N Design Patterns You Might Actually Use

Aja Hammerly

~~N~~5 Design Patterns You Might Actually Use

Aja Hammerly

Aja Hammerly
@thagomizer_rb
http://github.com/thagomizer
http://www.thagomizer.com

Motivations

http://www.fotopedia.com/items/flickr-2606920416

Singleton?

Factory?

Something Else?

https://www.flickr.com/photos/laughingsquid/7658733440/

re5et added a note on May 8, 2013

again, looks like strategies

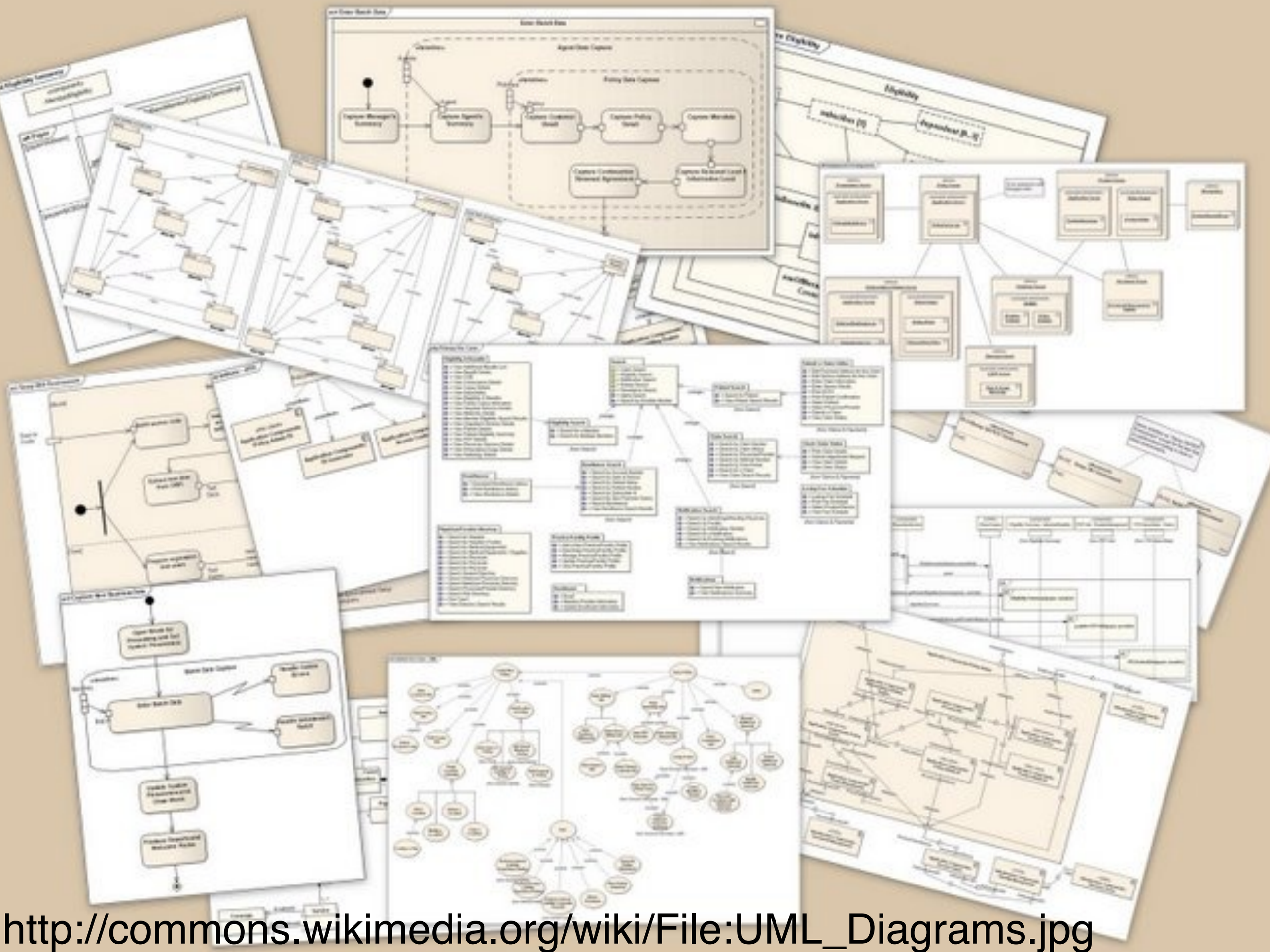
https://www.flickr.com/photos/kozumel/4918575268

Design Patterns

Common

Vocabulary

http://www.flickr.com/photos/54397539@N06/5034334027/



http://commons.wikimedia.org/wiki/File:UML_Diagrams.jpg

RASPBERRY IMITATION FLAVOR

Kool-Aid®

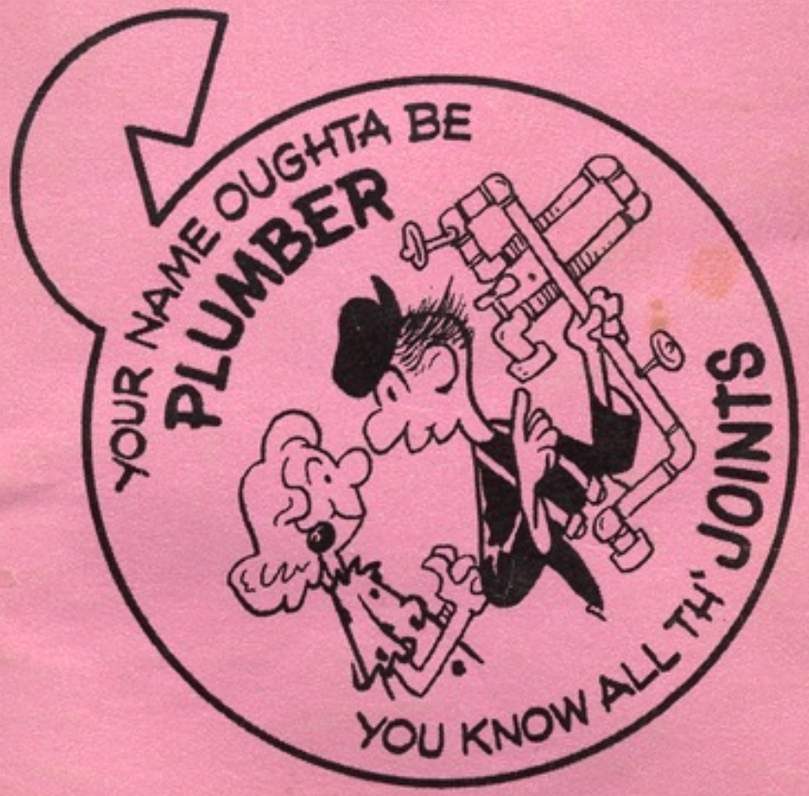
INSTANT SOFT DRINK MIX • ADD SUGAR

MAKES 2 QUARTS

K

Kool-Aid®

CUT out GAG TAG; brush off back of TAG; wear on pocket or button.



DIRECTIONS: Empty contents into large pitcher. Add 1 cup sugar. Add COLD water and ice to make 2 quarts.

GF GENERAL FOODS KITCHENS

Fumaric acid, sugar, artificial flavor and color, calcium carbonate, dioctyl sodium sulfosuccinate (DSS).

GENERAL FOODS CORP., WHITE PLAINS, N. Y., U. S. A.

P-8256 NET WT. 4.5 GM.

https://www.flickr.com/photos/roadsidepictures/4488040607

Setting Expectations

- ~140 Slides in 30 minutes
- Code Medium
- Advanced Beginner - Intermediate

Patterns

- Template
- Strategy
- Composite
- Decorator
- Command

Template

What Is It?

Outline For Similar Tasks

Examples

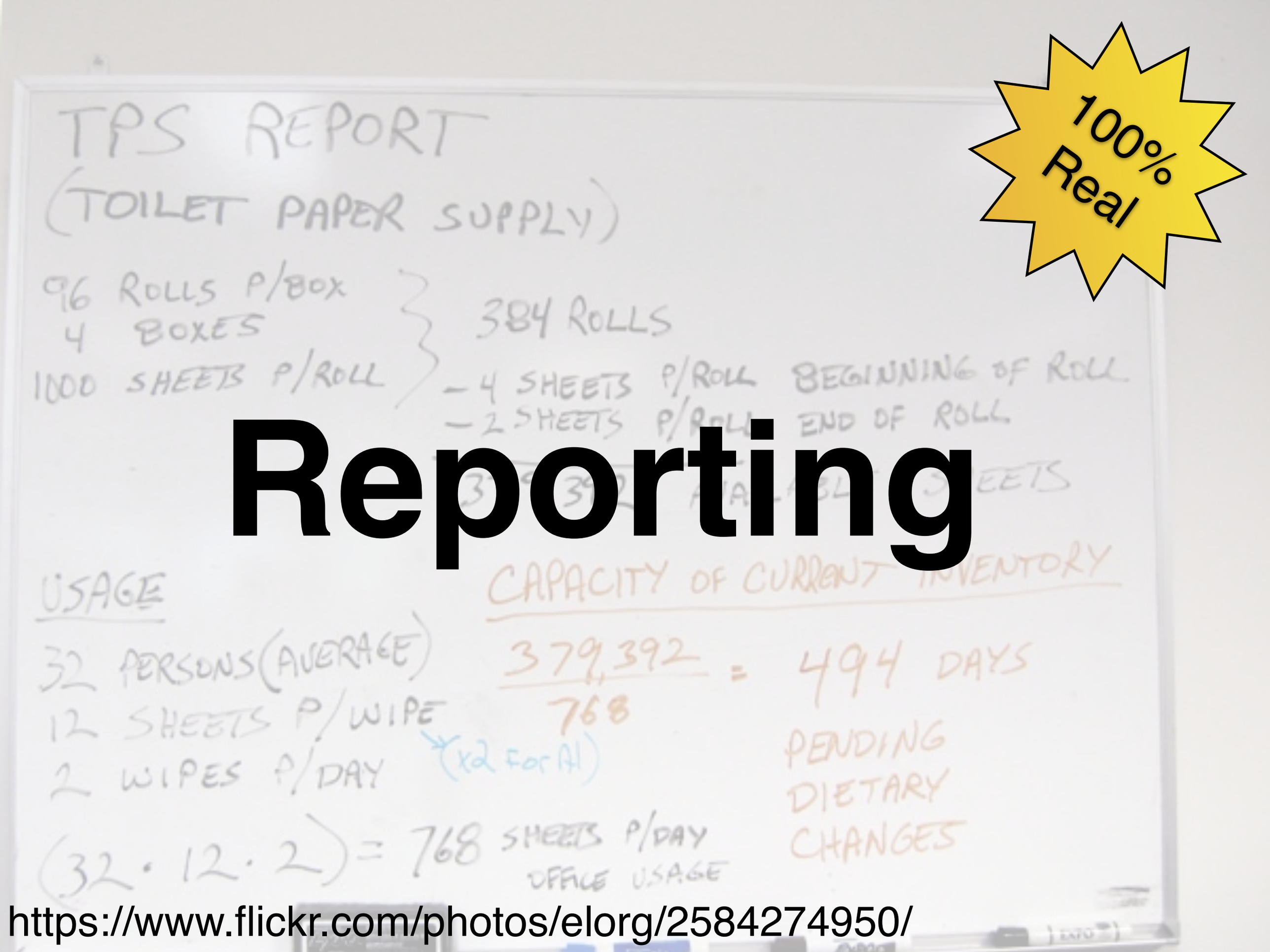
ETL

http://commons.wikimedia.org/wiki/File:BalticServers_data_center.jpg

Extract
Transform
Load

http://commons.wikimedia.org/wiki/File:BalticServers_data_center.jpg

Active Record Models



100% Real

Reporting

https://www.flickr.com/photos/elorg/2584274950/

Recognizing It

```ruby
class PastDueAccountsReport

  def extract
    @data = Accounts.past_due
  end

  def aggregate
    @data = @data.group_by { |a| a.owner }
  end

  def email
    Email.new("accounting@adequate.hq", @data)
  end

  def generate
    extract
    aggregate
    email
  end
end
```

```ruby
class NewUsersReport

  def extract
    @data = Users.new_in_last_24_hours
  end

  def aggregate
    @data = @data.group_by { |u| u.department }
  end

  def email
    Email.new("reports@adequate.hq", @data)
  end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class NewUsersReport

  def extract
    @data = Users.new_in_last_24_hours
  end

  def aggregate
    @data = @data.group_by { |u| u.department }
  end

  def email
    Email.new("reports@adequate.hq", @data)
  end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class ReportTemplate
  def extract; raise "NYI"; end
  def aggregate; raise "NYI"; end
  def email; raise "NYI"; end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class ReportTemplate
  def extract; raise "NYI"; end
  def aggregate; raise "NYI"; end
  def email; raise "NYI"; end

  def generate
    extract
    aggregate
    email
  end
end
```

```ruby
class ReportTemplate
  def extract; raise "NYI"; end
  def aggregate; raise "NYI"; end
  def email; raise "NYI"; end

  def generate
    extract
    aggregate
    email
  end
end
```

```
class NewUsersReport < ReportTemplate

  def extract
    @data = Users.new_in_last_24_hours
  end

  def aggregate
    @data = @data.group_by { |u| u.department }
  end

  def email
    Email.new("reports@adequate.hq", @data)
  end

end
```

```
class PastDueAccountsReport < ReportTemplate

  def extract
    @data = Accounts.past_due
  end

  def aggregate
    @data = @data.group_by { |a| a.owner }
  end

  def email
    Email.new("accounting@adequate.hq", @data)
  end

end
```

Share Common Code

Isolate Differences

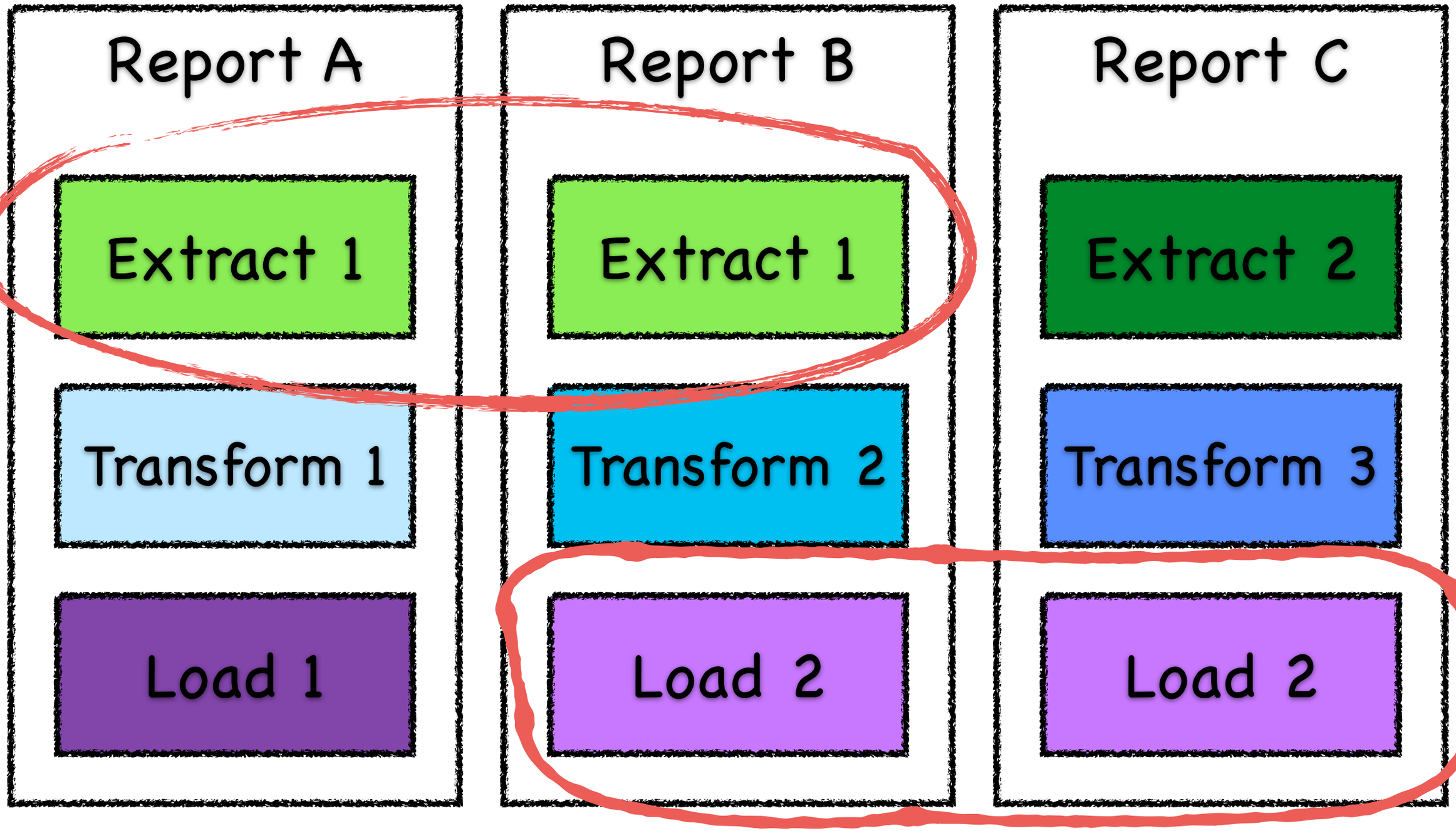
Easy To Add New Versions

Force Common Algorithm

Read multiple files

Inheritance

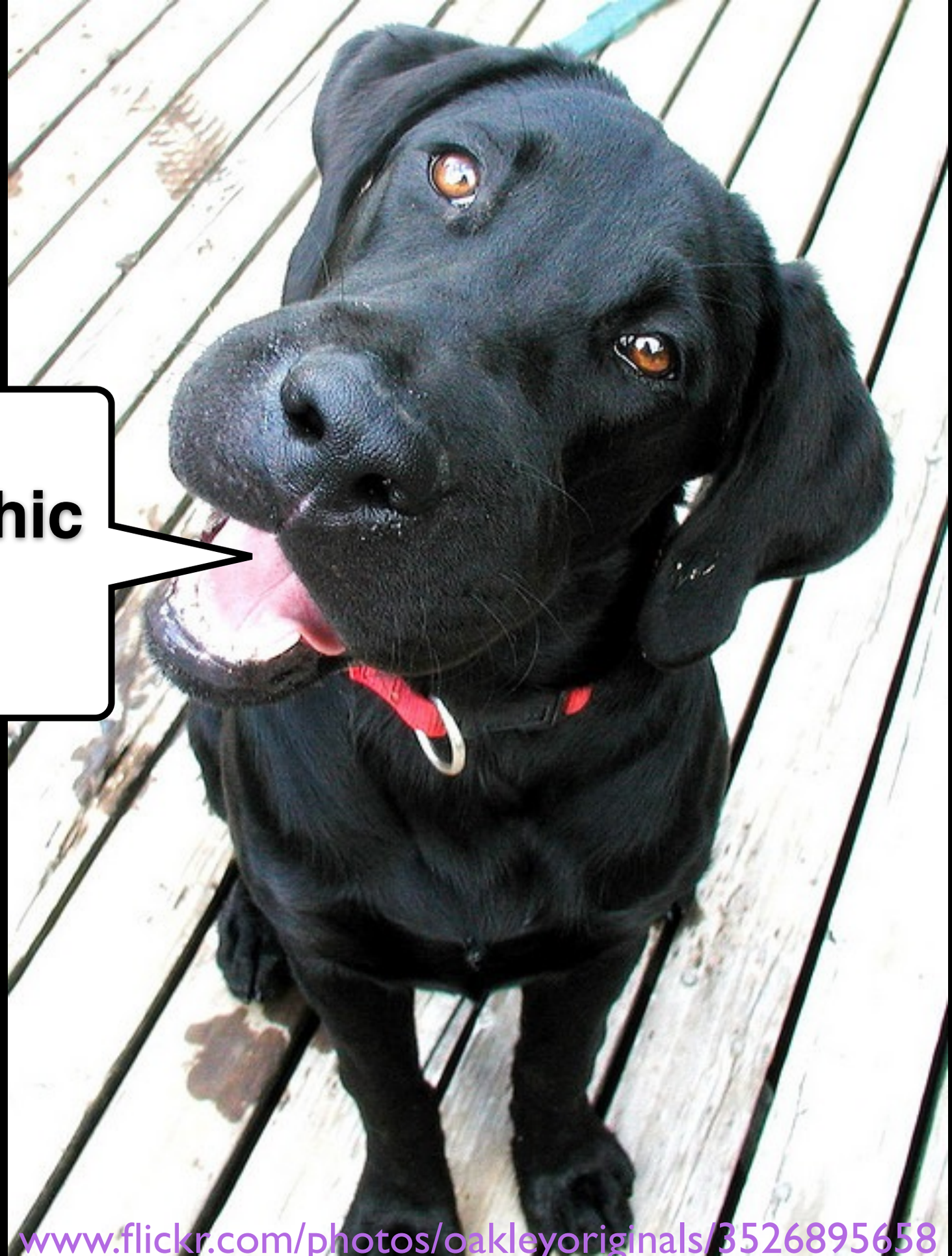
Inheritance


Report A
Report B
Report C
Extract 1
Extract 1
Extract 2
Transform 1
Transform 2
Transform 3
Load 1
Load 2
Load 2

Strategy

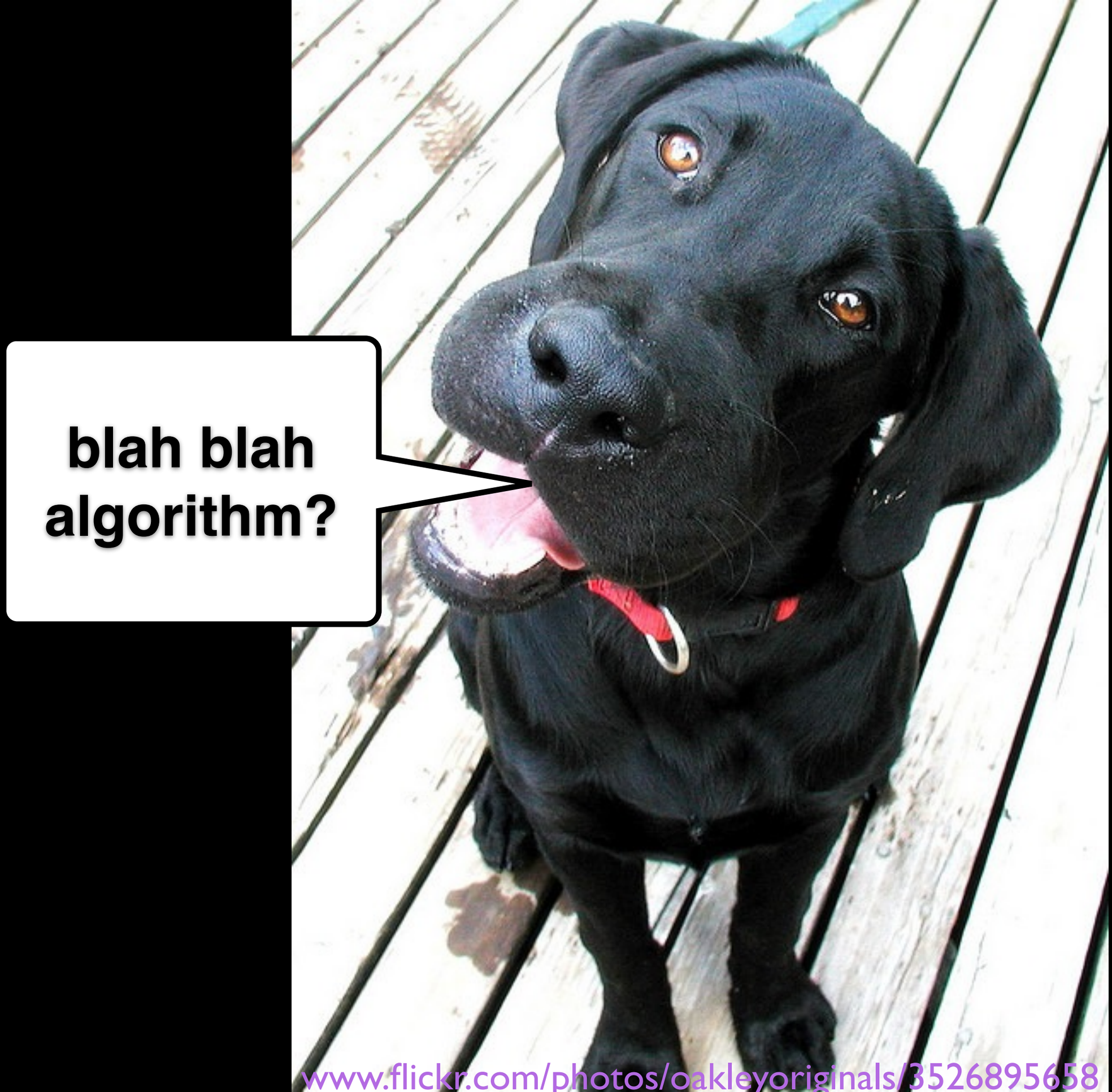
What Is It?

Polymorphic Algorithm



www.flickr.com/photos/oakleyoriginals/3526895658

Change The Algorithm At Runtime



www.flickr.com/photos/oakleyoriginals/3526895658

Extract Algorithm

Step 1

Step 2

Step 3a

Step 3b

Step 3c

Step 4

Step 1

Step 2

Choose Strategy

Step 4

Step 3a

Step 3b

Step 3c

Examples

Storage Strategy

http://upload.wikimedia.org/wikipedia/commons/8/85/Komórki_1_m_self_storage.jpg

Stock Market

http://simple.wikipedia.org/wiki/Stock_market#mediaviewer/File:E-ticker.jpg

Sales Tax

http://www.flickr.com/photos/phillip/345829246/

Recognizing It

```
def handle(a, b, c)
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  if a == 'x'
    results = x_request a, b
  elsif a == 'y'
    results y_request a
  elsif a == 'z' and b == '123'
    results z_request a, b, c
  end

  # step 4
  present results
end
```

```ruby
def handle(a, b, c)
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  if a == 'x'
    results = x_request a, b
  elsif a == 'y'
    results y_request a
  elsif a == 'z' and b == '123'
    results z_request a, b, c
  end

  # step 4
  present results
end
```

```ruby
def initialize
  @strategies = {}
  @strategies['x'] = XStrategy.new
  @strategies['y'] = YStrategy.new
  @strategies['z'] = ZStrategy.new
end

def handle a, b, c
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  @strategies[a].run a, b, c

  # step 4
  present results
end
```

```ruby
def initialize
  @strategies = {}
  @strategies['x'] = XStrategy.new
  @strategies['y'] = YStrategy.new
  @strategies['z'] = ZStrategy.new
end

def handle a, b, c
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  @strategies[a].run a, b, c

  # step 4
  present results
end
```

```
def initialize()
  @strategies = {}
  @strategies['x'] = XStrategy.new
  @strategies['y'] = YStrategy.new
  @strategies['z'] = ZStrategy.new
  @strategies['aa'] = AAStrategy.new
end

def handle(a, b, c)
  # step 1
  normalize_inputs a, b, c

  # step 2
  reticulate_splines

  # step 3
  @strategies[a].run(a, b, c)

  # step 4
  present results
end
```

```
def add_strategy(key, strategy)
  @strategies[key] = strategy
end
```

```
add_strategy 'aa', AAStrategy.new
```

Share Common Code

Isolate Differences

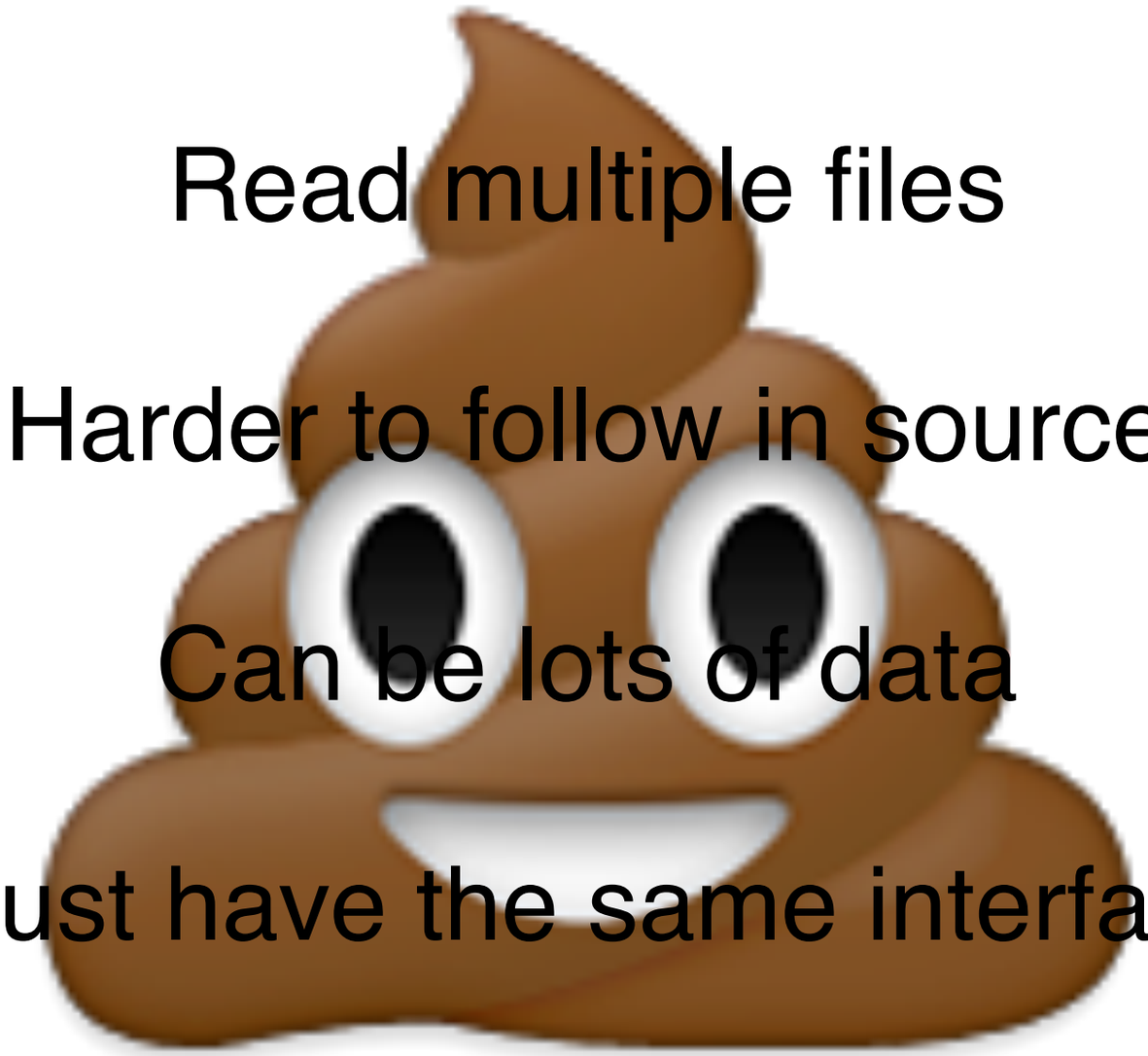
Easily Add New Strategies

Force Common Interface

Avoid Inheritance

Read multiple files

Harder to follow in source



Can be lots of data

Must have the same interface

Composite

What Is It?

Common Interface

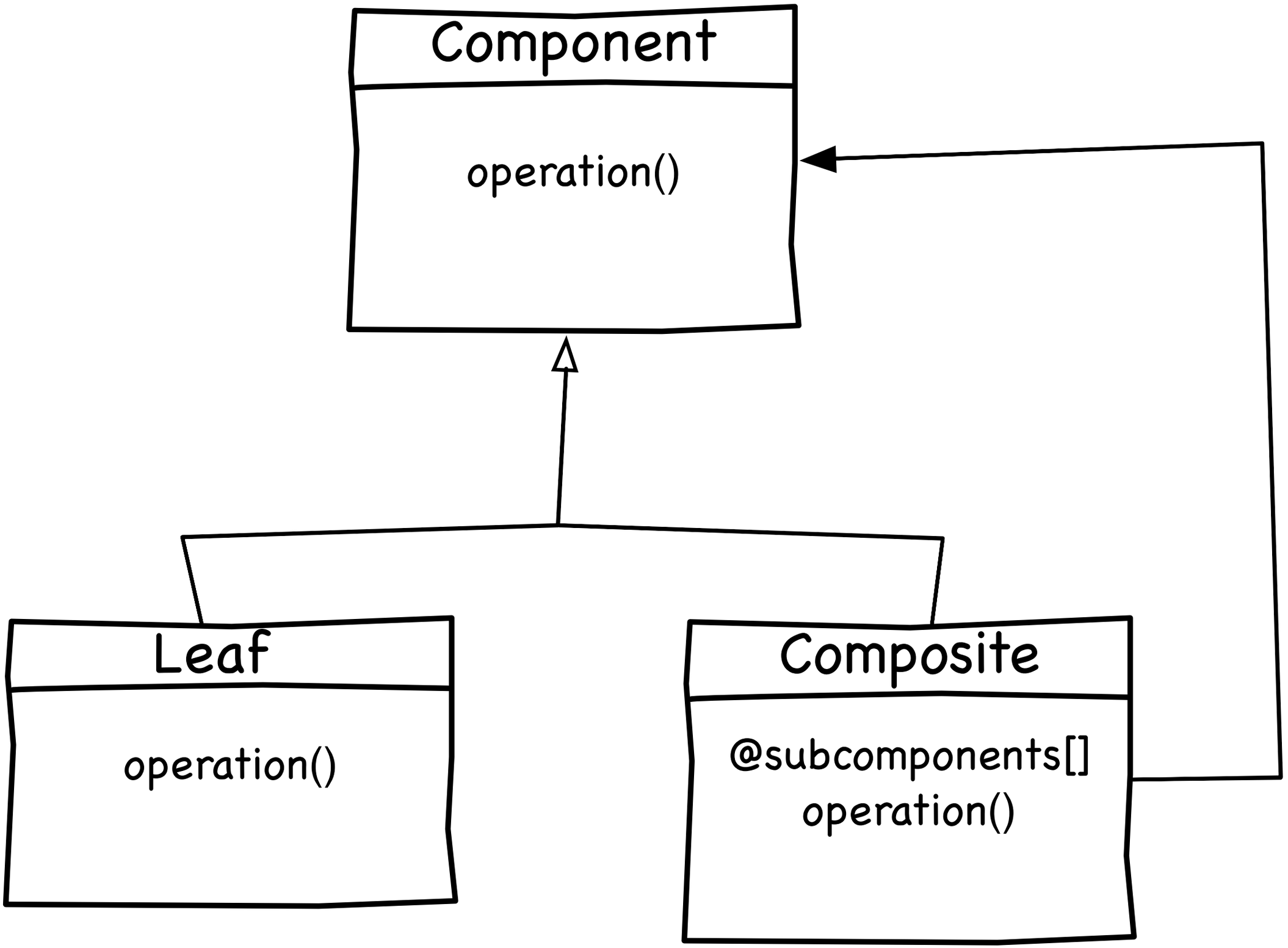
For One or Many

https://www.flickr.com/photos/joi/2569341936

Let Consumers

Ignore Quantity

Vocabulary


Component
operation()
Leaf
operation()
Composite
@subcomponents[]
operation()

Examples

Inbox: StuporSlow

Inbox | Mode | Due | Review | Clean Up | Standup | Waiting | Home | Work | Focus | Search

Project Filter: Remaining | Grouping: Ungrouped | Sorting: Unsorted | Availability Filter: Remaining | Status Filter: Any Status | Est… Any D

- Inbox
- Library
- Home
  - BlogTopics
  - Code
    - StuporSlow
    - Learn about…
    - Gosu
    - redmine_qa_con…
    - Talks
  - 72 hour kit 3
  - Misc 1
  - Self… 1
  - Clean Up… 2
  - Finances
- Nerd
  - omnifocus gems
  - ruby rogues 1
  - Racket 1
  - tic_tac_toe
- Work
  - Bluescape

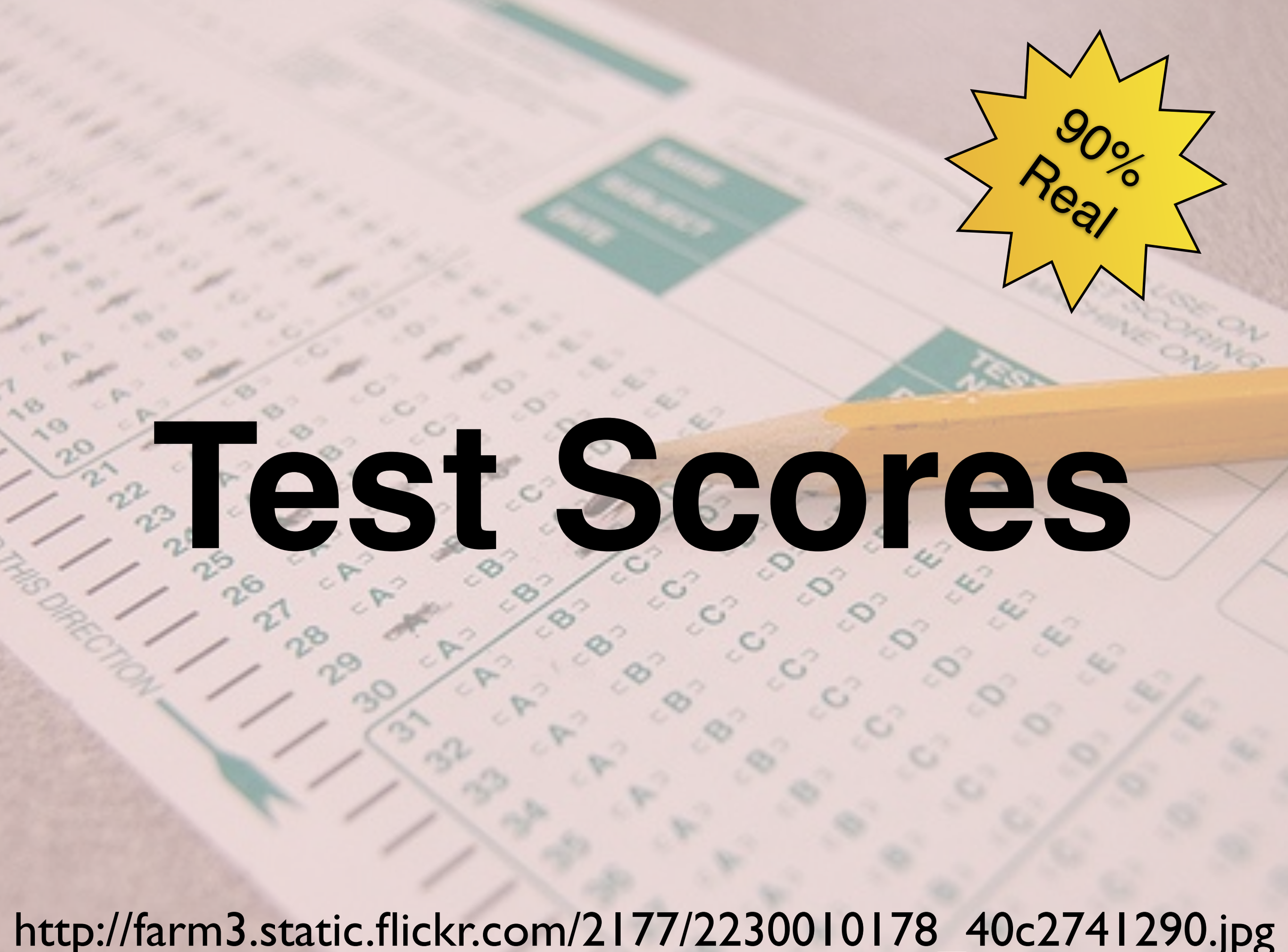
Name | Start | D

- StuporSlow
  - Improve testing
    - Add test for workout creation failure
    - Fix pending tests
    - Coverage and fix coverage
    - Consider using fo… simplify models
    - Move monkey patch of ar… to ad… Avg … b
  - Improve UI / Design
    - Put in bootstrap
    - Add better table views for workouts and work_template
    - Make homepage the graph, data, and an add workout button (+)
    - Have … EST
    - … Ry… late… visu… dat…
    - … jax to add … re … to workout and … mplat… …es
    - Add drop downs to the visualize page… restrict the graph/table to a specific exercise
    - Figure out which columns are required and which are optional — No Context — Start — D
    - Add indexes
    - Trainer model

Task Management

HTML

http://upload.wikimedia.org/wikipedia/commons/8/81/HTML5_de_Erick_Dimas.jpg



http://farm3.static.flickr.com/2177/2230010178_40c2741290.jpg

Recognizing It

```ruby
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

```ruby
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

```ruby
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

```ruby
def report_scores x
  case x
  when Student
    x.score
  when Classroom
    x.students.map(&:score).avg
  when School
    x.classes.map(&:students).flatten.map(&:score).avg
  end
end
```

Composite

```
class Composite
  def score
    # Some implementation
  end
end
```

```ruby
class Student
  attr_accessor :score
end
```

```ruby
class Classroom
  attr_accessor :students

  def score
    students.map(&:score).avg
  end
end
```

```ruby
class School
  attr_accessor :classes

  def score
    classes.map(&:score).avg
  end
end
```

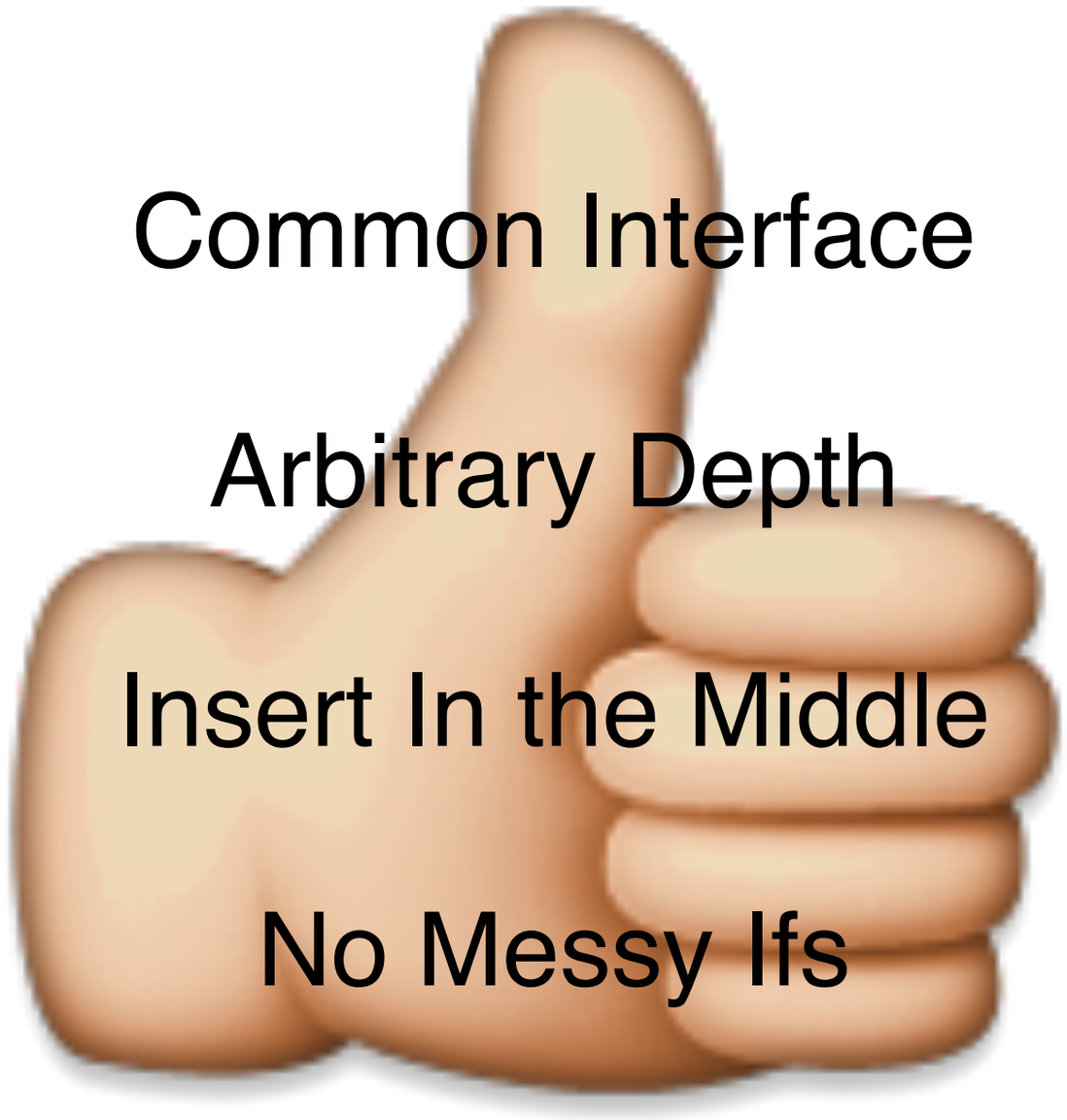
```
def report_scores(x)
  x.score
end
```

Common Interface

Arbitrary Depth

Insert In the Middle

No Messy Ifs



Harder to follow in source

Must remember hierarchy

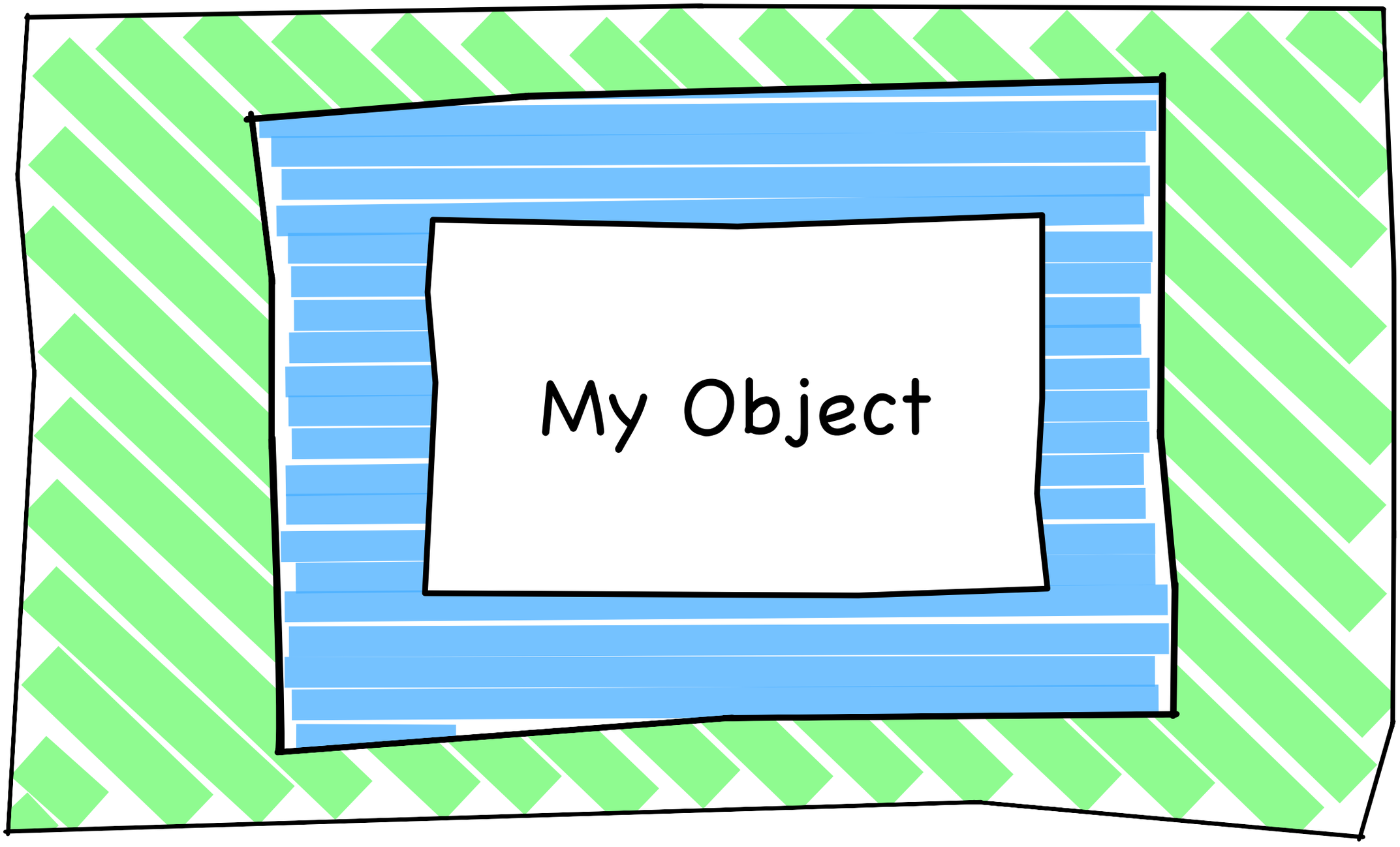
Decorator

What Is It?

Extend an Instance of an Object

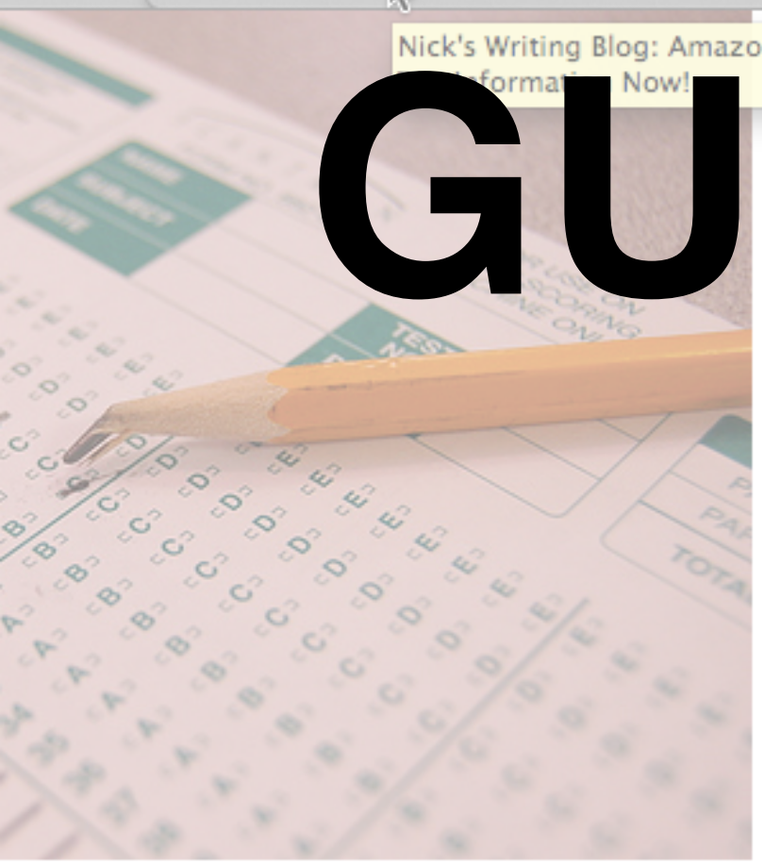
http://laura-c-f.deviantart.com/art/Confused-Dog-286725830

Wrapper Adding Functionality


My Object

Examples

GUI Elements



Calculating Discounts

http://therockinstallion.deviantart.com/art/Sparky-BOGO-2-390047802

RPG Characters

Recognizing It

```ruby
class Character
  attr_accessor :health, :damage, :xp

  def alive?
    health > 0
  end

  def attack enemy
    enemy.health -= self.damage
  end
end
```

POW!!

```ruby
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```ruby
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```ruby
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```ruby
require 'forwardable'

class DragonMightDecorator
  extend Forwardable

  def_delegators :@character, :health, :xp

  def initialize(character)
    @character = character
  end

  def damage
    @character.damage * 1.5
  end
end
```

```
> c = Character.new
> c.health = 100; c.damage = 10;

> d = DragonMightDecorator.new(c)
> d.damage
15.0

> d.health
100
```

Change Behavior Dynamically

Customize an Instance

Support Nesting

Offload Pain To Consumers

Multiple Files

Many Decorators -> Unpredictable

Command

What Is It?

Objectified Closure

https://www.flickr.com/photos/joi/2569341936

Object Containing Algorithm & Context

Examples

Delayed Job

Drawing Applications

http://upload.wikimedia.org/wikipedia/commons/1/1e/Tuxpaint-magic-tools.png

Active Record Migrations

Recognizing It

Undo

```ruby
class CreateRecipes < ActiveRecord::Migration
  def self.up
    create_table :recipes do |t|
      t.column :name, :string
      t.timestamps
    end
  end

  def self.down
    drop_table :recipes
  end
end
```

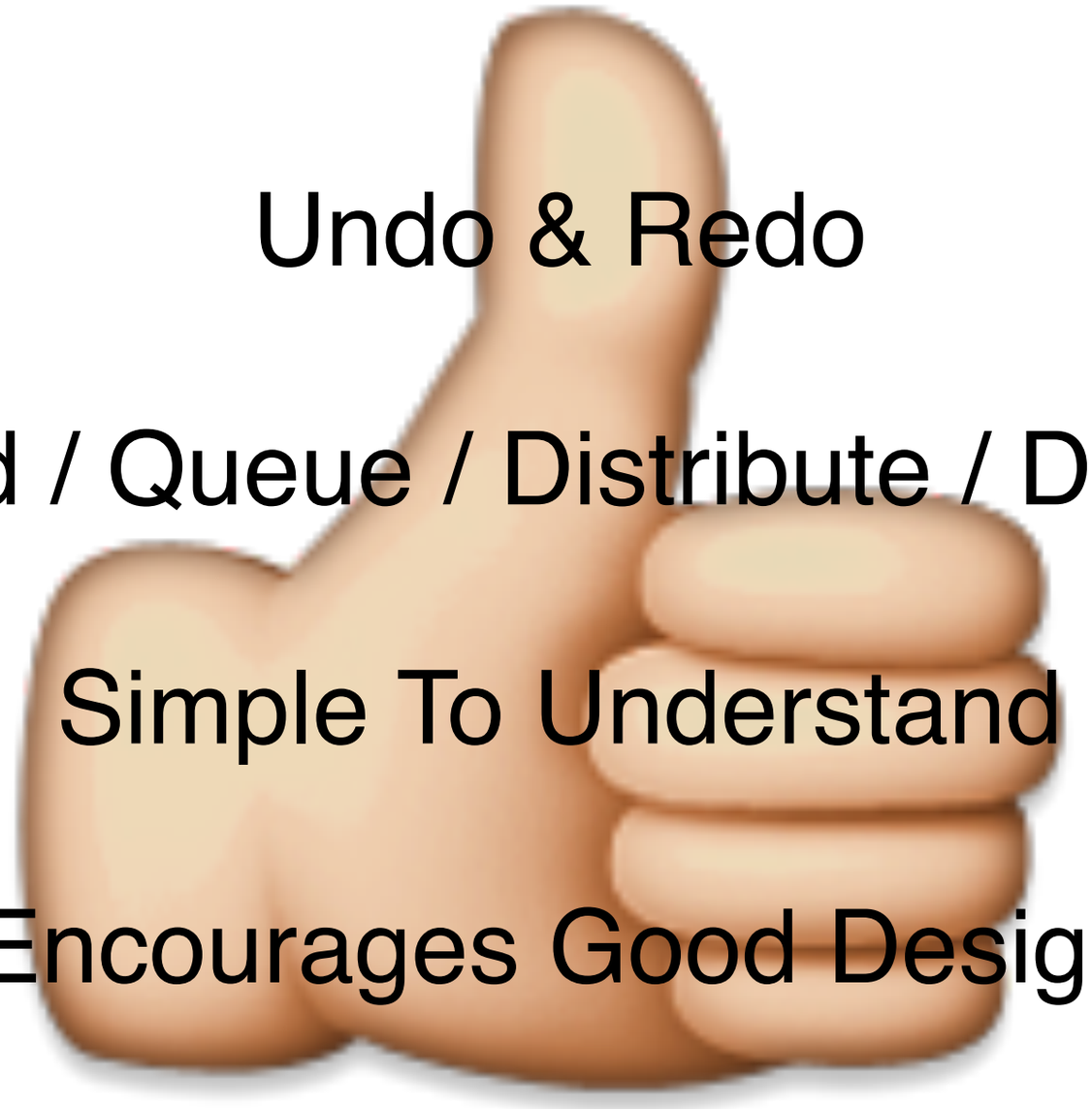
```ruby
class CreateRecipes < ActiveRecord::Migration
  def self.up
    create_table :recipes do |t|
      t.column :name, :string
      t.timestamps
    end
  end

  def self.down
    drop_table :recipes
  end
end
```

Delay

Delegation

Undo & Redo

Offload / Queue / Distribute / Delegate



Simple To Understand

Encourages Good Design

Overused

YAGNI

So What?

You Are Using

Patterns

Common Vocabulary

Cheat Sheet

Several Objects That Are Mostly The Same?

Template

Process The Same

Except *Step 3*?

Strategy

Hierarchical Objects?

Composite

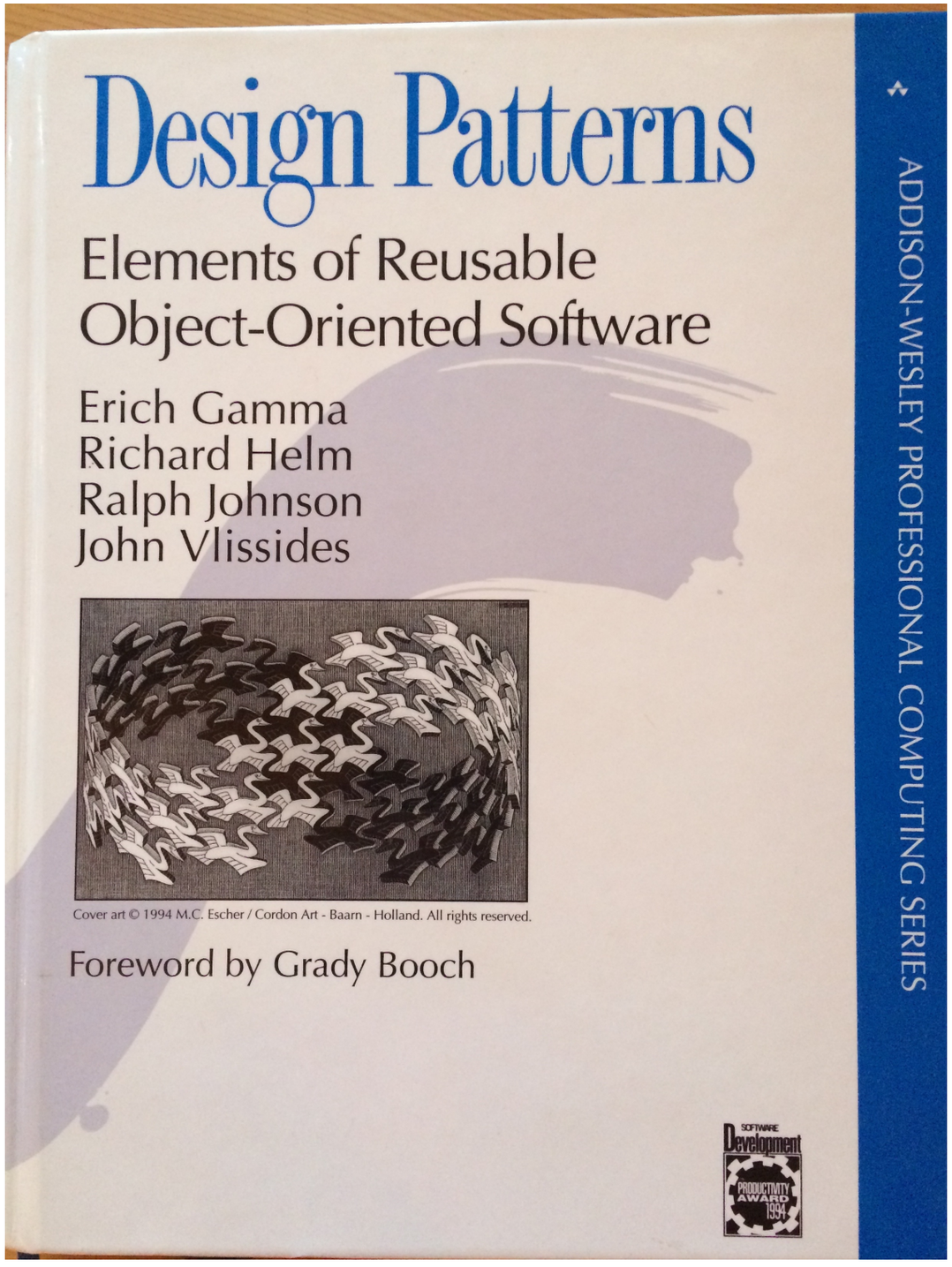
Sometimes Need Enhancements?

Decorator

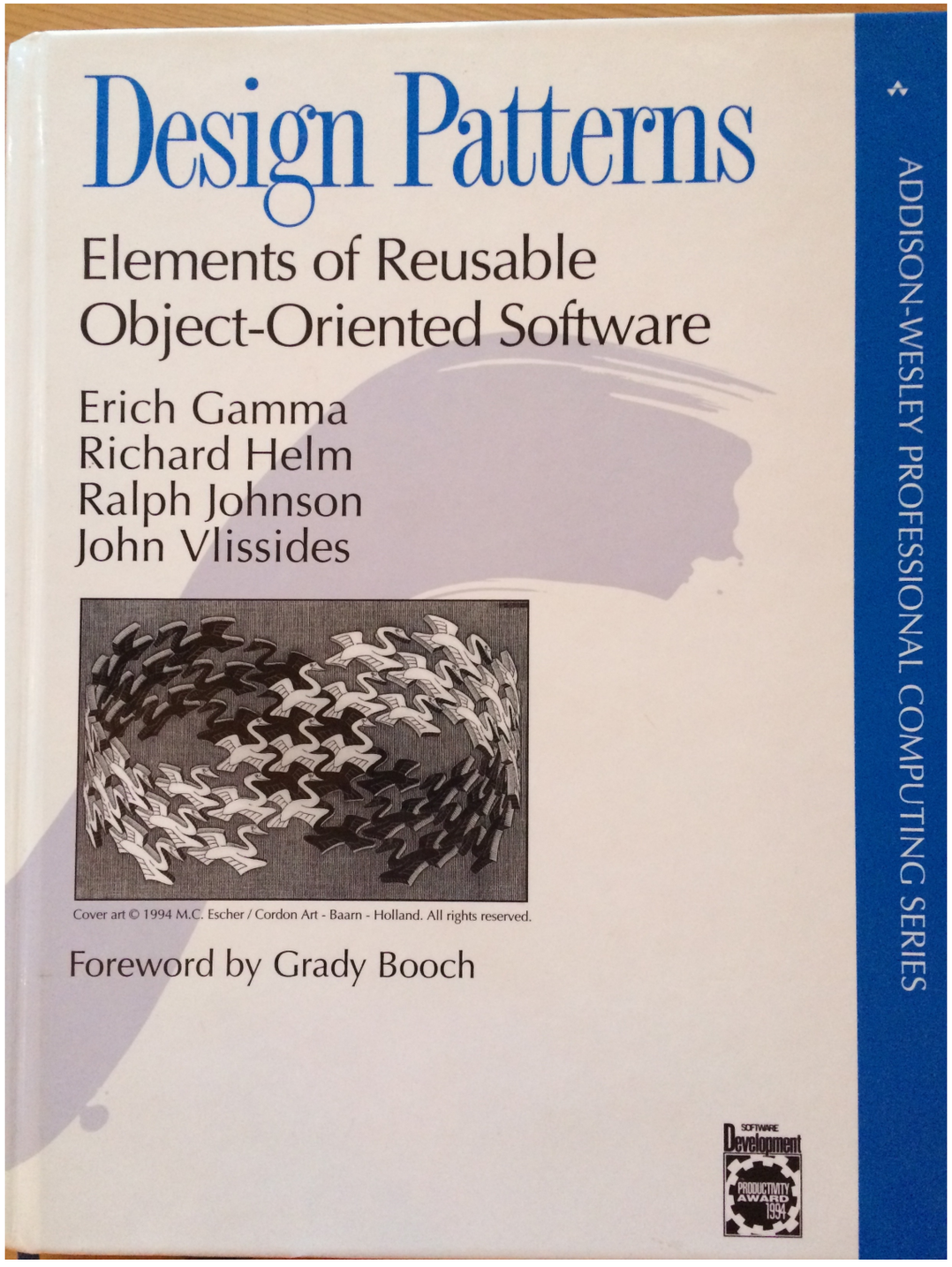
Undo or Delay or Delegate?

Command

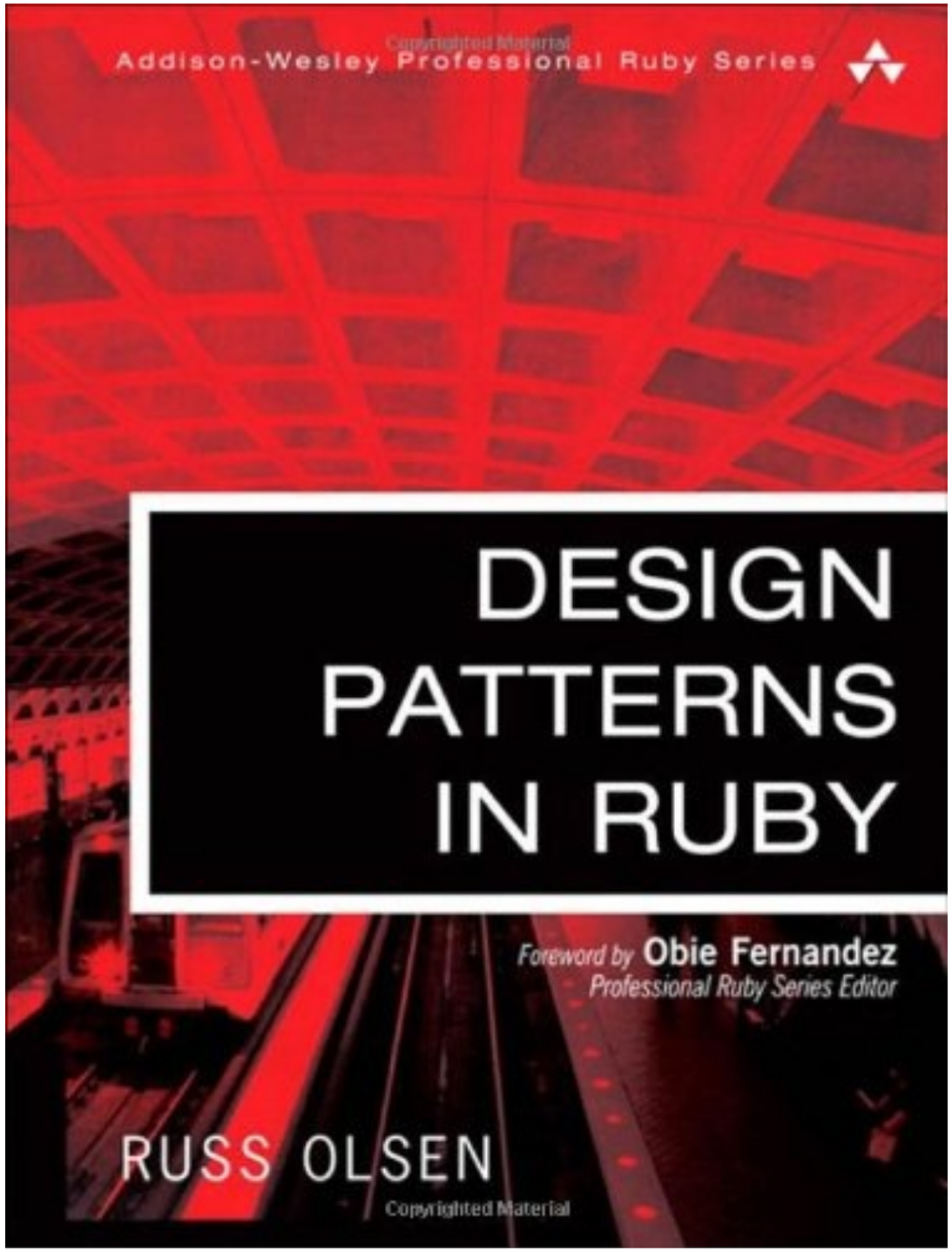
Learn More



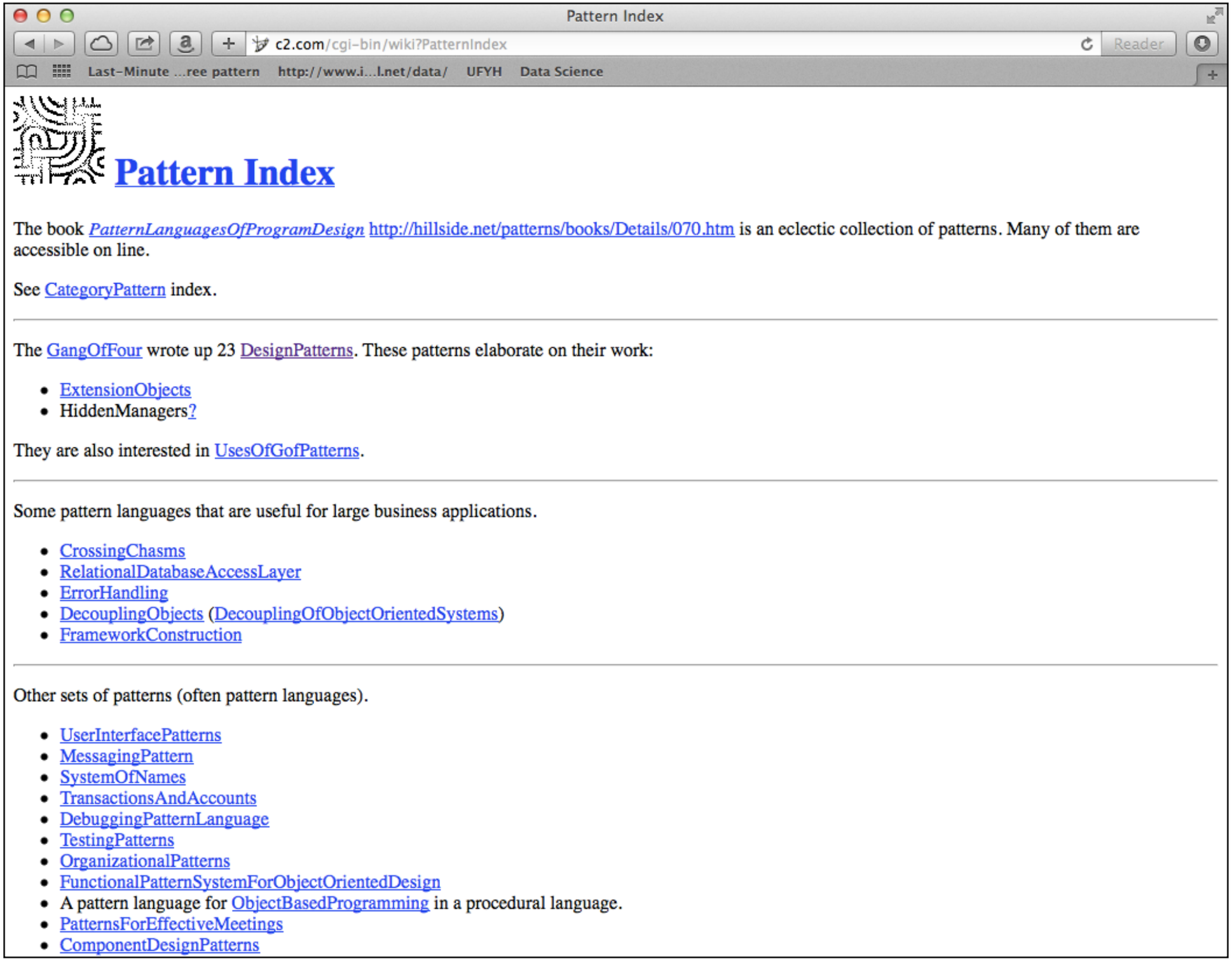
Design Patterns


Design Patterns
Elements of Reusable
Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.
Foreword by Grady Booch
SOFTWARE Development PRODUCTIVITY AWARD 1994
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Gang **O**f **F**our



Design Patterns In Ruby



Original Wiki

Thanks