Writing Embedded Domain Specific Languages in Java

Robert Chatley and Mike Hill

rchatley@google.com mike@mandu.co.uk

Google™

Mandu

Large projects have **lots** of code

Intention is often unclear

"…a good programmer in these times does not just write programs … a good programmer does language design, though not from scratch, but building on the frame of a base language"

- Guy Steele Jr.

Different **domains** have **specific languages**

Internal vs **External**

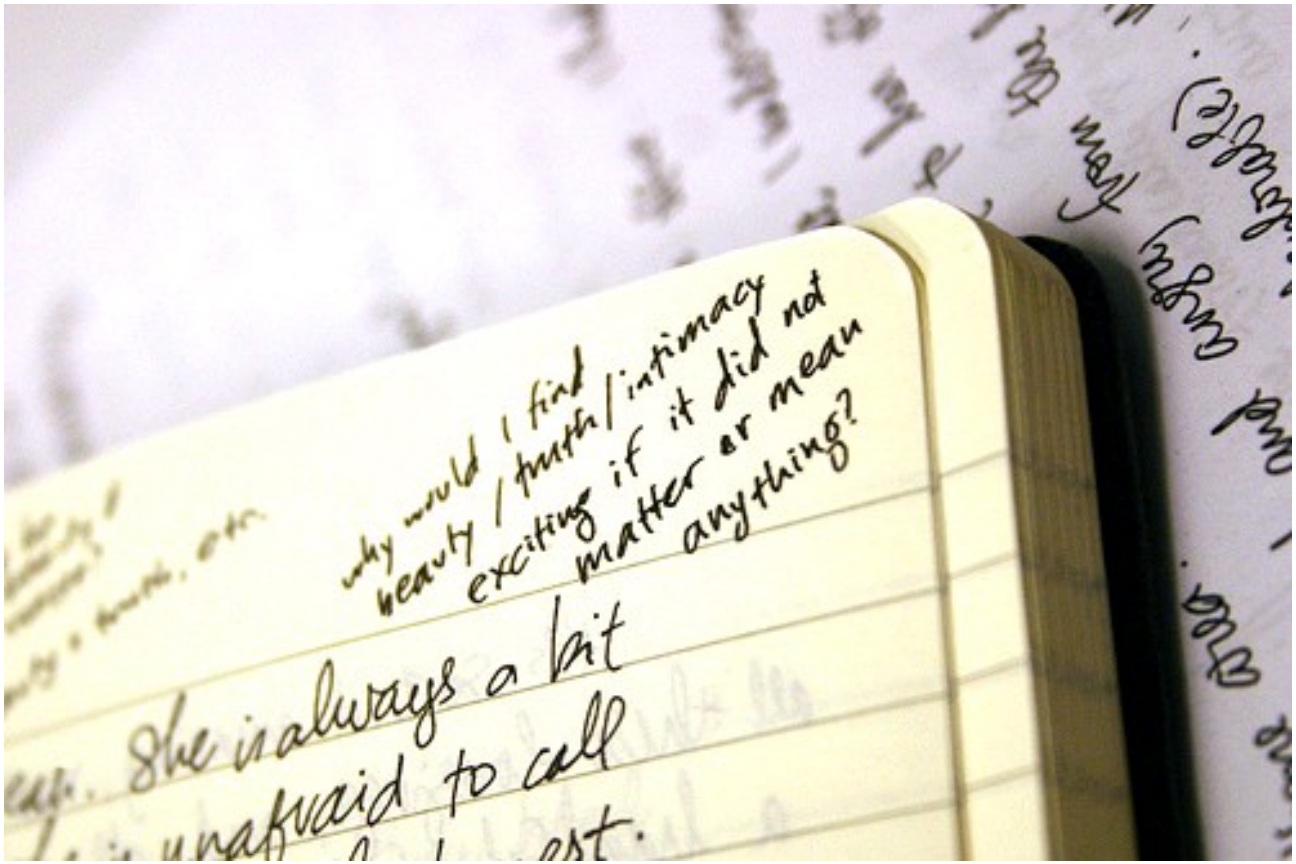
Make use of **factories**

```
class Coffee {
  Coffee(boolean milk, boolean sugar){
    ...
  }
}
drink = new Coffee(true, true);
```

```
class Coffee {
  Coffee(boolean milk, boolean sugar){
    ...
  }
}
drink = new Coffee(true, true);

class Coffee {
  static Coffee withMilkAndSugar() {
    return new Coffee(true, true);
  }
}
drink = Coffee.withMilkAndSugar();
```

Code to a **fluent interface**



```
class Coffee {
  void setMilk(boolean milk){
    this.milk = milk;
  }
}
drink.setMilk(true);
drink.setSugars(2);
```

```
class Coffee {
  void setMilk(boolean milk){
    this.milk = milk;
  }
  Coffee withMilk() {
    milk = true;
    return this;
  }
}
drink.setMilk(true);
drink.setSugars(2);

drink.withMilk().withSugars(2);
```

The **Builder** pattern…

```
Order order =
  Order.forDrinks(
          Coffee.black(),
          Tea.withMilk()
        )
       .toTakeAway();

order.build();

// or perhaps
order.make();
```

```
Order order =
   Order.forDrinks(
            Coffee.black(),
            Tea.withMilk()
          )
         .toTakeAway();

drinks = order.build();

// or perhaps
drinks = order.make();
```

a vararg method is nice here

Build or extract **layers**

Auction Exercise

We've coded up an auction and some bidders

- Bidders have different strategies
- Refactor bidders so their strategies can be described as a DSL
- We've provided acceptance tests to keep you safe

Retrospective

What did you learn?

What went well?

What went badly?

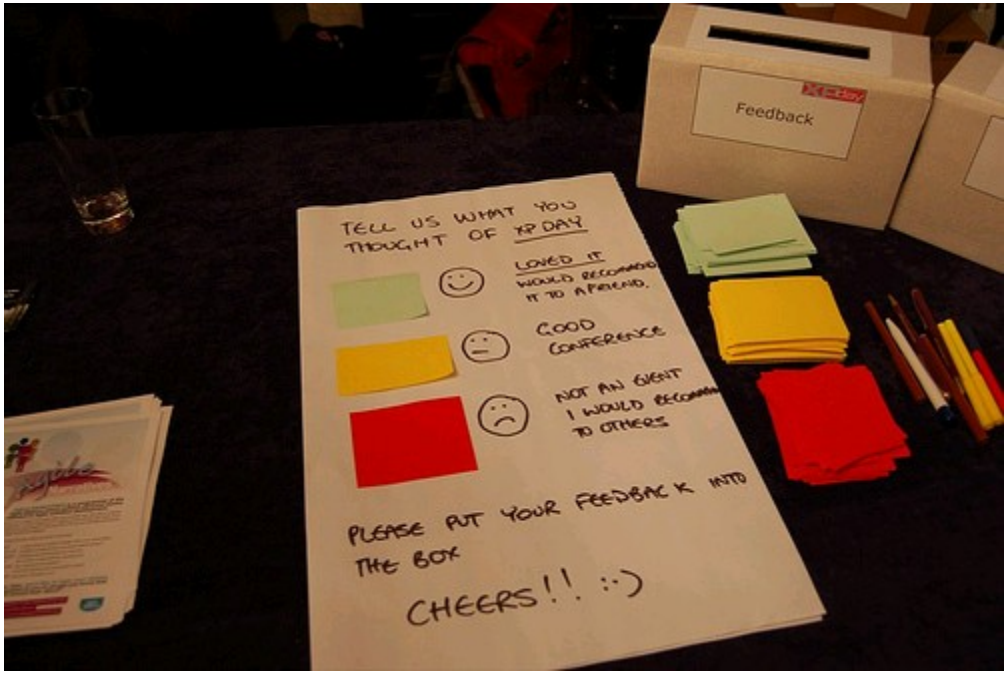
Puzzles?

Ideas?

Feedback

Interesting? Useful?

What could be improved?



Acknowledgments

John Ayres
Paul Carey
Steve Freeman
Nat Pryce
Joe Walnes
Tom White