

ContextR

cross-cutting concerns
in a dynamic world

Gregor Schmidt · euruko 2008 · www.nach-vorne.eu
2008-03-30

Introduction(s)

1.

Just seen a talk about
AOP in Ruby ...

**bad karma for
referring to something
I had not yet seen**

II.

“Nobody likes
conditionals”

“Nobody likes globals”

Object-oriented
programming is about
moving conditionals into
the inheritance tree

Object-oriented
programming is about
avoiding conditionals
by introducing globals

**Object-oriented
programming is about
avoiding conditionals
by introducing globals**

Not quiet right

III.





+

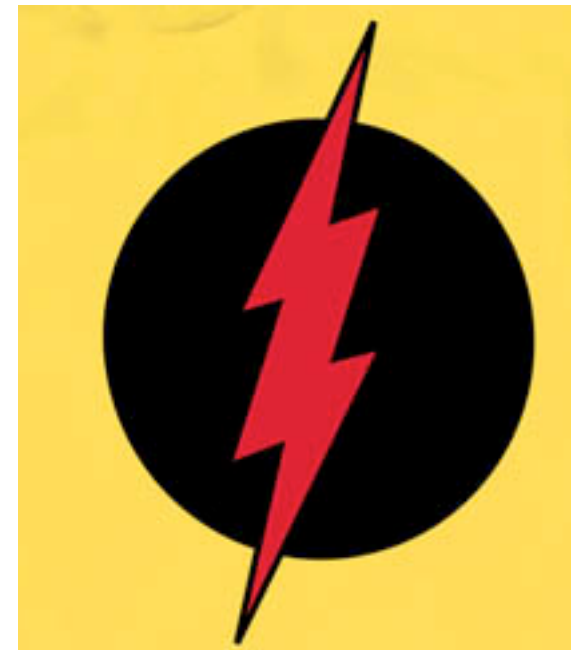




+



=

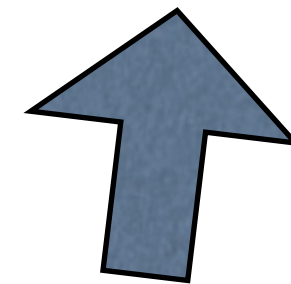
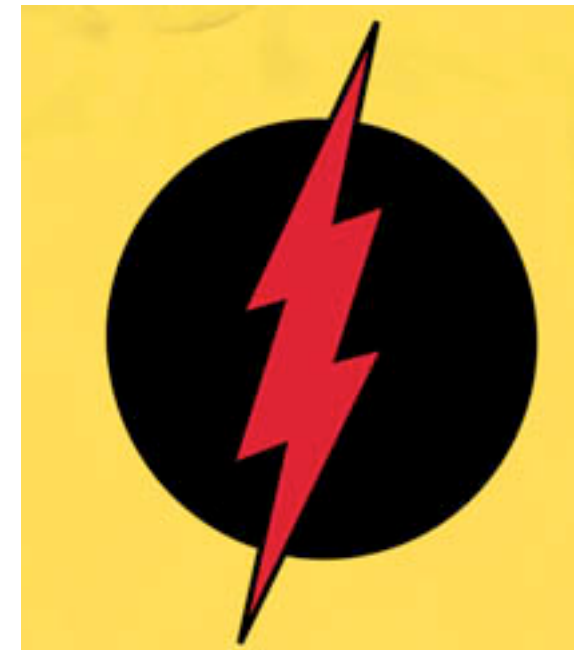




+



=



Conflict, not Lightning

IV.

The Nic Williams Way

“I have something for
ya, but ain't tell you
what it's for.”

V.

EURUKO 2005 REVISITED

Christian Neukirchen
introduced
context-oriented
programming to Ruby

VI.

My Master's Thesis

Addressing Cross-cutting Concerns

... in web applications

Scattering and Tangling

What's the deal?

**Requirements that
don't fit in your
inheritance tree**

Got Modules, Meta-programming, and friends

Remember
Introduction II ?

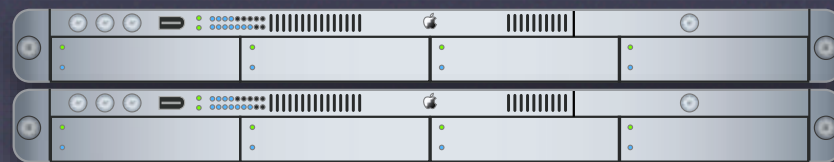
Object-oriented
programming is about
moving conditionals into
the inheritance tree

What about dynamic
conditions?

Conditions inspecting
the context

... of your server ...

... of your server ...



... of your network ...

... of your client

... of your client



... of your client



...lost my point ...

... some code ...


```
require "rubygems"  
require "contextr"
```

```
class Person  
  def say_hello  
    "hello"  
  end  
end
```

```
in_layer :enterprisy do  
  def say_hello  
    "<greeting><content>" +  
      super +  
      "</content></greeting>"  
  end  
end  
end
```



```
require "dust"
```

```
unit_tests do
```

```
  test "Acts nicely in the general case" do
```

```
    anybody = Person.new
```

```
    assert_equal "hello", anybody.hello
```

```
  end
```

```
  test "Acts enterprising, when asked to" do
```

```
    anybody = Person.new
```

```
    ContextR::with_layer :enterprising do
```

```
      assert(anybody.hello.size > 10)
```

```
    end
```

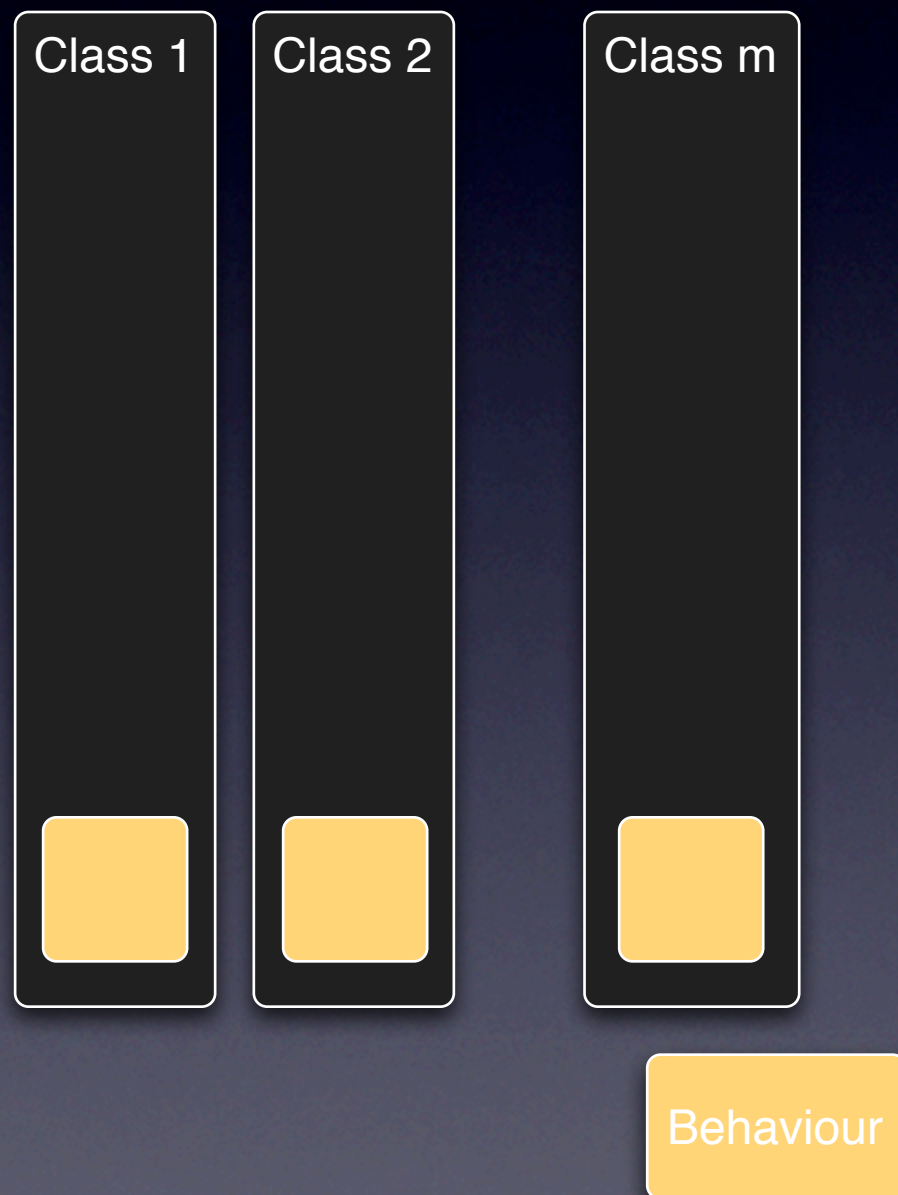
```
  end
```

```
end
```

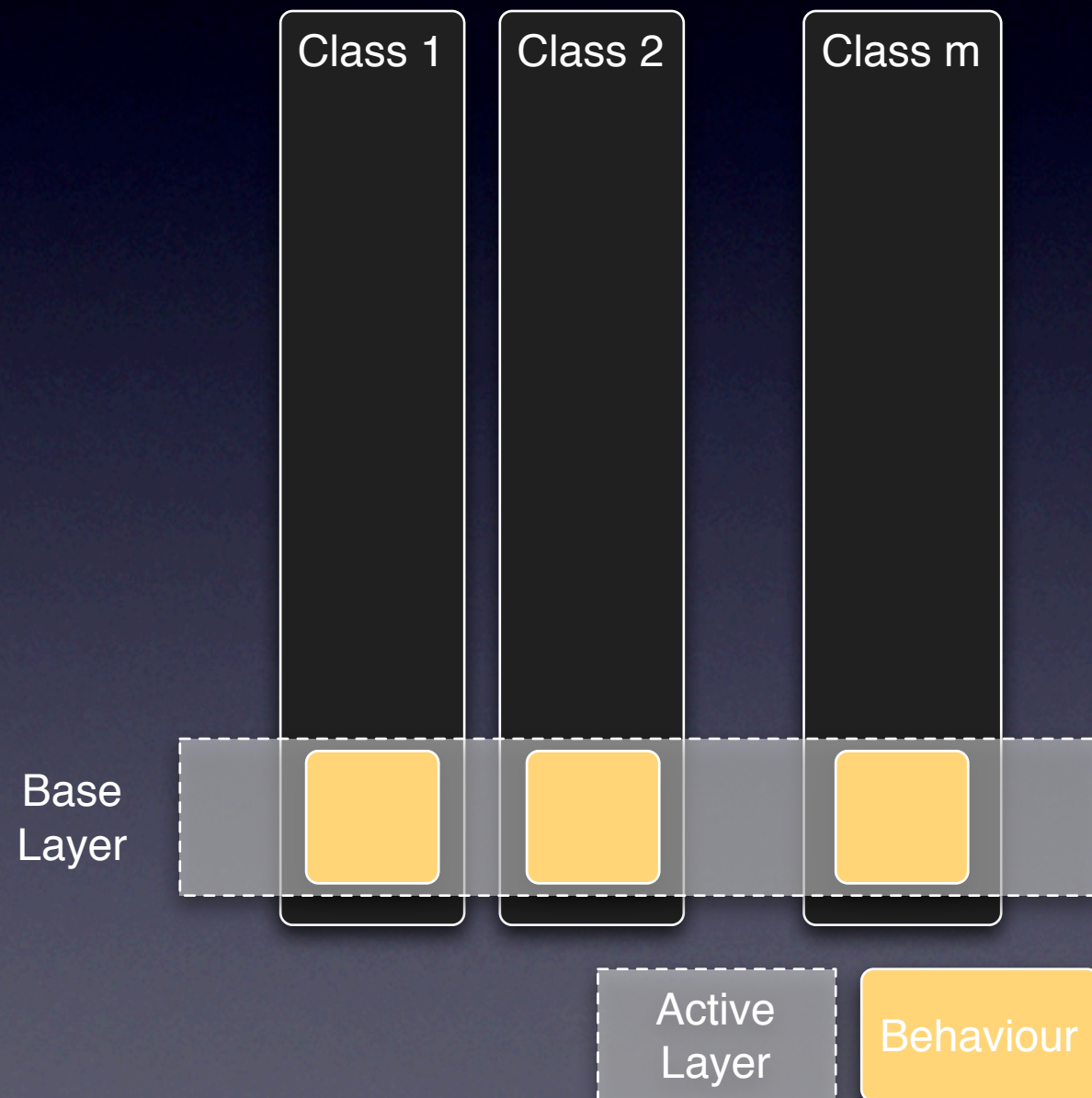

mind model

Context-oriented Design

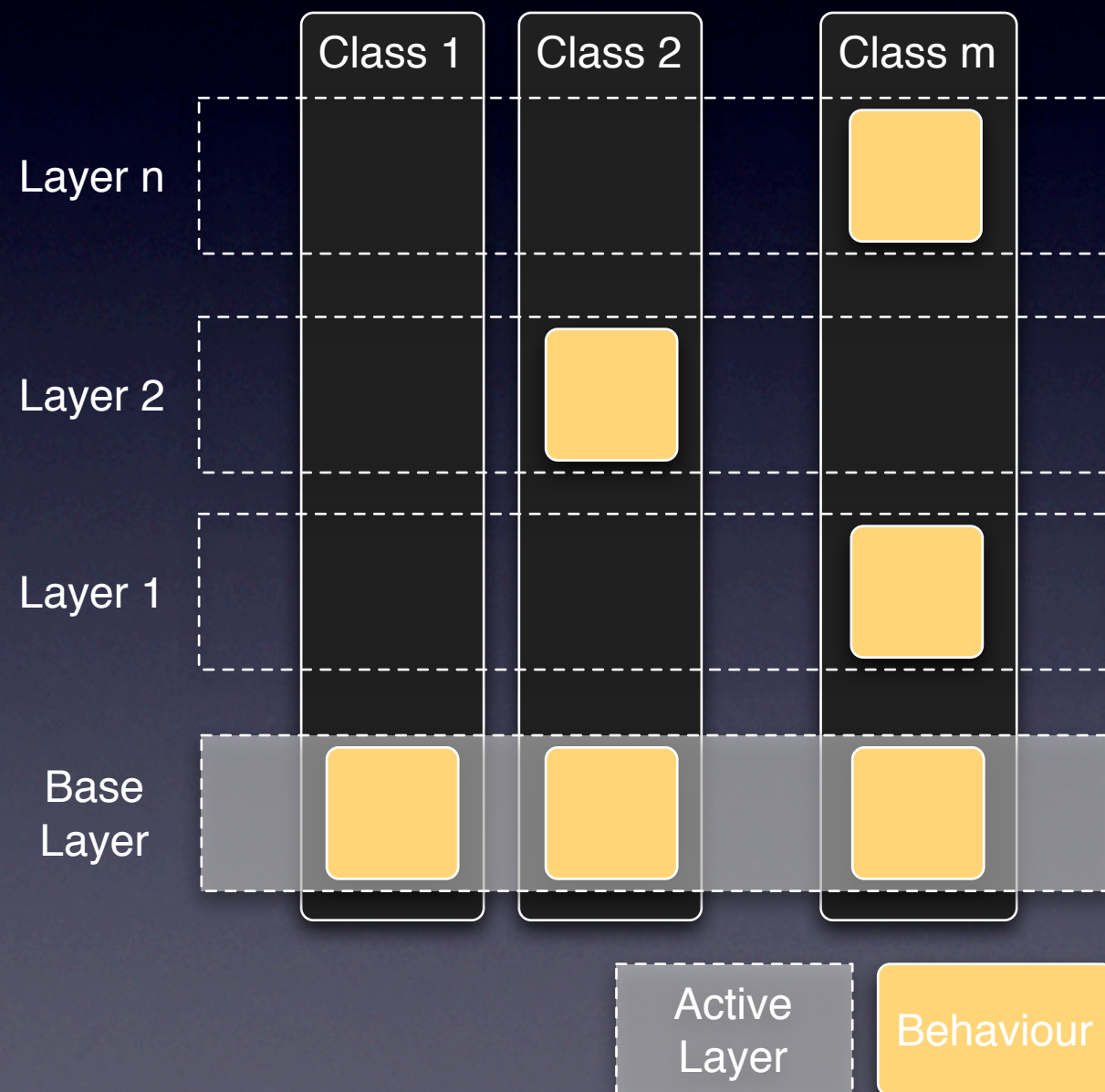
Context-oriented Design



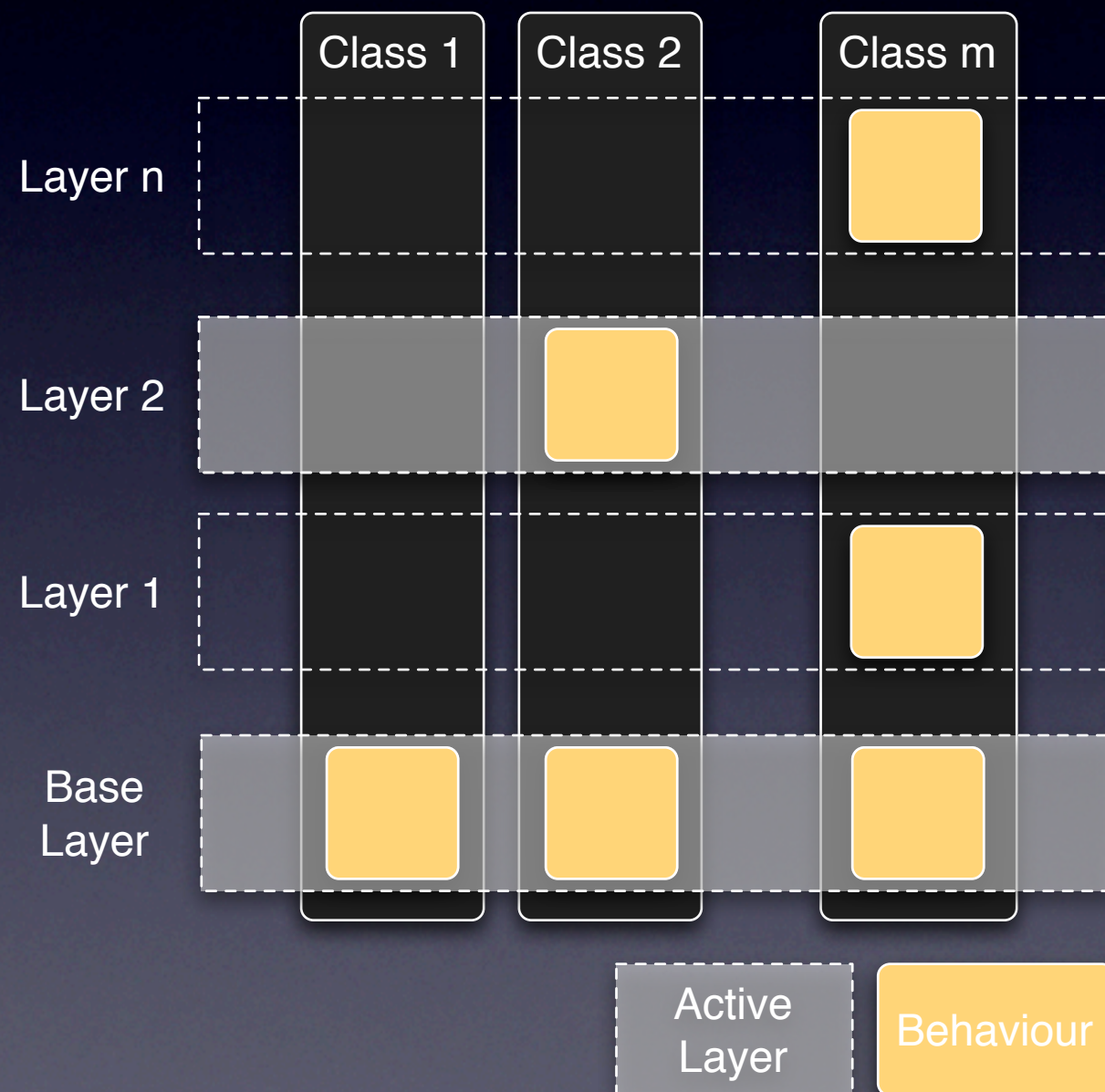
Context-oriented Design



Context-oriented Design



Context-oriented Design



**activation is
dynamically
scoped**

In web apps ...

... analyze context ...

... activate
behavior definitions
per request

COP \neq AOP || FOP

same domain policy

addressing
cross-cutting concerns

... using different
assumptions

www.contextr.org

more examples

my thesis
coming this summer

my thesis
coming this summer

unfortunately
in german

links to other
resources on
contextr.org




scientific
papers

links to other
resources on
contextr.org



scientific
papers

links to other
resources on
contextr.org



Lisp, Smalltalk,
Python, Java*

time is up

Questions?