State Machine

Designing components using a finite state machine model.

When State Matters

Some components behave differently depending on what *state* they are in.

Examples:

- Alarm Clock
- Calculator
- Stop Watch
- Point of Sale (POS) device
- most *parsers*

Simple Example: Stopwatch

Stopwatch *behaves differently* when it is running or stopped.

Identify States

Stopwatch states: RUNNING and STOPPED

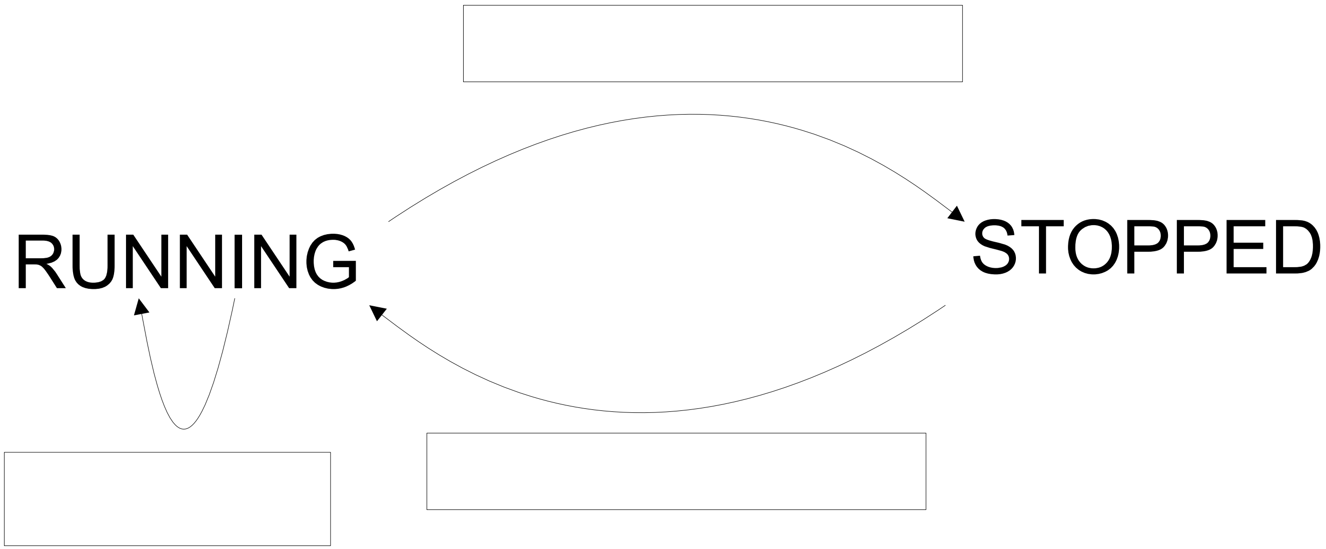
RUNNING

STOPPED

Events

Events are actions that can cause a state machine to change state.

What events cause a stopwatch to change state?

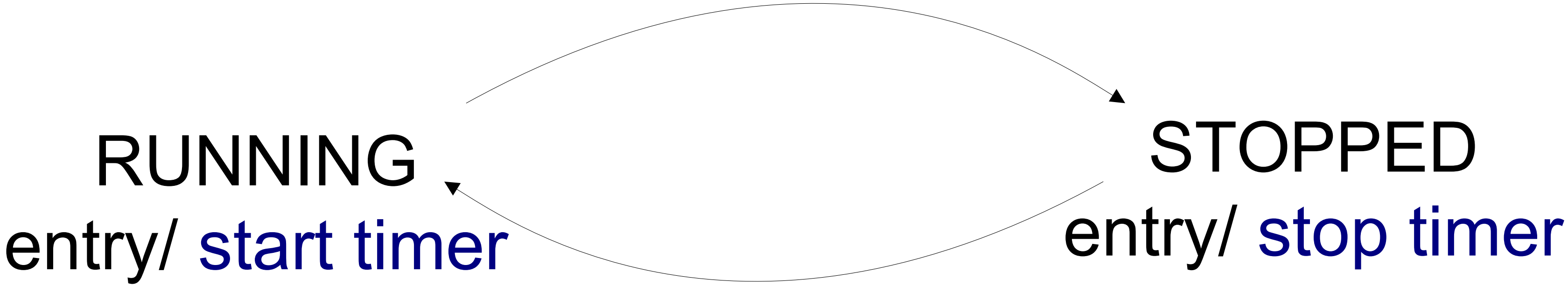


RUNNING

STOPPED

Action

The stopwatch performs some *action* in response to an event.



RUNNING

entry/ start timer

STOPPED

entry/ stop timer

Activities

An activity is something that lasts for some time.
An action is (nearly) instantaneous.

In the StopWatch UI, "*update display*" is an activity.

Programming a State Machine

Design the state machine first – step by step.

1. Identify the **states**
2. Identify **events***:* external and internally generated
3. Identify **actions** or **activities** the state machine performs in response to events or change in state.
4. Draw a state machine diagram.

Finally,

5. Code *state-dependent behavior* using state machine.

What behavior depends on state?

We use the attribute `running` to keep track of state.

```
class StopWatch {
    private boolean running;

    public void start( ) {
        if (running) return;
        startTime = System.nanoTime();
        running = true; // change state
    }
```

What behavior depends on state?

```
public double getElapsed() {
    if (running)
        return (System.nanoTime()-startTime)
                * NANOSECONDS;
    else
        return (stopTime-startTime)
                * NANOSECONDS;
}
public void stop() {
    if (! running ) return;
    stopTime = System.nanoTime();
    running = false;
}
```

The State Variable

We used a `boolean` variable to record the state.
This works when there are only 2 states.
For more states we need another type of state variable.

Consider: a StopWatch with Start, Stop, and Hold states.

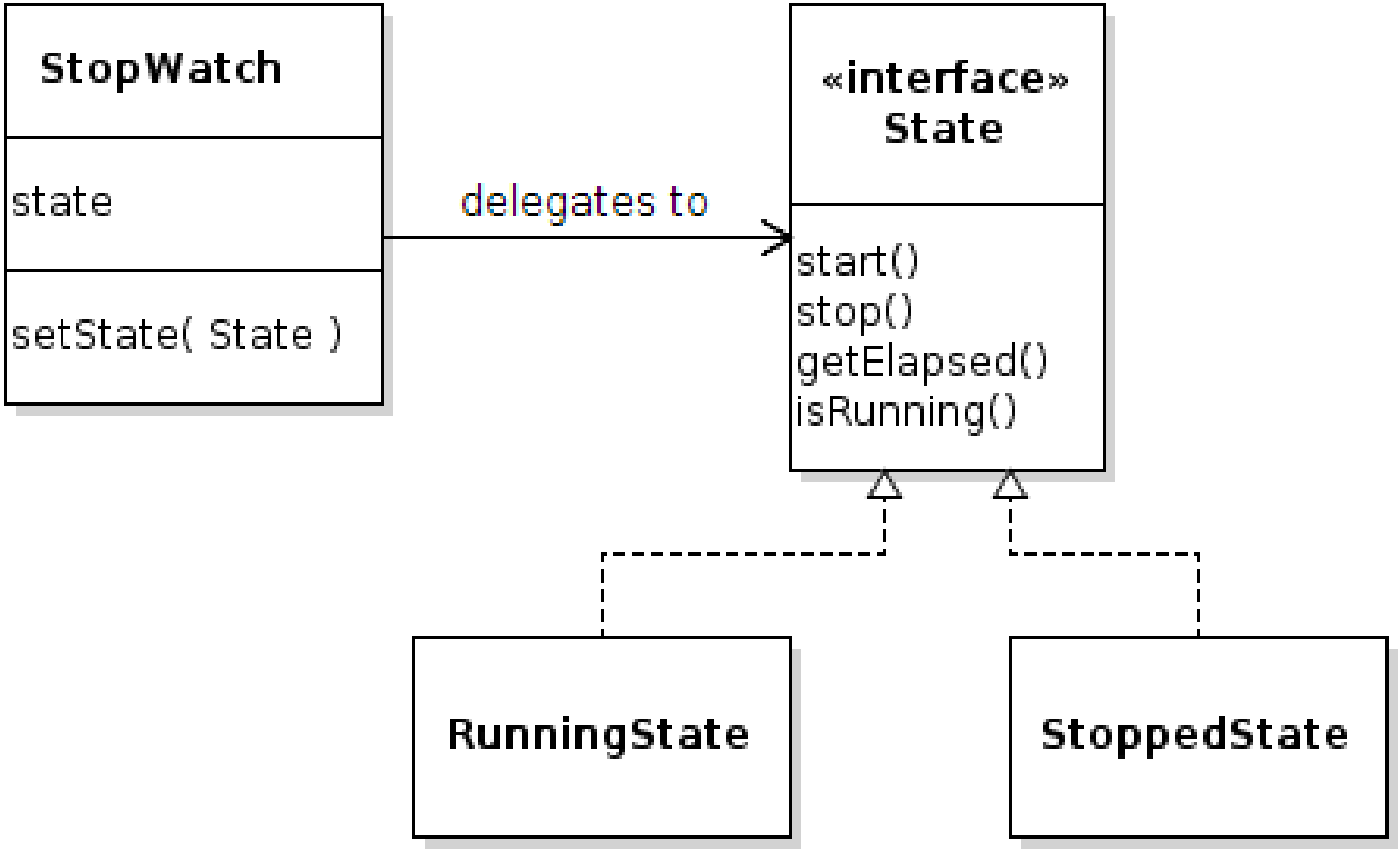
2 ways to represent state

```
// use "int" or "char"
class StopWatch {
    int state;
    final int STOPPED = 0;
    final int RUNNING = 1;
    final int HOLDING = 2;
    public void start( ) {
       if (state == RUNNING)
```

```
// use an enum
public enum State {
    STOPPED,
    RUNNING,
    HOLDING;
}
class StopWatch {
   State state;
   public void start() {
        if (state == RUNNING)
```

The O-O Approach

Use *Objects* to encapsulate state and the behavior that depends on state.
The *context* delegates behavior to state objects.



Delegating Behavior

Delegate means "let someone else do it".
Stopwatch delegates behavior to the **`state`**.

```
public class StopWatch {
    private State state;

    public void start( ) { state.start(); }
    public void stop()   { state.stop(); }
    public double getElapsed() {
        return state.getElapsed();
    }
```

State Objects and Changing State

The *context* (StopWatch) needs a setState method as a way of changing the state.

```
// Create states with a reference to the
// stopwatch
final State RUNNING = new RunningState(this);
final State STOPPED = new StoppedState(this);
private State state = STOPPED;

// provide a method for changing the state
public void setState(State newstate) {
    this.state = newstate;
}
```

Example of Changing State

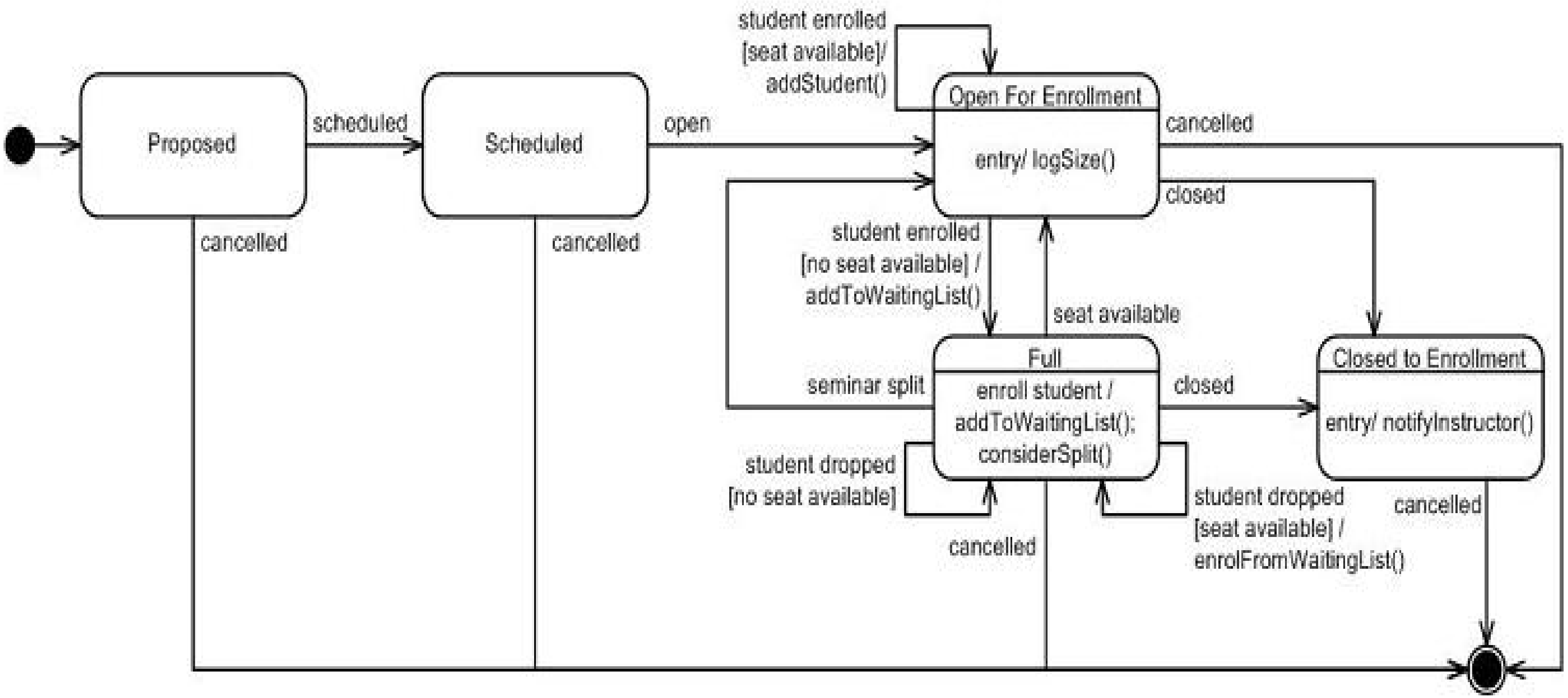
If the stopwatch is running and the Stop button is pressed, then change to stopped state...

```
class RunningState implements State {
   private StopWatch context;

   public void stop() {
      context.stopTime = System.nanoTime();
      context.setState( context.STOPPED );
   }

   public void start() {
      // already running so do nothing
   }
```

UML State Machine Diagram

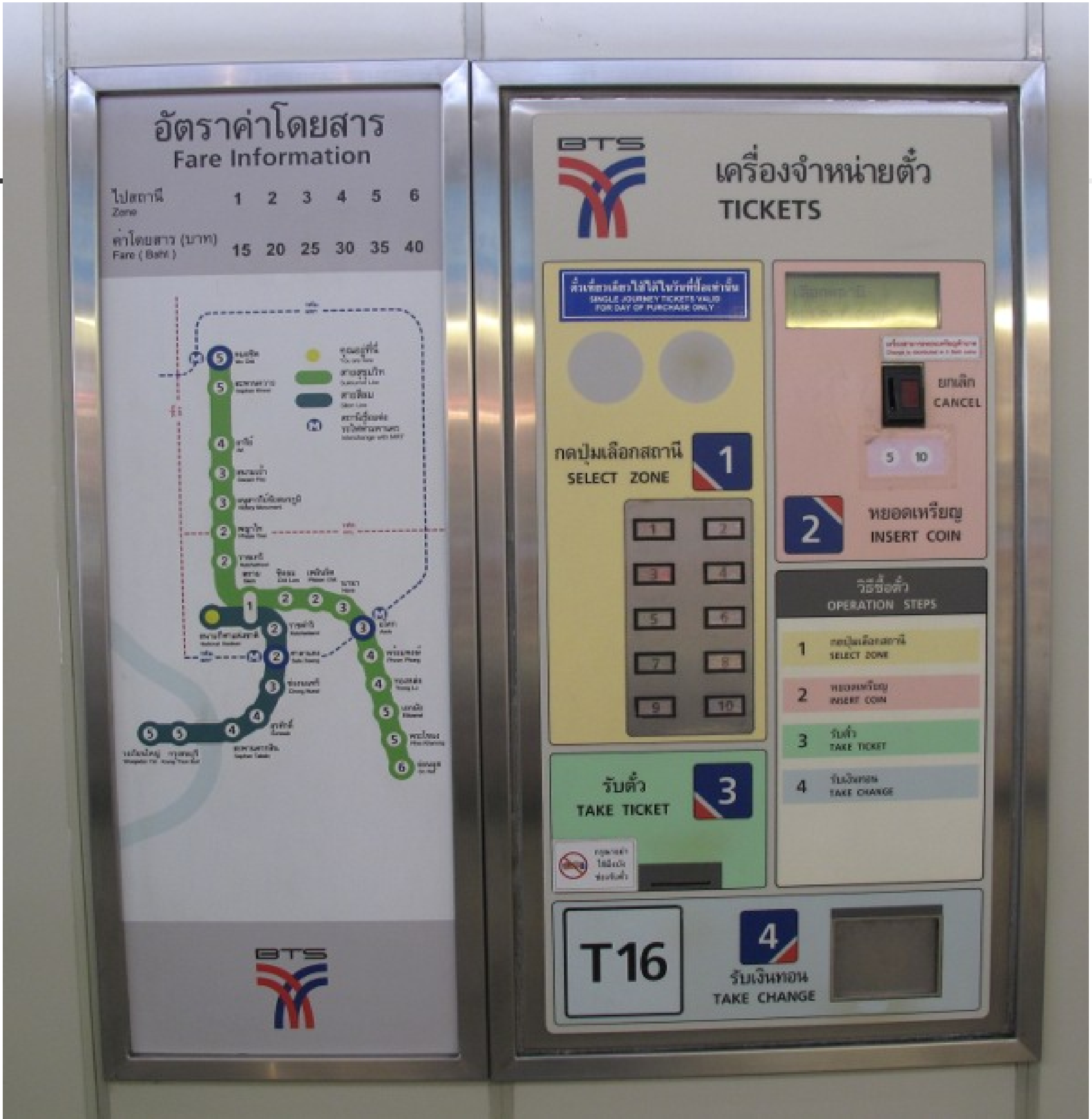


UML State Machine Diagram

Read *UML Distilled*, chapter 10.
Also good: *UML for Java Programmers,* chapter 10.

Exercise: Skytrain Ticket Machine

1. What are the states.
2. What are the events.
3. What actions/activities does ticket machine perform?
4. Draw a UML State Machine Diagram.


อัตราค่าโดยสาร
Fare Information
ไปสถานี
Zone
1 2 3 4 5 6
ค่าโดยสาร (บาท)
Fare (Baht)
15 20 25 30 35 40
BTS
BTS
เครื่องจำหน่ายตั๋ว
TICKETS
SINGLE JOURNEY TICKETS VALID FOR DAY OF PURCHASE ONLY
กดปุ่มเลือกสถานี
SELECT ZONE
1
1 2 3 4 5 6 7 8 9 10
ยกเลิก
CANCEL
5 10
2
หยอดเหรียญ
INSERT COIN
วิธีซื้อตั๋ว
OPERATION STEPS
1 กดปุ่มเลือกสถานี SELECT ZONE
2 หยอดเหรียญ INSERT COIN
3 รับตั๋ว TAKE TICKET
4 รับเงินทอน TAKE CHANGE
รับตั๋ว
TAKE TICKET
3
T16
4
รับเงินทอน
TAKE CHANGE

Exercise: Syllable Counter

Count the syllables in a word.
As a heuristic, we will count *vowel sequences*.
Example:
object = (o)bj(e)ct = 2 vowel sequences
beauty = b(eau)t(y) = 2 vowel sequences
Special cases:
l(a)y(ou)t = *treat "y" as consonant after other vowel*
l(a)the = don't count final "e" if it is a single vowel
m(o)v(ie) = 2 vowel seq. "final e" rule doesn't apply here.
th(e) = exception. count final "e" if it is *only* vowel
anti-oxident = (a)nt(i)-(o)x(i)d(e)nt   "-" is non-vowel

Example Words

How many vowel sequences in these words:

```
remarkable
selfie
county
coincidentally
she
mate
isn't
```

Exercise: Calculator

A calculator that behaves like Windows calc.
Use: http://www.online-calculator.com

1. What are the states.
2. What are the events.
3. What actions/activities does ticket machine perform?
4. Draw a UML State Machine Diagram.

(not so easy)

PA5: Cheap Digital Clock

cheap digital alarm clock.

Use states!