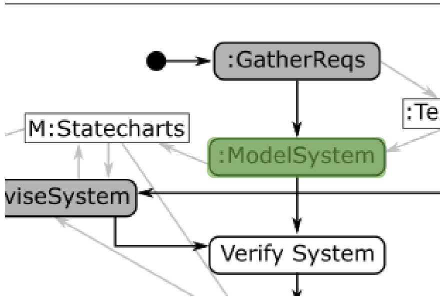




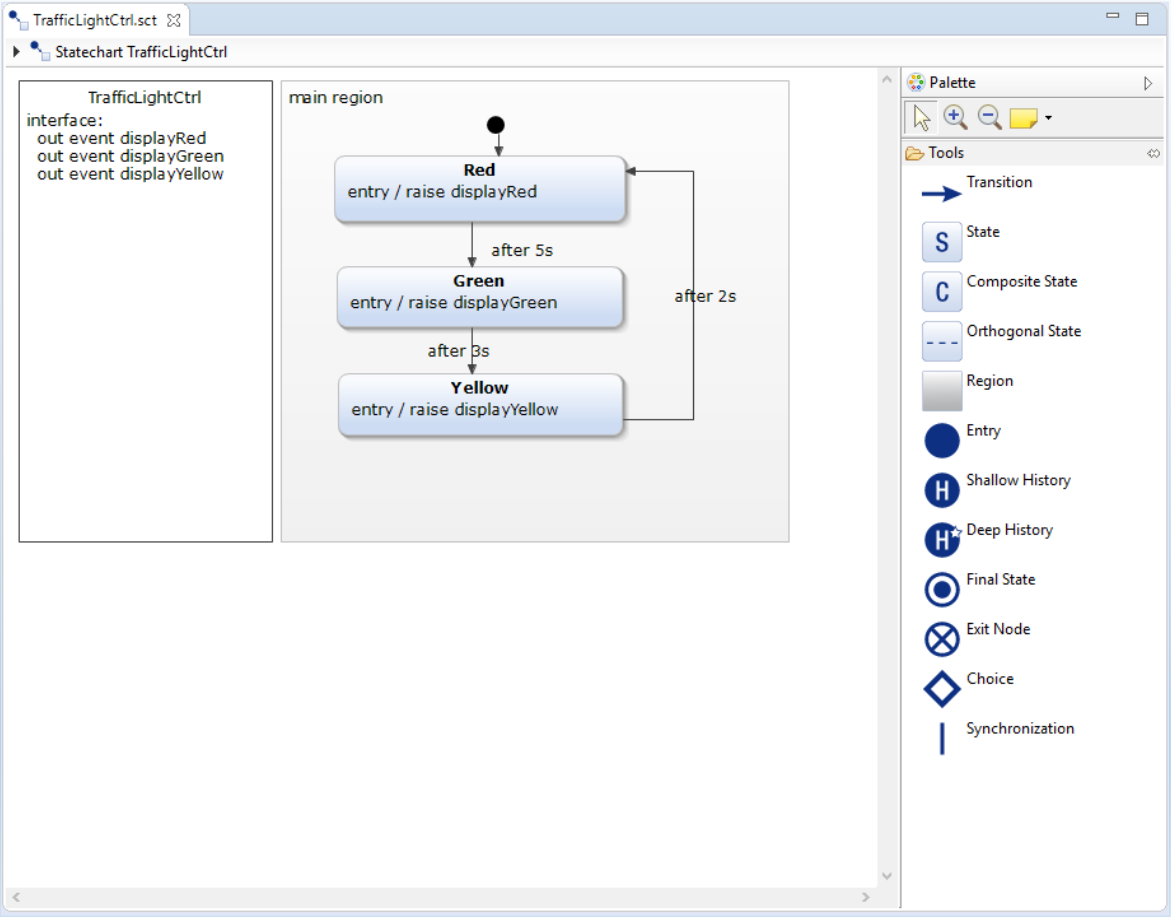
- R1: Three differently colored lights: red, green, yellow.
- R2: At most one light is on at any point in time.
- R3: Initially, the red light is on.
- R4: Cycles through red on, green on, and yellow on.
- R5: Duration: Red -> 5s, Green -> 3s, Yellow -> 2s

event(params) [guard] / output_action(params)





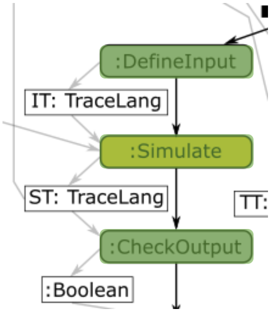
Yakindu[1]: Modelling

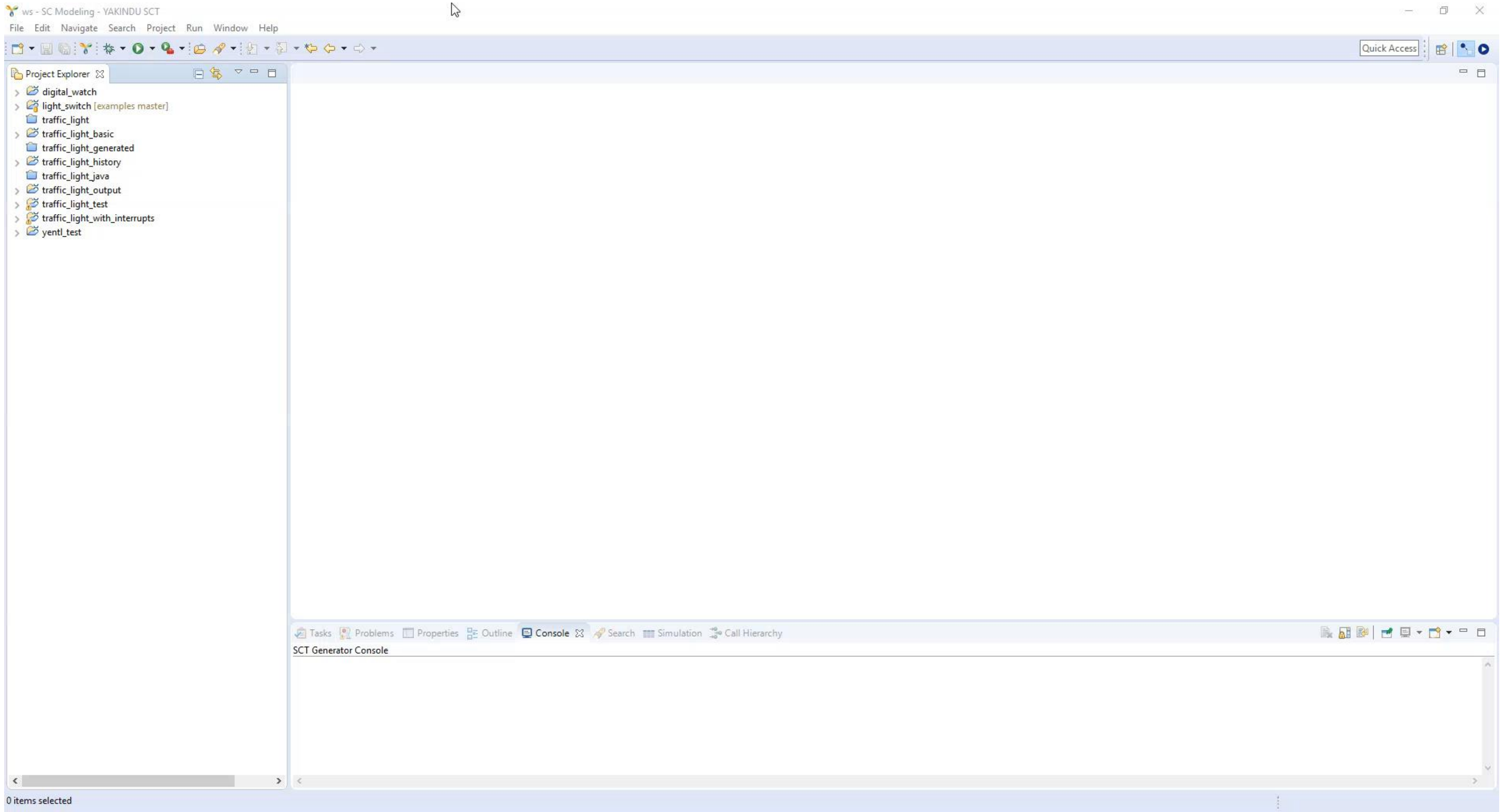


```
interface:
 out event displayRed
 out event displayGreen
 out event displayYellow
```

[5] https://www.itemis.com/en/yakindu/state-machine/

Yakindu: Simulation

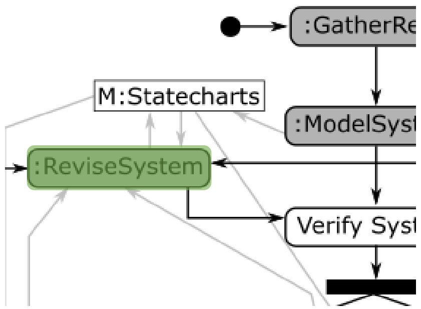



ws - SC Modeling - YAKINDU SCT
File Edit Navigate Search Project Run Window Help
Quick Access
Project Explorer
Tasks Problems Properties Outline Console Search Simulation Call Hierarchy

digital_watch
light_switch [examples master]
traffic_light
traffic_light_basic
traffic_light_generated
traffic_light_history
traffic_light_java
traffic_light_output
traffic_light_test
traffic_light_with_interrupts
yentl_test

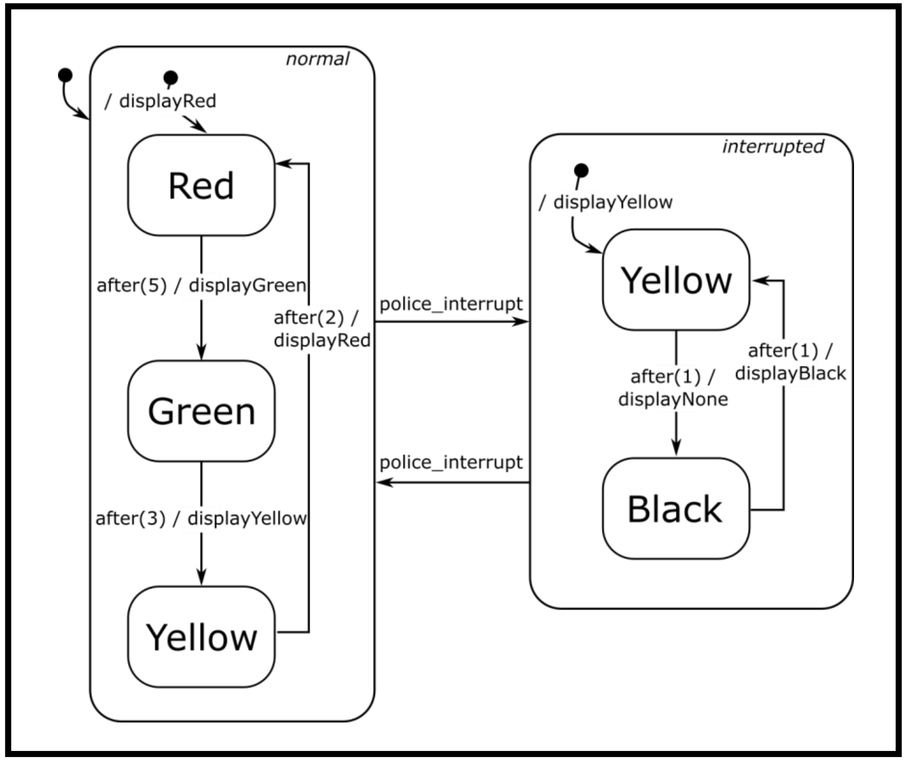
SCT Generator Console

0 items selected



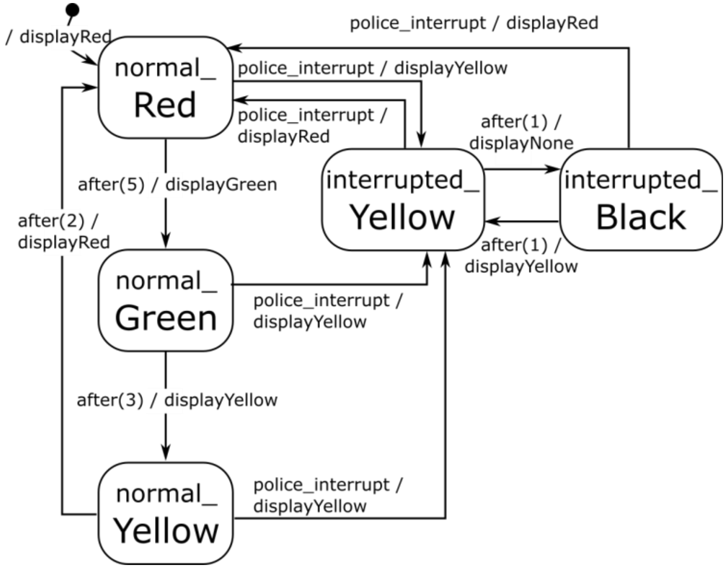
Hierarchy

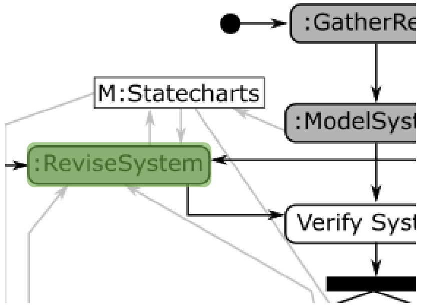
- R6: Police can interrupt autonomous operation.
  - Result = blinking yellow light (on -> 1s, off -> 1s).
- R7: Police can resume an interrupted traffic light.



Semantics/Meaning?

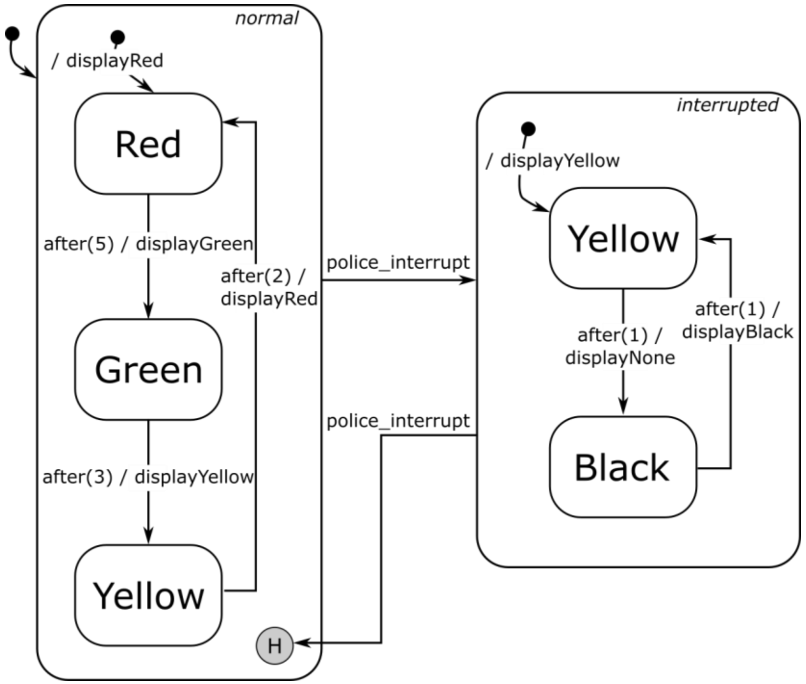
FLATTEN





History

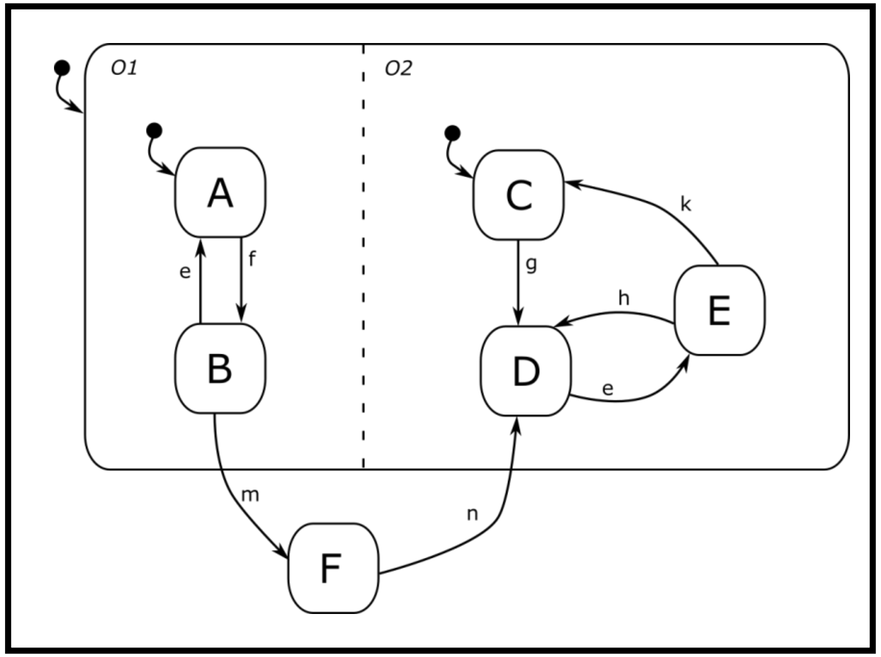
- R7: Police can resume an interrupted traffic light.
    - Result = light which was on at time of interrupt is turned on again.



H *shallow* history

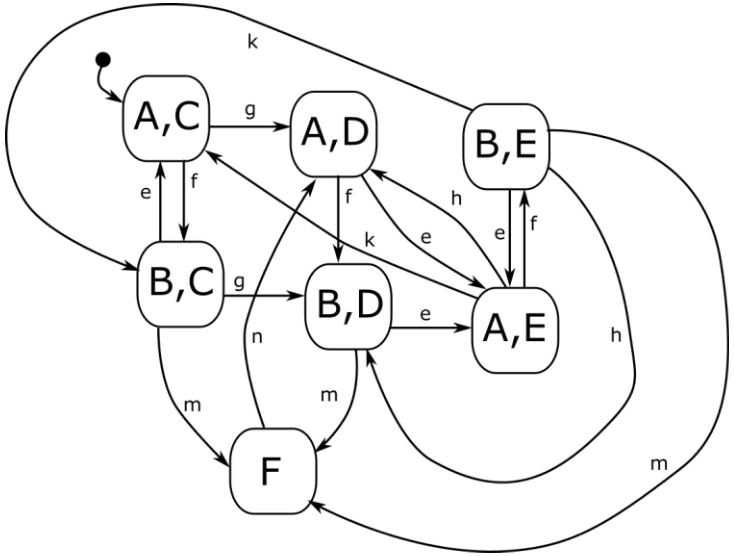
H* *deep* history

Concurrency

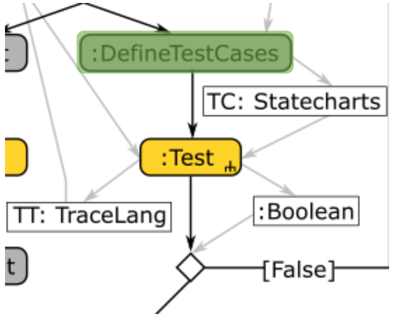


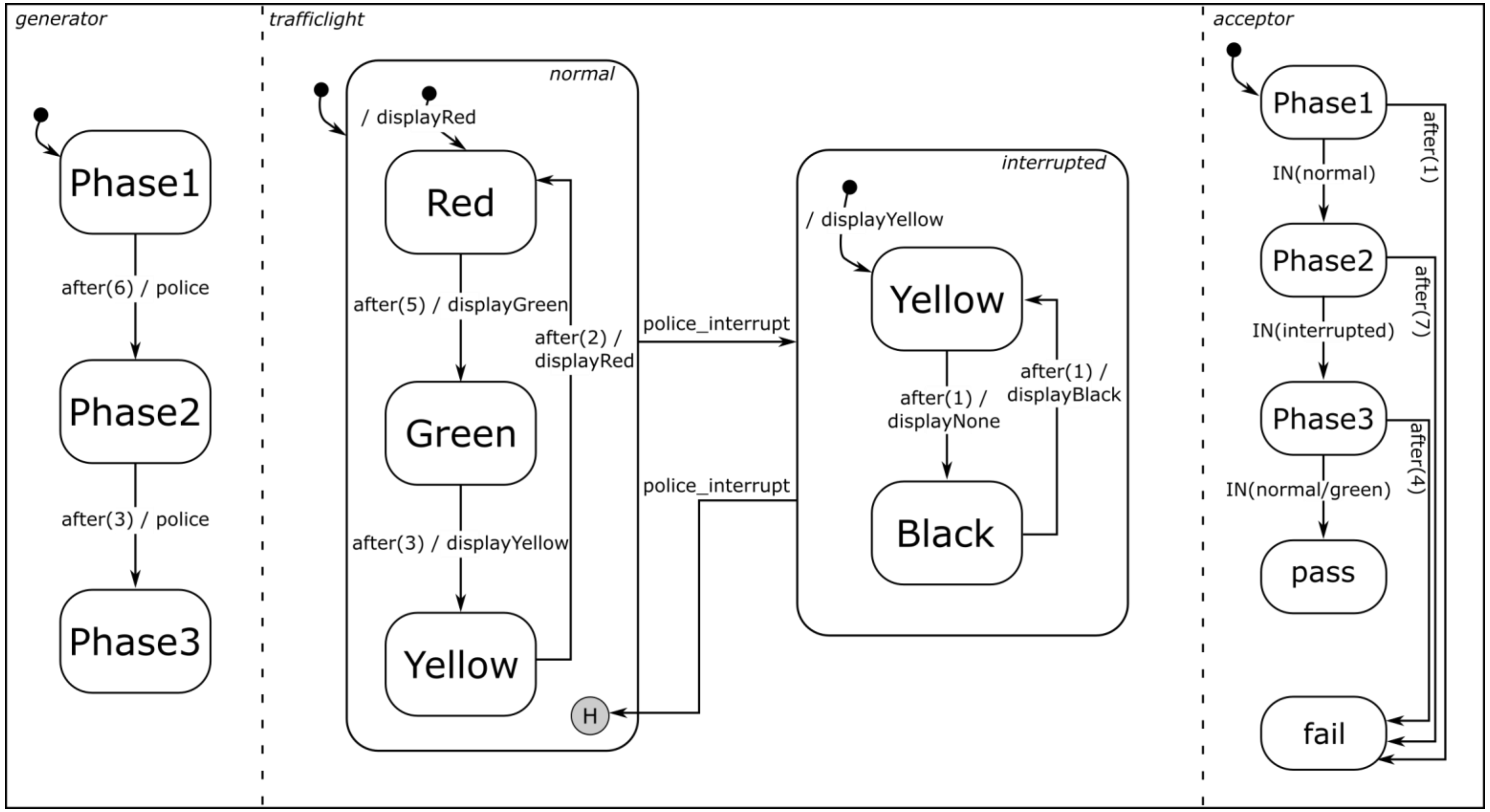
Semantics/Meaning?

FLATTEN

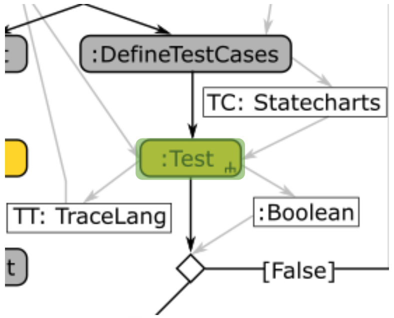


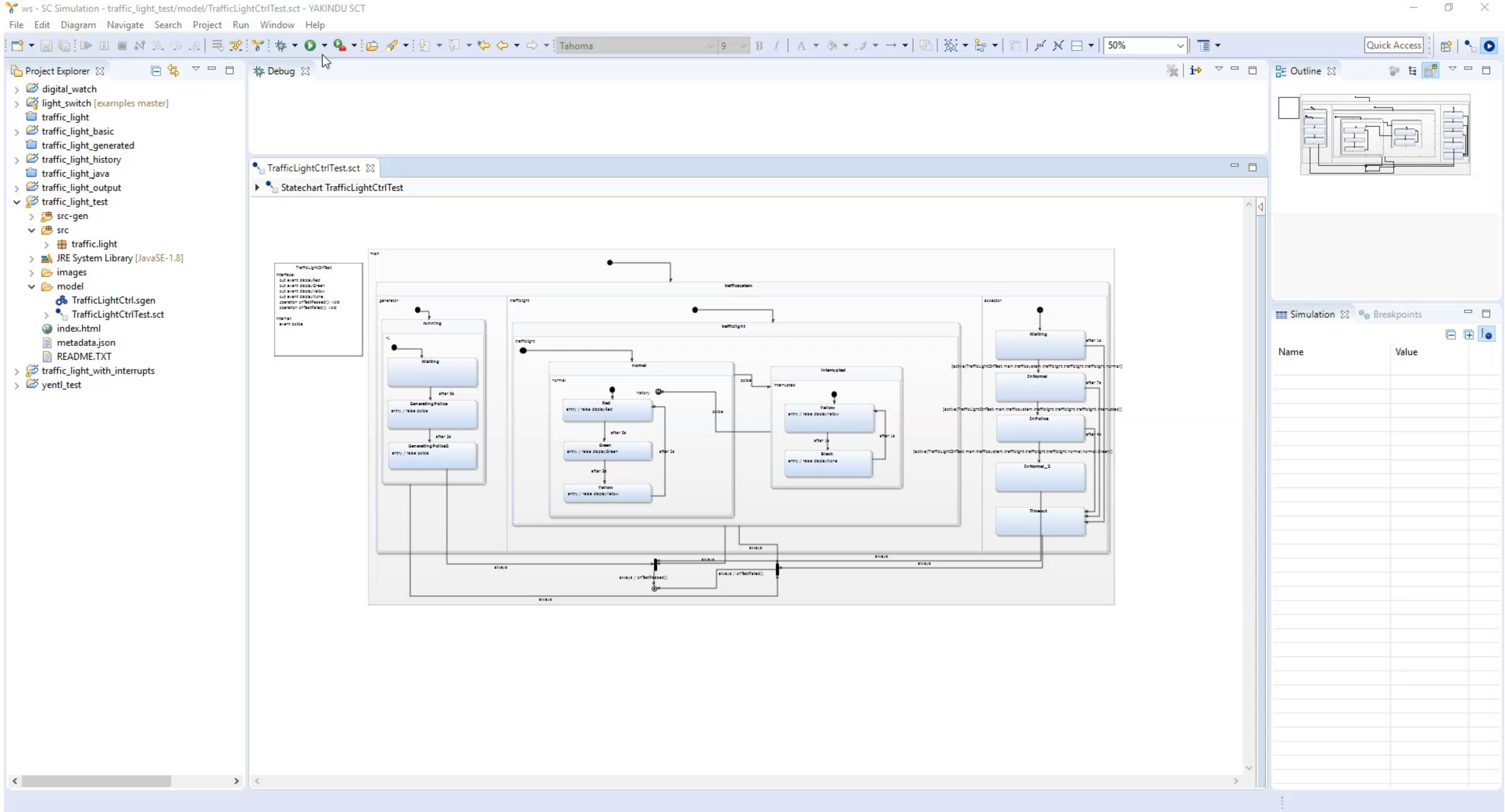
Statecharts Testing (White-Box)



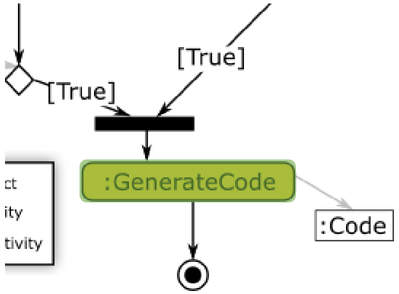


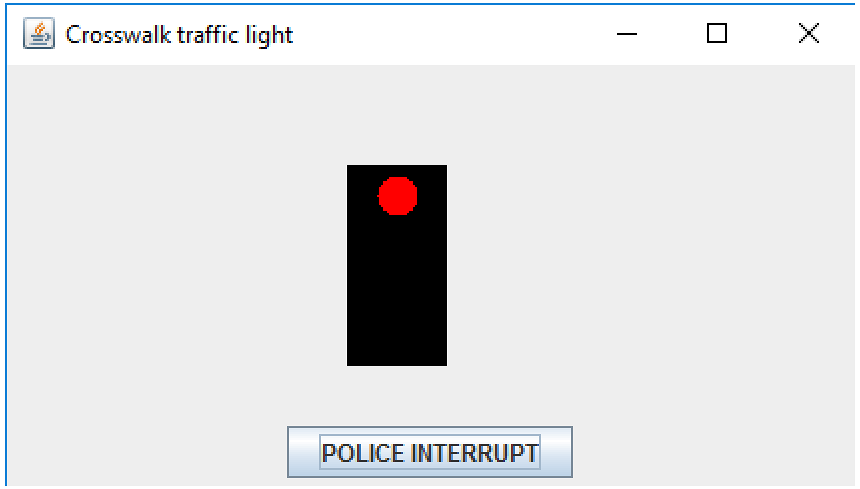
Yakindu: Testing





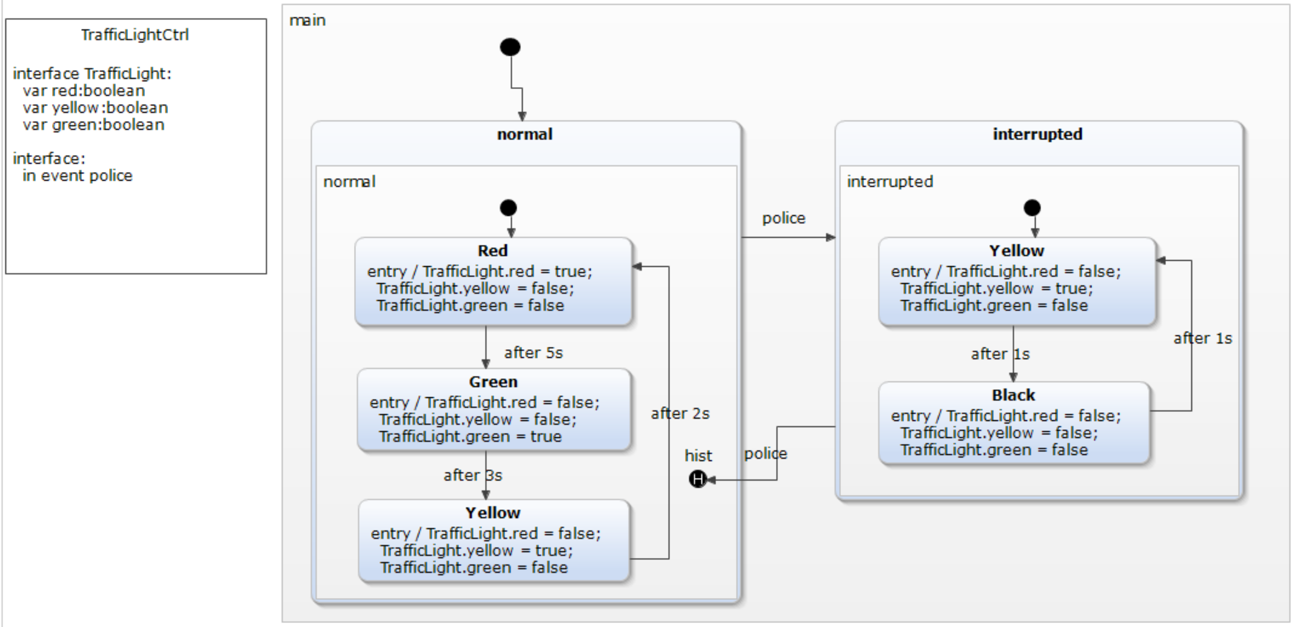
Code Generation





interrupts →

← events


TrafficLightCtrl

```
interface TrafficLight:
  var red:boolean
  var yellow:boolean
  var green:boolean

interface:
  in event police
```

main
normal
normal
Red
entry / TrafficLight.red = true;
TrafficLight.yellow = false;
TrafficLight.green = false
after 5s
Green
entry / TrafficLight.red = false;
TrafficLight.yellow = false;
TrafficLight.green = true
after 3s
Yellow
entry / TrafficLight.red = false;
TrafficLight.yellow = true;
TrafficLight.green = false
after 2s
hist
police
police
interrupted
interrupted
Yellow
entry / TrafficLight.red = false;
TrafficLight.yellow = true;
TrafficLight.green = false
after 1s
after 1s
Black
entry / TrafficLight.red = false;
TrafficLight.yellow = false;
TrafficLight.green = false

Interface:

- setRed(boolean)
- setGreen(boolean)
- setYellow(boolean)

Interface:

- in event *police*