Testing From State Machines

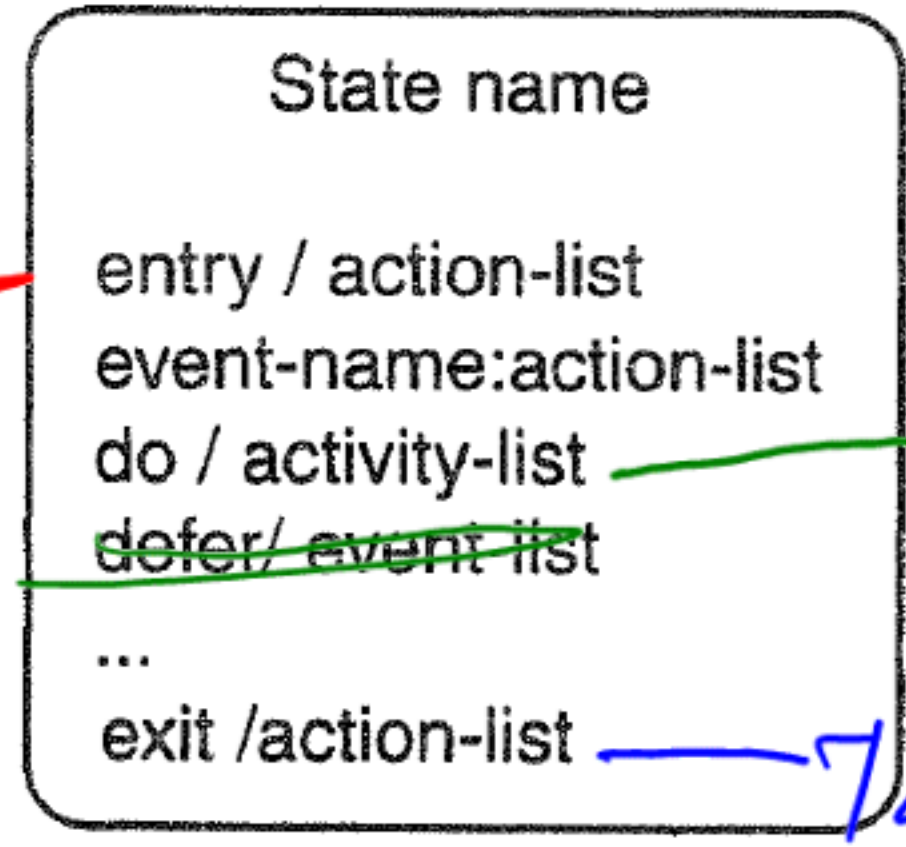
Lecture Objectives:

1) Define the concept of a finite state machine.
2) Define the terms state, transition, event, and action.
3) Explain the concept of a state transition table.
4) Given a state diagram, construct a state transition table for the problem.
5) Explain how state transition test cases can be constructed from a state machine definition.
6) Construct a set of test cases from a state diagram.
7) Implement a system which tests a state machine automatically using JUnit.

Defining Transitions

- Finite State Machine (FSM)
  - A Finite State Machine is a graph with nodes that represent states and edges that represent transitions
- UML State chart
  - The UML mechanism for representing finite state machines.

State Chart



State Icon

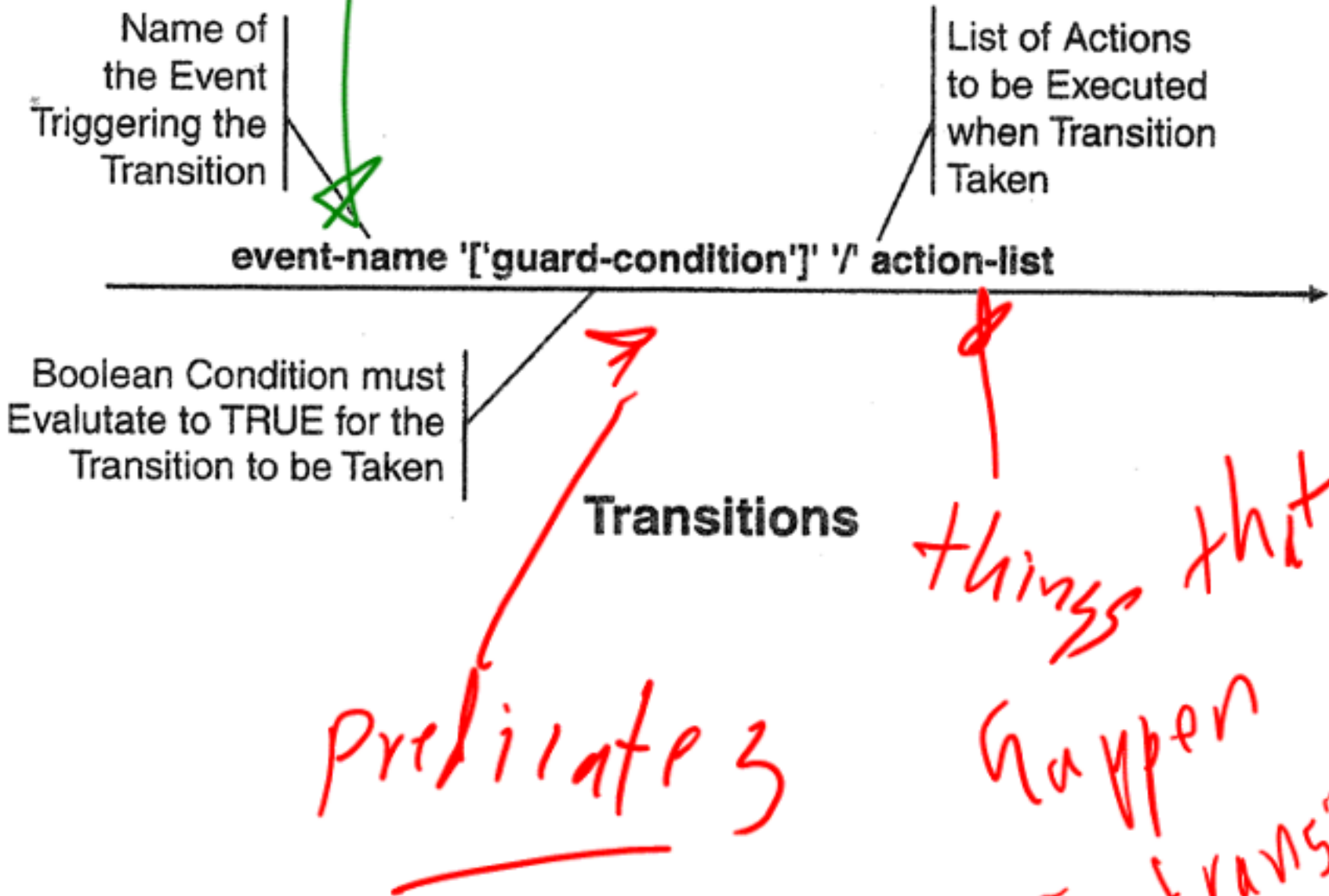
methods invoked when entering a state.

Indicates activity is performed.

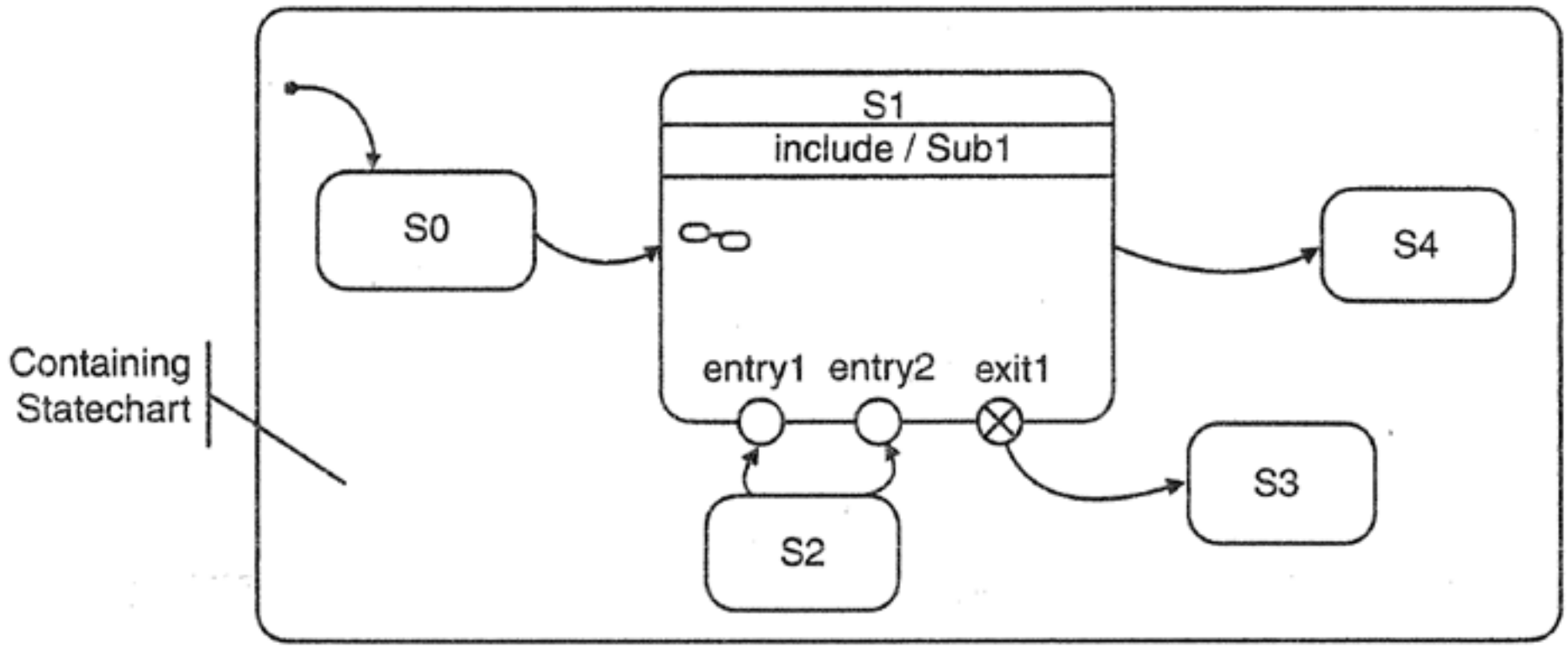
Things done when exiting state

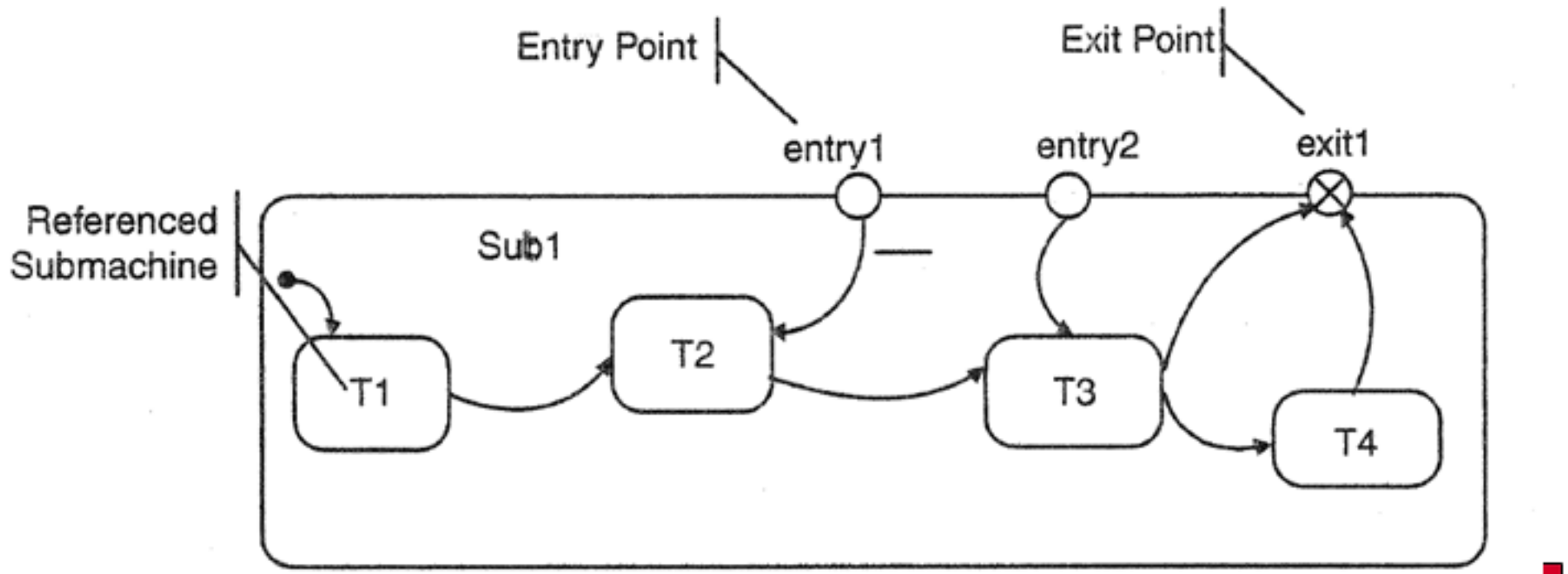
State Transitions

cause of the event.

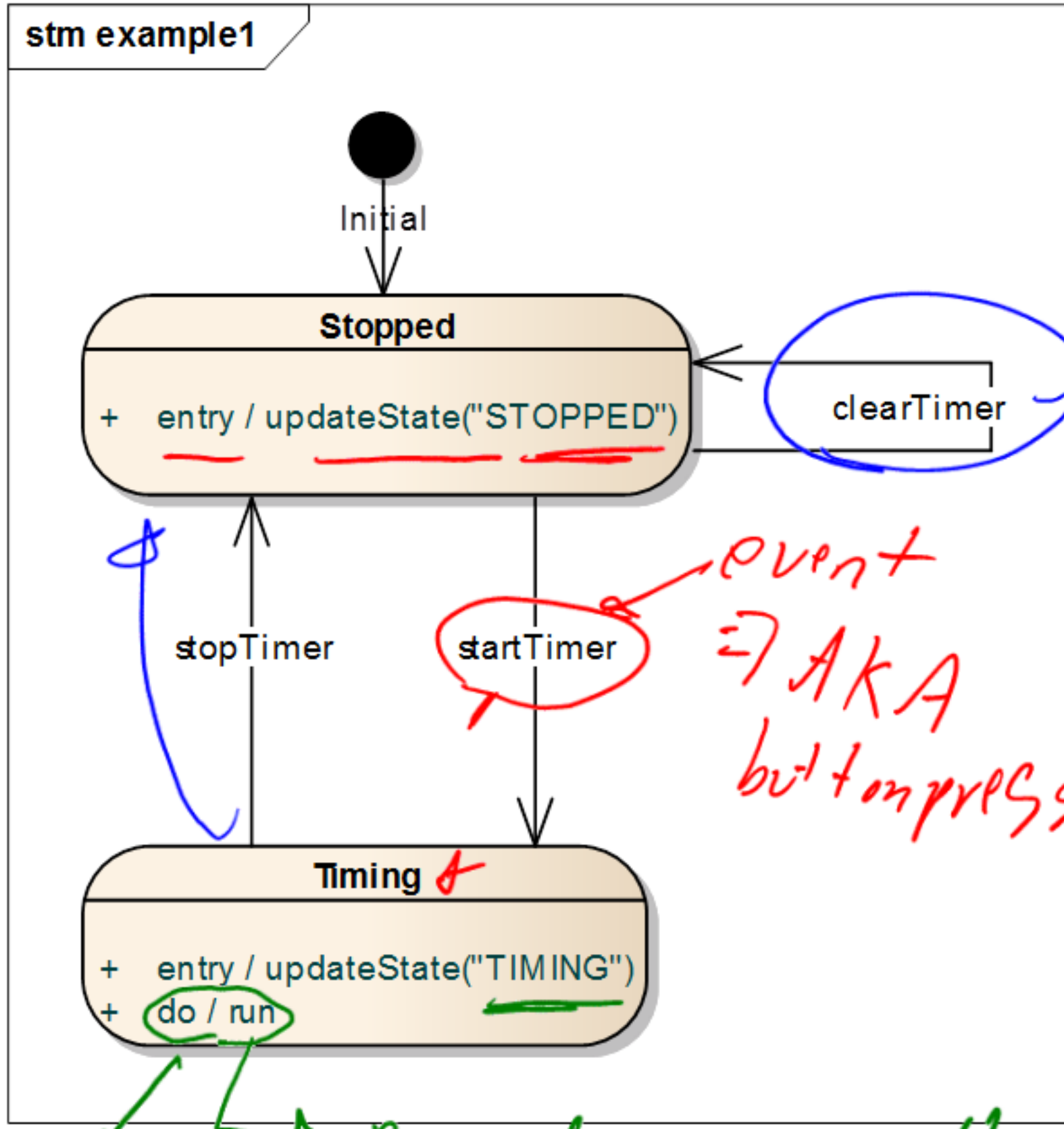


State Charts





A simple stopwatch



represents 1 impl.

class example1

«interface,observer»
StopwatchObserverInterface

+ updateTime(int) : void
+ updateState(String) : void

«interface»
StopwatchSubjectInterface

+ Subscribe(StopwatchObserverInterface) : void
+ unsubscribe, subjectInterface(StopwatchObserverInterface) : void

-observers

User interface

JPanel

«Real Observer»
StopWatchUI

- serialVersionUID: long = 1L {readOnly}
- timeDisplay: JTextField
- state: JTextField — show current state.
- ticker: TimeTicker
- CurrentTime: JTextField
+ updateTime(int) : void
+ StopWatchUI()
+ main(String[]) : void
+ updateState(String) : void

ticker

Runnable

«Real Subject»
TimeTicker

- keepRunning: boolean = true
- timing: boolean = false
- tmrThread: Thread
- time: int
- observers: List<StopwatchObserverInterface> = new LinkedList<...

+ updateTime(int) : void
+ updateState(String) : void
+ run() : void — Thread
+ clearTimer() : void
- startTimer() : void
+ stopTimer() : void
+ Subscribe(StopwatchObserverInterface) : void
+ ubsubscribe(StopwatchObserverInterface) : void

Methods invoked on entry/exit from states.

MSOE

State Table

State	Event	Guard Condition	Destination State
Stopped	startTimer	N/A	Timing
Stopped	clearTimer	N/A	Stopped
Timing	stopTimer	N/A	Stopped

Lets take a look at the code

Testing