



DSLs and LOP

Amanda Laucher

pcprogrammer@gmail.com

Twitter: @pandamonial

www.pandamonial.com


Objectives

- DSL/LOP Background
 - Internal/External
 - Visual/Textual
- Examine Tools
 - C#
 - F#
 - Oslo



DSL Background

- What is a DSL?
- What is LOP?
- Why should you care?



Is there really a
communication gap?



Domain Experts Type...

Add Kresten Krab Thorup to priority contacts

Phone: 555-555-5555

Change Don Syme's address

One Microsoft Way

Redmond WA 50505

Developers Type...

```
Contacts.Add("Kresten Krab Thorup", null, Priority,  
    "555-555-5555");
```

```
if (Contacts.Exists("Don Syme")) {  
    Contacts.Update("Don Syme", "One Microsoft Way", "Redmond",  
        "WA", "50505");  
}
```

DSL vs GPL

- Limited Expressiveness
- Not Turing complete (mostly)
- No ability to build abstractions
- Limited power
- Focused on a specific domain and make us more productive within that domain

Textual vs Visual





Textual vs Visual

- SQL
- Javascript
- XML
- Build Script Languages

Internal DSLs

- Molding general purpose language to look like what the domain experts type
- Fluent Interface
 - “The more the use of the API has that language-like flow, the more fluent it is.” - Martin Fowler

API

```
private void makeNormal(Customer customer) {
    Order o1 = new Order();
    customer.addOrder(o1);
    OrderLine line1 = new OrderLine(6, Product.find("TAL"));
    o1.addLine(line1);
    OrderLine line2 = new OrderLine(5, Product.find("HPK"));
    o1.addLine(line2);
    OrderLine line3 = new OrderLine(3, Product.find("LGV"));
    o1.addLine(line3);
    line2.setSkippable(true);
    o1.setRush(true);
}
```

Fluent Interface

```
private void makeFluent(Customer customer) {  
    customer.newOrder()  
        .with(6, "TAL")  
        .with(5, "HPK").skippable()  
        .with(3, "LGV")  
        .priorityRush();  
}
```



Use your languages features!

- Reflection
- Open Classes/Extension Methods
- Dynamic Typing
- Type Inference/Duck Typing
- Pattern Matching/Active Patterns
- HOF/Partial Application
- Quotations

F#

```
let bakeMuffins temp mix time = mix
let bake = bakeMuffins 180<c>
let sift (x,y) = y

let FruitMuffins fruit =
  let container = "muffinPan"
  let dryIngredients = sift (2.5<cups>,"flour")
                        |> mix (1.5<cups>,"sugar")
  let moistIngredients = mix (250<ml>,"milk") (1.3<cups>, "oil") (2,"eggs")
                        |> mix (1.25<cups>, fruit)
  mixture = combine dryIngredients moistIngredients
            |> fillContainer
  bake mixture 15<min>
            |> cool 10<min>
```

F#

```
let FruitMuffins fruit =  
    let container = "muffinPan"  
    let dryIngredients = sift (2.5< cups>, "flour")  
                            |> mix (1.5< cups>, "sugar")  
    let moistIngredients = mix (250< ml>, "milk") (1.3< cups>, "oil") (2, "eggs")  
                            |> mix (1.25< cups>, fruit)  
    mixture = combine dryIngredients moistIngredients  
                |> fillContainer  
    bake mixture 15< min>  
        |> cool 10< min>
```



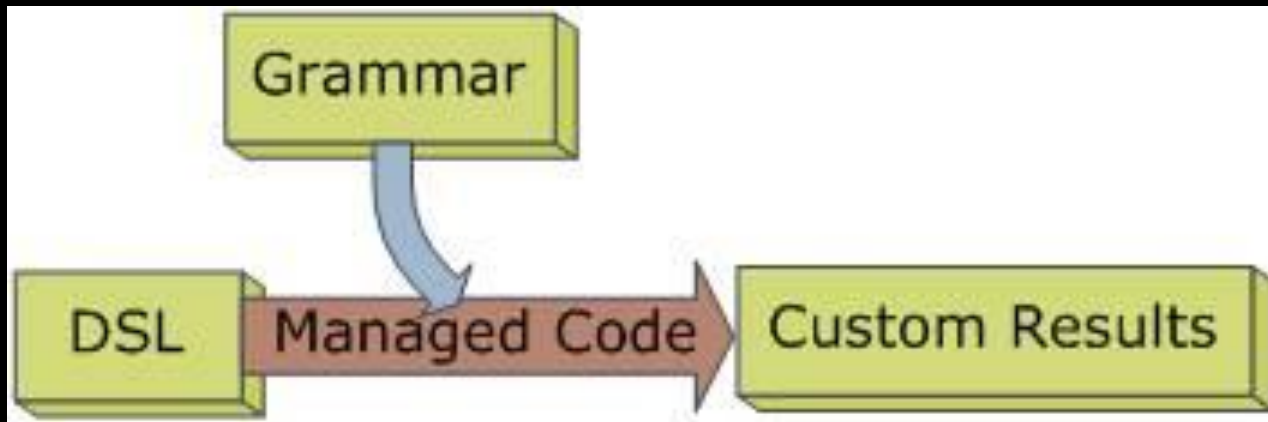
DSL Workbench

- Oslo
- Intentional Software



External Oslo Demo

Grammar Output → Results





Things to consider with DSLs

- Involve Domain Experts in development
- Life cycle
- DSLs evolve independently of other code and of each other
- DSLs are used in combination with other DSLs



Other Tools

- ANTLR
- Intentional Software
- XText

.NET folks also watch out for...

- Future Oslo
- Axum – Maestro
 - DSL for concurrency
- F#
 - Great for internal DSLs and walking ASTs
- DLR Languages
 - Iron – Python/Ruby
 - Clojure



Questions?

DSLs and LOP

Amanda Laucher

pcprogrammer@gmail.com

Twitter: @pandamonial

www.pandamonial.com