

# Ruby On Rails

Armin Sadeghi

Duncan Bayne

September 20, 2008

# What is Ruby on Rails?

- Rails is a web development framework optimised for RAD and Agile methodologies
- Ruby is the language used by the Rails framework
- Released in 2004
- Available for Linux, MS Windows, Solaris, FreeBSD and even ships with Mac OS X
- Strong support base and OSS community

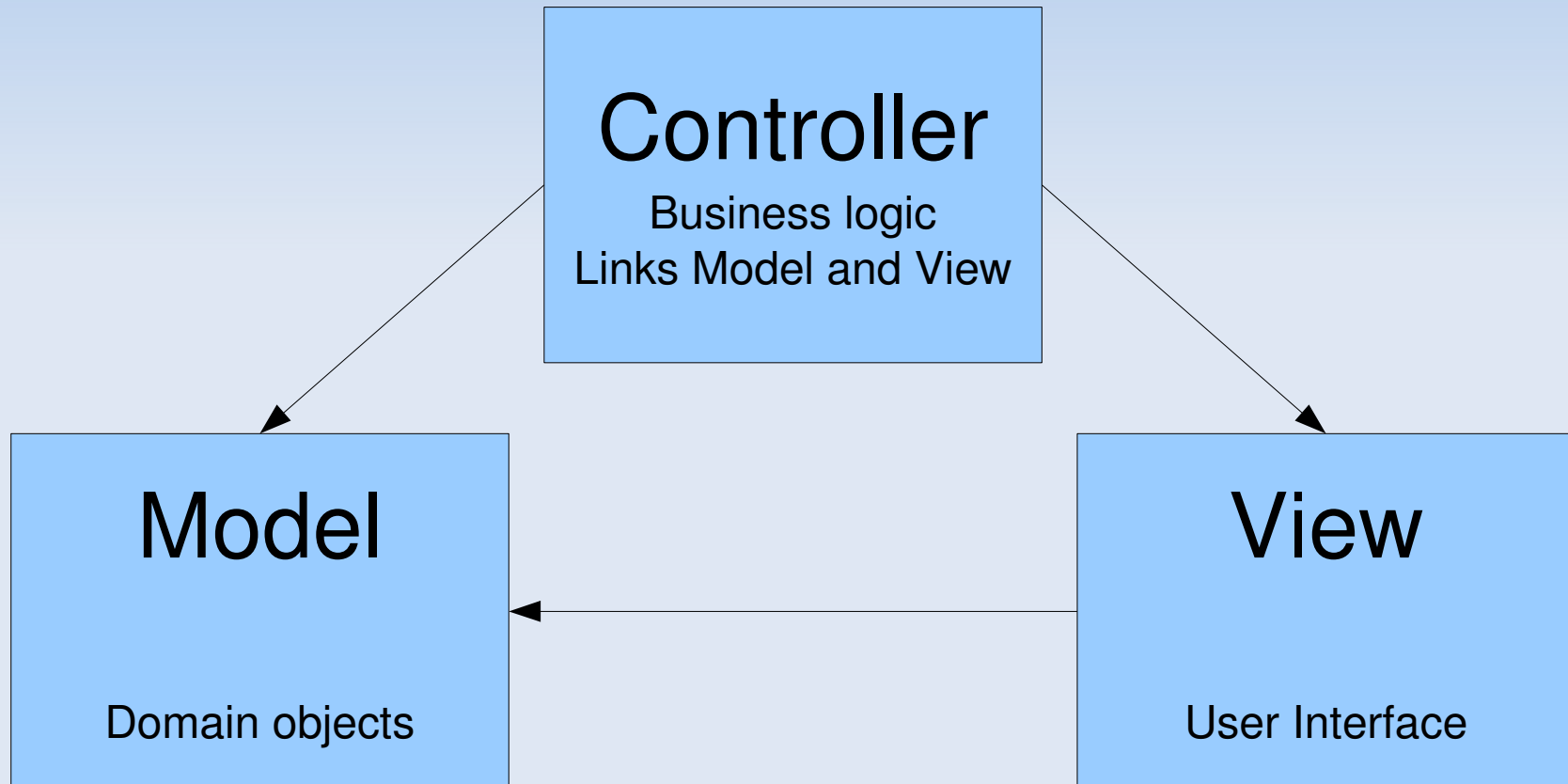
# Target Applications

- Small to medium size green-field applications
- Responsive AJAX user interfaces
- Can provide web and web service interfaces
- Unproven in very large scale deployments

# Architecture

- Model – View – Controller
- Ruby Language
- ActiveRecord
- Test Framework
- Migrations

# Model – View – Controller



# Ruby Language

- Free, open-source, OO interpreted language
- Development started in 1990s in Japan
- Automatic memory management
- Integrated package management (RubyGems)
- Dynamic Code
- Duck-typed

# Ruby Duck-Typing

```
class Dog
  def play
    print "Jump around barking\n"
  end
end
```

```
class Cat
  def play
    print "Scratch stuff\n"
  end
end
```

```
def game(dog)
  dog.play
end
```

```
game(Dog.new)
```

```
game(Cat.new)
```

```
~/ruby$ ruby duck.rb
```

```
Jump around barking
```

```
Scratch stuff
```



# Ruby Dynamic Code

```
class Deal  
end
```

```
dealA = Deal.new  
Deal.send(:define_method, :hi) do  
  print "Hello world!\n"  
end
```

```
dealA.hi
```

```
~/ruby$ ruby dynamic.rb
```

```
Hello world!
```





# ActiveRecord

- Dynamically generated ORM
- Database driven
- Uses Ruby's `method_missing` to add functionality:

```
class Dummy
  def method_missing(m, *args)
    puts("There's no method called #{m} here -- please try again.")
  end
end
```

- ActiveRecord adds missing methods as they're called, e.g. `User.find_by_street_address()`.

# Test Framework

- Three databases – Test, Development and Production
- Rails repopulates Test database from YAML fixtures before each test
- Unit tests for Models
- Functional tests for Controllers
- Optimised for Test Driven Development

# Migrations

- Script database changes using migrations
- Scripts are database-agnostic (in Ruby)
- Changes are atomic and can be rolled back

```
class AddNameToUser < ActiveRecord::Migration
  def self.up
    add_column :users, :name, :string, :null => false, :default => 'Unknown'
  end

  def self.down
    remove_column :users, :name
  end
end
```

# Tools

- Eclipse IDE
- Aptana plug-in for Eclipse
- Subversion source control

# Coding Demo

- Live coding demo that demonstrates:
  - Tests
  - Scaffolding
  - ORM
  - Migrations
  - Views

# Questions



# Resources

- <http://www.rubyonrails.org/>
- <http://www.railsforum.com/>
- <http://www.ruby-lang.org/>
- <http://instantrails.rubyforge.org/>
- <http://www.offbyzero.com/>