

<http://github.com/reborg/talk-macruby>



Writing Apps with HotCocoa and MacRuby

renzo.borgatti@agilepartners.com - ChiRb - April 6th 2009



MacRuby

Ruby 1.9 on Objective-C runtime

MacRuby

Primitive Ruby classes

\Leftrightarrow

Core Foundations & Cocoa equivalents



MacRuby

convenience, efficiency, flexibility,
performance over RubyCocoa

MacRuby

```
reborg:scripts reborg$ macirb  
irb(main):001:0> hello = "hello"  
=> "hello"  
irb(main):002:0> hello.class  
=> NSMutableString
```



```
static VALUE  
str_alloc(VALUE klass) {  
    VALUE str;  
    str = (VALUE)CFStringCreateMutable(NULL, 0);  
    if (klass != 0  
        && klass != rb_cNSString  
        && klass != rb_cNSMutableString  
        && klass != rb_cSymbol)  
        *(Class *)str = (Class)klass;  
    CFMakeCollectable((CFTyperef)str);  
    return (VALUE)str;  
}
```


MacRuby

MacRuby recently got a new VM!

[http://svn.macosforge.org/repository/
ruby/MacRuby/branches/experimental](http://svn.macosforge.org/repository/ruby/MacRuby/branches/experimental)



HotCocoa

Compact Ruby-based DSL for UI
development by Rich Kilmer

HotCocoa

```
require 'hotcocoa'
include HotCocoa
application :name => "Hello ChiRb" do |app|
  app.delegate = self
  window (
    :frame => [500, 300, 200, 100],
    :title => "Hello ChiRb") do |win|
    win << button(
      :title => "Click Me",
      :on_action => lambda do |sender|
        alert(:message => "Hello ChiRb!")
      end)
    win.will_close { exit }
  end
end
```




HotCocoa

command-line: creates rake-enabled
skeleton project



HotCocoa

Mappings: helpers for most used Cocoa classes. Add your own.

Right now, sources are your docs.

HotCocoa

Can be used from XCode but especially useful for programatic UI

Only way I know right now for testing-first UI under MacRuby

Pomodoro Technique



“The aim of the Pomodoro Technique is to use time as a valuable ally in accomplishing what we want to do in the way we want to do it, and to enable us to continually improve the way we work or study”

Francesco Cirillo

www.pomodorotechnique.com

Pomodoro Technique

25 + 5 and 4 + 1

The Pomodoro Is Indivisible

If a Pomodoro Begins, It Has to Ring

If More Than 5-7 Pomodoros, Break It Down

If Less Than One Pomodoro, Add It Up

The Next Pomodoro Will Go Better

Pomodoro Technique

	ACTIVITY INVENTORY	Lucy Banks
	...	
	Answer questions on thermodynamics in Ch 4	2
	Repeat laws of thermodynamics out loud to Mark	3
	Summarize laws of thermodynamics in writing	3
	Call Laura: invite her to the seminar on thermodynamics	
	Call Mark: give me my laptop back soon!	
	Call Andrew: buy tickets to concert?	
	Email Nick: how do you do ex. 2 p. 24?	
	...	

	TO DO TODAY	Rome, 12 July 2006 Lucy Banks Available Pomodoros: 8
	Answer questions on thermodynamics in Ch 4	☒☒
	Repeat laws of thermodynamics out loud to Mark	☒☒☐
	Summarize laws of Thermodynamics in writing	☒☒☒☒☐



Ring.....

5 minutes break, deep breath, relax



Pomodori

HotCocoa based tool for the Pomodoro
Technique (PT)



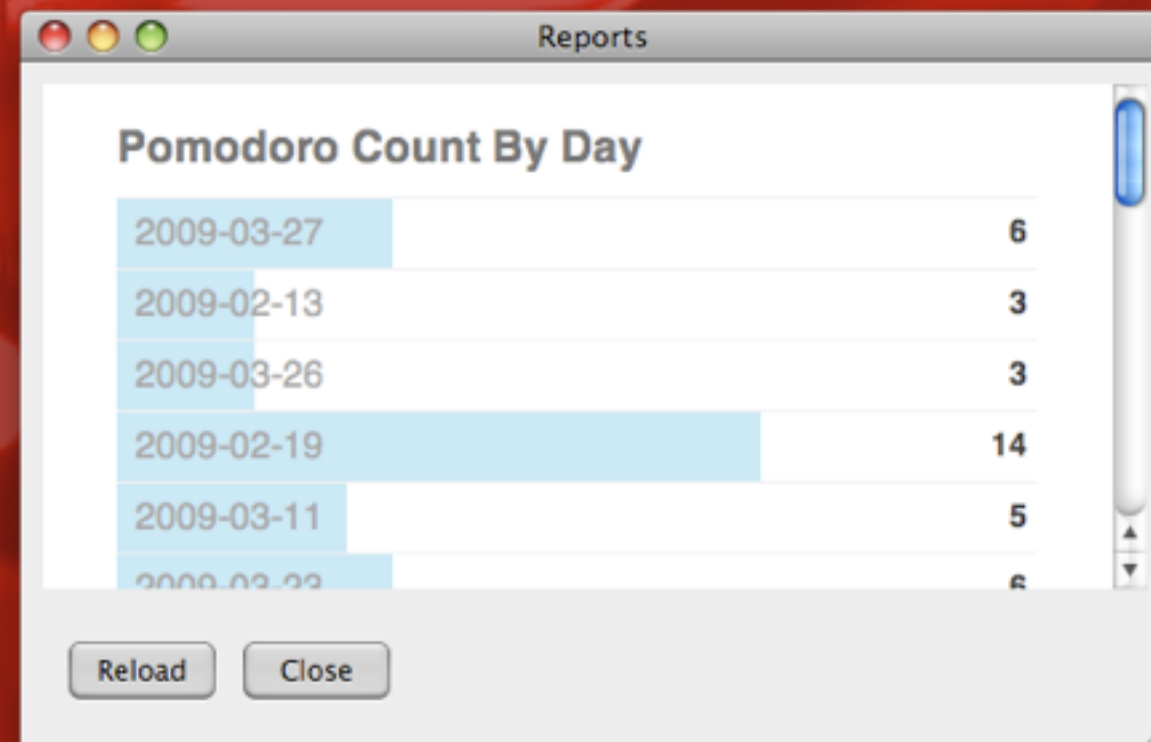
Pomodori

Very simple

Automatic timers

History

Charts



Pomodori

POMODORO!

Yesterday's pomodoros 7

Today's pomodoros so far 3

24:44

Report Void

Pomodori

Pomodoro description here

Done!

Report Submit



Architecture

Strict separation

view logic

workflow logic

business logic

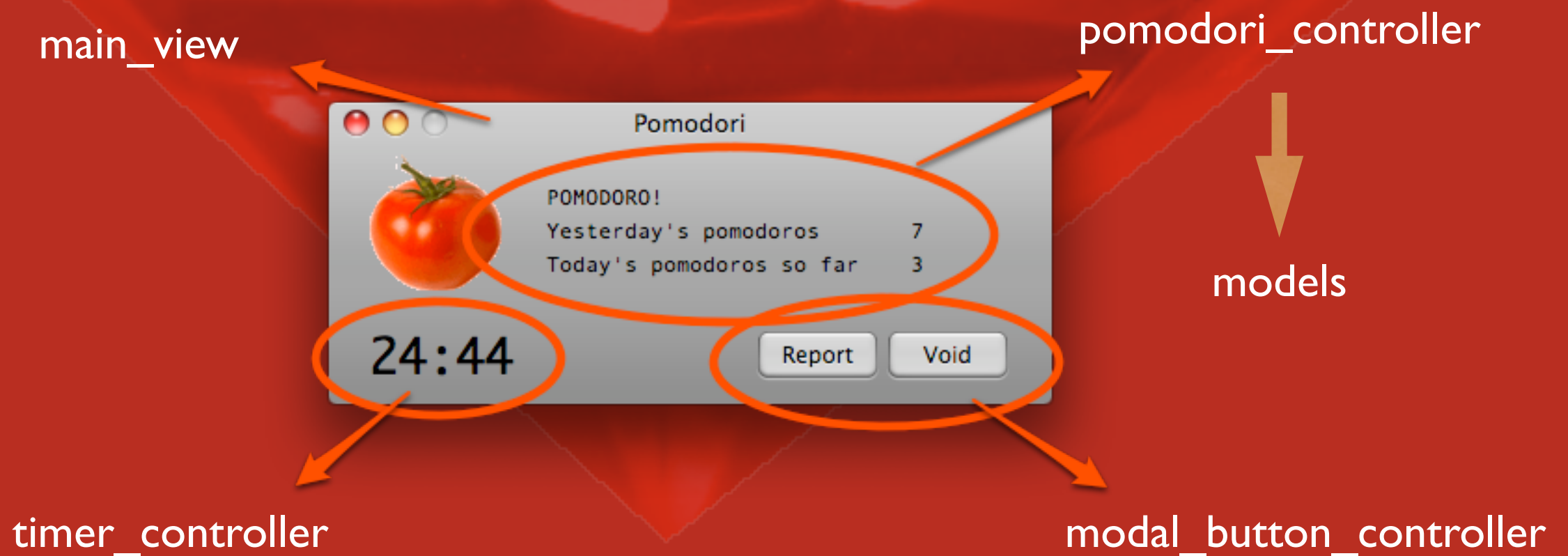
No architecture up-front

TDD => evolved to actual design



Architecture

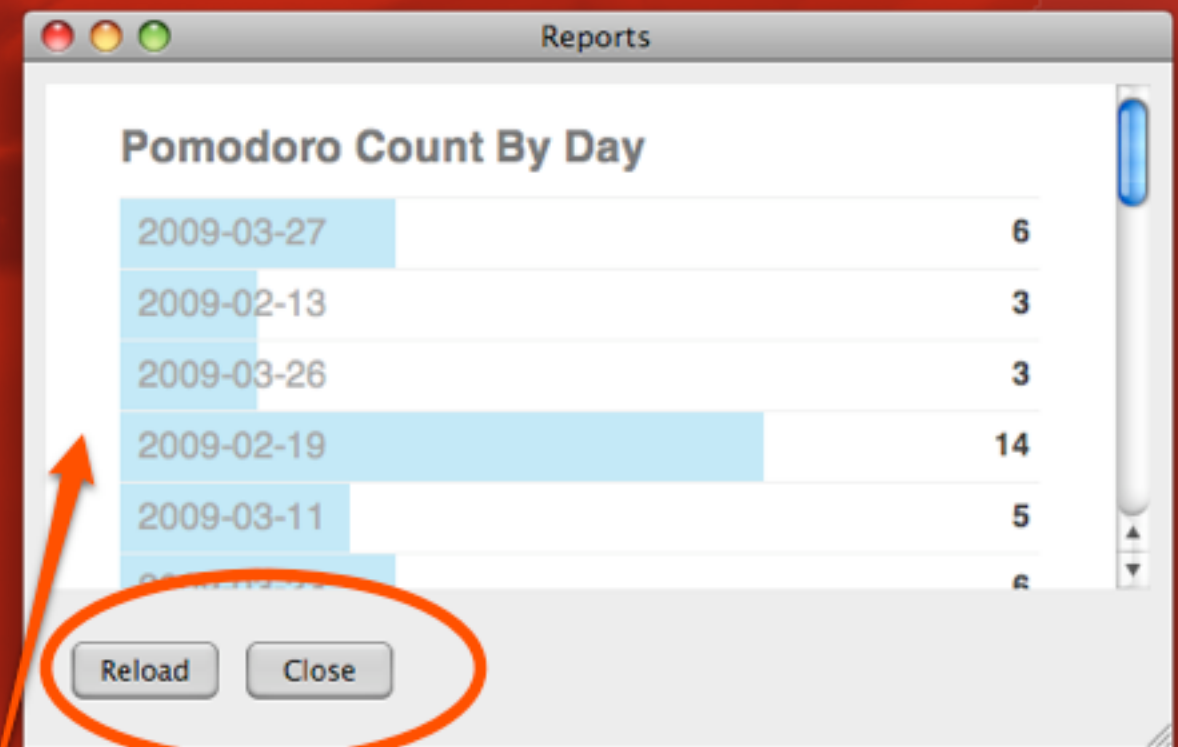
One view, one-many controllers





Architecture

View Switch: ask related controller



chart_controller

pom_by_count_model

TDD HotCocoa

Option #1: pure Ruby with everything
(RSpec, Stubba, etc.)

Option #2: MacRuby with limitations
(Test::Unit, Mocha and few more)

No options if the target class requires 'hotcocoa'

View TDD

```
def test_should_go_running_mode_on_init
  @main_view.expects(:running_mode)
  @main_view.send(:initialize)
end

def test_switch_on_input_box
  @main_view.send(:enable_input_box)
  assert_equal(true, @main_view.summary_label.editable?)
  assert_equal("Pomodoro description here",
    @main_view.summary_label.to_s)
end

def test_switch_to_submit_mode
  @main_view.expects(:enable_input_box)
  @main_view.submit_mode
  assert_equal("Submit", @main_view.modal_button.title)
end
```

Controller TDD

```
def setup
  @main_view = stub_everything
  @modal_button_controller = ModalButtonController.new(
    :main_view => @main_view)
end

def test_should_switch_to_break_on_submit
  when_goes_into_break_mode(:on_click_submit)
end

def when_goes_into_break_mode(action)
  @main_view.expects(:break_mode)
  @main_view.expects(:update_modal_button_action).with do |a_block|
    assert_match(/on_click_restart/, a_block.name)
  end
  @main_view.expects(:update_modal_button_label).with("Restart")
  @modal_button_controller.send(action, "sender")
end
```


Conclusions

MacRuby: young, 80% stable, promising

HotCocoa: cool (needs documentation)

Pomodori: the coolest app for the PT :)

PT: see you at Agile2009

Questions?

Resources

www.macruby.org - macruby main site

www.pomodorotechnique.com Pomodoro Technique main site

<http://blog.reborg.net> - my blog often talking about the PT

<http://reborg.github.com/pomodori> - Pomodori application main site

<http://svn.macosforge.org/repository/ruby/MacRuby/branches/experimental> - Laurent Sansonetti's work on the new MacRuby VM

Slides and examples: <http://github.com/reborg/talk-macruby>