

OO REACTIVE PROGRAMMING IS NOT REACTIVE OO PROGRAMMING

ELISA GONZALEZ BOIX, KEVIN PINTE, SIMON VAN DE WATER,
WOLFGANG DE MEUTER

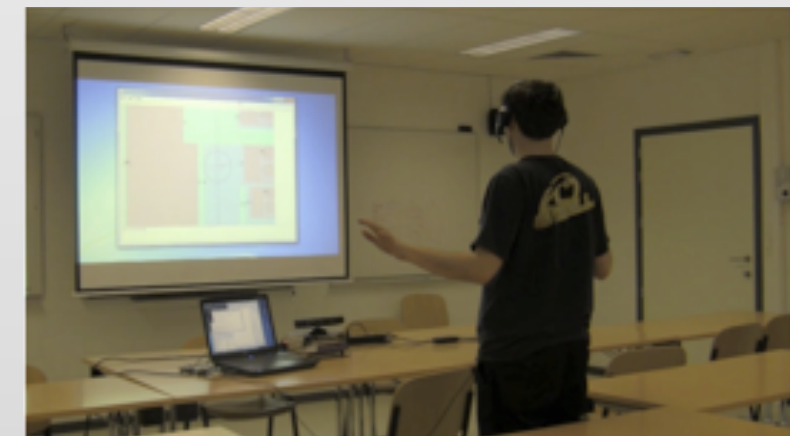


A RESEARCH AGENDA...



PROGRAMMING MOBILE WIRELESS APPLICATIONS

REM

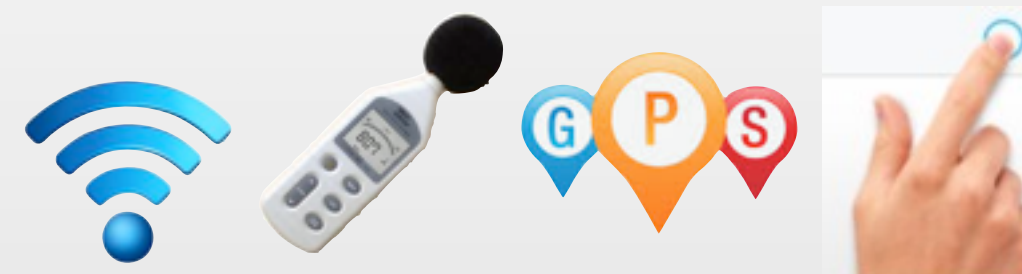
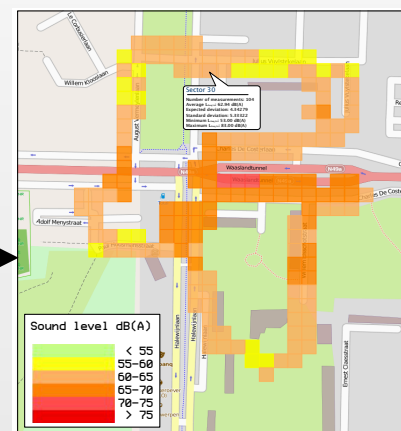


STATEFUL APPLICATIONS

NoiseTube

Server:
18.000 LOC

Mobile app:
13.000 LOC



MANAGING EVENT-
DRIVEN COMPUTATION
IN OO

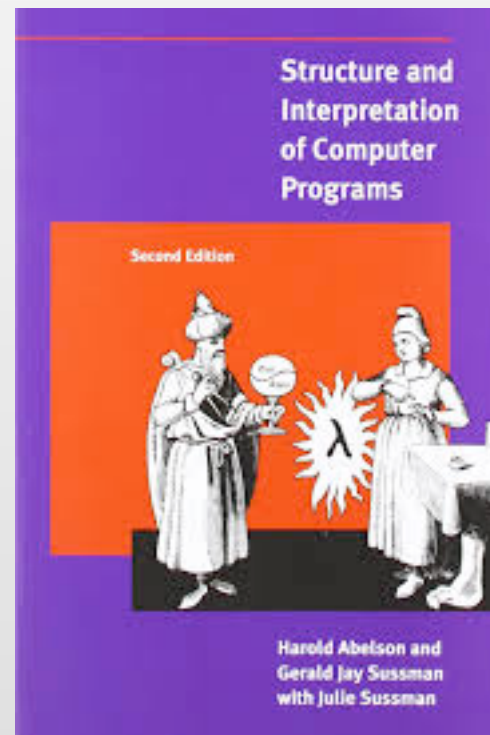


HOW TO RECONCILE RP WITH STATEFUL OBJECTS?

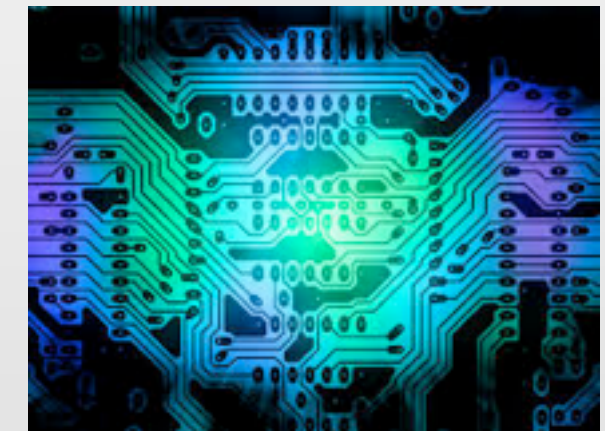
REM

(CHAPTER 3)

ORGANIZE A SYSTEM IN A MODULAR WAY



ACCORDING TO
OBJECTS THAT LIVE
IN THE SYSTEM

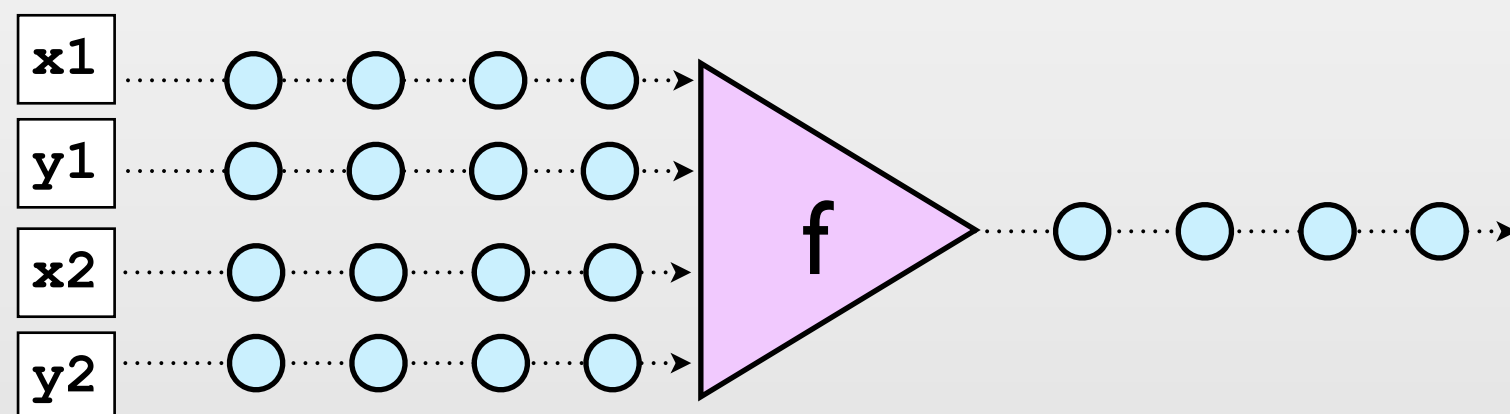


ACCORDING TO
STREAMS OF
INFORMATION THAT
FLOW IN THE SYSTEM

OBJECTS A.K.A. COMPOUND DATA

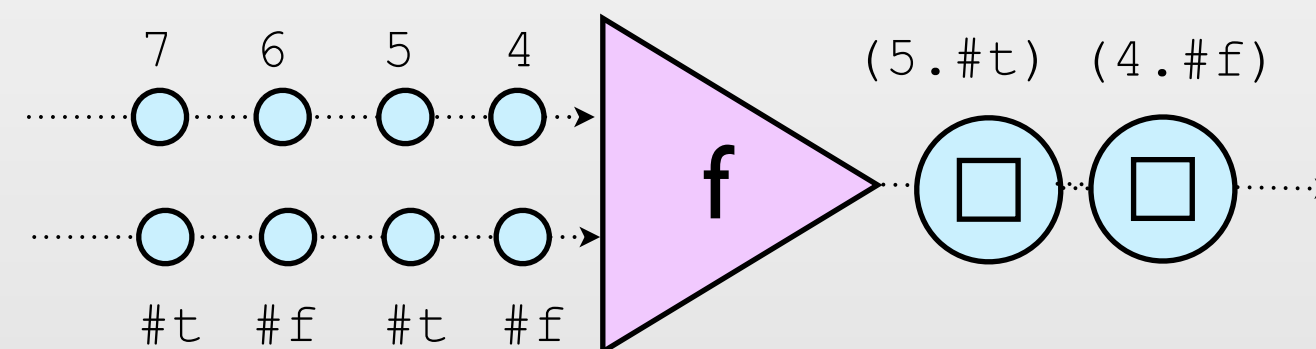
(PAIRS, ARRAYS, OBJECTS...)

Traditional FRP



```
>(distance x1 y1 x2 y2)
```

Traditional FRP on compound data



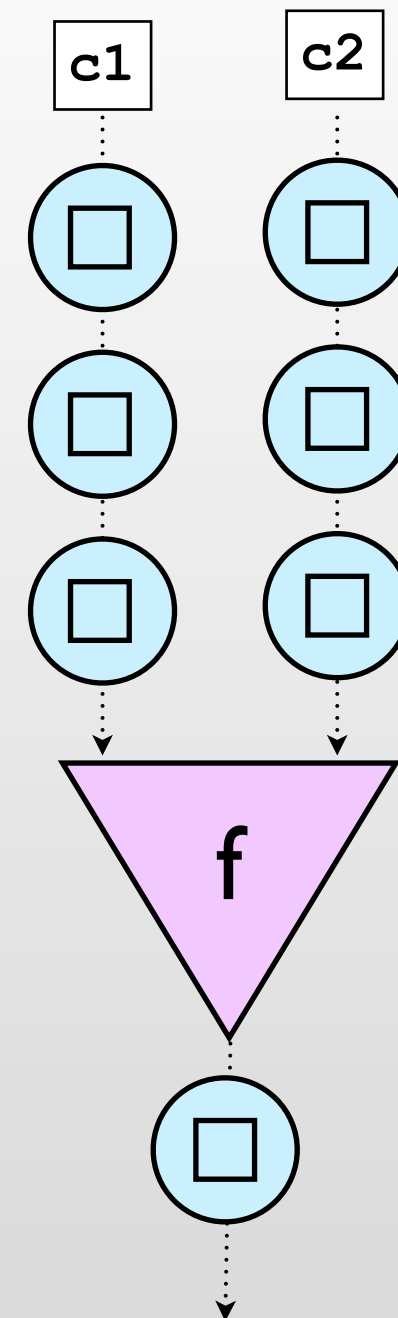
```
>(cons seconds
  (odd? seconds))
```


DATA STRUCTURE IDENTITY

REM

```
>(define c1 (cons seconds  
                  (odd? seconds)))  
>(define c2 (cons (+ seconds 1)  
                  (odd? (+ seconds 1))))
```

```
> c1  
'(1382901877 . #t)  
> c2  
'(1382901878 . #f)  
> (cons (cdr c1) (cdr c2))  
'(#t . #f)  
> (set-car! c2 42)  
✖ set-car!: undefined;  
> (set-cdr! c2 42)  
✖ set-cdr!: undefined;
```



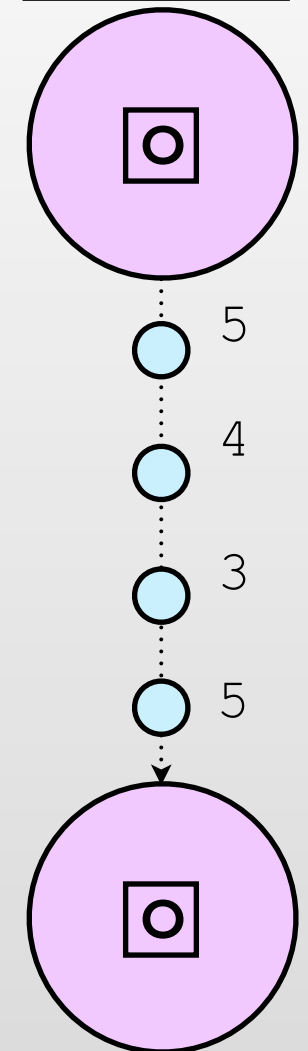
OBJECTS & REACTIVITY

ESTABLISHED WORK

- UNITING BOTH WORLDS
(FRTIME, FLAPJAX, ELM)

```
var nowB = timerB(1000);  
var startTm = nowB.valueNow();  
var clickTmsB = $E("reset", "click").snapshotE(nowB).startsWith(startTm);  
var elapsedB = nowB - clickTmsB;  
  
insertValueB(elapsedB, "curTime", "innerHTML");
```

DOM EL
"reset"

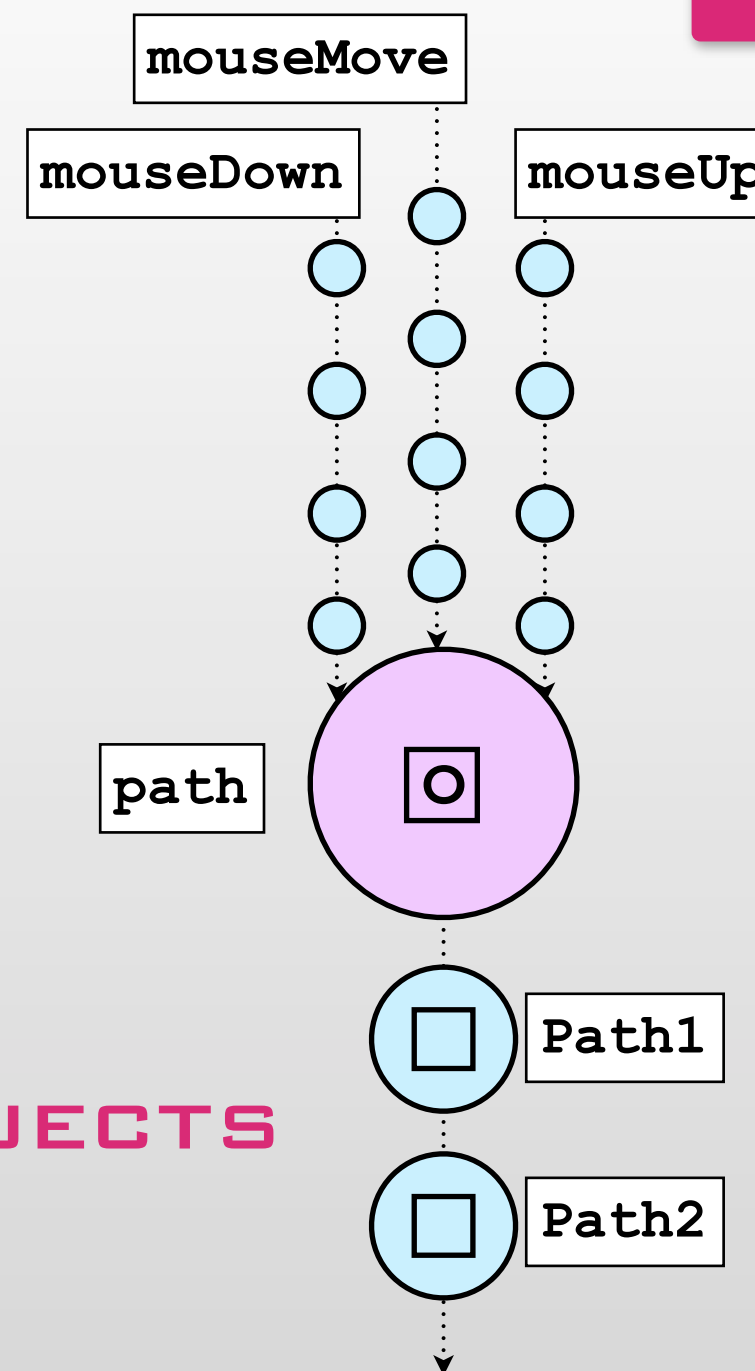


REM

OO REACTIVE PROGRAMMING

SCALA.REACT

```
val path: Signal[Path] = Signal.flow(new Path) { self =>
  val down = self await mouseDown
  self()= self.previous.moveTo(down.position)
  self.loopUntil(mouseUp) {
    val movement = self awaitNext mouseMove
    self()= self.previous.lineTo(movement.position)
  }
  self()= self.previous.close()
}
```



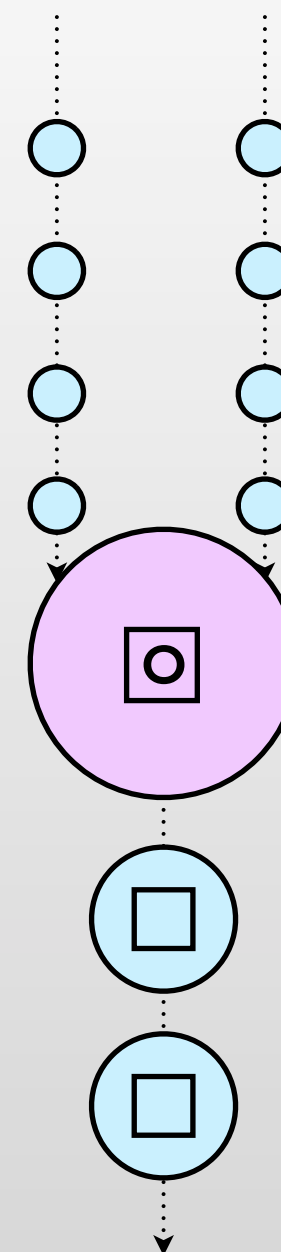
STREAMS OF DIFFERENT OBJECTS

OO REACTIVE PROGRAMMING

RESCALA

```
abstract class Animal(override implicit val world: World)
  extends BoardElement {
  ...
  val age = world.time.day.changed.iterate(1)(_ + 1)
  val isAdult = Signal { age() > Animal.FertileAge }
  val isFertile = Signal { isAdult() }
  ...
}
```

SIGNALS AS PART OF CLASS INTERFACE



REACTIVE OO PROGRAMMING

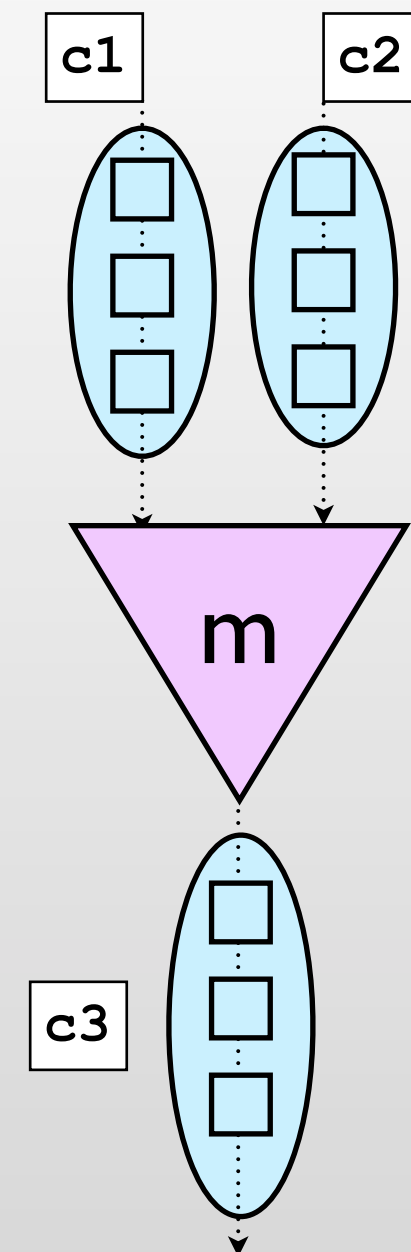
R
E
M

- UNIFYING BOTH WORLDS

```
def Coordinate := object: {  
  def x;  
  def y;  
  def move(c) {  
    Coordinate.new(x + c.x, y + c.y);  
  }  
}  
def c1 := Coordinate.new(1,2);  
def c2 := Coordinate.new(3,4);  
def c3 := c1.move(c2);  
c2.x := 42;
```

triggers
recomputation of
move

OBJECT AS UNITS OF REACTIVITY



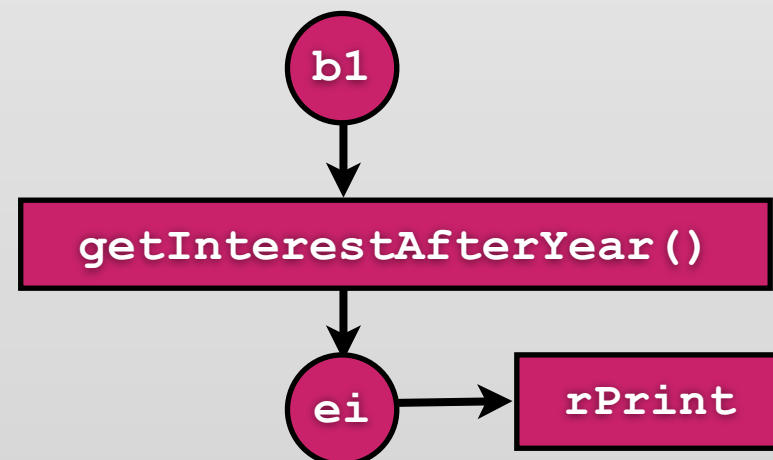
REACTIVE OBJECTS IN AMBIENTTALK (ROAM)

REM

- REACTIVE FIELDS
- REACTIVE METHODS

```
>def b1:= BankAccount.new(500,0.04);  
>def ei := b1.getInterestAfterYear()  
>rPrint(ei);  
>b1.deposit(50);  
>b1.withdraw(42);
```

```
>>20  
>>22  
>>20.32
```



```
def BankAccount := reactiveObject: {  
  def balance;  
  def interest;  
  def init(bal, int) {  
    self.balance := bal;  
    self.interest := int  
  };  
  def withdraw(amount) @Mutator {  
    if: (amount < balance) then: {  
      balance := balance - amount  
    }  
  };  
  def deposit(amount) @Mutator {  
    balance := balance + amount  
  };  
  def getInterestAfterYear() @Accessor{  
    balance * interest  
  }  
}
```

REACTIVE CIRCLES

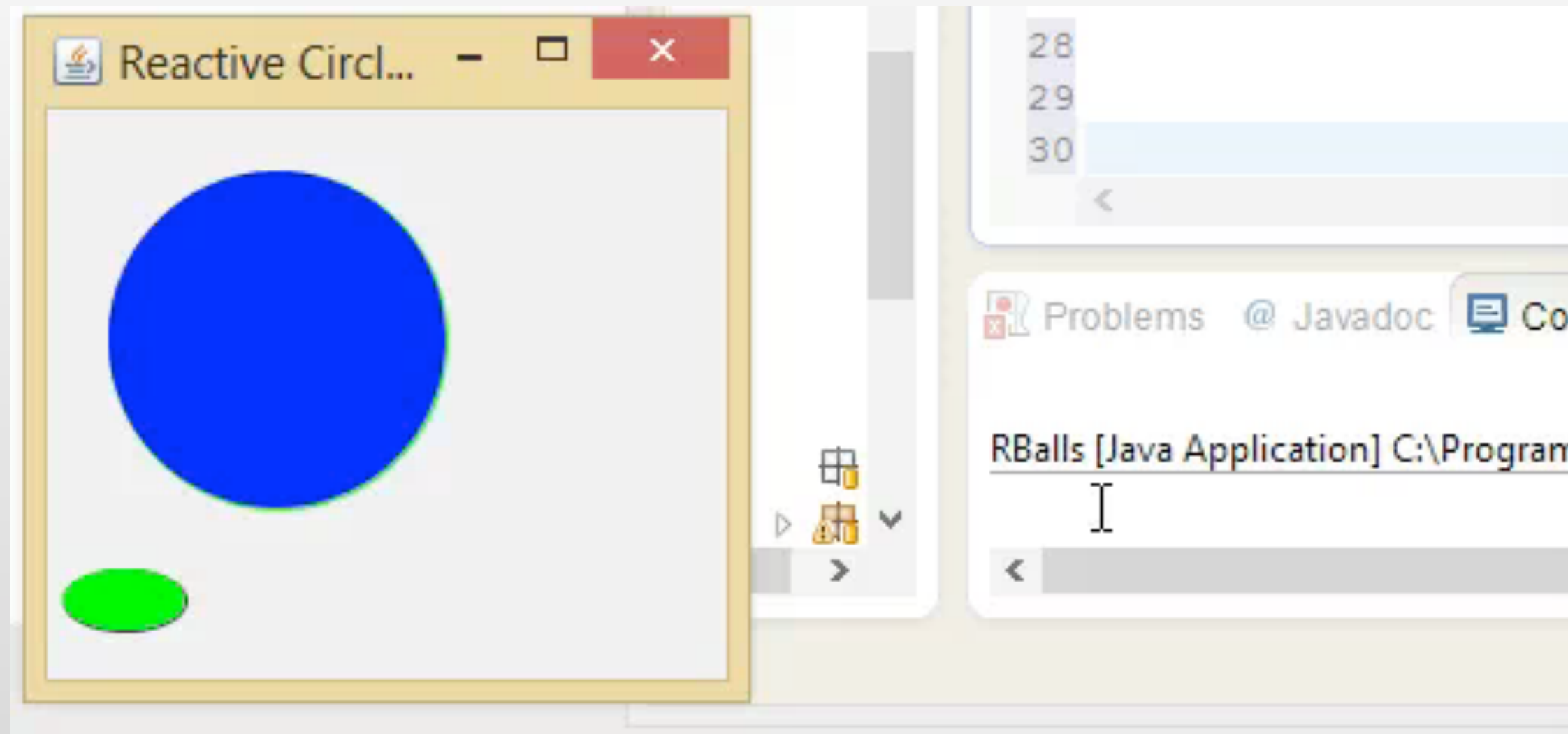
```
def mouse := reactiveObject: {  
  def [x,y,clicked];  
  ...  
};  
def ReactiveCircle := reactiveObject: {  
  def [jCircle, x, y, width, height, color];  
  def init(newx, newy,w,h,c) {  
    [x,y,width,height,color] := [newx, newy, w, h, c];  
    jCircle := JCircle.new(...);  
  };  
  def move(coordinates) {...};  
};  
def canvas := reactiveObject: {  
  def draw(c) {  
    c.jCircle.update(c.x, c.y, c.width, c.height, c.color);  
  };  
};
```

```
c1.width := 40;  
c2.color := `blue;  
c1.move(mouse);
```

```
def c1 := ReactiveCircle.new(5,150,20,20,`green);  
def c2 := ReactiveCircle.new(20,20,110,110,`yellow);  
canvas.draw(c1);  
canvas.draw(c2);
```

REACTIVE CIRCLES

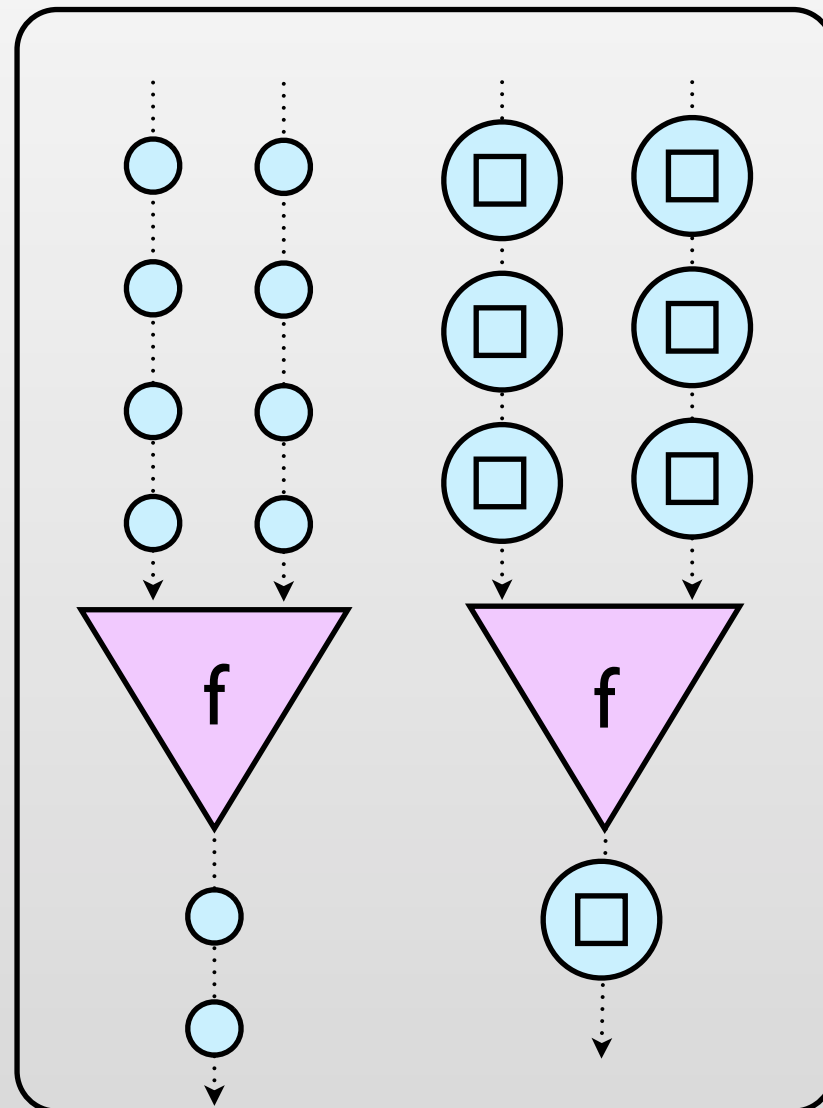
T
E
X
T



SUMMARY / DISCUSSION

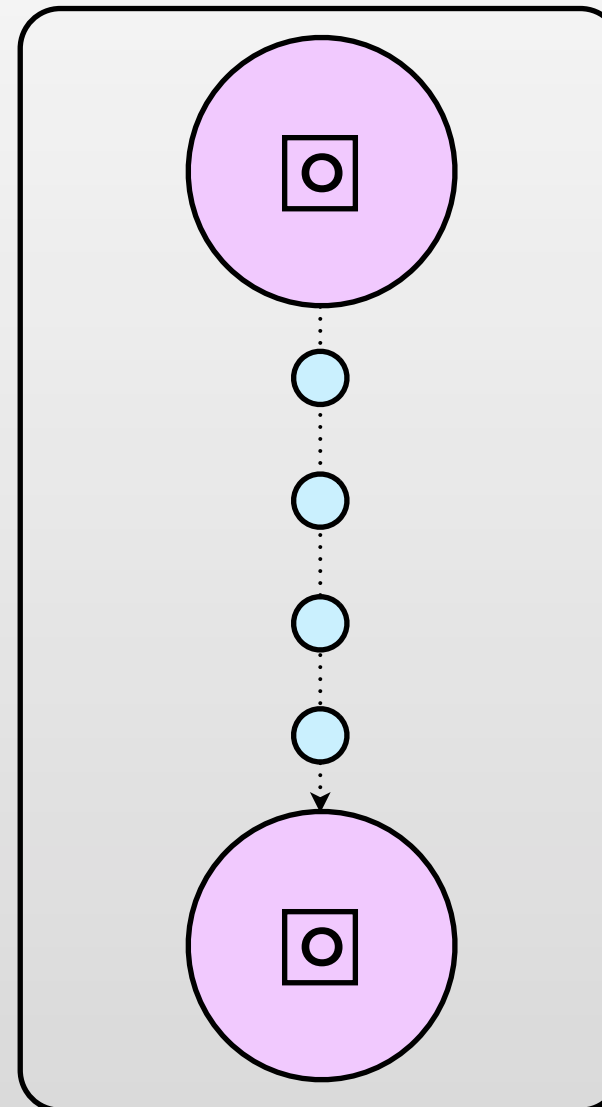
R
E
M

Functional/Procedural RP



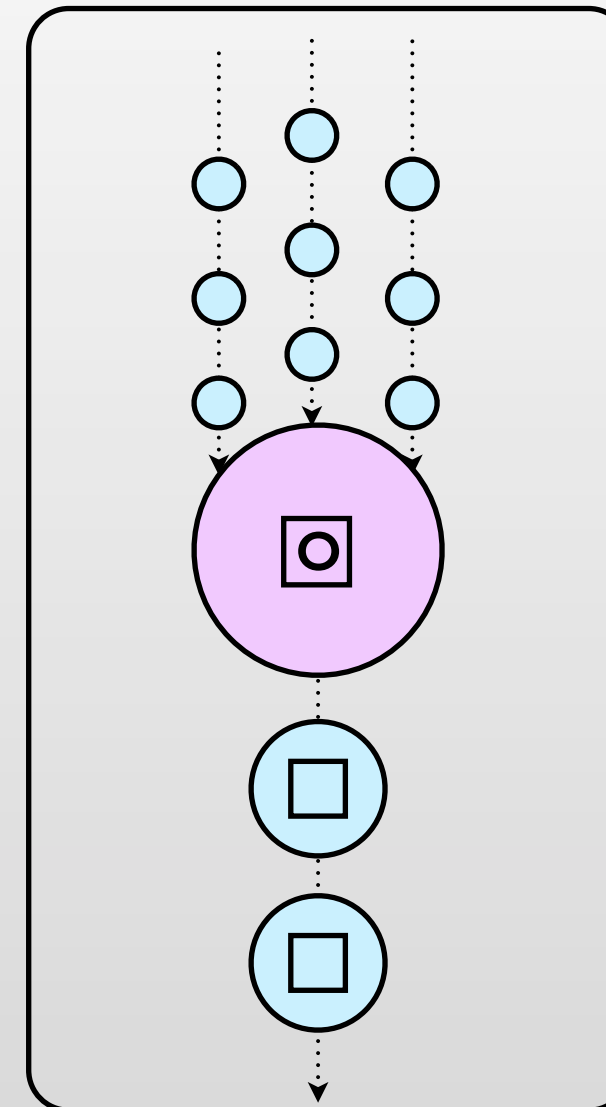
FRTIME

Unifying RP and OO



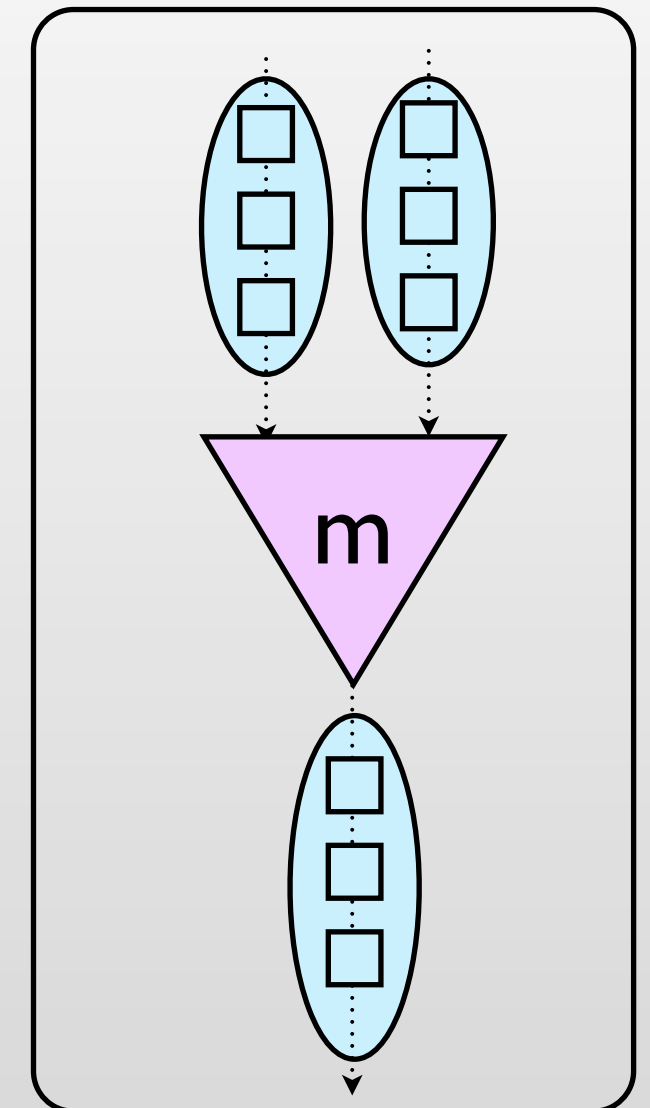
FLAPJAX

OORP



SCALA.REACT

ROOP



ROAM